

Conception d'un Système Multi-Agents Intelligent supporté par des Modèles de Langage (LLMs)

Consignes générales

- Chaque projet doit comporter une **partie théorique** présentant les fondements des concepts, méthodes et algorithmes utilisés.
- Une **phase de prétraitement et d'analyse exploratoire des données** est obligatoire.
- La **solution finale** doit être **automatisée et déployée**, si possible (ex. via Streamlit, Flask, une API,...)

L'objectif est de développer un système collaboratif multi-agents, où plusieurs agents intelligents (analyse, synthèse, décision, communication) interagissent entre eux à l'aide d'un LLM pour résoudre des tâches complexes de manière autonome et coopérative. L'objectif est aussi d'explorer comment un LLM peut servir de moteur cognitif permettant aux agents d'échanger, planifier et s'adapter.

Scénario possible :

Le système doit simuler une **équipe d'agents virtuels** chargés d'une mission donnée, par exemple :

- **Scénario 1 – Analyse financière** : un agent collecte les données, un autre les analyse, un troisième rédige un rapport explicatif.
- **Scénario 2 – Assistance à la recherche** : agents spécialisés dans la recherche, le résumé, la vérification et la reformulation d'informations scientifiques.
- **Scénario 3 – Gestion de projet** : agents planificateurs, exécutants et contrôleurs qui coordonnent leurs décisions via le LLM.

Tâches principales

1. **Fondements théoriques (IA agentic, Système multi-agents, LLMs, ...)**
2. **Modélisation du système multi-agents**
 - Définir le rôle et les compétences de chaque agent (analyseur, décideur, rapporteur, etc.).
 - Concevoir le protocole de communication entre agents (messages, intentions, contexte partagé).
3. **Intégration du LLM**
 - Utiliser un modèle open-source (LLaMA, Mistral, Phi-3, GPT-4 via API).
 - Gérer la compréhension, la génération et la coordination des réponses.
4. **Simulation des interactions**
 - Implémenter un scénario de collaboration.
 - Permettre aux agents de planifier, débattre et se corriger mutuellement.

Conception d'un Système Multi-Agents Intelligent supporté par des Modèles de Langage (LLMs)

5. Interface utilisateur

- Créer un tableau de bord (Streamlit) pour visualiser les échanges et le résultat final.

Technologies suggérées

Python – **LangChain**, **AutoGen**, **CrewAI**, ou **OpenDevin**

Streamlit – Hugging Face Transformers – Ollama – FAISS (si ajout d'un module RAG)

Livrables

- 1- Rapport technique
- 2- Code développé
- 3- Présentation vidéo avec une démonstration de la solution (15–20 minutes).
- 4- Présentation power point