

1. Introduction

Le projet consiste à développer une application web permettant la détection de plagiat entre un ensemble de documents préalablement enregistrés dans une base de données et un nouveau document uploadé par l'utilisateur. Grâce à l'utilisation des techniques avancées de traitement du langage naturel (NLP), le système analysera le contenu des fichiers afin de repérer les similitudes et de générer un rapport détaillé sur les éventuels cas de plagiat.

2. Objectifs du Projet

- **Développement d'une plateforme web conviviale :** Permettre à l'utilisateur d'ajouter et de gérer ses fichiers dans une base de données.
- **Détection de plagiat automatisée :** Mettre en place un mécanisme basé sur des techniques NLP pour comparer un nouveau document aux documents existants.
- **Génération d'un rapport détaillé :** Fournir une analyse complète mettant en évidence les passages similaires et le taux de plagiat.
- **Sécurité et confidentialité :** Assurer la protection des données et des documents déposés par les utilisateurs.
- **Performance et scalabilité :** Garantir une réponse rapide même en cas d'importante volumétrie de données.

3. Description Fonctionnelle

3.1 Gestion de la Base de Données Documentaire

- **Ajout de fichiers multiples :**
 - Permettre aux utilisateurs d'uploader plusieurs fichiers simultanément dans leur espace de stockage.
 - Formats supportés : PDF, DOCX, TXT, etc.
- **Organisation des fichiers :**
 - Possibilité de classer et de rechercher les fichiers existants via des filtres (date, type, nom, etc.).

3.2 Upload et Analyse d'un Nouveau Document

- **Téléversement du document à vérifier :**
 - Interface simple pour uploader le fichier devant être comparé aux fichiers existants.
- **Prétraitement des documents :**
 - Extraction du texte, normalisation et segmentation du contenu.
- **Comparaison et détection de plagiat :**
 - Utilisation d'algorithmes NLP pour analyser les similitudes textuelles entre le document uploadé et ceux stockés dans la base.
 - Possibilité d'utiliser des techniques telles que le calcul de similarité sémantique, le fingerprinting ou l'indexation vectorielle.

3.3 Génération du Rapport de Plagiat

- **Détail des similarités :**
 - Présentation des passages ou des segments de texte similaires.

- Indication du taux de plagiat global et par sections du document.
- **Visualisation intuitive :**
 - Mise en évidence graphique (par exemple, surlignage, barres de pourcentage) pour faciliter la compréhension du rapport.
- **Export du rapport :**
 - Possibilité de télécharger le rapport au format PDF ou DOCX.

4. Contraintes Techniques

- **Technologies Web :**
 - **Backend :** Python (Django, Flask) ou Node.js.
 - **Frontend :** JavaScript avec frameworks tels que React ou Angular.
- **Modules NLP :**
 - Intégration de bibliothèques de traitement du langage naturel (par exemple, spaCy, NLTK, ou des modèles transformer).
- **Base de Données :**
 - Utilisation d'un SGBD relationnel (MySQL, PostgreSQL) ou NoSQL pour stocker les documents et métadonnées.
- **Sécurité :**
 - Mise en œuvre de protocoles de chiffrement, gestion des droits d'accès et sauvegarde régulière des données.
- **Performance et Scalabilité :**
 - Optimisation des requêtes et utilisation de techniques de mise en cache pour améliorer le temps de réponse.
- **Compatibilité :**
 - Application responsive, accessible sur les principaux navigateurs web (Chrome, Firefox, Edge, Safari).

5. Livrables Attendus

- **Documentation Technique :**
 - Description détaillée de l'architecture, des algorithmes utilisés et des choix technologiques.
- **Code Source :**
 - Développement complet de l'application web avec gestion de versions (Git).
 - Instructions pour le déploiement, la configuration et la maintenance.
- **Interface Web Fonctionnelle :**
 - Site web opérationnel permettant la gestion des fichiers, le téléchargement et l'analyse de plagiat.
- **Rapport de Stage :**
 - Document détaillant le processus de conception, les développements réalisés, les tests effectués et les perspectives d'évolution.
- **Démonstration et Tests :**
 - Présentation du système en conditions réelles, y compris les résultats des tests unitaires, d'intégration et de performance.

6. Planning Prévisionnel

Le planning sera défini en collaboration avec l'encadrant et pourra se décliner en plusieurs phases :

- **Phase 1 : Analyse et spécifications**
 - Recherche, collecte des besoins et rédaction du cahier des charges.
- **Phase 2 : Conception de l'architecture et prototypage**
 - Élaboration des maquettes et choix des technologies.
- **Phase 3 : Développement de l'application**
 - Implémentation des modules de gestion des fichiers, analyse NLP et génération du rapport.
- **Phase 4 : Tests et optimisation**
 - Validation fonctionnelle et optimisation des performances.
- **Phase 5 : Rédaction du rapport final et préparation de la soutenance**
 - Documentation complète et préparation de la présentation du projet.

7. Critères de Réussite

- **Fonctionnalité complète :**
 - Le système doit permettre l'ajout de fichiers, l'upload du document à vérifier et la génération d'un rapport précis sans dysfonctionnement.
- **Interface utilisateur intuitive :**
 - L'ergonomie de la plateforme doit faciliter la navigation et l'utilisation des fonctionnalités.
- **Performance et rapidité :**
 - Les analyses de plagiat doivent être effectuées rapidement, même pour un volume important de documents.
- **Sécurité des données :**
 - Protection efficace des fichiers et respect des normes en matière de confidentialité.
- **Documentation exhaustive :**
 - Documentation technique et utilisateur complète et à jour.