Prompt Engineering :
comment mieux parler aux IA pour les rendre plus intelligentes

basé sur l'article : "A Systematic Survey of Prompt Engineering in Large Language Models (Sahoo et al., 2025)"

Enzo PERRIN
Arthur MERMET

POLYTECH GRENOBLE
GRENOBLE INP
UGA Université Grenoble Alpes
La Région Auvergne-Rhône-Alpes

Contexte

- Les grands modèles de langage (LLMs) comme ChatGPT ou Claude comprennent le langage naturel.
- Problème : leur comportement dépend fortement de la formulation de la demande.
- Solution : le prompt engineering, l'art de concevoir des instructions efficaces pour orienter la réponse.

→ Exemple :

- "Résume ce texte" vs "Résume ce texte en trois points clés adaptés à un lycéen".

Principe de base

- Le prompt agit comme un programme linguistique : il configure temporairement le modèle sans changer ses poids.
- Le prompt = une interface de contrôle du modèle.
- Deux grandes familles :
  - Prompts manuels : instructions en langage naturel.
  - Prompts appris : représentations vectorielles qui activent certaines parties

→ Avantage : adapter un modèle sans ré-entraînement coûteux.

Les techniques fondamentales

- Zero-shot: aucune donnée d'exemple
  - "Traduis en espagnol : Bonjour."
- Few-shot: quelques exemples dans le prompt
  - "Texte : ... Résumé : ..." ×3 puis nouveau texte
- Chain-of-Thought (CoT)
  - "Explique ton raisonnement avant la réponse."

→ CoT précision de 90 % sur problèmes mathématiques complexes.

Nouvelles approches de raisonnement

- Tree-of-Thought (ToT) : explore plusieurs chemins de raisonnement, comme un arbre de décision.
- Graph-of-Thought (GoT) : relie des idées non linéaires, imite la pensée humaine.
- Self-Refine : le modèle critique et corrige sa propre réponse.

Nouvelles approches de raisonnement

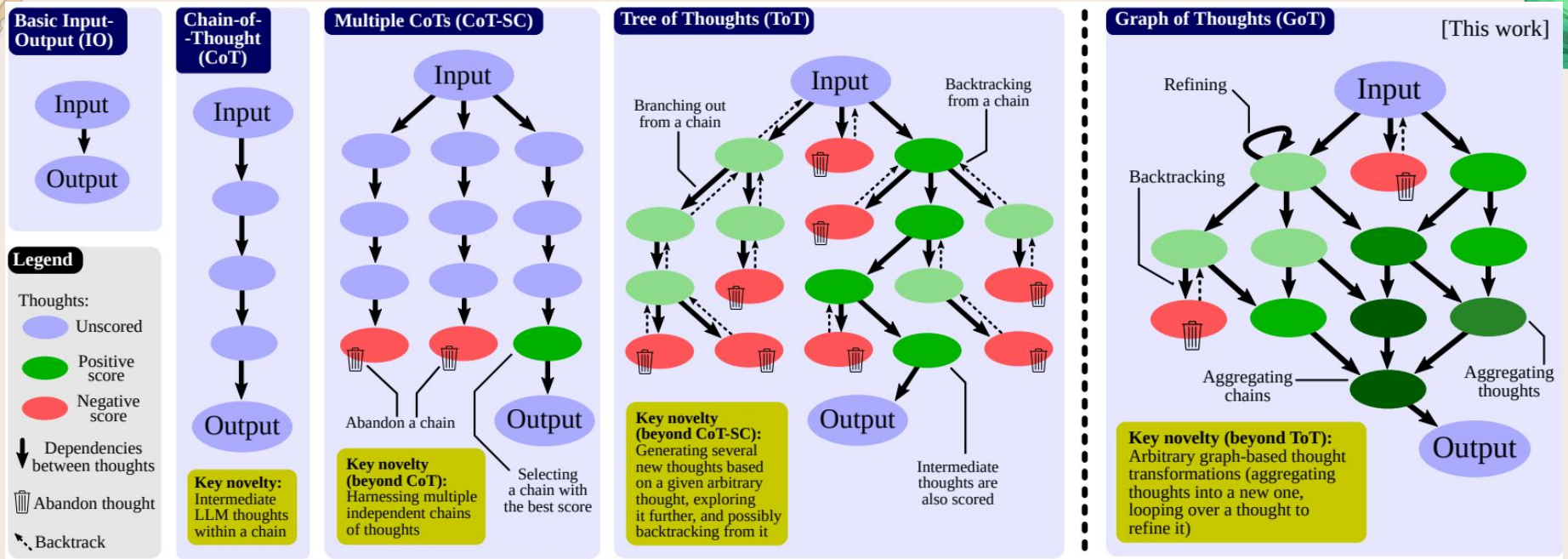


Figure 1: Comparison of Graph of Thoughts (GoT) to other prompting strategies.

