
DEVELOPPEMENT D'UN AGENT IA AVEC LANGCHAIN ET UN LLM

Rapport de ETCHECOPAR Maxime

3^{ème} année BUT Informatique

Table des matières

1.	Description de l'objectif et de l'architecture globale	3
1.1.	Objectif de l'agent.....	3
1.2.	Architecture globale	3
2.	Spécifications des outils	4
3.	Exemples concrets de besoins et collaborations des outils.....	4
4.	Liste de requêtes de démonstration.....	5

Lien du google colab :

https://colab.research.google.com/drive/14uoh4KmQ4fz3lejEjGYex_INUHkdUQ1i?usp=sharing

1. Description de l'objectif et de l'architecture globale

1.1. Objectifs de l'agent

L'objectif de mon idée pour ce projet est de concevoir un Agent Éducatif Intelligent capable d'accompagner un utilisateur dans son apprentissage. Contrairement à un chatbot classique, cet agent est conçu pour utiliser des outils spécifiques pour garantir la véracité des informations et l'interactivité.

Il répond à quatre besoins principaux :

- Explication pédagogique : Fournir des cours clairs basés sur des sources fiables (Wikipedia).
- Génération d'exercices (QCM) : Créer des quiz personnalisés selon un sujet et une difficulté donnée.
- Correction et Évaluation : Corriger les réponses de l'élève, calculer précisément son score et fournir des explications détaillées.
- Accessibilité et Suivi : Vocaliser les corrections (Text-to-Speech) et historiser les résultats pour suivre la progression.

1.2. Architecture globale

L'agent repose sur LangChain et utilise le paradigme ReAct (Reason + Act) vu en cours. Cela signifie que pour chaque requête utilisateur, l'agent suit une boucle de pensée : il observe la demande, raisonne pour choisir l'outil adéquat, agit, observe le résultat, et formule une réponse finale.

Les composants clés sont :

- Le Cerveau (LLM)
- La Mémoire : Une ConversationBufferMemory qui permet à l'agent de conserver le contexte des échanges précédents.
- L'Interface : Développée avec Gradio, il y a une navigation par onglets (Discussion, QCM, Historique).

2. Spécifications des outils

L'agent dispose de 6 outils distincts lui permettant d'interagir. Les voici :

Outil	Rôle
Wikipédia	C'est notre base de connaissance encyclopédique. Utilisé pour le mode « Cours »
Recherche web (Tavily)	Accès aux informations en temps réel (actualités, faits récents non présents sur Wikipedia).
Calculatrice	Effectue des calculs mathématiques. Le LLM délègue le calcul du pourcentage de réussite au QCM à cet outil pour éviter les problèmes.
Python executor	Exécute des scripts Python si le raisonnement demande de la programmation.
Heure actuelle	Fourni le contexte temporel, indispensable pour horodater l'historique des scores des qcm.
Lecteur vocal TTS	Pour l'accessibilité. Convertit le texte de correction en fichier audio pour que l'utilisateur puisse écouter les explications.

3. Exemples concrets de besoins et collaborations des outils

Voici comment l'agent orchestre ses outils pour répondre à des scénarios complexes.

Scénario 1 : La correction d'un QCM

Besoin : L'utilisateur soumet ses réponses ("1A, 2B") du qcm généré. L'agent doit corriger, expliquer, noter et permettre l'écoute de la correction.

Collaboration des outils (Chaîne de pensée ReAct) :

1. *Analyse* : L'agent reçoit les réponses et les compare à la bonne solution (connaissance interne ou Wikipedia).
2. *Raisonnement* : Il rédige le texte explicatif ("La réponse A est fausse car...").
3. *Action 1 (avec la calculatrice)* : Il identifie qu'il doit calculer un score (ex: 2 bonnes réponses sur 3). Il appelle l'outil Calculatrice avec $(2/3)*100$.
4. *Observation 1* : L'outil renvoie le résultat du calcul soit 66.67%.
5. *Action 2 (lecteur vocal)* : Il prend tout le texte de la correction générée et l'envoie à l'outil TTS.
6. *Observation 2* : L'outil confirme la création du fichier correction_audio.mp3.
7. *Réponse finale* : L'agent affiche le texte, le score calculé, et l'interface charge le fichier audio.

Scénario 2 : Recherche d'informations d'actualité

Besoin : L'utilisateur demande "Quelles sont les dernières découvertes sur Mars ?".

Collaboration :

1. L'agent détecte que sa connaissance interne peut être obsolète.
2. Il appelle l'outil de recherche web (Tavily).
3. Il synthétise les résultats retournés pour faire un résumé.

4. Liste de requêtes de démonstration

Voici 10 requêtes variées pour tester l'étendue des capacités de l'agent dans l'interface :

Dans l'onglet Discussion & Cours :

1. Explique-moi le principe de la photosynthèse de manière simple.
2. Quels sont les dernières découvertes sur Mars ?
3. Quelle heure est-il exactement ?
4. Peux-tu me calculer la racine carrée de 1444 ?

Dans l'onglet Entraînement & QCM :

5. *Sujet : « Révolution Française », Niveau : « Facile », Nombre de question : 3, Puis clic sur générer, Réponse : « 1A 2C 3B », Puis clic sur valider*
6. *Sujet : « Capitales du monde », Niveau : « Expert », Nombre de question : Comme vous le souhaitez, Puis clic sur générer, Réponse : Essayez de répondre au qcm, Puis clic sur valider.*
7. *Dans le champ réponse : « Je ne sais pas ». Examiner la correction.*

Dans l'onglet Historique & Progression :

8. Cliquer sur « Actualiser le tableau » pour voir apparaître l'historique du dernier qcm effectué

Tests techniques :

9. *Dans la discussion : « Écris-moi un script Python qui affiche les nombres premiers jusqu'à 100. »*
10. *Dans la discussion : « Calcule $54 * 89 / 2 + 10$ ».*