

Parcours Makers - Module 1 - Prompt engineering avancé

Cours 1 : Implémenter un raisonnement Chain-of-Thought pas à pas avec Self-Refine

Ce notebook vous guide étape par étape pour structurer un raisonnement explicite en Python avec un LLM (modèle de langage). Vous apprendrez à :

- Construire un prompt clair qui guide le raisonnement du modèle
- Récupérer une réponse structurée
- Déclencher une relecture automatique (Self-Refine) pour corriger et améliorer la réponse
- Comparer les deux versions selon des critères simples

✓ 1. Préparation de l'environnement

Commencez par installer les dépendances nécessaires (OpenAI, gestion de variables d'environnement, encodage).

```
# 📄 Exécutez cette cellule une seule fois
# exécutez une seule fois
!pip install -q \
    openai==1.96.1 \
    python-dotenv==1.1.1 \
    tiktoken==0.9.0
```

757.5/757.5 kB 12.2 MB/s eta 0:00:00

```
import os
from getpass import getpass
from dotenv import load_dotenv
```

```
load_dotenv() # charge un éventuel .env
if "OPENAI_API_KEY" not in os.environ: # invite si la variable manque
    os.environ["OPENAI_API_KEY"] = getpass("🔑 Entrez votre clé OpenAI : ")
```

🔑 Entrez votre clé OpenAI :

```
from openai import OpenAI
client = OpenAI() # le client utilisera directement la variable d'environnement
```

✓ 2. Créer un prompt Chain-of-Thought

Nous composons un prompt en trois parties :

- Rôle : précise la posture du modèle.
- Consigne : détaille la tâche et impose un raisonnement explicite.
- Format : indique comment structurer les étapes et la conclusion.

```
def create_cot_prompt(question: str) -> str:
    """Construit un prompt explicite en trois sections."""
    return f"""
    Tu es un consultant expert en organisation.
    Ta tâche : répondre à la question ci-dessous en détaillant ton raisonnement étape par étape.
    Pour chaque étape, commence par 'Étape :' puis termine par 'Conclusion :' sur une ligne à part.

    Question : {question}
    """,strip()
```

✓ 3. Générer une première réponse (version 1)

Le modèle est appelé avec une température basse pour privilégier la cohérence.

```
def ask_openai(prompt: str, model="gpt-4o-mini", temperature=0.3):
    response = client.chat.completions.create(
        model=model,
        temperature=temperature,
        messages=[{"role": "user", "content": prompt}],
    )
    return response.choices[0].message.content

# Exemple : question métier
question = (
    "Comment estimer le budget annuel nécessaire pour déployer un CRM "
    "dans une PME de 100 utilisateurs, en tenant compte des licences, "
    "de la formation et de la maintenance ?"
)

prompt_v1 = create_cot_prompt(question)
answer_v1 = ask_openai(prompt_v1)
print("◆ Réponse initiale (v1)\n")
print(answer_v1)
```

↩ ◆ Réponse initiale (v1)

Étape : Identifier les besoins spécifiques de la PME en matière de CRM. Cela inclut la définition des fonctionnalités re

Conclusion :

Étape : Rechercher les différentes options de CRM disponibles sur le marché. Il existe plusieurs types de CRM, allant de

Conclusion :

Étape : Estimer le coût des licences. Pour une PME de 100 utilisateurs, il faut multiplier le coût par utilisateur par l

Conclusion :

Étape : Évaluer les coûts de formation. La formation est essentielle pour garantir que les utilisateurs adoptent efficac

Conclusion :

Étape : Prendre en compte les coûts de maintenance et de support. Cela peut inclure les mises à jour logicielles, le sup

Conclusion :

Étape : Additionner tous les coûts estimés pour obtenir le budget total. Cela inclut les licences, la formation et la ma

Conclusion :

Étape : Prévoir une marge pour les imprévus. Il est sage d'ajouter un pourcentage (environ 10 %) pour couvrir les coûts

Conclusion :

Étape : Finaliser le budget en intégrant toutes les estimations et la marge pour imprévus. En reprenant l'exemple précéd

Conclusion :

En résumé, pour estimer le budget annuel nécessaire pour déployer un CRM dans une PME de 100 utilisateurs, il faut prend

✓ 4. Créer un prompt Self-Refine

Le modèle relit sa propre réponse, détecte les incohérences ou lacunes, puis produit une version révisée. Nous conservons la même question et insérons la réponse v1 dans un nouveau prompt critique.

```
def create_self_refine_prompt(question: str, first_answer: str) -> str:
    return f"""
Tu es le même consultant, mais tu dois maintenant relire ta réponse.
Identifie toute erreur, toute ambiguïté ou tout manque de précision,
puis propose une version améliorée. Utilise le format :

Analyse :
- (liste de points à corriger)

Réponse révisée :
- Étape 1 :
- Étape 2 :
...
- Conclusion :

Question : {question}
```

Ta réponse initiale :

```
{first_answer}
"".strip()
```

```
prompt_v2 = create_self_refine_prompt(question, answer_v1)
answer_v2 = ask_openai(prompt_v2)
print("◆ Réponse révisée (v2)\n")
print(answer_v2)
```

↩ ◆ Réponse révisée (v2)

Analyse :

- La réponse initiale manque de structure et de clarté, car chaque étape est suivie d'une "conclusion" qui n'apporte pas
- Certaines étapes pourraient être regroupées pour éviter des répétitions inutiles.
- Les exemples chiffrés sont utiles, mais il serait préférable de les présenter de manière plus cohérente.
- Il manque une introduction qui contextualise la réponse.
- La conclusion finale pourrait être plus synthétique et explicative.

