Développement d'une application web basée IA pour de la recherche d'information dans des documents

Étudiante : Marie Brunetto | 4IR | Stage de Juin à Septembre 2024
Entreprise : AVL Software and functions, Ratisbonne, Allemagne | Tutrice : Dominique LOUARN-PIOCH

Mots-Clefs: rag, base de données vectorielle, intelligence artificielle, full-stack, application web

Contexte

Stage effectué dans l'entreprise **AVL Software and Functions**, spécialisée dans la conception de solutions à destination de l'automobile. Je travaillais dans le département **Process, Methods and Tools**, plus précisément dans une équipe visant à créer des outils utilisant de l'**Intelligence Artificielle**.

L'objectif de mon stage était d'ajouter à l'application existante **Maestra AI**, une page permettant aux utilisateurs de **questionner en langage naturel** des **documents volumineux** afin de retrouver des informations précises et/ou générales.

Travail Réalisé

1. Concept

Principe du **RAG (Retrieval-Augmented Generation)** en trois parties :
Indexation : Les documents sont divisés en extraits (chunk) puis convertis en vecteurs de sens sémantique (embedding) par un Large Language Model (LLM)
Stockage : Base de données vectorielle dans laquelle sont stockés chunks et embeddings.
Recherche : Recherche par similarité en trois étapes : reformuler la question utilisateur, rechercher dans la base, traduire les extraits en langage naturel grâce à un LLM.

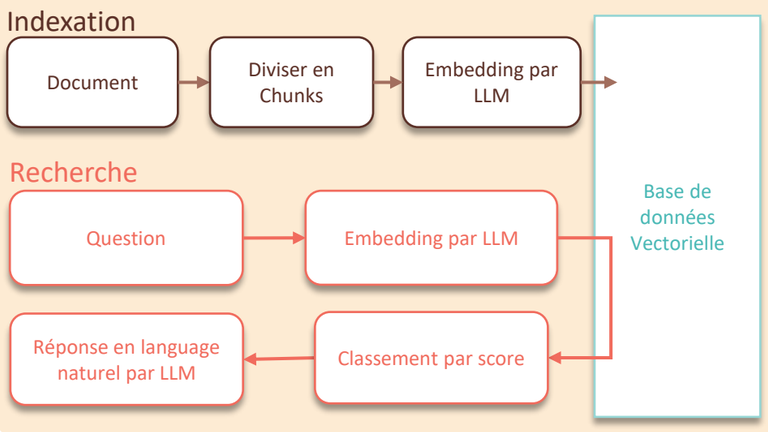
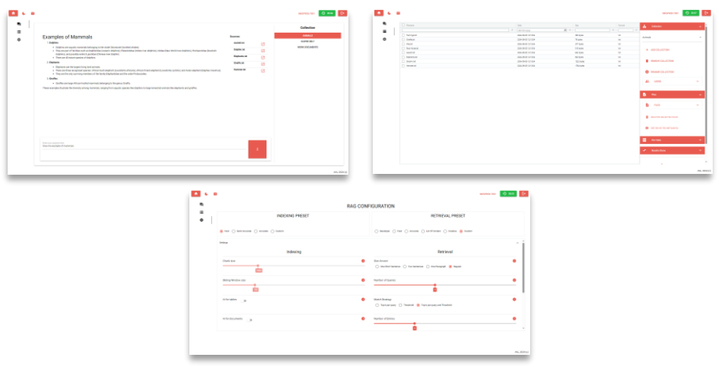


Schéma simplifié décrivant le fonctionnement du RAG

2. Front-End



Aperçu des trois onglets décrits ci-dessous

Développement du Front-End avec NICEGUI, framework Python utilisé pour le développement web. En suivant la charte graphique de l'application Maestra AI, création d'une page de trois onglets:

- **Talk To Your Data**, pour poser ses questions et rechercher les données,
- **Manage Data**, pour gérer les collections et fichiers dans lesquels les recheches sont réalisées,
- **Settings**, pour régler les paramètres d'indexation et de recherche.

Il fallait aussi s'assurer que la page soit intuitive et complète pour l'utilisateur.

3. Back-End

Le Back-End, réalisé en Python, consiste principalement en l'implémentation d'une base de données vectorielle. Deux systèmes ont été utilisés :
Qdrant : Base de données vectorielle efficace et simple d'utilisation.
PGVector : Extension PostGreSQl, le choix s'est finalement tourné vers PGVector pour plus de contrôle sur la structure de la base (voir schéma) ainsi que la capacité à stocker desBinary Large Objects (BLOB) pour les fichier sources.

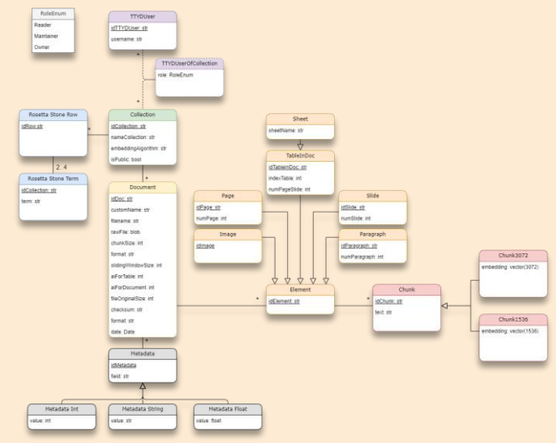


Diagramme Relationnel de la base de données vectorielle PGVector

Résultats

Personnel

- **Application de concepts** appris en cours dans un contexte professionnel,
- Approfondissement des connaissances du **monde de l'entreprise** et de la **gestion de projet**,
- Découverte du **développement orienté IA**,
- **Échange culturel** dans une entreprise internationale et découverte de la culture Allemande.

Entreprise

- Développement d'une application permettant à des utilisateurs de différents départements de **chercher efficacement dans des documents**,
- Développement **d'outils automatisant les Q&A** et génération automatique de texte basé sur des documents.