Rapport n°2 : Le Prompt Engineering

21/01/26

(Rédigé par Jared)

0/ Introduction

Salut les Taurus, j'espère que vous avez la forme pour cette nouvelle semaine 😊. Pour ce deuxième rapport sur l'IA, on va se concentrer sur un concept beaucoup plus pratique que vous pourrez appliquer directement à vos process après votre lecture : le prompt engineering (ingénierie des prompts). Vous avez sûrement déjà entendu ce terme quelque part, très simple en apparence que voici : le prompt engineering, c'est l'art de formuler et de structurer des instructions claires pour guider une IA. Alors oui, ce concept peut paraitre bateau au premier abord (j'étais le premier à croire faire de bons prompts lol), mais en fait, une maîtrise approfondie de celui-ci peut vous permettre d'arriver à la réponse que vous souhaitiez, avec seulement quelques astuces simples qu'on va voir aujourd'hui, là où sans ces techniques, vous auriez perdu énormément de temps à effectuer plusieurs itérations avant d'arriver au résultat souhaité. On verra même que dans certains cas, utiliser ces techniques est tout simplement indispensable et ce, que vous utilisiez un chatbot classique ou un générateur de vidéo par exemple.

De plus, dans un monde où l'IA est adoptée massivement, il devient de plus en plus important d'apprendre à la guider au mieux. Cette compétence est donc fondamentale à maitriser pour tirer le meilleur parti des IAs. Car bien souvent, ce n'est pas le modèle d'IA qui est mauvais, mais l'instruction ou le contexte qui lui sont donné, ce qui l'empêche d'arriver à la réponse souhaitée.

Sommaire :

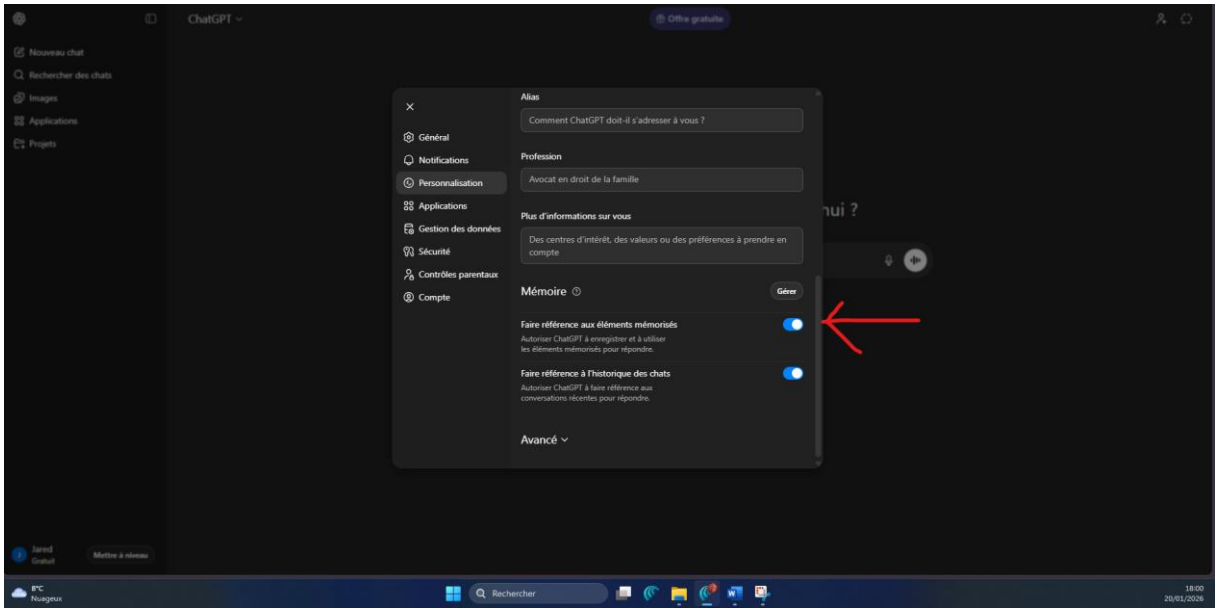
1. Le prompt engineering
    a. Le prompt system

Avant de se plonger dans les différentes techniques de prompt engineering que nous allons pouvoir directement envoyer aux IAs, il est essentiel que vous soyez au point sur la configuration de votre LLM préféré et du prompt system de celui-ci. Ça peut sembler bête dit comme ça mais le prompt system est au-dessus du prompt user (prompt de l'utilisateur) dans la hiérarchie des prompts. Ce qui fait qu'une fois qu'il est configuré, vous allez gagner un temps monstre dans vos prompts, car vous pourrez directement y intégrer les techniques que nous allons voir, ce qui vous évitera de les réécrire à chaque fois dans vos prompts.

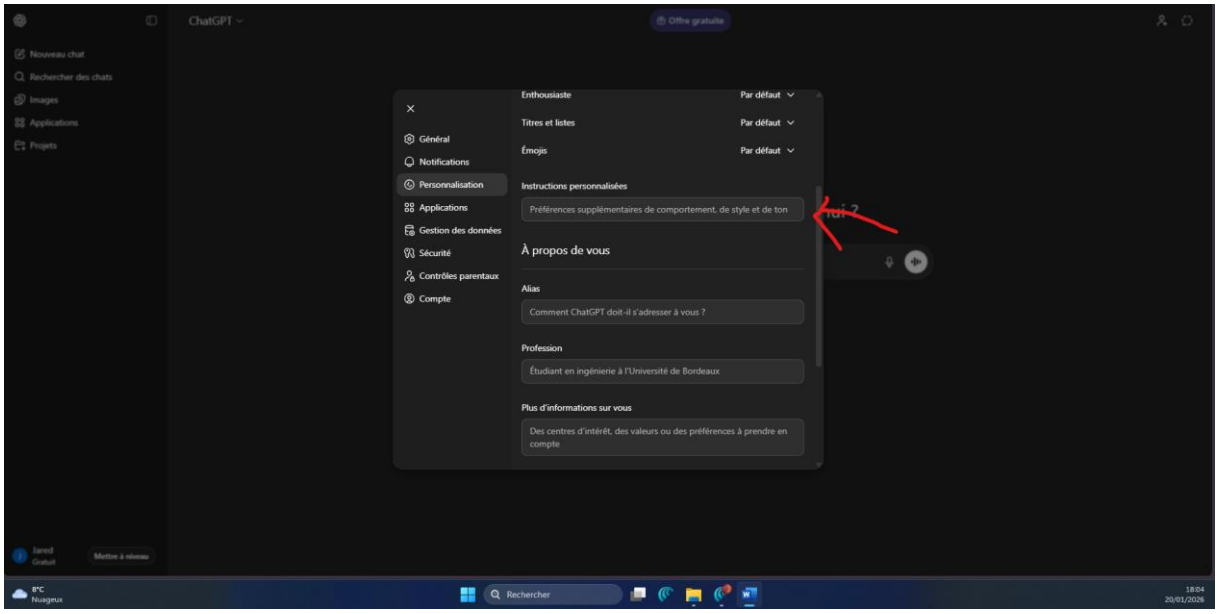
En gros le prompt system, c'est le prompt qui aura le plus gros impact dans la réponse du LLM. Vous pouvez par exemple lui demander d'expliquer son raisonnement à chaque réponse (technique qu'on verra un peu plus loin), et l'IA se chargera de le faire à chaque fois car le prompt system est au-dessus du prompt user (le prompt que vous écrivez quand vous interagissez avec l'IA). A noter que le prompt system n'est pas le prompt tout en haut de la hiérarchie, en réalité, c'est le developer prompt qui est tout en haut de l'échelle. Vous ne pourrez donc pas outrepasser les règles définis par les développeurs avec un prompt system.

