

Exercice Scientifique et technique

LLM pour l'analyse automatique des documents

SENE Mouhamed

Contexte du problème

- Les projets d'ingénierie génèrent de grands volumes de documents techniques : spécifications, exigences, normes
- L'analyse manuelle est coûteuse en temps et sujette aux erreurs humaines
- Les documents sont hétérogènes, volumineux et fortement interdépendants
- Besoin industriel : automatiser l'extraction d'information et la détection d'incohérences
- Enjeu : réduire le temps d'analyse tout en maintenant un haut niveau de fiabilité

Problématique scientifique

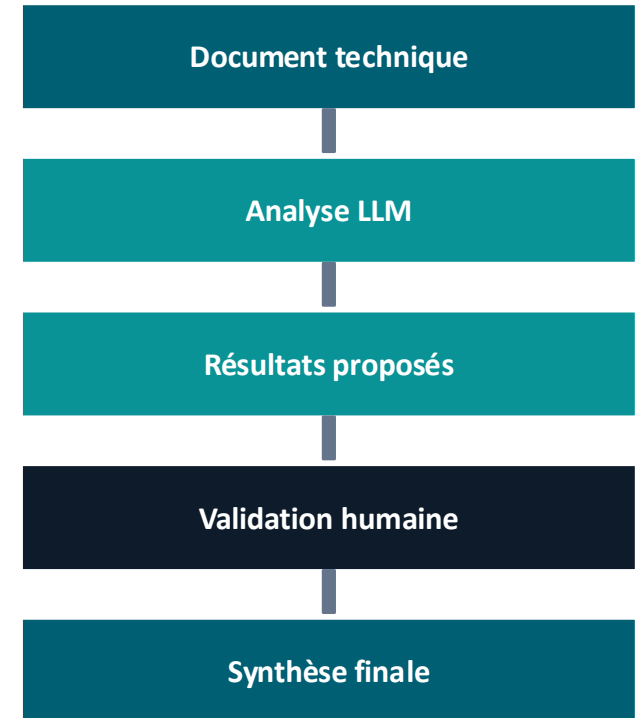
- **Question centrale** : Comment automatiser l'analyse de documents techniques avec un LLM tout en garantissant la fiabilité des résultats ?
- Les LLMs peuvent produire des hallucinations ou des erreurs factuelles sur des données techniques précises
- Les documents d'ingénierie exigent précision, cohérence et traçabilité des décisions
- Tension fondamentale : automatisation (efficacité) vs validation humaine (fiabilité)
- **Hypothèse** : un système hybride IA + humain permet de dépasser les limites des LLMs seuls

Fondements scientifiques : NLP et LLMs

- **NLP** : traitement automatique du langage — tokenisation, embeddings vectoriels, analyse sémantique
- **LLM** : modèles pré-entraînés sur corpus massifs (GPT-4, Claude, LLaMA) basés sur l'architecture Transformer
- **Mécanisme d'attention** : (Vaswani et al., 2017) — capture les dépendances longue distance dans un texte
- **Prompt engineering** : technique de formulation des requêtes pour orienter le comportement du modèle
- **RAG** : Retrieval-Augmented Generation — enrichit le contexte du LLM avec des documents externes pertinents

Méthodologie : Human-in-the-Loop (HITL)

- **HITL** : paradigme où un humain intervient dans la boucle de décision du système IA
- Le modèle propose une analyse → l'humain valide, corrige ou rejette
- Réduit l'impact des hallucinations sur les résultats critiques
- Issu des travaux sur l'apprentissage actif (Settles, 2009)
- Amélioration itérative : les retours humains affinent les prompts et règles métier