Réponse révisée :

- Introduction : Pour estimer le budget annuel nécessaire à la mise en place d'un CRM dans une PME de 100 utilisateurs,
- Étape 1 : Identifier les besoins spécifiques de la PME en matière de CRM. Cela inclut la définition des fonctionnalités
- Étape 2 : Rechercher les différentes options de CRM disponibles sur le marché. Il existe plusieurs types de CRM, allant
- Étape 3 : Estimer le coût des licences. Pour une PME de 100 utilisateurs, il faut multiplier le coût par utilisateur par
- Étape 4 : Évaluer les coûts de formation. La formation est essentielle pour garantir que les utilisateurs adoptent eff
- Étape 5 : Prendre en compte les coûts de maintenance et de support. Cela peut inclure les mises à jour logicielles, le
- Étape 6 : Additionner tous les coûts estimés pour obtenir le budget total. En reprenant les estimations précédentes, l
 - Licences : 36 000 €
 - Formation : 3 000 € (moyenne)
 - Maintenance : 7 200 €
 - Budget total avant imprévus : 46 200 €.
- Étape 7 : Prévoir une marge pour les imprévus. Il est sage d'ajouter un pourcentage (environ 10 %) pour couvrir les co
- Conclusion : En intégrant toutes les estimations et la marge pour imprévus, le budget final pour déployer un CRM dans

✓ 5. Comparer les deux versions

Nous effectuons une comparaison simple : longueur (tokens), nombre d'étapes et clarté perçue. Vous pouvez compléter avec vos propres critères.

```
import tiktoken, textwrap, re

enc = tiktoken.encoding_for_model("gpt-4o-mini")

def count_tokens(text):
    return len(enc.encode(text))

def count_steps(text):
    return len(re.findall(r"Étape\s*\d*", text, flags=re.IGNORECASE))

def summary_stats(label, text):
    print(f"{label}\nTokens : {count_tokens(text)} | Étapes : {count_steps(text)}\n")

summary_stats("Version 1", answer_v1)
summary_stats("Version 2", answer_v2)

print("-"*60)
print("A vous d'évaluer maintenant la clarté et la cohérence de chaque version.")
```


↩ Version 1
Tokens : 625 | Étapes : 8

Version 2
Tokens : 787 | Étapes : 9

A vous d'évaluer maintenant la clarté et la cohérence de chaque version.

✓ 6. Activité guidée

1. Dans la cellule suivante, remplacez la variable question par votre propre question métier.
2. Réexécutez l'ensemble des cellules depuis la création du prompt pour observer le raisonnement et la version Self-Refine.
3. Notez les écarts entre v1 et v2 :
 - Les étapes sont-elles plus détaillées ?
 - La conclusion est-elle plus précise ?
 - La structure respecte-t-elle le format attendu ?

 Personnalisez ici votre question, puis relancez tout
question = "Définis un plan de migration vers le cloud pour une startup SaaS de 50 personnes."

```
prompt_v1 = create_cot_prompt(question)
answer_v1 = ask_openai(prompt_v1)
print("◆ Réponse initiale (v1)\n")
print(answer_v1)
```

◆ Réponse initiale (v1)

Étape 1 : Évaluation des besoins et des objectifs

Dans cette première étape, il est crucial de comprendre les besoins spécifiques de la startup SaaS. Il faut identifier l

Conclusion : Une évaluation approfondie des besoins et des objectifs permettra de définir une stratégie de migration ada

Étape 2 : Sélection du modèle de cloud

Il existe plusieurs modèles de cloud (public, privé, hybride). Il est essentiel de choisir celui qui correspond le mieux

Conclusion : Le choix du modèle de cloud influencera les coûts, la sécurité et la flexibilité de la solution choisie.

