Méthodologies d’extraction de données sur du texte utilisant les « modèles de langues » (LLM)

# Préambule

Qu’est-ce qu’un grand modèle de langage (LLM) ?

Les grands modèles de langage (LLM) sont des modèles d’apprentissage automatique capables de comprendre et de générer des textes en langage humain. Ils fonctionnent en analysant des ensembles de données linguistiques massives. (1)

# Table des matières

[Préambule 1](#_Toc167793870)

[Table des matières 1](#_Toc167793871)

[1. Présentation des documents 2](#_Toc167793872)

[2. Analyse des documents 2](#_Toc167793873)

[2.1 Xebia – How to extract structured data from unstructured text using LLMs 2](#_Toc167793874)

[2.2 Langchain – Use cases : Extraction 3](#_Toc167793875)

[2.3 Medium - Harness The Power of LLMs : How to extract data from legacy documents using LLMs 5](#_Toc167793876)

[2.4 Towardsai - Demystifying Information Extraction using LLM. 6](#_Toc167793877)

[Bibliographie 7](#_Toc167793878)

# 1. Présentation des documents

Les documents ayant servi au recueil d’informations proviennent du site Cloudflare, Xebia (article de septembre 2023) et du site Langchain (date inconnue mais copyrigt 2024 en bas du site).

Cloudflare est une entreprise de cyber-sécurité américaine, ils ont une section de vulgarisation des termes relatifs à l'IA sur leur site, ce sont des spécialistes en informatique.

Xebia est une entreprise de consultation en transformation digitale, il s’agit de professionnels du domaine, leurs informations peuvent s’avérer utiles et donner une vue du sujet traité.

Langchain est une librairie connue, open source, de développement d’applications autour des LLM. La documentation à travers cette librairie est indispensable pour savoir comment traiter le sujet en pratique.

Medium est une plateforme de partage de connaissance, où les gens peuvent voter pour populariser les articles utiles.

Towardsai est un site de publications relatives à l'IA.

# 2. Analyse des documents

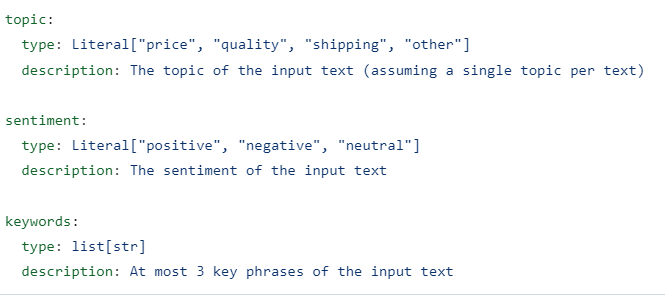
## 2.1 Xebia – How to extract structured data from unstructured text using LLMs

L’intérêt des LLM est qu’ils permettent de compléter une tâche sans entraîner de modèles.

L’extraction de données utilisant les LLM se déroule en 4 étapes :

* Créer une demande
* Donner le texte non structuré au LLM
* Récupérer la sortie structurée du LLM
* La sauvegarder dans une base de données

On peut demander une sortie structurée au modèle (comme couleur = noir ou blanc) avec un fichier « .yaml ». La structure demandée sera chargée par des classes « Pydantic BaseModel ».



On peut indiquer au modèle les indications de recherche et les sorties possibles pour chaque caractéristiques.

Figure 1 : Exemple de structure de fichier yaml

La sortie du modèle est validée par un modèle pydantic, si on a une erreur, on peut tenter de s’adapter à cette erreur en adaptant le prompt.

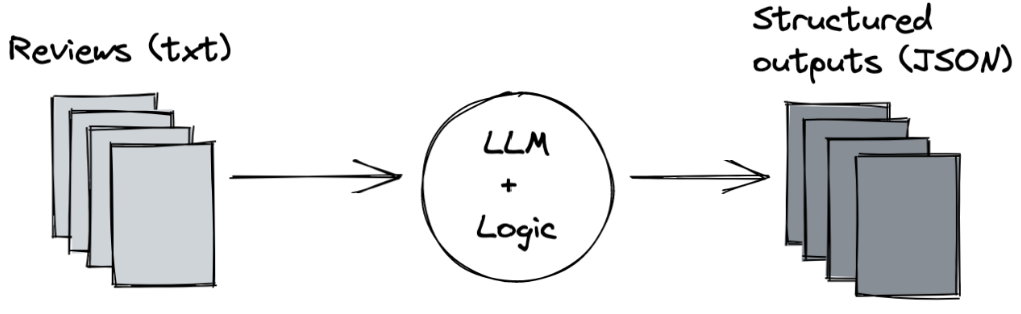


Figure 2 : Séparation de la logique modèle d’avec les données

## 2.2 Langchain – Use cases : Extraction

Il y a deux types d’approches pour extraire des données en utilisant les LLMs :

* **Fonctions** : Certains LLM ont des fonctions intégrées d’extraction arbitraires d’entités (peuvent donner des attributs que l’on n’a pas demandé).
* **Parsing** : On définit soit même les fonctions d'extraction (Pydantic).