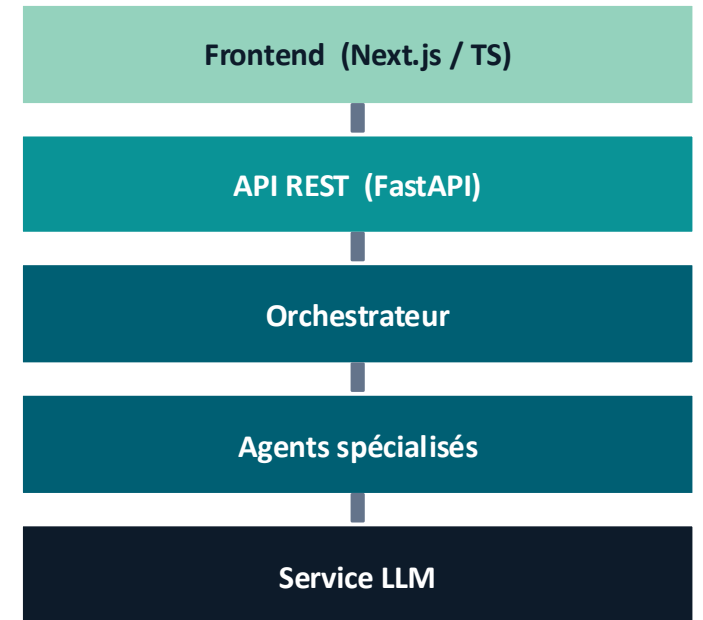


Cas d'étude : application chez Capgemini Engineering

- Application d'analyse documentaire assistée par IA développée en contexte industriel réel
- **Workflow** : import de documents → analyse automatique → validation utilisateur → synthèse finale
- **Analyses réalisées** : extraction d'exigences, détection d'incohérences, génération de résumés
- Boucle HITL : l'utilisateur valide ou rejette chaque incohérence détectée par le système
- Les résultats sont retournés au frontend sous format JSON via l'API
- Cas concret de mise en production d'un LLM sur des documents normatifs industriels

Architecture technique du système

- **Frontend** : Next.js + TypeScript — interface utilisateur réactive et fortement typée
- **Backend** : Python + FastAPI — API REST haute performance, retours JSON structurés
- **Orchestrateur** : coordonne le flux de traitement entre les agents spécialisés
- **Agents spécialisés** : modules indépendants dédiés à chaque tâche (extraction, cohérence, résumé)
- **Service LLM** : couche d'abstraction pour les appels au modèle de langage



Lien avec les enseignements académiques

- **Projet académique full-stack** : application React + backend + API — architecture identique au projet en entreprise
- **Développement frontend** : React (cours) → appliqué via Next.js (surcouche React) chez Capgemini Engineering
- **Architecture client–serveur** : séparation frontend/backend, communication via API REST — pattern étudié en cours
- **Intégration de LLMs** : projet académique utilisant des modèles de langage — approfondissement direct en contexte professionnel
- **Qualité et typage** : TypeScript en entreprise, extension naturelle des bonnes pratiques vues en cours

Références scientifiques et état de l'art

- **Vaswani et al. (2017)** Attention Is All You Need — fondement de l'architecture Transformer
- **Brown et al. (2020)** GPT-3 : few-shot learning et capacités émergentes des grands modèles
- **Lewis et al. (2020)** RAG : Retrieval-Augmented Generation — réduction des hallucinations
- **Settles (2009)** Active Learning Literature Survey — base théorique du HITL
- **Ouyang et al. (2022)** InstructGPT / RLHF — alignement des LLMs par retour humain
- **Ji et al. (2023)** Survey of Hallucination in NLG — état de l'art sur les limites des LLMs

Conclusion & Perspectives

Bilan

- Les LLMs offrent un potentiel fort pour l'analyse automatique de documents techniques
- L'approche HITL est essentielle pour garantir fiabilité et confiance dans les résultats
- L'application développée chez Capgemini Engineering valide la faisabilité industrielle de cette approche

Perspectives

- **Limites** : coût des appels API, latence, dépendance au modèle sous-jacent
- **Perspectives** : fine-tuning sur corpus métier, métriques de qualité automatiques
- **Ouverture** : intégration de RAG avancé pour réduire davantage les hallucinations