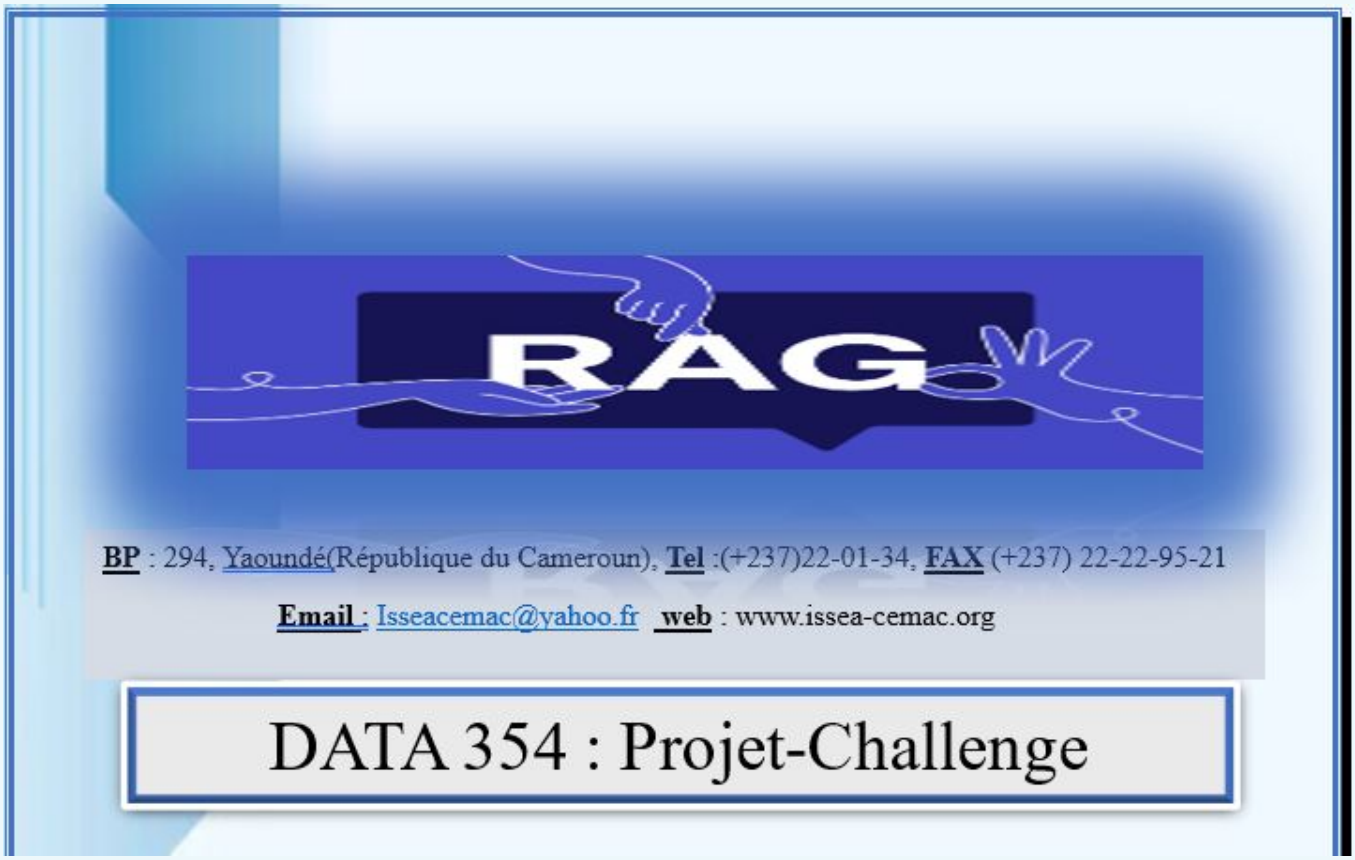


GEN AI, construction d'un agent conversationnel avec approche RAG : cas du site ecofoin

About The Authors

Rédigé par : TAGNE TCHINDA Rinel (Ingénieur Statisticien Economiste-Finissant et Data-Scientist)

Lien Github du Travail :GitHub - Data 354-GEN$_I$A – 2025

Lien Streamlit du deploiement de l'application : Dans le code (deploiement en local)

Résumé

Ce projet a pour but de créer un agent conversationnel spécialisé utilisant la Génération Augmentée par Récupération (RAG) pour répondre à des questions sur des documents. Le code permet de scraper des articles du site Ecofin, de les prétraiter et de les intégrer dans un pipeline RAG avec un modèle de langage (Mistral-7B-Instruct-v0.3) et un système de récupération (FAISS). Une interface conviviale développée avec Streamlit permet aux utilisateurs de poser des questions et d'obtenir des réponses. Les objectifs principaux ont été atteints : le scraping et le pipeline RAG fonctionnent, et l'interface est opérationnelle. Cependant, des améliorations peuvent être apportées pour ammeliorer la gestion des erreurs et l'interface utilisateur.