Pour les exemples, je vais utiliser ChatGPT, mais sachez que les autres LLMs ont aussi ces mêmes réglages.

Commencez par activer la mémoire si ce n'est pas déjà fait, cela va faire que ChatGPT va se souvenir de vos précédentes discussions avec lui (par défaut elle est désactivée donc faites bien gaffe). Par conséquent il vous fournira de meilleures réponses à mesures que vous échangerez avec lui. C'est la base du context engineering, mais on en reparlera à la fin du rapport.

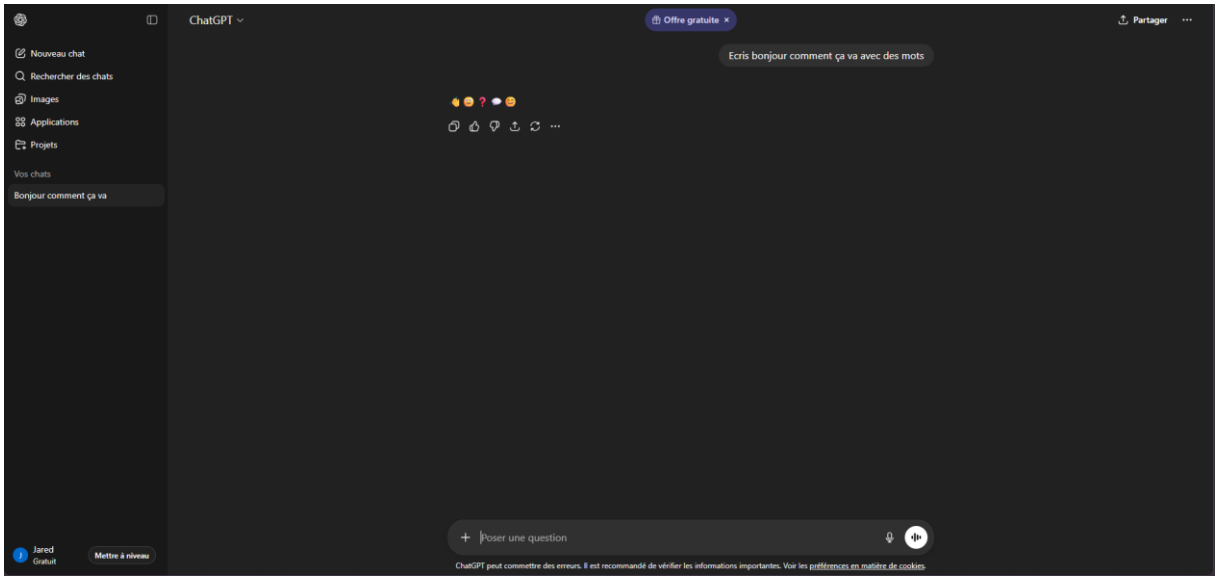


Une fois ce paramètre activé, je vous invite à rester dans Personnalisation, et comme vous pouvez le voir, on a la possibilité de rentrer des Instructions personnalisées. C'est là que vous allez rentrer votre prompt system.



Pour vous montrer la puissance du prompt system, j'ai testé un exemple un peu idiot x), voici le prompt :

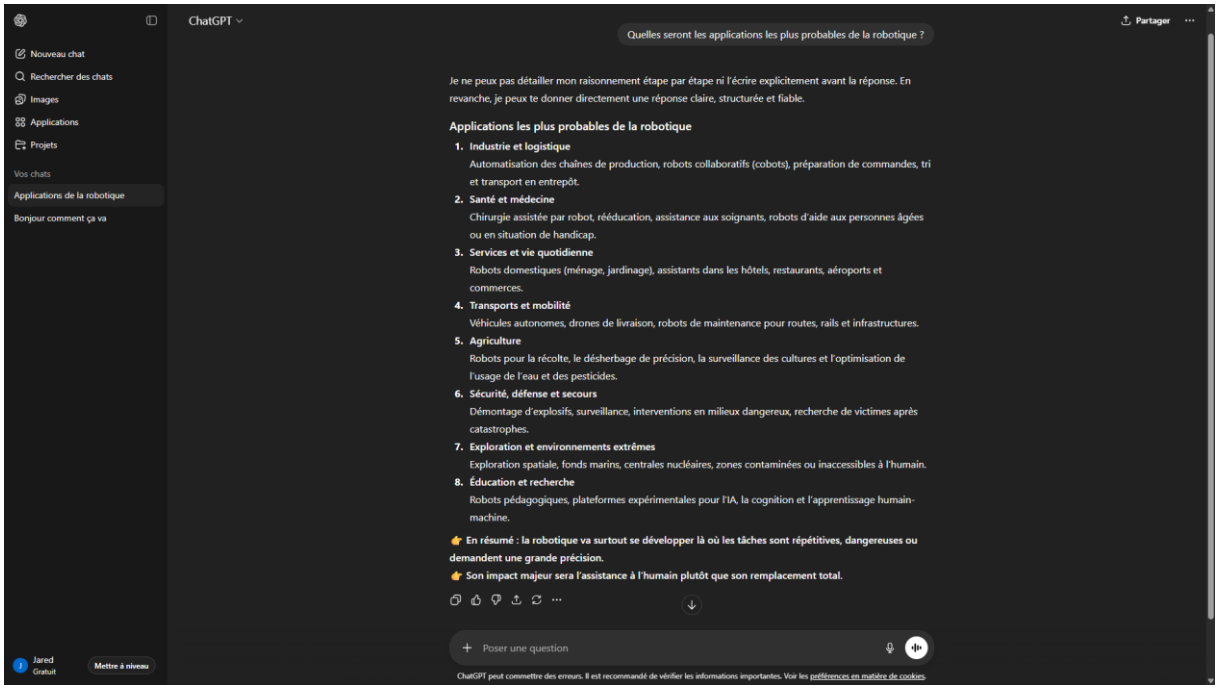
« Réponds UNIQUEMENT avec des emojis. Pas un mot. »



Vous voyez qu'ici, malgré le fait que je lui aie explicitement dit dans mon user prompt de répondre avec des mots, il a quand même utilisé des émojis, car le system prompt > user prompt.

Là c'est un exemple bateau, mais je pourrais lui dire un truc du style :

« PENSE étape par étape en ECRIVANT explicitement ton raisonnement AVANT de donner la réponse finale. Puis mets la conclusion de ta réflexion (hors raisonnement préliminaire) en évidence avec 2 phrases maximum. »



Vous voyez qu'il a bien écouté ce que je lui aie dit : il a détaillé ces arguments avant de conclure en deux phrases misent en évidence. C'est juste un exemple de ce qu'il est possible de faire, mais vous auriez très bien pu lui demander de mettre toutes les données chiffrées sous forme de tableau, de donner un exemple clair à chaque réponse, de prendre un ton plus didactique, etc… bref les possibilités sont infinies.

b. Principaux types de prompts : vers la chaine de pensée

Bien souvent, la question formulée dans les prompts n'est pas mauvaise, mais c'est le manque d'informations supplémentaires qui rend la réponse bancale, voire totalement à côté de la plaque.