Source: Graph of Thoughts: Solving Elaborate Problems with Large Language Models (Gerstenberger et al., 2024)

Réduire les erreurs et hallucinations

- Les LLMs peuvent “inventer” des faits.
- Solutions :
  - RAG (Retrieval-Augmented Generation) : enrichissement contextuel, en ajoutant des documents externes au prompt.
  - CoVe (Chain-of-Verification) : le modèle se relit et se corrige.

→ Résultat : -20 % d’erreurs factuelles en moyenne.

Vers l'automatisation

- APE (Automatic Prompt Engineer) : le modèle crée ses propres prompts par renforcement.
- OPRO : les LLMs deviennent eux-mêmes des outils d'optimisation.

→ Objectif : passer de la création du prompt à une ingénierie systématique.

Passons à la demo !!!!

Objectif

- Roles :
  - Ce que doit accomplir le model
- Pourquoi :
  - Les LLMs ne “devinent” pas
  - Sans objectif explicite, ils restent génériques
- Comment :
  - Peu / Pas de décision du model

Format

- Roles :
  - Spécifier comment la réponse doit être structurée ou livrée
- Pourquoi :
  - Un modèle de langage ne fournit pas la même réponse en fonction du format
- Comment :
  - Sur / Sous spécification

Contraintes

- Roles :
  - Donner des règles ou des erreurs à éviter
- Pourquoi :
  - Permet d’obtenir des réponses plus fiables, cohérentes, et vérifiées
- Comment :
  - Validation
    - Ex : “ Vérifie que le texte ait la bonne intonation “

Contexte

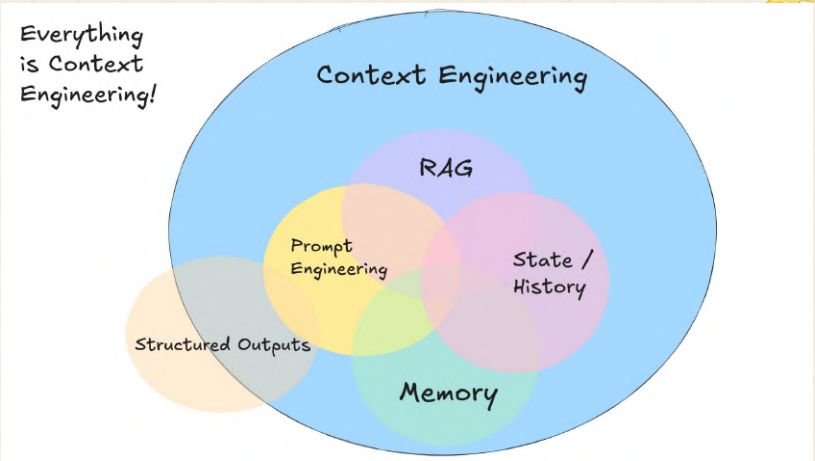
- Roles :
  - Fournir les données nécessaires pour que le modèle comprenne la situation
- Pourquoi :
  - Un LLM n’a ni mémoire ni perception du monde
  - Le contexte agit comme un sens pour un LLM
- Comment :
  - Il vaut mieux trop que pas assez

Bonnes Pratiques

- Ne pas entreméler Objectif, format et contraintes
  - Ex : *“Je suis étudiant et je veux un mail qui soit court mais motivant, sans être trop formel et avec un ton cool mais pas familier et en même temps il faut que ce soit clair et avec mes compétences et disponible en mars.”*
- Pas de décision du model

Conclusion et perspectives

- Le prompt engineering = nouvelle interface entre humain et IA.
- Impact : plus de contrôle, moins d’erreurs, meilleure adaptabilité.
- Tendances futures :
    - Contexte engineering



Sources

“A Systematic Survey of Prompt Engineering in Large Language Models” (Sahoo et al., 2025)”

“Graph of Thoughts: Solving Elaborate Problems with Large Language Models” (Gerstenberger et al., 2024)