Étape 3 : Choix des fournisseurs de services cloud

Il est important de comparer les différents fournisseurs de services cloud (AWS, Azure, Google Cloud, etc.) en fonction

Conclusion : Sélectionner le bon fournisseur de services cloud est essentiel pour garantir une migration réussie et une

Étape 4 : Élaboration d'un plan de migration

Un plan de migration détaillé doit être élaboré, incluant un calendrier, les ressources nécessaires, et les étapes de la

Conclusion : Un plan de migration bien structuré facilitera le déroulement de la migration et minimisera les risques de

Étape 5 : Mise en place de la sécurité et de la conformité

Avant de migrer, il est crucial de mettre en place des mesures de sécurité adéquates pour protéger les données sensibles

Conclusion : La sécurité et la conformité doivent être intégrées dès le début pour éviter des problèmes juridiques et de

Étape 6 : Migration et tests

La migration doit être effectuée par étapes, en commençant par les applications moins critiques pour tester le processus

Conclusion : Une migration progressive et des tests rigoureux garantiront que les applications fonctionnent comme prévu

Étape 7 : Formation et adoption par les utilisateurs

Une fois la migration terminée, il est essentiel de former les employés à l'utilisation des nouvelles solutions cloud. C

Conclusion : La formation des utilisateurs est cruciale pour assurer une adoption réussie des nouvelles technologies et

Étape 8 : Suivi et optimisation

Après la migration, il est important de surveiller les performances des applications dans le cloud et d'optimiser les re

Conclusion : Un suivi régulier et des optimisations permettront de tirer pleinement parti des avantages du cloud et d'as

En résumé, un plan de migration vers le cloud pour une startup SaaS doit être soigneusement élaboré et exécuté en plusie

```
prompt_v2 = create_self_refine_prompt(question, answer_v1)
answer_v2 = ask_openai(prompt_v2)
print("◆ Réponse révisée (v2)\n")
print(answer_v2)
```

◆ Réponse révisée (v2)

Analyse :

- La réponse initiale présente une structure claire, mais elle manque de précision dans certains domaines, notamment sur
- Les conclusions à la fin de chaque étape sont redondantes et pourraient être consolidées pour éviter la répétition.
- Certaines étapes pourraient bénéficier d'exemples concrets ou de recommandations spécifiques pour renforcer la pertinence.
- L'ordre des étapes pourrait être légèrement ajusté pour mieux refléter le processus de migration.

Réponse révisée :

- Étape 1 : Évaluation des besoins et des objectifs
 - Identifier les applications et services à migrer.
 - Définir les objectifs de la migration (réduction des coûts, scalabilité, sécurité, etc.).
 - Réaliser une analyse des performances actuelles et des limitations des infrastructures existantes.
- Étape 2 : Sélection du modèle de cloud
 - Évaluer les différents modèles de cloud (public, privé, hybride) en fonction des besoins de la startup.

- Considérer des facteurs comme le coût, la sécurité des données et les exigences de conformité.
- Étape 3 : Choix des fournisseurs de services cloud
 - Comparer les fournisseurs (AWS, Azure, Google Cloud, etc.) sur des critères tels que le coût, les fonctionnalités, l
 - Considérer des études de cas ou des témoignages d'autres startups pour guider le choix.
- Étape 4 : Élaboration d'un plan de migration
 - Créer un plan détaillé incluant un calendrier, les ressources nécessaires, et les étapes de migration (préparation,
 - Définir clairement les rôles et responsabilités de chaque membre de l'équipe.
- Étape 5 : Mise en place de la sécurité et de la conformité
 - Établir des mesures de sécurité adéquates (chiffrement des données, gestion des accès, etc.) avant la migration.
 - S'assurer que toutes les réglementations pertinentes (comme le RGPD) sont respectées dès le début.
- Étape 6 : Migration et tests
 - Effectuer la migration par étapes, en commençant par des applications moins critiques.
 - Réaliser des tests après chaque étape pour garantir le bon fonctionnement des applications dans le nouvel environnem
- Étape 7 : Formation et adoption par les utilisateurs
 - Organiser des sessions de formation pour les employés sur les nouvelles solutions cloud.
 - Fournir des ressources telles que des guides d'utilisation et un support technique pour faciliter l'adoption.
- Étape 8 : Suivi et optimisation
 - Mettre en place un système de surveillance des performances des applications dans le cloud.
 - Ajuster les configurations et gérer les coûts en fonction des besoins réels pour optimiser l'utilisation des ressour

Conclusion : Un plan de migration vers le cloud pour une startup SaaS doit être soigneusement élaboré et exécuté en plus

```
summary_stats("Version 1", answer_v1)
summary_stats("Version 2", answer_v2)
```

```
↔ Version 1
  Tokens : 807 | Étapes : 13

Version 2
  Tokens : 684 | Étapes : 15
```

✓ Pour aller plus loin

- Testez la même boucle avec différents modèles (gpt-4o, o3) et comparez coûts et qualités.
- Expérimentez avec des paramètres temperature et top_p plus élevés pour observer l'impact sur la créativité et la longueur des réponses.
- Ajoutez un compteur de tentatives Self-Refine pour arrêter la boucle si le modèle n'améliore plus son texte.
- Intégrer un score de qualité : demander au modèle de s'autoévaluer, ou intégrer une métrique externe pour noter la qualité de chaque réponse. Cela permet d'industrialiser la boucle Self-Refine.

✓ Conclusion

Vous disposez maintenant d'un socle simple pour générer un raisonnement pas à pas et le faire relire par le modèle lui-même. Cette technique améliore la transparence et la fiabilité des réponses, tout en restant accessible à un public non technique. Adaptez-la à vos propres cas d'usage pour renforcer la qualité des livrables générés par un LLM.