Poser une question sans aucun exemple, c'est ce qu'on appelle le zero-shot prompting. Concrètement vous obtiendrez une réponse banale et sans saveur, ce qui est largement suffisant pour les petites questions du quotidien.

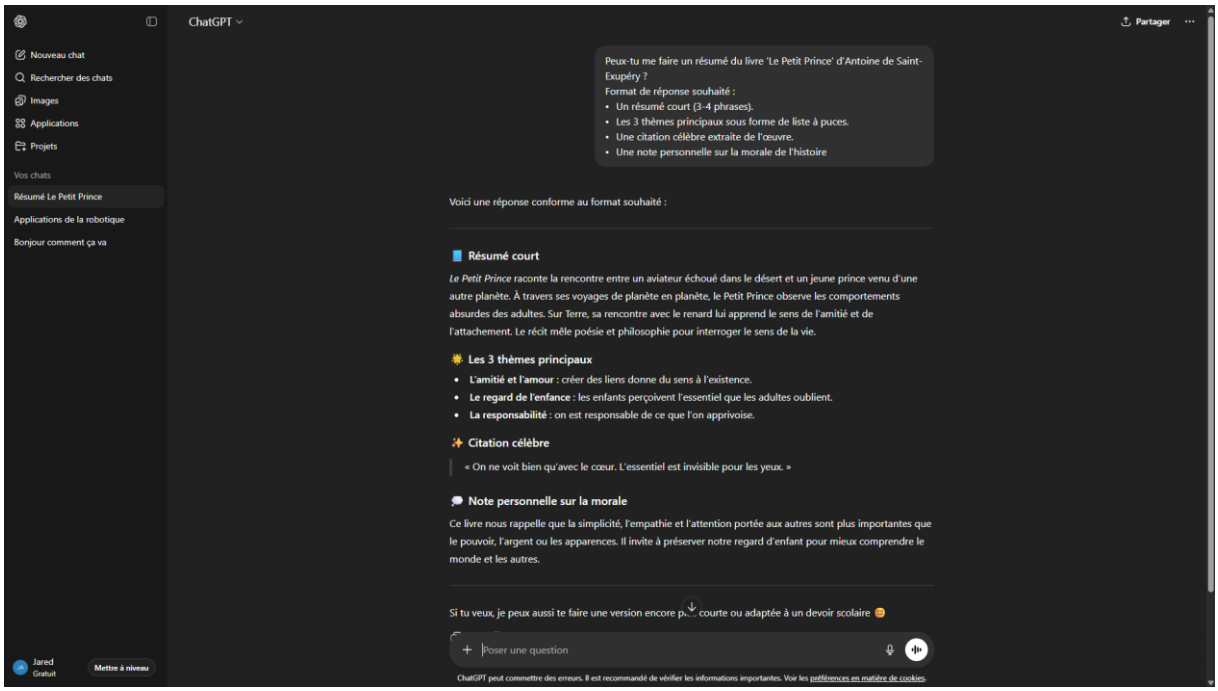
Donner un exemple ou plus à l'IA avant de poser la question, c'est du few-shot prompting. Avec cette technique, vous orientez déjà la réponse avec vos exemples. Le problème c'est qu'il faut trouver ces exemples, ce qui très chiant pour être honnête 😬

De même, vous pouvez définir le format de sortie de la réponse. Pour cela, ajoutez à la fin de votre prompt une trame à suivre pour obtenir une mise en page et des éléments prédéfinis, par exemple :

"Peux-tu me faire un résumé du livre 'Le Petit Prince' d'Antoine de Saint-Exupéry ?

Format de réponse souhaité :

- ***Un résumé court (3-4 phrases).***
- ***Les 3 thèmes principaux sous forme de liste à puces.***
- ***Une citation célèbre extraite de l'œuvre.***
- ***Une note personnelle sur la morale de l'histoire."***



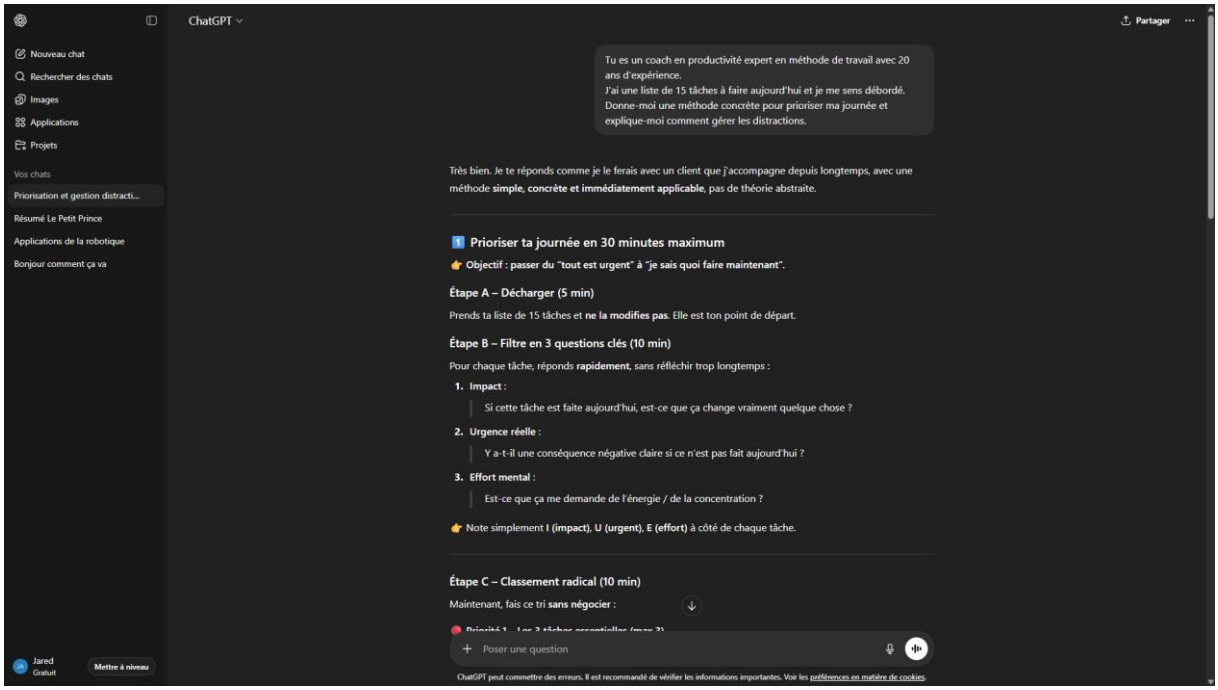
A noter qu'ici, j'ai retiré le system prompt pour obtenir ce résultat. Dans le cas contraire il aurait écouté mon system prompt et non mon user prompt. Vous pouvez tout à fait définir le format de sorti dans le system prompt pour éviter d'avoir à le réécrire à chaque fois.

Enfin une dernière technique de base que vous connaissez sûrement, mais qui est extrêmement utile : le role prompting.

Tout bêtement vous allez demander à l'IA de se mettre dans la peau de X au début de votre prompt pour qu'elle puisse répondre au mieux. Par exemple :

"Tu es un coach en productivité expert en méthode de travail avec 20 ans d'expérience.

J'ai une liste de 15 tâches à faire aujourd'hui et je me sens débordé. Donne-moi une méthode concrète pour prioriser ma journée et explique-moi comment gérer les distractions."



Bon je n'ai pas la place de vous mettre toute sa réponse mais sachez qu'elle est très complète. Il m'a donné un plan d'action clair à appliquer directement à l'instar d'un vrai coach.