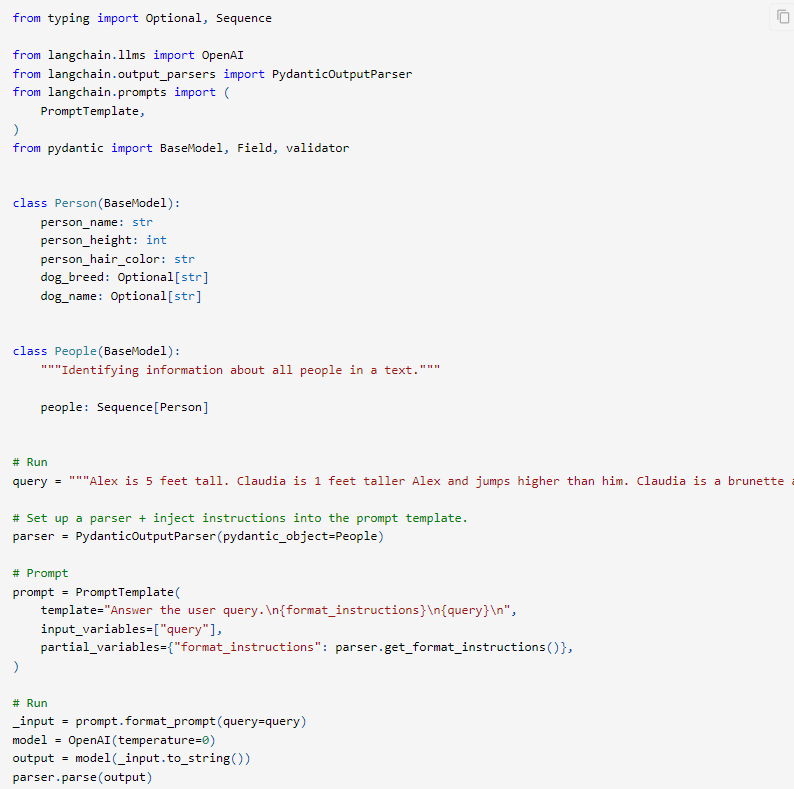
C’est le parsing qui nous intéresse car personnalisable, voici un exemple de parsing :

Figure 3 : Demande de parsing sur un prompt

La classe BaseModel (pydantic) est utilisée pour définir les attentes de sorties.

Dans le modèle de prompt (langchain), définition d’un prompt général contenant toutes les informations, avec la variable d'entrée dans l'argument "input\_data".

Chargement du modèle, run du modèle avec le prompt, segmentation de la sortie avec le parser d'entrée.

## 2.3 Medium - Harness The Power of LLMs : How to extract data from legacy documents using LLMs

La première étape est de transformer le document en texte brut, pour entrer dans les LLMs.

Organisation du script d'extraction des données :

- Chargement texte

- Créer prompt LLM

- Envoyer le prompt au LLM par l'API openAI

- Imprimer les résultats

Exemple d'un prompt :

Figure 4: Exemple de prompt

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

"lease" est le texte brut dans lequel chercher les informations

Une image contenant texte, Police, capture d’écran, algèbre

Description générée automatiquement

Figure 2 : Envoi du prompt

Une image contenant texte, Police, ligne

Description générée automatiquement

Figure 3 : Résultat

Le prompt est segmenté en 3 parties :

- Contexte : exemple : tu es un agent de génération de texte travaillant avec des contrats de location

- Task specification : tâche à exécuter, format attendu

- Texte source

## 2.4 Towardsai - Demystifying Information Extraction using LLM.

On peut utiliser des modèles OCR (Optical Character Recognition)

# Bibliographie

1. **Cloudflare.** [En ligne] https://www.cloudflare.com/fr-fr/learning/ai/what-is-large-language-model/.

2. **Langchain.** [En ligne] https://python.langchain.com/docs/use\_cases/extraction.

3. **Xebia.** [En ligne] https://xebia.com/blog/archetype-llm-batch-use-case/.

4. **Lights, Brightest.** Harness The Power of LLMs: How to extract data from legacy documents using LLMs. *Medium.* [En ligne] https://medium.com/@brightestlights/harness-the-power-of-your-how-to-extract-data-from-legacy-documents-using-llms-2841f5835359.

5. **Mohan, Aditya.** Demystifying Information Extraction using LLM. *Towardsai.* [En ligne] https://pub.towardsai.net/demystifying-information-extraction-using-llm-f1a551f01f66.