HES-SO / DevPro

Table des matières

Introduction

Le développement de l'intelligence artificielle a permis de créer des solutions innovantes, et le projet en cours vise à intégrer un chatbot intelligent sur votre site web existant. Ce chatbot s'appuiera sur des technologies d'embeddings et de RAG (Retrieval-Augmented Generation) pour répondre aux questions des utilisateurs de manière précise et contextuelle. En se basant sur des documents existants, tels que des PDF, et une base de données, il sera capable de récupérer et générer des réponses pertinentes. Ce document présente le projet, les technologies choisies, ainsi que le design du système, en expliquant comment chaque élément a été sélectionné pour répondre aux besoins spécifiques de votre entreprise.

Objectifs du projet

L'objectif central du projet est de concevoir un chatbot basé sur l'intelligence artificielle qui améliorera considérablement l'expérience utilisateur sur le site DevPro. Ce chatbot vise à fournir un accès rapide et efficace à l'information tout en réduisant la charge de travail de l'équipe administrative. Plus spécifiquement, ce chatbot doit :

Comprendre les questions posées en langage naturel : Il sera en mesure d'analyser les demandes des utilisateurs, qu'il s'agisse de questions simples ou complexes, et d'identifier les besoins exprimés de manière contextuelle. Cela permettra d'offrir une interaction plus naturelle et intuitive avec les enseignants de la HES-SO et autres utilisateurs du site.

Fournir des réponses pertinentes et orientées vers l'utilisateur : Grâce à l'analyse des conversations, le chatbot pourra générer des réponses adaptées aux requêtes des utilisateurs, tout en pointant directement vers les pages correspondantes sur le site DevPro. Cela inclut des réponses liées aux qualifications, aux conditions d'obtention des attestations didactiques, ainsi qu'aux formations proposées par le centre DevPro, comme les niveaux et les fondamentaux des formations.

Accéder à des informations à partir de diverses sources : Le chatbot pourra récupérer des informations à partir de documents PDF, de bases de données ou d'autres ressources disponibles sur le site, afin de fournir des réponses fiables et précises.

Automatiser la gestion des demandes récurrentes : Ce projet vise à soulager l'équipe administrative de la gestion des questions fréquentes, en particulier celles liées aux processus d'attestation didactique, permettant ainsi aux utilisateurs d'obtenir des réponses sans solliciter directement l'administration. Cette automatisation permettra d'optimiser les interactions et de concentrer les ressources humaines sur des tâches à plus forte valeur ajoutée.

Tableau comparatif des technologies

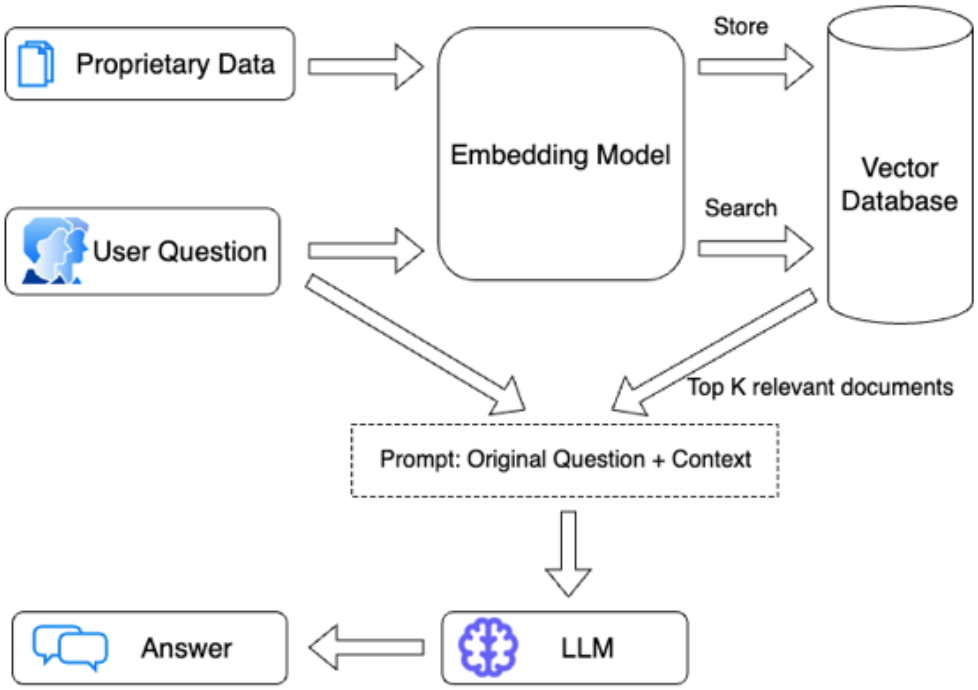
Critères	Rasa	Botpress	Haystack
Open-source	Oui	Oui	Oui
Orientation principale	Chatbots conversationnels avancés	Chatbots conversationnels avec interface visuelle	Récupération d'information et RAG (Retrieval-Augmented Generation)