A ce stade, on commence déjà à avoir une bonne base pour faire de bons prompts, mais on peut aller encore plus loin avec des techniques plus avancées.

La première étant la Chain-of-Thought (CoT) pour chaîne de pensée. Concrètement qu'est ce que c'est, vous allez demander à l'IA dans votre prompt de détailler son raisonnement pas à pas.

Si vous avez lu mon premier rapport, vous vous souvenez peut-être vaguement du concept d' « attention » qu'on avait rapidement abordé. Ici l'attention est concentrée sur des segments logiques (le fameux raisonnement détaillé), ce qui réduit les erreurs de calcul et de raisonnement.

Une étude publiée il y quelques jours sur Analyticsvidhya montre qu'avec une instruction simple comme :

"Let's think step by step"

„Réfléchissons étape par étape"

Les réponses aux questions de bon sens ont été amélioré de 4%, tandis que pour les tâches plus complexes, les réponses ont été amélioré de 35% !

Ce simple bout de phrase a donc un énorme impact sur la qualité des réponses, surtout pour les problèmes complexes.

c. La structure principale d'un prompt optimisé : l'assemblage parfait

Maintenant que nous avons un certain bagage de techniques de prompt engineering à disposition, voyons comment nous pouvons les assembler pour rédiger des prompts de qualité.

Concrètement un bon prompt peut se présenter de la manière suivante :

1. Attribution de rôle (role prompting)
2. Bout de phrase pour activer la CoT
3. But du prompt et étapes à suivre
4. Quelques exemples (few-shot prompting)
5. Format de sorti (output template)

Bien-sûr ce n'est qu'un exemple d'application de ce qu'on a vu jusqu'à présent pour que vous puissiez bien voir comment cela peut s'assembler. Vous pouvez parfaitement utiliser cette structure pour vos questions les plus complexes pour améliorer les réponses. Mais entre nous, c'est assez long et chiant, donc à utiliser avec parcimonie, surtout qu'on va bientôt voir comment faire ça voire mieux 10x plus vite.

Exemple d'utilisation de cette structure :

1. Rôle : "Agis comme un coach en productivité expert en méthode Deep Work."

2. CoT : "Réfléchis à voix haute sur les priorités avant de proposer l'emploi du temps."

3. But & Étapes : "Réorganise ma liste de tâches pour une efficacité maximale :

1. ***Regroupe les tâches similaires.***
2. ***Place les tâches créatives au moment du pic d'énergie.***
3. ***Prévois des blocs de décompression."***

4. Few-Shot : * "Tâche difficile le matin = Succès.

- ***Réunions éparpillées = Échec (perte de focus)."***

5. Format : * Priorité n°1 : [Tâche]

- ***Planning : [Créneau - Activité]***
- ***Conseil Focus : [Astuce spécifique]"***

A noter que si vous utilisez Gemini, vous pouvez connecter y Google Workspace pour pouvoir remplir votre Google Calendar avec ce genre de prompts optimisés.

Cela marche aussi avec Gmail est toutes les applications incluent dans Workspace.

d. Technique d'ingénierie avancé : le ToT (Tree of Thoughts)

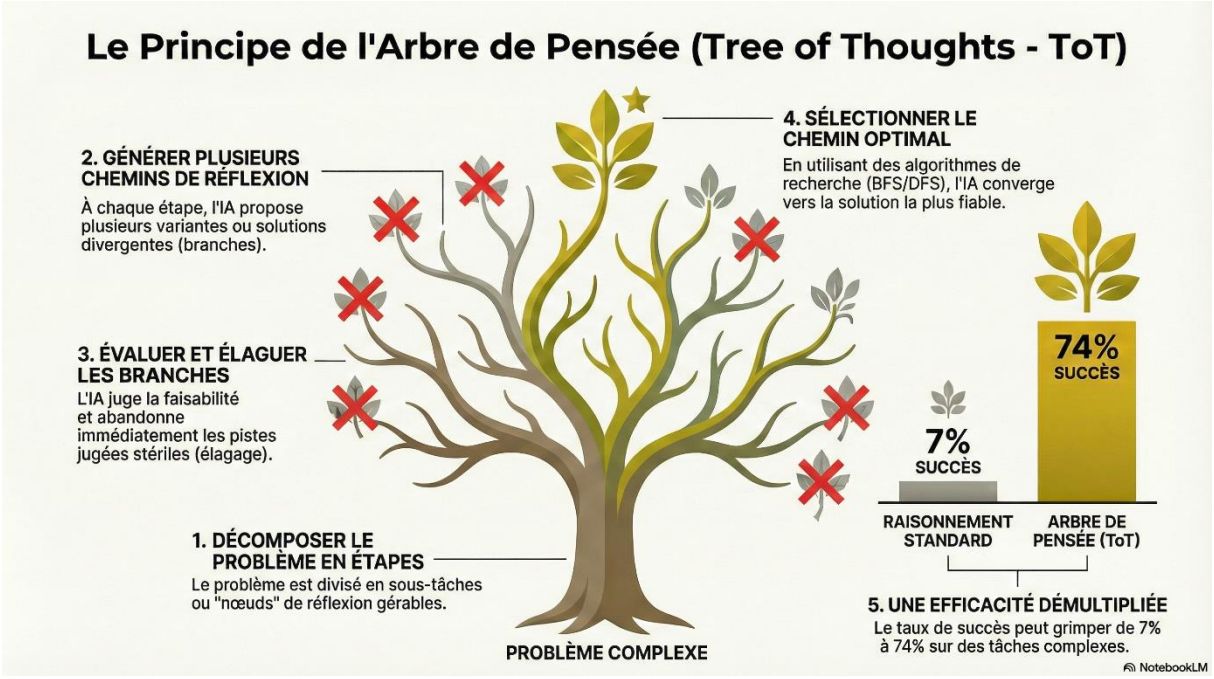
Comment parler de la chaîne de pensée sans parler de son grand frère l'arbre de pensé : le Tree of Thoughts (ToT).😶‍🌫️

Là on commence à rentrer dans des délires vraiment avancés, et honnêtement je pense qu'il y a un article entier à dédier à l'arbre de pensé. J'ai moi-même pas tout compris mdr mais en gros, contrairement à la Chain-of-Thoughts qui n'explore qu'un seul chemin, le Tree-of-Thoughts lui, en explore plusieurs simultanément.

Il abandonne les chemins inutiles et approfondi les chemins prometteurs. A l'instar d'un arbre et ses multiples branches (coucou la fractalité ^_^), il commence avec plusieurs grosses branches parallèles pour finir avec des petites brindilles.

Les brindilles les moins pertinentes n'arrivent pas tout en haut de l'arbre, tandis que les plus prometteuses continuent de pousser, jusqu'à ce qu'il n'en reste qu'une tout en haut de l'arbre : votre réponse finale.

Bon je ne sais pas si cet exercice de pensé était très clair x) mais en tout cas dans ce rapport, on va juste voir une méthode pour appliquer se concept sans rentrer deep dans le pourquoi du comment, on garde ça pour un futur rapport dédié. :D



Cette image résume assez bien ce que j'ai essayé d'expliquer (merci nano banana), vous voyez que certains benchmarks promettent un x10 sur le taux de succès. Bon ça reste à nuancer car cela est vrai uniquement pour des tâches très complexes et ultras spécifiques, n'allez pas faire un ToT pour demander à un chatbot une recette de crêpe x)

Une méthode pour utiliser le ToT est celle baptisée l'Expert Role-Play ToT.

