

PROJET LLM – CRÉATION D’UN CHATBOT PERSONNEL

Introduction

Aujourd’hui, le portfolio ne se limite plus à une simple page statique présentant un parcours et des compétences. Grâce aux avancées de l’intelligence artificielle, il est désormais possible de transformer ce support en une expérience interactive.

L’objectif de ce projet est de concevoir un **portfolio interactif sous forme de chatbot**, capable de dialoguer avec les visiteurs comme un véritable **jumeau virtuel professionnel**.

Ce chatbot permet à l’utilisateur de poser des questions telles que « *Qui es-tu ?* », « *Quelles sont tes compétences ?* » ou « *Sur quels projets as-tu travaillé ?* », et d’obtenir des réponses précises, personnalisées et cohérentes.

Cependant, un défi majeur se présente : bien que les modèles de langage (LLMs) disposent de vastes connaissances générales, ils n’ont pas accès aux données privées ou spécifiques à un profil professionnel. Pour pallier cette limitation, ce projet repose sur une approche appelée **Retrieval Augmented Generation (RAG)**, qui combine la puissance des LLMs avec des données internes et personnalisées.

Mise en place de l’environnement

La première étape du projet a consisté à mettre en place un environnement Python. Sur macOS, l’installation de bibliothèques via `pip` nécessite l’utilisation d’un environnement virtuel. J’ai donc installé **Python 3.13** et configuré un environnement Python afin de pouvoir installer les dépendances nécessaires au projet.

Par la suite, j’ai créé un compte **Upstash** ainsi qu’une base de données vectorielle nommée « *base Andrea* ». Cette base est destinée à stocker les vecteurs générés à partir des documents utilisés.

Enfin, j’ai effectué plusieurs tests afin de vérifier le bon fonctionnement de la clé OpenAI, de l’URL de la base de données ainsi que du token d’authentification.

Préparation des données

Concernant la préparation des données, j’ai commencé par créer **quatre fichiers au format Markdown** (`.md`).

Chaque fichier contient plusieurs sections délimitées par des titres précédés du symbole `#`. Ces

balises Markdown servent de repères permettant de structurer et de séparer les différentes informations.

Indexation des données

L'étape suivante a été l'**indexation des documents**. Grâce à la structure Markdown, les fichiers ont été découpés en plusieurs petites parties appelées **chunks**.

Ces chunks ont ensuite été envoyés vers la base de données Upstash, qui s'est chargée de transformer automatiquement leur contenu en **vecteurs numériques** (embeddings).

À l'issue de cette étape, la base de données contenait l'ensemble des informations issues des fichiers Markdown, mais sous forme vectorielle, permettant une recherche sémantique efficace.

Création de l'agent IA

Cette étape est centrale dans le projet : elle consiste à créer un **agent intelligent** capable d'interroger la base de données vectorielle et de générer des réponses pertinentes.

Le fonctionnement est le suivant :

- l'utilisateur pose une question à l'agent ;
 - la question est transformée en vecteur ;
 - ce vecteur est comparé à ceux stockés dans la base de données ;
 - selon le paramètre **k** (qui définit le nombre de résultats retournés), l'agent récupère les correspondances les plus pertinentes ;
 - enfin, à l'aide d'une instruction (prompt), l'agent génère une réponse adaptée, par exemple : « *répondre en français* » ou « *répondre comme si tu étais Andrea* ».
-

Interface utilisateur

La dernière étape consiste à intégrer l'agent IA dans une **interface utilisateur**. Pour cela, j'ai utilisé **Streamlit**, qui permet de créer rapidement une application web interactive.

Cette interface offre une expérience utilisateur plus agréable, avec une véritable conversation en temps réel entre l'utilisateur et le chatbot.

Conclusion

Au début, j'ai réalisé ce projet entièrement à l'aide de l'intelligence artificielle, sans réellement comprendre ce que je faisais. Cela rendait son utilisation compliquée, car même si l'IA est un outil très puissant, elle exécute uniquement ce qu'on lui demande : sans compréhension précise, les résultats manquent de clarté et de pertinence.

Après avoir consulté la documentation et demandé des éclaircissements au professeur, j'ai progressivement acquis une meilleure compréhension du projet et j'ai pu développer le code par moi-même. Je me suis néanmoins appuyé sur l'IA de manière plus ciblée, en lui posant des questions précises. Cette approche m'a permis à la fois de mieux comprendre mon travail et d'obtenir un code plus optimisé, passant de 270 lignes à seulement 70.

Ce projet m'a ainsi permis de renforcer significativement mes compétences, notamment dans le fonctionnement d'un chatbot.