

Projet IA Générative

Agents Intelligents, Raisonnement Avancé & Streamlit

Objectif du Projet

L'objectif est de concevoir et développer une application interactive via **Streamlit** qui utilise un ou plusieurs **Agents LLM** pour accomplir une tâche complexe.

Le cœur du projet repose sur la capacité de votre agent à **raisonner** (planifier, critiquer, décider) avant d'agir, et à présenter le résultat via une interface utilisateur.

1. Thématiques

Le choix du sujet est libre. Vous allez proposer un cas d'usage pertinent. Voici des exemples pour guider votre réflexion, avec un accent particulier sur les domaines nécessitant une haute fiabilité.

- **L'agent de diagnostic :** L'utilisateur entre des symptômes. L'agent génère des hypothèses, pose des questions complémentaires (boucle interactive), et classe les pathologies probables en justifiant chaque choix.
- **L'investigateur financier :**
Un agent qui prend le nom d'une entreprise, recherche ses dernières actualités boursières, analyse son rapport annuel (PDF) donné ou recherché puis génère une recommandation d'investissement argumentée sur Streamlit.
- **Le Planificateur de voyage autonome :**
Un agent qui demande une destination, cherche la météo en temps réel, suggère des activités adaptées au climat prévu, et crée un itinéraire jour par jour téléchargeable.
- **Programmateur :**
Un agent qui prend une demande de fonctionnalité, écrit le code Python correspondant, génère les tests unitaires associés, et affiche le code avec la syntaxe directement dans l'interface.
- **Une architecture Multi-Agents :**
Une architecture mettant en scène deux agents qui débattent d'un sujet proposé par l'utilisateur. Un troisième agent "modérateur" analyse les échanges et synthétise une conclusion équilibrée.
- **Vulgarisateur de rapports cliniques :** Analyse de comptes-rendus complexes (biologie/imagerie) pour les traduire en langage patient, avec détection automatique des termes techniques.
- **Vérificateur d'interactions des médicaments :** Un agent qui croise une liste de médicaments avec une base de connaissances pour signaler les contre-indications via un raisonnement logique.
- **Éducation :** Un agent qui aide l'étudiant à résoudre un problème sans jamais donner la réponse directement. Le problème peut être de diverse nature.

2. Les Techniques de Raisonnement

Votre application ne doit pas être une simple "boîte de chat". Vous devez implémenter explicitement des techniques de raisonnement pour guider le modèle (LLM).

Chain of Thought (CoT)

Forcer le modèle à décomposer le problème.

Exemple : "Pense étape par étape. 1. Analyse les données. 2. Identifie les variables. 3. Calcule la solution."

Tree of Thoughts (ToT)

Pour l'exploration complexe. L'agent génère plusieurs pistes de solutions, les évalue, et ne conserve que la plus prometteuse (élagage de l'arbre de décision).

ReAct (Reason + Act)

Le standard des agents. Une boucle itérative :

Pensée → Action (Outil) → Observation → Réponse

Self-Correction (réflexion)

L'agent produit une première réponse, la critique lui-même pour trouver des erreurs (hallucinations, logique), et génère une version finale corrigée.

Autres techniques de réflexion / raisonnement

3. Livrables

1. **Page expliquant votre sujet** : Une page contenant un paragraphe expliquant votre sujet, les fonctionnalités ainsi que les membres du projet à rendre au plus tard le **samedi 22 novembre** à minuit
2. **Code Source** : Repository GitHub (propre, avec `requirements.txt`) ou les sources .zip
3. **Démo Vidéo** : Une démonstration video de votre projet. Sur streamlit vous avez la possibilité de prendre une capture video.
4. **Readme** : Explication du choix de la technique de raisonnement.
5. **Date limite du rendu** : **19 décembre à minuit**.

Bon courage !