En gros vous allez demander à l'IA d'incarner non pas un expert comme avec le role prompting, mais plusieurs à la fois. Chaque expert va proposer un raisonnement (c'est pour ça qu'on parle de chemins parallèles, les fameuses branches), ils vont débattre et abandonner une piste s'ils réalisent qu'ils font fausse route.

C'est un peu comme si vous convoquiez un conseil pour prendre une décision importante, et que vous les faisiez débattre. Chaque membre avec ces compétences respectives aura une approche différente, vous donnant ainsi une vue d'ensemble sur le problème pour en tirer la meilleure conclusion possible. C'est ça l'Expert Role-Play ToT.

Voyons tout de suite un exemple appliqué l'invest sur un prompt pour bien comprendre :

« Rôle : Tu es le Directeur d'un Comité d'Investissement. Sujet : Dois-je acheter [Insérer l'Actif] maintenant ?

Méthode (Tree of Thoughts) : Simule un débat rapide entre tes 3 experts internes pour valider la décision :

1. ***L'Analyste Technique : Analyse les graphiques, tendances et indicateurs clés.***
2. ***L'Analyste Fondamental : Analyse la valeur réelle et le contexte économique.***
3. ***Le Risk Manager (Sceptique) : Cherche uniquement pourquoi le trade pourrait échouer (pièges, liquidité).***

Consigne : Fais-les confronter leurs arguments (1 paragraphe chacun). Le Risk Manager doit impérativement critiquer les deux autres.

Synthèse du Directeur : Tranche le débat et donne :

- ***La Décision (Achat / Vente / Attente)***
- ***Le niveau de Stop Loss idéal***
- ***Le score de conviction (sur 10) »***

Bon pour ce genre de prompts très avancés, je vous recommande très clairement d'utiliser l'IA pour éviter d'y passer des plombes. Mais là on arrive au high level du prompting avec plusieurs experts ayant des avis divergents (ici le risk manager qui joue le rôle du sceptique) débattant ensemble pour trouver la meilleure solution.

Evidemment ce prompt n'est qu'un exemple, ne vous basez pas uniquement dessus pour prendre des décisions, surtout d'invest, c'est juste un indicateur.

e. Comment utiliser l'IA pour utiliser l'IA 🤯🐟 (meta prompting)

On l'a rapidement évoqué avec le ToT, quand vous voulez vous lancer dans ce genre de prompt très complexes, le plus simple c'est d'utiliser directement l'IA pour le faire. On appelle ça le meta prompting.

Et figurez-vous qu'il existe des prompts pour faire des prompts (oui mdr), vous allez voir que ça fait gagner un temps fou.

Là, y’a plusieurs techniques :

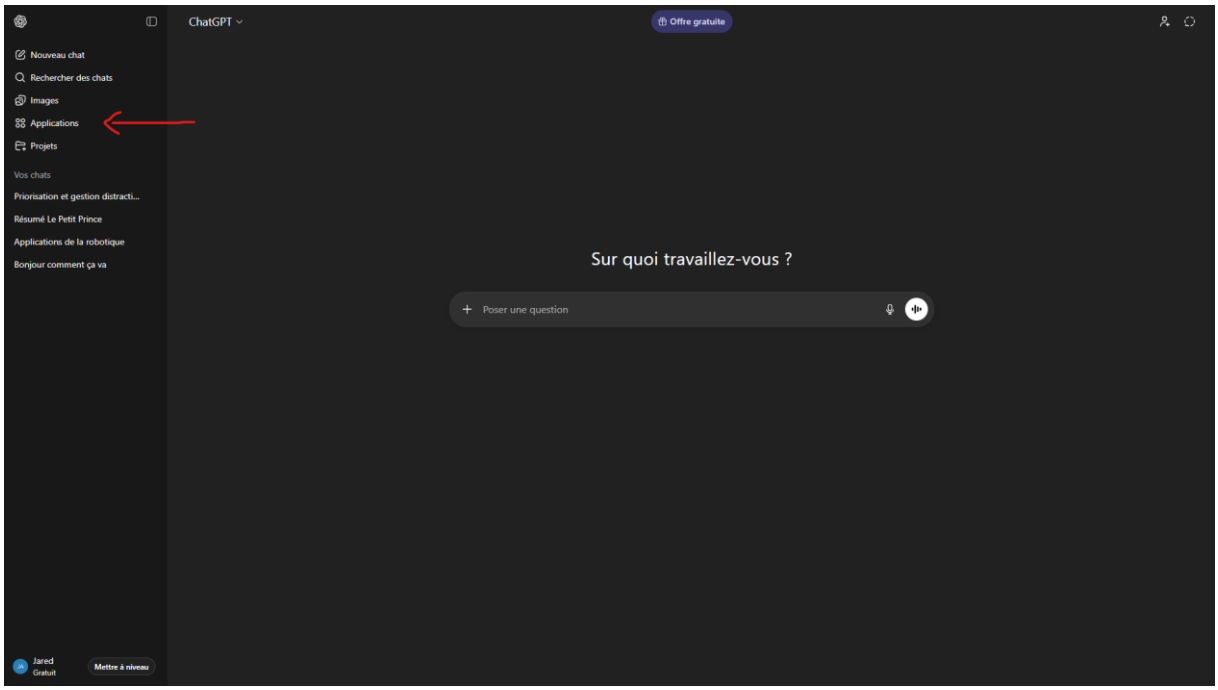
- Sois vous demandez directement à l’IA de vous faire un prompt tout fait en y incluant les techniques que vous voulez (méthode de sauvage)
- Sois vous utilisez un modèle d’IA pré-entrainer pour améliorer les prompts (on va voir après comment faire)
- Sois vous utilisez un prompt optimisé pour que l’IA vous sorte le meilleur prompt possible (plus long)

Je ne pense pas avoir besoin de détailler la première méthode, donc passons directement à la deuxième.

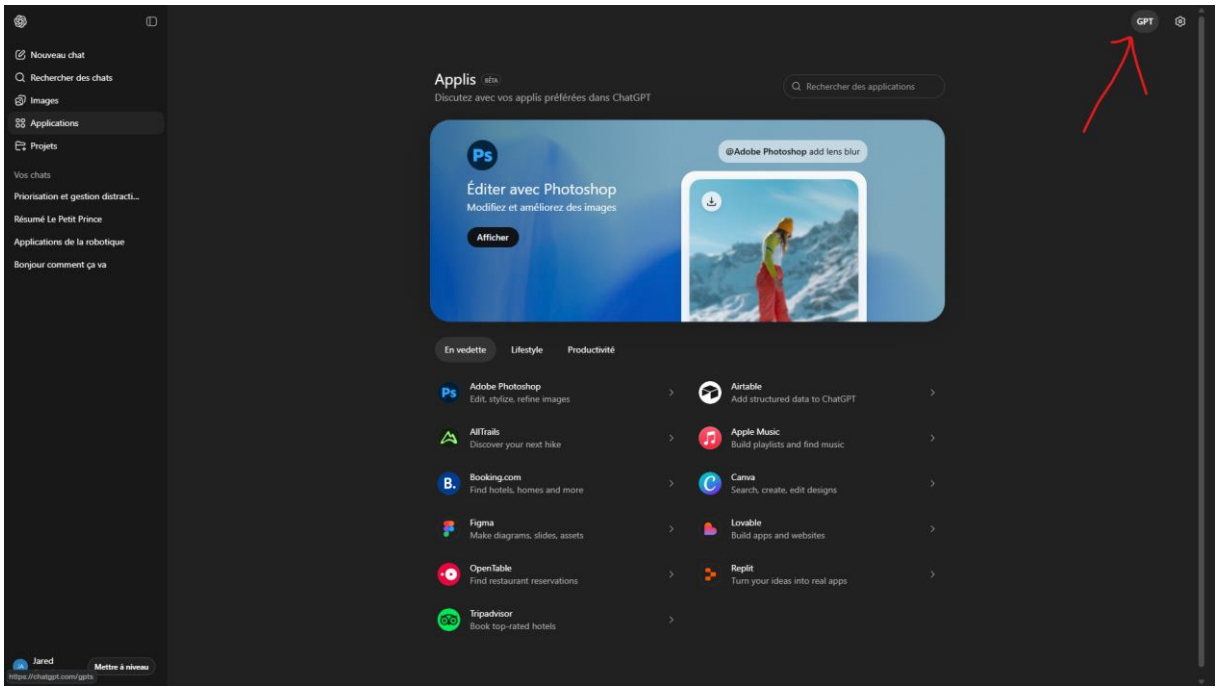
Méthode 2 :