Abstract

This project developed a conversational agent using Retrieval Augmented Generation (RAG) to answer questions about specific documents. It scrapes and preprocesses articles from the Ecofin website, integrating them into a RAG pipeline with a language model (Mistral-7B-Instruct-v0.3) and a document retrieval system (FAISS). A Streamlit interface enables users to ask questions and receive answers. Key goals were achieved : scraping works, the RAG pipeline is operational and effective.

1 Introduction

L'intelligence artificielle a connu une avancée majeure avec l'émergence des modèles de langage de grande taille (LLM) tels que ChatGPT d'OpenAI. Ces modèles ont démontré des capacités impressionnantes pour générer du texte et engager des conversations cohérentes. Cependant, une limitation importante de ces modèles est leur incapacité à fournir des réponses basées sur des données spécifiques, à jour ou propriétaires qui ne faisaient pas partie de leur corpus d'entraînement. Cette limitation a conduit au développement de techniques comme la Génération Augmentée par Récupération (RAG), qui combine les forces des LLM avec la récupération de données externes pour fournir des réponses plus précises et contextuellement pertinentes.

2 Revue de la littérature

1.Fondements Théoriques et Empiriques : Analyse des discours politiques en NLP

L'analyse des discours politiques a longtemps été un domaine privilégié des sciences sociales, notamment en linguistique, en science politique et en sociologie. Avec l'avènement du Natural Language Processing (NLP), cette analyse a gagné en précision et en objectivité, grâce à des méthodes quantitatives et automatisées. Le NLP permet de traiter de grands volumes de textes, d'identifier des motifs récurrents, de mesurer les émotions et de comparer les discours à grande échelle. Cette section explore les fondements théoriques et les méthodes clés en NLP appliquées à l'analyse des discours politiques.

1.Revue de la littérature

Génération Augmentée par Récupération (RAG) :

Lewis et al. (2020) ont introduit le modèle RAG, qui combine un modèle de langage pré-entraîné avec un système de récupération de documents. Cette approche permet au modèle de générer des réponses basées sur des documents externes, ce qui est particulièrement utile pour les tâches nécessitant des informations à jour ou spécifiques à un domaine. L'étude a démontré que RAG surpasse les modèles de langage traditionnels dans les tâches nécessitant une précision factuelle et une pertinence contextuelle.

Guu et al. (2020) ont exploré l'utilisation des modèles augmentés par récupération dans le cadre de réponses à des questions en domaine ouvert. Leur travail a mis en lumière l'importance de l'intégration de sources de connaissances externes pour améliorer la capacité du modèle à répondre à des questions complexes.

Modèles de Langage de Grande Taille (LLM) :

Brown et al. (2020) ont introduit GPT-3, un modèle de langage de pointe avec 175 milliards de paramètres. L'étude a montré la capacité du modèle à effectuer une large gamme de tâches de traitement du langage naturel avec un minimum de réglage spécifique à la tâche. Cependant, les auteurs ont également noté les limites du modèle dans la gestion des tâches nécessitant l'accès à des données externes ou propriétaires.

Raffel et al. (2020) ont présenté T5 (Text-To-Text Transfer Transformer), un cadre unifié pour diverses tâches de NLP. L'étude a souligné l'importance de l'apprentissage préalable sur des ensembles de données divers.

Applications de RAG dans l'industrie :

Borgeaud et al. (2021) ont discuté de l'application de RAG dans les environnements d'entreprise, où le besoin de réponses précises et contextuellement pertinentes est critique. L'étude a mis en avant l'utilisation de RAG dans les systèmes de support client et de gestion des connaissances, démontrant son efficacité pour améliorer la précision des réponses et la satisfaction des utilisateurs.

2. Collecte et prétraitement des données :

Scraper les articles du site Ecofin pour la dernière semaine. La paticularité de notre projet est sa capacité à scrapé les données actualisées à chaque fois que le modele est lancé. mais pour des soucis de memoire notre code ne stocke pas les données de maniere à nous constituer une solide base de données dans le temps.

