Rapport de conception du prompt

Devoir 1 — Thèmes techniques autour des LLM (cas OpenClaw)

---

1. Chronologie de conception du prompt

1.1 Instructions de base

La conception a débuté par la définition d'**instructions de base** visant à cadrer la tâche :

- **Rôle assigné** : « Tu es un expert en architectures LLM et en systèmes multi-agents. »
- **Consigne principale** : rédiger une note de synthèse académique sur OpenClaw comme framework d'agents LLM, à partir **exclusivement** des sources fournies et en respectant **strictement** la structure imposée.
- **Périmètre des sources** : les articles sont injectés dans le prompt via le fichier `articles.txt` (six articles + synthèse générale), afin que le modèle n'utilise que ce corpus.

Ces instructions ont été placées dans un **seul bloc de prompt** (après la structure et les articles) pour éviter que l'outil d'évaluation (promptfoo) ne traite plusieurs entrées comme plusieurs prompts distincts.

1.2 Rôle du système

Le **rôle du système** est implicite dans le prompt utilisateur : le modèle est invité à endosser une posture d'expert en architectures LLM et systèmes multi-agents. Ce choix permet :

- de fixer un niveau d'exigence technique (registre master / ingénieur) ;
- de limiter les formulations trop vulgarisatrices ou hors-sujet ;
- de favoriser un ton académique et neutre, cohérent avec une note de synthèse.

Aucun message système séparé n'est utilisé dans la configuration promptfoo actuelle : tout le cadrage est dans le prompt unique.

1.3 Contraintes de format

Les contraintes de format ont été introduites en plusieurs temps :

1. **Structure imposée** : un fichier `structure.txt` définit six grandes parties (I à VI) avec sous-points obligatoires. Ce fichier est injecté en tête du prompt sous la section « STRUCTURE OBLIGATOIRE À RESPECTER ».
2. **Format de rédaction** : pour obtenir une sortie lisible et structurée, des instructions explicites ont été ajoutées : chaque grande partie doit apparaître avec un **titre sur sa propre ligne** (chiffres romains +

intitulé exact) ; **saut de ligne** entre chaque grande partie ; développement sous chaque titre en respectant les sous-points de la structure.

3. **Contraintes rédactionnelles** : utilisation exclusive des sources, citations sous la forme [ARTICLE X], mention explicite des incertitudes, comparaison avec une architecture LLM simple, et contrainte de longueur (nombre de mots minimal).

L'ordre des blocs dans le prompt est donc : **structure** → **articles** → **instructions** (dont format de rédaction et contraintes).

---

2. Choix pédagogiques pour fixer le registre technique

2.1 Niveau et public cible

- **Niveau visé** : master / ingénieur, avec un style académique neutre.
- **Registre** : technique mais lisible, sans jargon superflu, en restant précis sur les concepts (agents, orchestration, outils, boucle raisonnement–action–observation).

2.2 Ancrage dans les sources

- Les **sources sont fournies in-context** dans le prompt (articles.txt), ce qui évite de laisser le modèle « inventer » des références.
- La consigne « à partir exclusivement des sources ci-dessus » et « ne pas inventer d'informations absentes des sources » vise à ancrer la synthèse dans le corpus et à limiter les extrapolations non vérifiables.

2.3 Structure comme guide d'apprentissage

- La structure en six parties (contexte → architecture → orchestration → cas d'usage → limites/risques → conclusion) suit une **logique pédagogique** : du général au particulier, puis aux limites et à la synthèse.
- Les sous-points (ex. « Boucle raisonnement → action → observation », « Comparaison avec une architecture LLM simple ») servent de **checklist** pour couvrir les notions attendues et maintenir le registre technique.

2.4 Comparaison avec une architecture LLM simple

La demande explicite de **comparer avec une architecture LLM simple** oblige à clarifier la valeur ajoutée des architectures multi-agents (OpenClaw) et à consolider la compréhension par contraste.