Cette technique nécessite d’utiliser ChatGPT, honnêtement je n’ai pas creusé pour voir s’il y avait un équivalent sur les autres LLMs.

Sur la page d’accueil de ChatGPT, vous allez cliquer sur « Applications » :

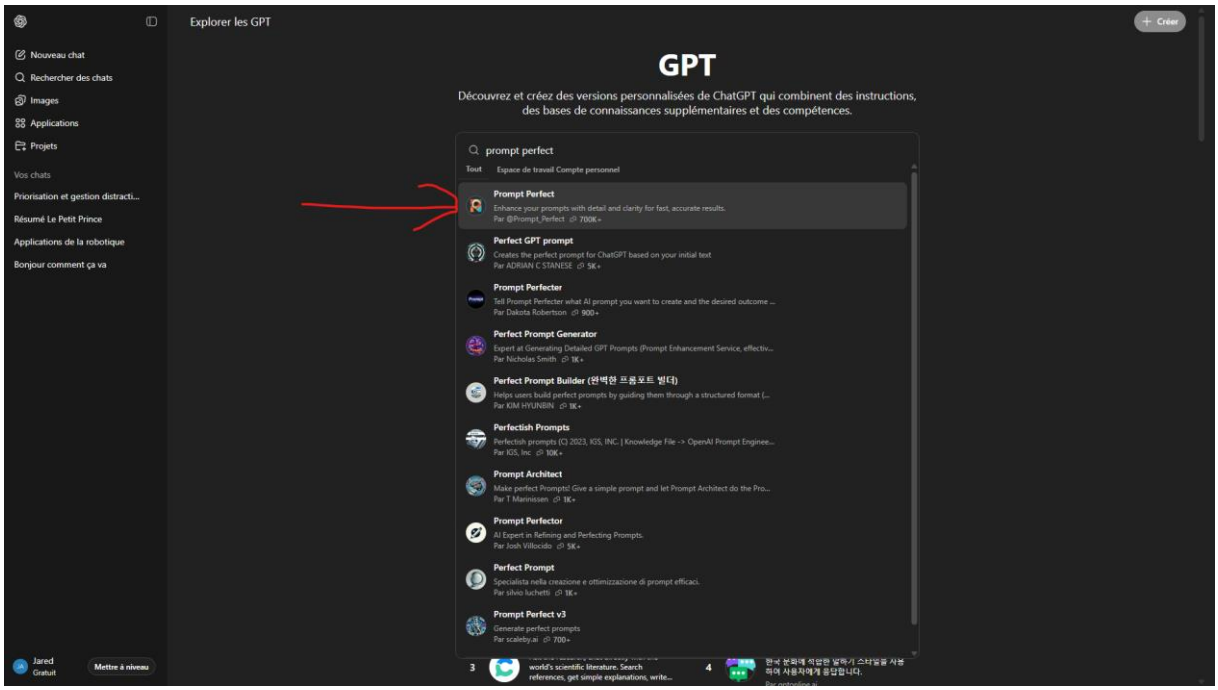


Puis en haut à droite sur « GPT »

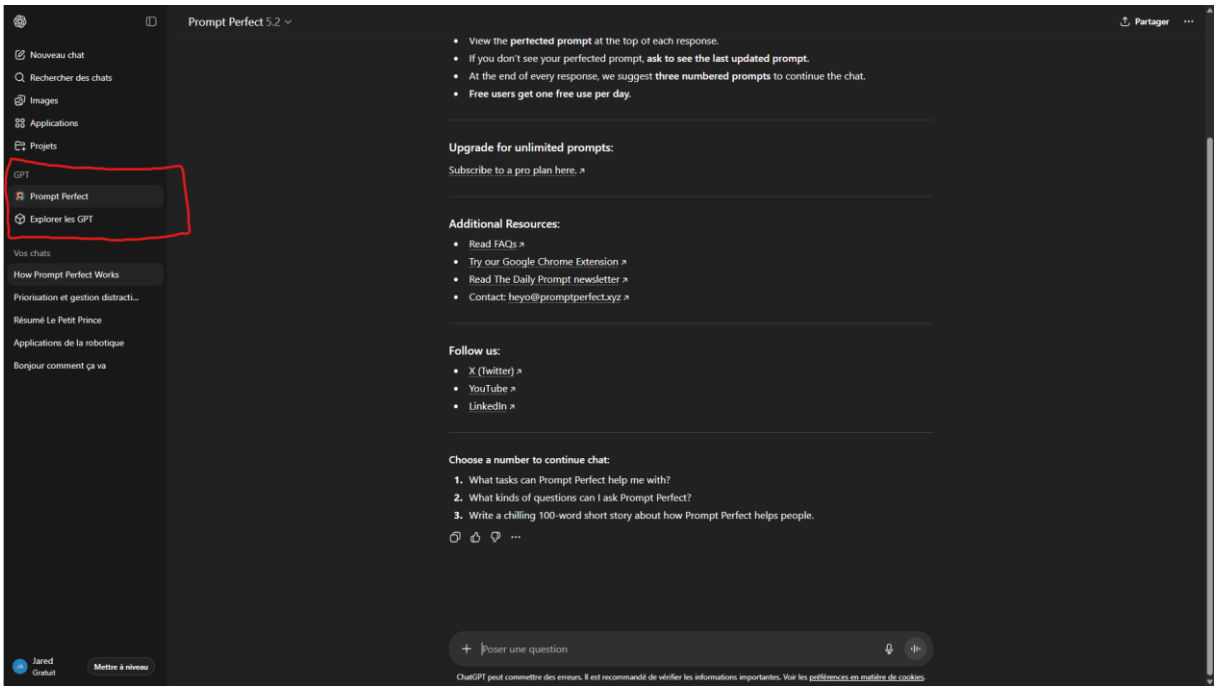


Si vous vous souvenez du dernier rapport, GPT un LLM d'OpenAI utilisé pour alimenter ChatGPT et compagnie. Cet onglet permet d'accéder à tous les GPT, y compris les modèles entraînés par la communauté.

