Prompt Engineering ou Comment discuter avec les modeles de langue derriere les plateformes comme ChatGpt

Les plateformes telles que **ChatGPT, Gemini, DeepSeek** et d'autres utilisent des modèles LLM (*Large Language Models* ou *modèles de langue à grande échelle*). Ce sont des algorithmes entraînés sur des milliards de données variées disponibles sur Internet.

Ces LLM comprennent le langage humain, mais pour les interroger efficacement, il faut utiliser un **prompt** (ou langage de requête).

Pour rédiger un bon prompt, il est essentiel de respecter une certaine structure afin de réduire les biais et d'éviter les **hallucinations** des modèles (réponses hors contexte).

Template de prompt

```
contexte_role: "⟨Qui est le modèle ? ex : Prof MBA en marketing durable⟩"
objectif: "⟨Ce que tu veux obtenir⟩"
public_cible: "⟨Étudiants, décideurs, ingénieurs, etc.⟩"
contraintes: "⟨Langue, niveau, style, longueur, format⟩"
produit_attendu: "⟨Cours structuré, plan, code, email, tableau, etc.⟩"

structure:
  - "Introduction"
  - "Développement"
  - "Conclusion"
  - "Format attendu : ⟨Markdown, JSON, tableau⟩"

exemples_few_shot:
  - input: "⟨Exemple de question⟩"
    output: "⟨Exemple de réponse attendue⟩"
```

Paramètres de génération

```
modèle: "GPT-5.1"
temperature: 0.3        # contrôle la créativité (0 = rigide, 1 = créatif)
top_p: 1.0              # filtre le vocabulaire (0.9 = plus restreint)
max_tokens: 1200        # longueur max de la sortie
presence_penalty: 0.2   # encourage l'introduction de nouvelles idées
frequency_penalty: 0.3  # réduit les répétitions
stop_sequences: []      # séquences d'arrêt (optionnel)
```

Explication des paramètres

1. **temperature**
   - Contrôle le niveau de créativité et de hasard.
   - `0.0 – 0.3` : réponses très factuelles et précises.
   - `0.4 – 0.7` : équilibre précision/créativité.
   - `0.8 – 1.2` : réponses variées, créatives, parfois moins prévisibles.
2. **top_p** (*nucleus sampling*)
   - Limite le choix aux mots les plus probables.
   - `1.0` : aucun filtre (tout le vocabulaire disponible).
   - `0.9` : uniquement les mots les plus probables.
3. **max_tokens**
   - Définit la longueur maximale de la réponse.
   - Exemple : 1 000 tokens ≈ 750 mots.
4. **presence_penalty**
   - Encourage ou non l'introduction de nouvelles idées.
   - `0` : pas de pénalité (peut répéter).
   - `>0` : pousse le modèle à explorer de nouveaux concepts.
5. **frequency_penalty**
   - Réduit les répétitions de mots ou phrases.
   - `0` : pas de limite.
   - `0.5 – 1` : limite les redites, rend le texte plus fluide.
6. **stop_sequences**
   - Définit quand le modèle doit s'arrêter.
   - Exemple : `["###"]` → la génération s'arrête au symbole `###`.
7. **modèle**
   - C'est le "cerveau" utilisé (ex. `gpt-5.1`, `gpt-4o-mini`).
   - Chaque modèle a ses forces (code, vitesse, créativité, etc.).

Résumé rapide

- **temperature** : dose de créativité.
- **top_p** : filtre les mots les plus probables.
- **max_tokens** : longueur maximale de la sortie.
- **presence_penalty** : encourage la nouveauté.
- **frequency_penalty** : limite les répétitions.
- **stop** : fixe un point d'arrêt.
- **modèle** : cerveau utilisé.

Exemple appliqué

```
consignes:
  contexte_role: "Tu es Professeur d'université spécialisé en marketing
durable."
  objectif: "Construire un plan de cours complet en 12 séances."
  public_cible: "Étudiants MBA (niveau Master 2)."
  contraintes: "Langue : français, longueur 1500-2000 mots, format
académique, tableau planning."
```

```
  produit_attendu: "Plan de cours structuré avec objectifs pédagogiques,
modalités d’évaluation et bibliographie APA."

  structure:
    - "Présentation générale"
    - "Objectifs pédagogiques"
    - "Compétences visées"
    - "Plan des 12 séances (tableau)"
    - "Modalités d’évaluation"
    - "Bibliographie APA"
    - "Conclusion"

  exemples_few_shot:
    - input: "Donne un plan de cours court (3 séances)."
      output: "Séance 1 : Fondements du marketing – Séance 2 : Digitalisation
– Séance 3 : Durabilité."

parametres_generation:
  modèle: "GPT-5.1"
  temperature: 0.3
  top_p: 1.0
  max_tokens: 1800
  presence_penalty: 0.2
  frequency_penalty: 0.3
  stop_sequences: []
```