---

3. Itérations effectuées et raisons des ajustements

3.1 Passage de trois prompts à un seul

- **Problème initial** : la configuration listait trois entrées sous `prompts:` (structure.txt, articles.txt, et un bloc d'instructions). Promptfoo traitait chaque entrée comme un **prompt distinct**, générant trois évaluations au lieu d'une.
- **Ajustement** : regroupement en **un seul prompt** composé de trois sections (structure, articles, instructions), avec injection de la structure et des articles via `defaultTest.inputs` et les variables `{{structure}}` et `{{articles}}`.
- **Raison** : obtenir une seule note de synthèse cohérente qui combine structure, sources et consignes.

3.2 Renforcement du format de sortie (titres I à VI)

- **Problème** : les sorties n'étaient pas clairement structurées avec des parties « Grand I », « Grand II », etc.
- **Ajustement** : ajout d'une section **« FORMAT DE RÉDACTION OBLIGATOIRE »** dans les instructions, avec liste des six titres exacts à faire apparaître sur une ligne dédiée ; consigne de saut de ligne entre les grandes parties ; rappel de respecter les sous-points sous chaque titre.
- **Raison** : améliorer la lisibilité et la conformité à la structure demandée, et faciliter la correction ou l'évaluation.

3.3 Contrainte de longueur

Une contrainte de **longueur minimale** (en nombre de mots) a été précisée dans les contraintes pour éviter des synthèses trop courtes. La valeur exacte peut être ajustée selon le cahier des charges du devoir.

---

4. Risques identifiés et mesures de mitigation

4.1 Hallucinations et informations inventées

- **Risque** : le modèle peut générer des détails techniques (composants, noms de modules, scénarios) non présents dans les articles.
- **Mitigation** : instructions explicites (« Utiliser uniquement les informations issues des sources », « Ne pas inventer d'informations absentes des sources ») ; sources fournies in-context ; exigence de **citer les sources sous la forme [ARTICLE X]** pour encourager l'ancrage et permettre un contrôle a posteriori.

4.2 Dérives de registre ou de contenu

- **Risque** : ton trop vulgarisé, hors-sujet, ou au contraire trop spéculatif.
- **Mitigation** : rôle « expert » et consigne de style académique neutre ; structure et sous-points imposés qui canalisent le contenu ; demande de **mentionner explicitement les incertitudes** pour éviter des affirmations trop péremptoires.

4.3 Non-respect de la structure

- **Risque** : parties manquantes, ordre différent, ou titres absents.
- **Mitigation** : structure fournie en tête de prompt avec intitulés exacts ; section « FORMAT DE RÉDACTION OBLIGATOIRE » avec la liste des six titres ; évaluation possible via promptfoo pour vérifier la présence de marqueurs (ex. « I. », « II. ») ou la longueur.

4.4 Dépendance au modèle et à la qualité des sources

- **Risque** : un modèle plus faible ou des articles insuffisants peuvent conduire à des synthèses partielles ou imprécises.
- **Mitigation** : choix d'un modèle adapté (ex. via OpenRouter) et reproductibilité par configuration YAML ; corpus d'articles variés pour couvrir technique, sécurité et usages ; itérations sur le prompt et tests répétés pour valider la stabilité des sorties.

---

5. Synthèse

La conception du prompt a suivi une chronologie claire : instructions de base et rôle d'expert, intégration de la structure et des articles en un seul prompt, puis renforcement des contraintes de format et de rédaction. Les choix pédagogiques visent un registre technique de niveau master/ingénieur, ancré dans les sources et guidé par une structure en six parties. Les itérations ont porté sur la fusion des trois entrées en un prompt unique et sur l'explicitation des titres et de la mise en forme. Les risques (hallucinations, dérives, non-respect de la structure) sont mitigés par des consignes explicites, des citations obligatoires et une structure imposée, complétées par des tests et ajustements itératifs.

Rapport généré dans le cadre du devoir 1 — Promptfoo / OpenClaw.

Pour enregistrer en PDF : Ctrl+P (ou Fichier → Imprimer) puis « Enregistrer au format PDF » ou « Microsoft Print to PDF ».