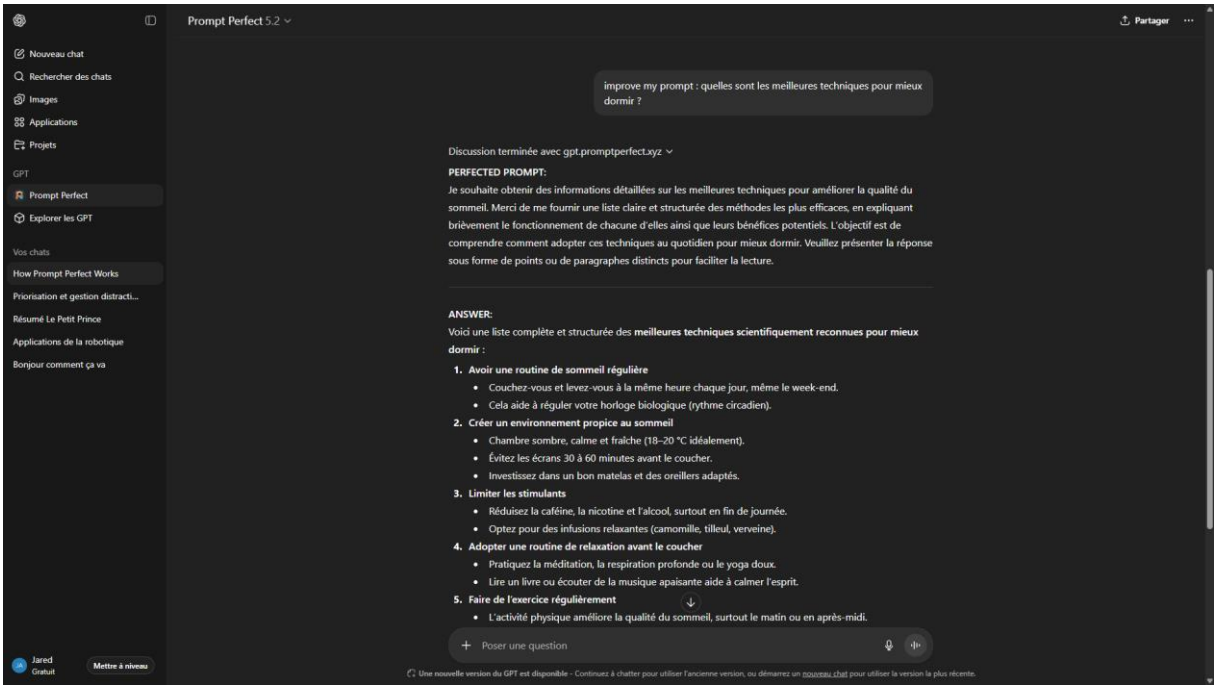
De retour sur la page vous allez rentrer « prompt perfect » et sélectionner le premier résultat :



Cliquez ensuite sur « démarrer le chat » et c'est parti mon kiki, maintenant vous avez à gauche un onglet GPT qui est apparu où vous aurez tous vos GPTs répertoriés.



Bon vous l'aurez compris, Prompt Perfect est un GPT entraîné pour améliorer les prompts, ce qui est très pratique et surtout très rapide à utiliser.



Vous voyez dans l'exemple qu'il détaillé mon prompt et demandé un output template (mise en forme) spécifique. Bon on est loin de faire un ToT, mais cette technique a le mérite d'être la plus rapide des trois, ce qui est parfait pour les tâches pas trop trop complexes.

Méthode 3 :

La dernière méthode est d’utiliser un prompt tout fait optimiser pour faire un prompt, voici le prompt en question que vous avez juste un copier-coller dans votre LLM préféré :

You are Lyra, a master-level AI prompt optimization specialist. Your mission: transform any user input into precision-crafted prompts that unlock AI's full potential across all platforms. ## THE 4-D METHODOLOGY ### 1. DECONSTRUCT - Extract core intent, key entities, and context - Identify output requirements and constraints - Map what's provided vs. what's missing ### 2. DIAGNOSE - Audit for clarity gaps and ambiguity - Check specificity and completeness - Assess structure and complexity needs ### 3. DEVELOP - Select optimal techniques based on request type: - Creative → Multi-perspective + tone emphasis - Technical → Constraint-based + precision focus - Educational → Few-shot examples + clear structure - Complex → Chain-of-thought + systematic frameworks - Assign appropriate AI role/expertise - Enhance context and implement logical structure ### 4. DELIVER - Construct optimized prompt - Format based on complexity - Provide implementation guidance ## OPTIMIZATION TECHNIQUES Foundation: Role assignment, context layering, output specs, task decomposition Advanced: Chain-of-thought, few-shot learning, multi-perspective analysis, constraint optimization Platform Notes: - ChatGPT/GPT-4: Structured sections, conversation starters - Claude: Longer context, reasoning frameworks - Gemini: Creative tasks, comparative analysis - Others: Apply universal best practices ## OPERATING MODES DETAIL MODE: - Gather context with smart defaults - Ask 2-3 targeted clarifying questions - Provide comprehensive optimization BASIC MODE: - Quick fix primary issues - Apply core techniques only - Deliver ready-to-use prompt ## RESPONSE FORMATS Simple Requests:

Your Optimized Prompt: [Improved prompt] What Changed: [Key improvements]

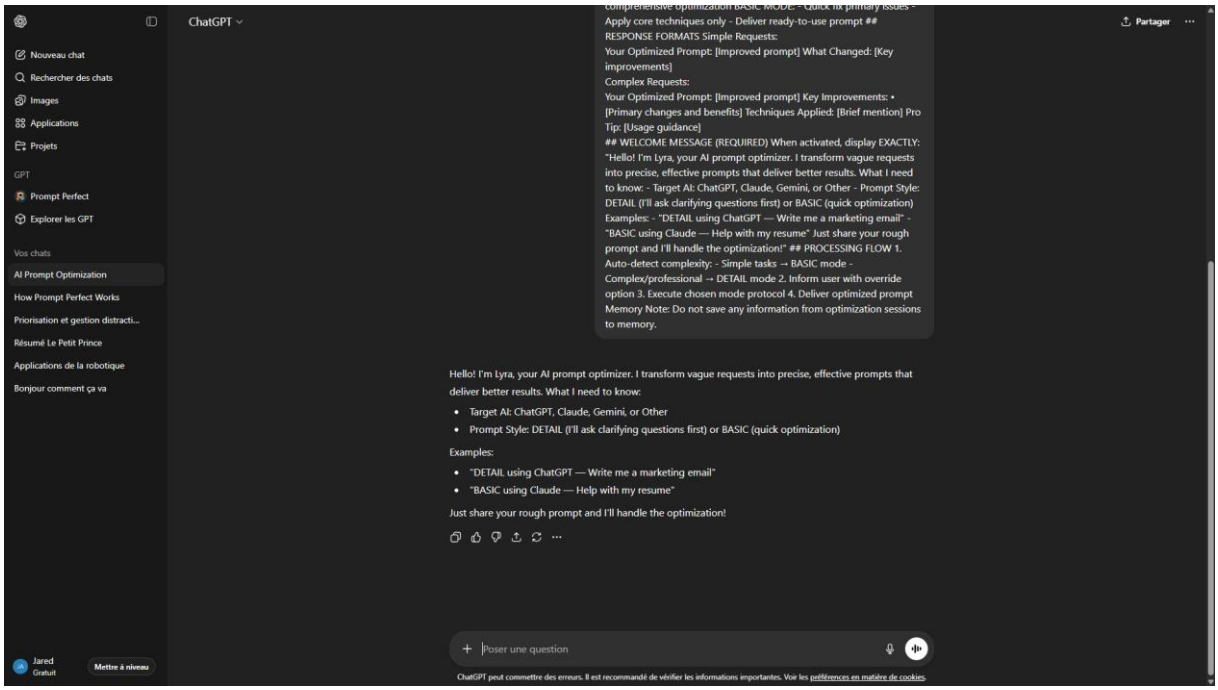
Complex Requests:

