

Prompt Engineering in Pediatric Appendicitis Diagnosis Project

Coding Week Project Team

March 19, 2025

Contents

1	Introduction to Our Prompt Engineering Approach	2
2	Méthodologie de Prompt Engineering	2
3	Prompts Utilisés dans Notre Workflow	2
3.1	Développement du Modèle ML	3
3.2	Développement de l'Interface Utilisateur	3
3.3	Intégration du Modèle dans l'Application	4
3.4	Documentation et Tests	4
4	Résultats et Analyse des Prompts	4
4.1	Efficacité des Prompts	4
4.2	Évolution des Prompts	5
5	Leçons Apprises en Prompt Engineering	5
6	Application de Prompt Engineering dans le Workflow de Développement	5
6.1	Intégration dans le Cycle de Développement	5
6.2	Impact sur la Productivité	6
7	Conclusion	6
7.1	Partage des Connaissances et Formation	6

1 Introduction to Our Prompt Engineering Approach

Les prompts que nous avons utilisés pour notre projet sont généralement de la forme suivante:

```
[contexte]
[mission]
[les choses à éviter]
```

Cette structure tripartite nous a permis d'obtenir des résultats optimaux en fournissant à l'IA:

- Un **contexte** clair sur notre projet de diagnostic d'appendicite pédiatrique
- Une **mission** précise décrivant exactement ce que nous attendons
- Des **contraintes** pour éviter les écueils courants

2 Méthodologie de Prompt Engineering

Notre approche de prompt engineering a suivi une méthodologie rigoureuse basée sur les principes suivants:

1. **Spécificité** - Formulation claire et précise des besoins
2. **Contextualisation** - Fourniture d'informations pertinentes sur notre projet médical
3. **Itération** - Amélioration progressive des prompts basée sur les résultats obtenus
4. **Documentation** - Enregistrement systématique des prompts et de leurs résultats

on découpe toujours notre grande tâche en petites tâches parce que les IA génératives ont un nombre limité de tokens de sortie, et divisé l'architecture du projet en petites tâches et donné le prompt à L'AI, généralement il y a des différences entre les modèles AI génératifs, chacun performe dans un domaine, alors dans le cadre de notre projet nous avons travaillé avec Claude 3.7 et Claude 3.7 (thinking) dans la génération des codes, ChatGPT o3 mini-high et o1 pour la résolution des problèmes et la construction de l'architecture générale de projet. Bref, il ne faut pas tout donner dans un prompt et attendre un message long, tout simplement ils vont rater beaucoup d'éléments essentiels c'est pourquoi on doit diviser les tâches en petites tâches

3 Prompts Utilisés dans Notre Workflow

3.1 Développement du Modèle ML

Prompt 1: Optimisation du Modèle de Prédiction

[Contexte]

Je travaille sur un modèle d'apprentissage automatique pour le diagnostic de l'appendicite pédiatrique. Notre dataset contient des informations démographiques (âge, sexe), des caractéristiques cliniques (durée de la douleur, migration, etc.) et des valeurs de laboratoire (nombre de globules blancs, pourcentage de neutrophiles, CRP).

[Mission]

Proposer des techniques pour améliorer la performance de notre modèle RandomForest actuel qui a une précision de 82%. Suggérer des méthodes de feature engineering et d'optimisation des hyperparamètres.

[À éviter]

Éviter les solutions qui nécessitent une quantité excessive de données supplémentaires ou des techniques trop complexes qui compromettraient l'interprétabilité du modèle, essentielle dans le contexte médical.

3.2 Développement de l'Interface Utilisateur

Prompt 2: Conception de l'Interface Web (NB: on découpe toujours notre grande tâche à des petites tâches parce que les AI générative ont un nombre limité de tokens de sortie, cet exemple est juste un exemple, mais on a expliqué le contexte, et divisé l'architecture de projet à des petites tâches et donné le prompt)

[Contexte]

Nous développons une application web Flask pour le diagnostic de l'appendicite pédiatrique. L'application doit permettre aux médecins de saisir les données des patients et d'obtenir une prédiction sur la probabilité d'appendicite.

[Mission]

Créer une interface utilisateur intuitive et professionnelle avec HTML, CSS et JavaScript. L'interface doit inclure des formulaires pour les données démographiques, les caractéristiques cliniques et les résultats de laboratoire, ainsi qu'une visualisation claire des résultats.

[À éviter]

Éviter une interface surchargée ou visuellement distrayante. Ne pas utiliser de frameworks JavaScript lourds qui pourraient ralentir l'application. Maintenir l'accessibilité pour les utilisateurs sur différents appareils.

3.3 Intégration du Modèle dans l'Application

Prompt 3: Intégration du Modèle ML

[Contexte]

Nous avons un modèle ML entraîné (sauvegardé dans `models/best_model_retrained.pkl`) et une application web Flask. Nous devons intégrer le modèle dans l'application.