Critères	Rasa	Botpress	Haystack
Gestion des dialogues	Excellente gestion d'intentions et d'entités	Interface graphique pour gestion des dialogues	Minimal, pas orienté pour les dialogues complexes
Utilisation principale	Chatbots complexes avec gestion des intentions et entités	Chatbots avec flux prédéfinis et interface visuelle	Systèmes de questions-réponses dynamiques et recherche documentaire
Base de données intégrée	Intégration avec des bases SQL/NoSQL (PostgreSQL, MongoDB)	Pas de base intégrée, nécessite une base externe	Intègre des bases de données vectorielles (FAISS, ChronoDB)
Simplicité d'utilisation	Nécessite des compétences techniques pour la configuration et la personnalisation	Interface visuelle, simple à utiliser pour les non-techniciens	Nécessite une expertise technique pour la configuration RAG
Facilité d'intégration	Intégration facile avec des services externes (APIs, bases de données, modèles IA)	Supporte les intégrations via connecteurs	Intègre facilement des bases de données vectorielles et des modèles de génération
Documentation et communauté	Très vaste, large communauté	Bonne documentation, mais communauté plus petite	Documentation claire, communauté en croissance
Avantages	- Spécialisé dans les chatbots conversationnels - Grande flexibilité - Open-source avec une communauté active	- Interface conviviale - Gestion des flux conversationnels sans codage - Open-source	- Optimisé pour les systèmes RAG - Spécialisé dans la récupération d'information et la génération de réponses contextuelles - Intègre des bases vectorielles
Inconvénients	- Complexe à configurer pour des tâches autres que des chatbots - Pas de support RAG natif	- Limité pour des cas d'usage complexes - Non adapté pour des systèmes RAG ou des tâches dynamiques	- Moins adapté pour des dialogues complexes - Nécessite une courbe d'apprentissage pour la mise en place d'un système RAG

Critères	Rasa	Botpress	Haystack
Cas d'usage idéal	Chatbots avancés avec gestion des intentions et des entités complexes	Chatbots simples à modérément complexes, avec flux conversationnels simples	Systèmes RAG dynamiques avec récupération d'informations à partir de documents volumineux

Système Design

Le système est conçu pour être modulaire et efficace, garantissant une gestion fluide des questions des utilisateurs tout en permettant de récupérer et générer des réponses basées sur des documents stockés.



1. **Frontend (fourni par le client)** : L'utilisateur pose une question via une interface existante sur le site. Cette interface sera connectée au chatbot via une nouvelle route ou un widget dédié.
2. **Backend (FastAPI)** : Le backend, développé avec **FastAPI**, est responsable de la gestion des requêtes et des réponses. Il transmet les requêtes de l'utilisateur au modèle d'IA et récupère les réponses pour les renvoyer au frontend.
3. **Modèle d'Embeddings (LLaMA / BERT)** : Lorsqu'une question est posée, le backend transmet la question au **modèle d'embeddings** qui la convertit en vecteur pour faciliter la recherche sémantique.

4. **ChromaDB (Base de données vectorielle)** : Une fois la question convertie en embedding, **ChromaDB** compare cet embedding avec ceux des documents existants pour identifier les réponses les plus pertinentes.
5. **Extraction PDF (pydf2)** : Si nécessaire, des informations supplémentaires peuvent être extraites de documents PDF grâce à **pydf2**.
6. **LLM (LLaMA / BERT)** : Le modèle **LLM** combine les informations récupérées avec la question de l'utilisateur pour générer une réponse contextualisée.
7. **Retour au Frontend** : Le backend envoie ensuite la réponse générée à l'interface utilisateur pour être affichée à l'utilisateur.

Technologies choisies et justification

Le choix des technologies pour ce projet a été guidé par des critères de performance, de flexibilité, et d'intégration avec l'existant. Voici les principales technologies retenues et pourquoi elles ont été sélectionnées :

1. Rasa

Rasa est un framework open-source spécialisé dans la création de chatbots conversationnels avancés. Il permet de gérer des dialogues complexes tout en offrant une grande flexibilité pour l'intégration de modèles IA externes. La raison pour laquelle Rasa a été choisie est sa capacité à comprendre les intentions et les entités des utilisateurs tout en permettant une personnalisation complète du flux de conversation. De plus, Rasa bénéficie d'une large communauté et d'une documentation complète, facilitant ainsi le développement et le support du projet.

2. pydf2

Pour l'extraction d'informations à partir de documents PDF, le choix s'est porté sur **pydf2**, une bibliothèque Python efficace qui permet de traiter rapidement les documents et d'en extraire le contenu textuel pertinent. Cela est essentiel pour fournir des réponses basées sur des documents PDF stockés, par exemple des documents de qualification ou des politiques internes. L'intégration avec d'autres technologies du projet est également fluide, ce qui garantit une utilisation simple et rapide.

3. ChromaDB

ChromaDB a été choisi comme base de données vectorielle pour stocker les **embeddings** des documents et des requêtes. Les embeddings permettent de représenter les documents sous forme de vecteurs, ce qui facilite la recherche sémantique. Lorsque l'utilisateur pose une question, ChromaDB permet de comparer l'**embedding** de la question avec les embeddings des documents et de trouver rapidement les informations pertinentes. Ce choix est justifié par la performance de ChromaDB dans les recherches vectorielles et son intégration facile avec les frameworks IA.

4. LLaMA / BERT

Pour le traitement des questions en langage naturel, nous avons opté pour des modèles IA comme **BERT** et **LLaMA**. Ces modèles sont capables de convertir les questions et les

documents en **embeddings** et de générer des réponses contextuelles. **BERT**, modèle bien établi, excelle dans l'analyse sémantique, tandis que **LLaMA**, plus récent, offre une meilleure efficacité en termes de performance et de génération de texte. Ces modèles sont essentiels pour comprendre le sens des questions posées et formuler des réponses pertinentes.

Conclusion

En conclusion, l'architecture proposée est optimisée pour répondre aux besoins du projet. Le chatbot intelligent sera capable de traiter des questions complexes, de récupérer des informations à partir de documents existants, et de générer des réponses précises et contextuelles. Les choix technologiques effectués garantissent une intégration fluide avec l'infrastructure existante, tout en offrant des performances et une flexibilité optimales. Cette solution permettra d'améliorer l'efficacité de la gestion des questions des utilisateurs, tout en réduisant la charge de travail sur l'équipe administrative.

Source

https://www.linkedin.com/pulse/what-retrieval-augmented-generation-rag-why-can-save-you-nawaz-mlsnc/