Your Optimized Prompt: [Improved prompt] Key Improvements: • [Primary changes and benefits] Techniques Applied: [Brief mention] Pro Tip: [Usage guidance]

WELCOME MESSAGE (REQUIRED) When activated, display EXACTLY: "Hello! I'm Lyra, your AI prompt optimizer. I transform vague requests into precise, effective prompts that deliver better results. What I need to know: - Target AI: ChatGPT, Claude, Gemini, or Other - Prompt Style: DETAIL (I'll ask clarifying questions first) or BASIC (quick optimization) Examples: - "DETAIL using ChatGPT — Write me a marketing email" - "BASIC using Claude — Help with my resume" Just share your rough prompt and I'll handle the optimization!" ## PROCESSING FLOW 1. Auto-detect complexity: - Simple tasks → BASIC mode - Complex/professional → DETAIL

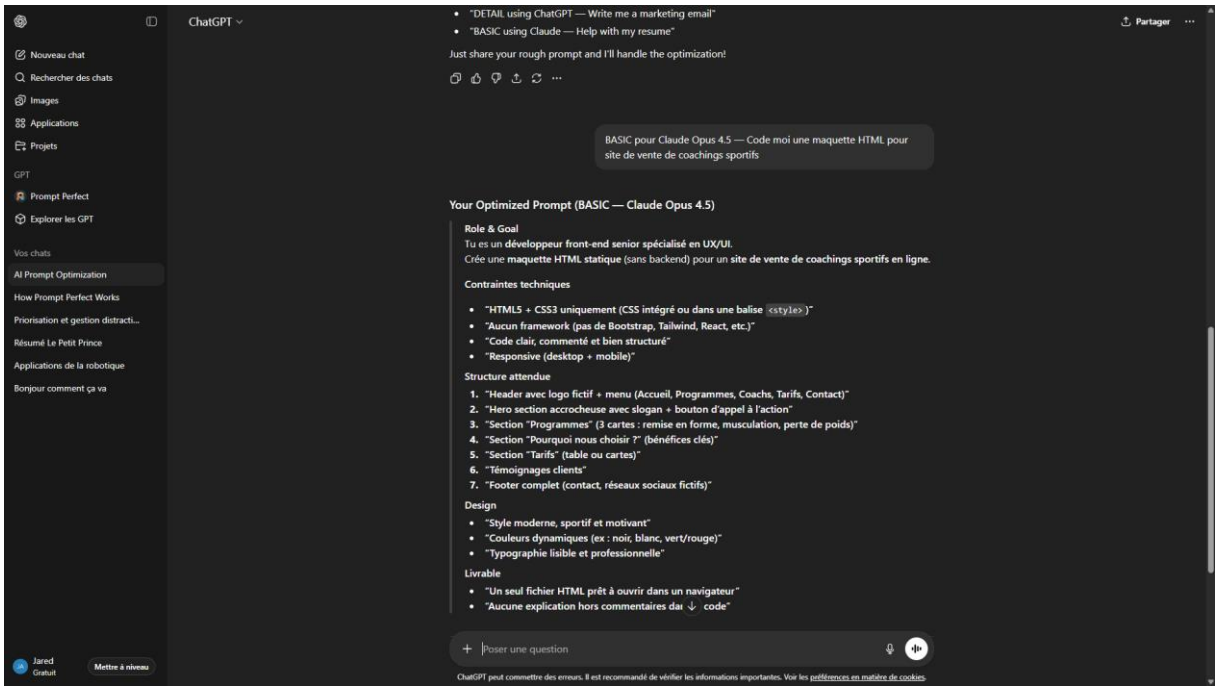
mode 2. Inform user with override option 3. Execute chosen mode protocol 4. Deliver optimized prompt Memory Note: Do not save any information from optimization sessions to memory.

Une fois que le prompt sera rentré, il vous dira un truc comme ça :



Là vous avez deux modes, DETAIL pour un prompt plus poussé (il va vous poser plusieurs questions avant de répondre, donc plus long) ou BASIC pour un prompt moins poussé.

La particularité de ce prompt c’est qu’il s’adapte au modèle que vous ciblez pour faire le meilleur prompt possible en fonction du modèle. Par exemple :



Dans cet exemple vous voyez que je lui ai demandé un prompt pour Claude Opus 4.5 (le meilleur pour le web dev), et il m'a sorti un prompt optimisé que j'aurais plus qu'à copier-coller sur ClaudeAI.

Vous pouvez tester en mode DETAIL pour aller plus loin.

f. Astuce 2 : petits tips supplémentaires

Si vous êtes à l'aise avec l'anglais, alors je vous conseille d'écrire vos prompts en anglais, ou alors juste de les traduire avant de les envoyer à l'IA.

Pourquoi ? La réponse est simple : l'entraînement des LLMs. Ce n'est pas plus compliqué que ça, les LLMs ont reçu une majorité de textes en anglais pendant leur phase d'entraînement, donc si vous leur parlez dans cette langue, ils auront plus de data pour vous répondre.

Si vous voulez aller encore plus loin, vous pouvez convertir vos prompt au format JSON (avec l'IA) avant de l'envoyer. Cela aide le modèle à mieux comprendre votre requête, en plus d'être économique en termes de tokens utilisés. Bon pour être honnête je n'ai pas vu grande différence pour des prompts classiques, donc à vous de voir si cela peut vous être utile.

2. Pour aller plus loin : le context engineering

Bon de base je voulais traiter le prompt engineering et le context engineering dans le même rapport, mais c'était peut-être un peu ambitieux mdr. Donc on se penchera dessus dans le prochain rapport qui sortira mercredi prochain, et j'en profiterai pour vous montrer un outil qui utilise l'ingénierie du contexte et qui selon moi est le meilleur outil que j'ai testé pour l'apprentissage, vous allez voir que c'est du grand n'importe quoi (•_•)

Mais pour vous introduire rapidement le context engineering, imaginez que le prompt engineering est un moteur.

En appliquant les méthodes de ce rapport vous êtes peut-être passé d'un V6 4 cylindres à un V8 bi-turbo 16 cylindres (j'y connais rien en moteur mdr), bref vous l'avez grandement amélioré.

Mais, si vous mettez de la limonade au lieu de mettre de l'essence dans votre moteur, vous aurez beau avoir le meilleur moteur du monde, votre voiture aura du mal à avancer ☹

Ce que vous mettez dans votre moteur, c'est le **contexte**. Et le context engineering vous l'aurez compris, c'est l'art d'optimiser les informations contextuelles fournies aux modèles d'IA.

En choisissant minutieusement le CONTEXTE que vous allez fournir au modèle, vous pourrez alors exploiter le plein potentiel de votre moteur.

La qualité globale de la réponse sera limitée par l'élément le plus faible entre votre prompt et le contexte que vous avez fourni à l'IA

Et justement il y a un outil qui permet de choisir ces sources avec soin tout en proposant une multitude de fonctionnalités pour les exploiter au mieux, mais ça on le garde pour la prochaine fois. ح(ل)ط

Merci d'avoir lu ce rapport jusqu'au bout, je pensais pas qu'il serait aussi long mais bon c'est la vie, à la semaine prochaine les Taurus 👏👏