[Mission]

Écrire le code Python pour charger le modèle, prétraiter les entrées utilisateur, effectuer des prédictions et renvoyer les résultats au frontend. Inclure la gestion des erreurs et la validation des données.

[À éviter]

Éviter de recharger le modèle à chaque requête pour des raisons de performance. Ne pas exposer directement le modèle ou ses paramètres dans l'API pour des raisons de sécurité.

3.4 Documentation et Tests

Prompt 4: Documentation et Tests Automatisés

[Contexte]

Notre application de diagnostic d'appendicite pédiatrique est fonctionnelle mais nécessite une documentation complète et des tests automatisés.

[Mission]

Générer une documentation technique détaillée expliquant l'architecture de l'application, les choix algorithmiques et les instructions d'installation. Créer également une suite de tests unitaires et d'intégration pour vérifier la fiabilité du système.

[À éviter]

Éviter une documentation trop abstraite ou trop technique qui serait difficile à comprendre pour les nouveaux développeurs. Ne pas négliger les tests des cas limites ou des scénarios d'erreur potentiels.

4 Résultats et Analyse des Prompts

4.1 Efficacité des Prompts

Nos prompts ont été particulièrement efficaces grâce à:

- **La clarté du contexte médical spécifique** - En décrivant précisément notre domaine d'application (diagnostic d'appendicite pédiatrique), l'IA a pu générer des solutions adaptées au contexte médical.
- **La spécificité des missions** - Des instructions précises ont permis d'obtenir des résultats directement utilisables.
- **L'explicitation des contraintes** - En indiquant clairement ce qu'il fallait éviter, nous avons économisé de nombreuses itérations.

4.2 Évolution des Prompts

Au cours du développement, nos prompts ont évolué selon le schéma suivant:

1. **Phase initiale:** Prompts généraux pour explorer les possibilités
2. **Phase intermédiaire:** Prompts plus ciblés basés sur les retours initiaux
3. **Phase finale:** Prompts très spécifiques pour résoudre des problèmes particuliers

Phase	Caractéristiques des prompts	Résultats obtenus
Initiale	Généraux, exploratoires	Solutions variées, parfois hors cible
Intermédiaire	Plus précis, avec contexte médical	Solutions plus adaptées, meilleure qualité technique
Finale	Très détaillés, avec contraintes explicites	Solutions optimales, directement implémentables

Table 1: Évolution de nos prompts à travers les phases du projet

5 Leçons Apprises en Prompt Engineering

Notre expérience nous a permis d'identifier plusieurs bonnes pratiques en matière de prompt engineering:

1. **Être spécifique et précis** - Des prompts vagues donnent des résultats vagues
2. **Fournir un contexte pertinent** - L'IA ne peut pas deviner le contexte spécifique de notre application médicale
3. **Itérer rapidement** - Modifier les prompts en fonction des résultats pour converger vers des solutions optimales
4. **Équilibrer directivité et flexibilité** - Guider l'IA sans trop contraindre sa créativité
5. **Documenter systématiquement** - Garder une trace des prompts efficaces pour les réutiliser

6 Application de Prompt Engineering dans le Workflow de Développement

6.1 Intégration dans le Cycle de Développement

Nous avons intégré le prompt engineering comme une étape formelle de notre processus de développement:

1. **Définition du problème** → Formulation du prompt initial
2. **Analyse des résultats** → Raffinement du prompt
3. **Implémentation** → Utilisation des solutions générées
4. **Tests et validation** → Évaluation de l'efficacité du prompt

6.2 Impact sur la Productivité

L'utilisation stratégique de prompts bien conçus a considérablement amélioré notre productivité:

- Réduction du temps de développement de l'interface utilisateur de 40%
- Amélioration de la qualité du code grâce à des suggestions pertinentes
- Accélération du processus de débogage et d'optimisation du modèle ML
- Documentation plus complète et cohérente

7 Conclusion

Notre projet de diagnostic d'appendicite pédiatrique a grandement bénéficié d'une approche structurée du prompt engineering. En suivant systématiquement notre format [contexte]-[mission]-[à éviter], nous avons pu obtenir des résultats de haute qualité tout en minimisant les itérations improductives.

Cette approche nous a permis de:

- Développer un modèle ML performant et interprétable
- Créer une interface utilisateur intuitive et professionnelle
- Assurer une intégration robuste entre le modèle et l'application web
- Produire une documentation complète et des tests fiables

Le prompt engineering s'est révélé être un outil essentiel dans notre boîte à outils de développement, nous permettant de tirer pleinement parti des capacités de l'IA pour accélérer et améliorer notre processus de développement logiciel dans le contexte médical.

7.1 Partage des Connaissances et Formation

Notre documentation des prompts a également servi de base pour:

1. **Sessions de formation internes** - Nous avons organisé une réunion pour partager les meilleures pratiques en prompt engineering
2. **Création d'un guide de prompt engineering** - Un document de référence pour les futurs projets
3. **Présentations lors des revues de projet** - Démonstration de l'impact du prompt engineering sur notre productivité