Prétraiter les données scrapées pour les rendre utilisables dans le pipeline RAG.

3. Mise en place du pipeline RAG :

Configurer un système de récupération de documents pertinents en fonction des requêtes utilisateur.

Intégrer le système de récupération avec un modèle de langage pré-entraîné (par exemple, Mistral-7B-Instruct-v0.3) pour générer des réponses contextuelles.

4. Développement de l'interface utilisateur :

Créer une interface web conviviale avec Streamlit pour permettre aux utilisateurs d'interagir avec le chatbot.

S'assurer que l'interface fournit des réponses claires et concises, avec des références aux sources utilisées.

5. Tests et évaluation :

Effectuer des tests approfondis pour s'assurer que le chatbot fournit des réponses précises et pertinentes.

Évaluer la performance du système sur la base des retours utilisateurs et apporter les ajustements nécessaires.

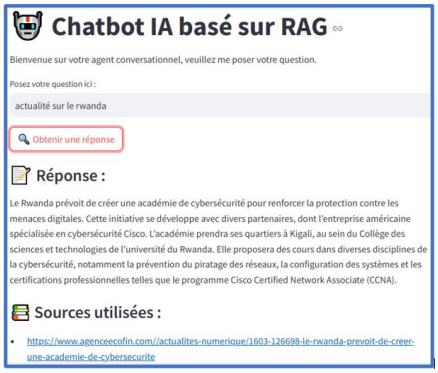
6. Documentation et déploiement :

Nous avons déployé la solution en utilisant ngrok, en l'exécutant sur Google Colab, qui offre des capacités de mémoire supérieures et des processeurs plus puissants pour améliorer la vitesse de réponse aux requêtes.

Nous avons également préparer une documentation complète pour guider les utilisateurs dans la configuration et l'exécution du système. De plus, nous avons fourni un accès au dépôt GitHub contenant le code et la documentation pour faciliter l'utilisation et la collaboration.

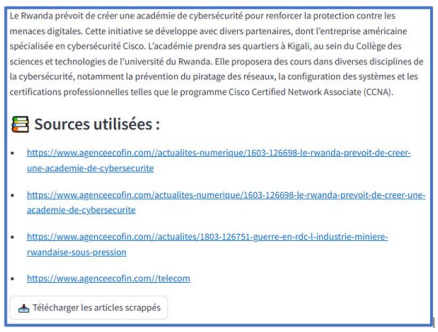
7. Prise en main de l'application :

lees images qui suivent permettent de presenter un extrait de l'interface de l'application de l'agent conversationnel.



Source : nos travaux sous python à partir des données scrapés su le site ecofoin

L'image suivante montre que : les sources de données ayant permis à l'agent de repondre sont proposées à l'utilisateurs ; qui peut donc directement avoir acces à celà et meme consulté celà.



L'image suivante monte comment touts les données scrappées peuvent etre telecharger directement.

Source : nos travaux sous python à partir des données scrapés su le site ecofoin

3 Recommandations et ammeliorations

Limites du Projet

Dépendance à la Structure du Site Web : Le scraping repose sur la structure HTML actuelle du site. Tout changement dans cette structure peut entraîner des erreurs ou des données manquantes.

Gestion des Erreurs : Bien que le code inclut une gestion basique des erreurs, des cas extrêmes ou des erreurs inattendues peuvent toujours survenir, notamment en raison de problèmes de réseau ou de modifications sur

le site cible.

Limitations du Modèle de Langage : Le modèle de langage utilisé (Mistral-7B) a ses propres limites en termes de compréhension contextuelle et de génération de réponses, ce qui peut entraîner des informations inexactes ou peu pertinentes.

Ressources de Calcul : L'utilisation de modèles de langage et de l'indexation FAISS peut nécessiter des ressources de calcul significatives, limitant l'accessibilité pour des utilisateurs disposant de capacités techniques ou financières limitées.

Recommandations

Mises à Jour du Scraper : Prévoir des mises à jour régulières du scraper pour s'adapter aux changements du site ou intégrer de nouveaux sites d'actualités pertinents.

Formation Supplémentaire du Modèle : Envisager d'affiner le modèle de langage sur des données spécifiques à la thématique pour améliorer la pertinence des réponses.

Sécurité des Données : Mettre en œuvre des mesures de sécurité pour protéger les données scrappées et les informations des utilisateurs, en respectant les réglementations sur la protection des données.

4 references :

Documentation du modele utilisé :

https ://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3