

1. Introduction

Ce projet de stage a pour objectif de développer une application web permettant d'exploiter des techniques avancées de traitement du langage naturel (NLP) associées au modèle Retrieval Augmented Generation (RAG). Le système proposé offrira aux utilisateurs la possibilité de télécharger plusieurs fichiers, sur lesquels le modèle s'auto-entraînera, afin de pouvoir ensuite interagir avec ces données via une interface de chat intuitive.

2. Objectifs du Projet

- **Développer un système web interactif permettant :**
 - L'upload et le prétraitement de documents multiples (PDF, .txt, .docx).
 - L'entraînement d'un modèle NLP de type RAG sur ces documents.
 - Une interface de chat pour poser des questions et obtenir des réponses basées sur le contenu des documents.
- **Entraînement automatique du modèle :** Implémenter un mécanisme permettant au modèle de s'auto-entraîner sur les fichiers importés.
- **Interface de chat interactive :** Offrir aux utilisateurs la possibilité de poser des questions et d'obtenir des réponses contextuelles basées sur le contenu des fichiers.
- **Sécurité et confidentialité :** Garantir la protection des données utilisateurs, tant lors du téléchargement que lors de l'interaction.
- **Performance et scalabilité :** Concevoir une architecture évolutive et performante pour répondre à un nombre croissant d'utilisateurs et de données.

3. Description Fonctionnelle

3.1 Téléchargement de Fichiers

- **Formats acceptés :** PDF, TXT, CSV etc.
- **Interface utilisateur :** Interface intuitive pour la sélection et le téléchargement de plusieurs fichiers simultanément.
- **Validation :** Contrôle de la validité et du format des fichiers téléchargés.

3.2 Entraînement Automatique du Modèle

- **Extraction de contenu :** Analyse et extraction des données pertinentes à partir des fichiers.
- **Auto-entraînement :** Mise en place d'un pipeline permettant au modèle NLP de s'auto-entraîner sur le contenu des fichiers téléchargés.
- **Mise à jour du modèle :** Possibilité de ré-entraîner ou de mettre à jour le modèle en fonction des nouveaux fichiers ajoutés.

3.3 Interface de Chat

- **Chatbot intégré :** Interface de discussion permettant aux utilisateurs de poser des questions.
- **Réponses contextuelles :** Le modèle RAG utilise les informations extraites pour générer des réponses pertinentes et contextualisées.
- **Historique des échanges :** Conservation des sessions de chat pour consultation ultérieure.

3.4 Administration et Gestion

- **Interface administrateur :** Outils de gestion des utilisateurs, des fichiers téléchargés et des sessions de chat.
- **Reporting :** Statistiques sur l'utilisation du système et les interactions.

4. Contraintes Techniques

- **Technologies web :**
 - **Backend :** Python (Django, Flask) ou Node.js.
 - **Frontend :** JavaScript avec frameworks comme React ou Angular, ou bootstrap.
- **Modèle NLP :** Implémentation ou intégration d'un modèle RAG ou équivalent.
- **Base de données :** Système de gestion de bases de données relationnelles ou NoSQL pour stocker les fichiers, les métadonnées et l'historique des chats.
- **Sécurité :** Protocoles de chiffrement, gestion des accès et protection des données sensibles.
- **Scalabilité :** Conception modulaire permettant d'ajouter de nouvelles fonctionnalités et de supporter une montée en charge.
- **Compatibilité :** Application responsive, compatible avec les principaux navigateurs web (Chrome, Firefox, Edge, Safari).

5. Livrables Attendus

- **Documentation Technique :**
 - Description de l'architecture du système.
 - Choix technologiques et méthodes d'implémentation.
- **Code Source :**
 - Développement du site web avec gestion de versions (Git).
 - Instructions claires pour le déploiement et la maintenance.
- **Interface Web Fonctionnelle :**
 - Prototype opérationnel incluant le téléchargement de fichiers, l'auto-entraînement du modèle et l'interface de chat.
- **Rapport de Stage :**
 - Documentation détaillée du processus de développement, des tests effectués et des perspectives d'évolution.
- **Démonstration et Tests :**
 - Présentation de la solution en conditions réelles.
 - Résultats des tests unitaires, d'intégration et de performance.

6. Planning Prévisionnel

Un planning détaillé sera élaboré en concertation avec l'encadrant, comprenant les phases suivantes :

- **Phase 1 : Analyse et spécifications** – Recherche, définition des besoins et rédaction du cahier des charges .
- **Phase 2 : Conception de l'architecture et du design** – Élaboration des maquettes et de l'architecture technique
- **Phase 3 : Développement et intégration** – Implémentation des fonctionnalités principales et intégration du modèle NLP .
- **Phase 4 : Tests et débogage** – Vérification de la qualité et de la performance du système .
- **Phase 5 : Rédaction du rapport et préparation de la soutenance** – Documentation finale et préparation à la présentation .

7. Critères de Réussite

- **Fonctionnalités complètes et opérationnelles** : Le système doit permettre le téléchargement, l'auto-entraînement et l'interaction via le chat sans dysfonctionnement majeur.
- **Interface utilisateur intuitive** : L'application doit être ergonomique et facile à utiliser.
- **Performance et rapidité** : Réponses générées rapidement et pertinence des interactions.
- **Sécurité des données** : Protection effective des informations sensibles et respect des normes de confidentialité.
- **Documentation exhaustive** : Documentation technique et utilisateur complète.