Réalisation d’une application de type RAG

**      **

Certification : Développeur d’application en Intelligence Artificiel

Marseille - Nice

LOMBARDI Joachim

Table des matières

[1. Réaliser la collecte, le stockage et la mise à disposition des données d’un projet en intelligence artificielle 3](#_Toc185683848)

[I. Programmer la collecte de données depuis plusieurs sources pour un projet en intelligence artificielle 3](#_Toc185683849)

Afin de réaliser mon application de manière sécurisée, efficace et structuré, j’utilise le Framework Django que l’on utilise dans mon entreprise.

Django est un Framework populaire, datant de 2005, qui permet de créer des applications robustes et sécurisées.

Django comprend :

* Un moteur ORM utilisant des objet python plutôt que des requêtes SQL, ce qui simplifie l’interaction avec la base de données.
* Une interface d’administration pour gérer les données. (Notamment le CRUD).
* Un système de routage URL séparé de la vue.
* Un système de templates HTML qui permet de séparer la logique métier (dans les vues et modèles) de l’affichage (dans les templates). Il intègre également des variables, des filtres pour transformer les données et des balises pour la logique.
* La sécurité intégrée avec la protection contre les vulnérabilités courantes comme :
* Les attaques XSS (Cross-Site Scripting) : Django échappe automatiquement les variables rendues dans les templates pour empêcher l'exécution de code malveillant.
* Les injections SQL.
* Les attaques CSRF (Cross-Site Request Forgery) : Django utilise un système de jetons CSRF pour valider l'origine des requêtes POST, PUT, PATCH et DELETE.
* Le clickjacking : Django inclut un middleware qui ajoute l'en-tête X-Frame-Options pour empêcher l'intégration des pages web dans des cadres (frames).
* Le stockage sécurisé des mots de passe.
* La modularité qui permet de diviser un projet en modules réutilisables.
* Un système de migration de base de données. Les migrations permettent de synchroniser facilement les changements apportés aux modèles Python avec la structure de la base de données.

Le langage utilisé est le Python car c’est celui utilisé pour appli en IA, de plus c’est celui qu’on utilise dans mon entreprise.

L’outil de versioning Git est utilisé pour stocker le script de l’application en ligne.

# Réaliser la collecte, le stockage et la mise à disposition des données d’un projet en intelligence artificielle

## Programmer la collecte de données depuis plusieurs sources pour un projet en intelligence artificielle

### Identification des contraintes techniques propres aux sources de données

Tout média numérique est susceptible d’être utilisée comme source de donnée, les documents bien évidemment, mais aussi les sites internet, les vidéos, les bandes sonores, les images…

Afin de pouvoir en extraire des données, il faut d’abord connaître les règles de confidentialité.

#### Règles de confidentialité

Ces règles servent à protéger les données personnelles comme :

* Le nom, le prénom, l’adresse, le numéro de téléphone.
* Les informations d'identification (numéro de carte d'identité, numéro de sécurité sociale, etc.).
* Les données numériques : adresses IP, cookies, géolocalisation.
* Les données sensibles : informations de santé, opinions politiques, orientation sexuelle, croyances religieuses, etc.

Selon la CNIL, les 7 grands principes des règles de protection des données personnelles sont les suivants :

**Principe 1** : Licéité, limitation des finalités, minimisation des données

Licéité : Les données personnelles ne doivent être collectées que pour des finalités légitimes, explicites et déterminées.

Limitation des finalités : Les données collectées ne peuvent être utilisées que pour les finalités pour lesquelles elles ont été collectées.

Minimisation des données : La quantité de données collectées doit être réduite au minimum nécessaire pour atteindre les finalités déterminées.

En d'autres termes, les entreprises doivent être transparentes sur la raison pour laquelle elles collectent des données et ne collecter que les informations dont elles ont réellement besoin. Les mentions légales sur votre site internet sont là pour les informer.

**Principe 2** : Exactitude

Les données personnelles doivent être exactes et, si nécessaire, mises à jour. Des mesures doivent être prises pour que les données inexactes soient rectifiées ou effacées.

Cela signifie que les entreprises doivent prendre des mesures pour s'assurer que les données personnelles qu'elles détiennent sont exactes et à jour.

**Principe 3** : Limitation du traitement

Les données personnelles ne doivent être conservées que pendant une durée proportionnelle aux finalités pour lesquelles elles sont traitées.

En d'autres termes, les entreprises doivent avoir une politique claire de conservation des données et supprimer les données lorsqu'elles ne sont plus nécessaires.

**Principe 4** : Intégrité et confidentialité

Les données personnelles doivent être traitées de manière à garantir leur sécurité, y compris la protection contre le traitement non autorisé ou illicite et contre la perte, la destruction ou les dommages accidentels, par des mesures techniques ou organisationnelles appropriées.

Cela signifie que les entreprises doivent mettre en place des mesures de sécurité adéquates pour protéger les données personnelles contre les accès non autorisés, les utilisations abusives et les pertes accidentelles.

**Principe 5** : Responsabilisation

Le responsable du traitement est responsable du respect des principes susmentionnés et doit être en mesure de démontrer sa conformité.

Cela signifie que les entreprises doivent désigner un responsable de la protection des données (DPO) qui sera responsable de la mise en œuvre et du respect du RGPD.

**Principe 6** : Respect des droits des personnes

Les personnes ont le droit d'accéder à leurs données personnelles, de les rectifier, de les effacer, de limiter leur traitement, de s'opposer à leur traitement et de les transférer.

En d'autres termes, les entreprises doivent informer leurs clients de leurs droits en matière de protection des données et leur permettre d'exercer ces droits facilement.

**Principe 7** : Protection des données dès la conception et par défaut :

Le responsable du traitement doit mettre en œuvre des mesures techniques et organisationnelles appropriées pour protéger les données personnelles dès la conception du traitement et tout au long des produits et services.

Cela signifie que les entreprises doivent intégrer le traitement des données dans leur stratégie globale d'entreprise.

Ensuite, il existe également des contraintes lors de l’extraction des données :

#### Collecte des données

Comme indiqué dans l’introduction, il existe différents types de données outre les différents médias, un même type de source peux avoir plusieurs supports, par exemple ils peuvent être en format texte, JSON, CSV, base de données relationnelles… Ainsi, les méthodes d’extraction sont différentes et peuvent nécessiter du prétraitement comme pour les PDF afin d’avoir des données propres.

Ces méthodes d’extraction peuvent avoir des limites, comme lors du Scraping (extraction de données depuis une page web) où le nombre de requêtes peut être limité, le téléchargement de fichier là aussi peut être limité ou également l’accès à des bases de données protégées par mot de passe. Il peut également y avoir des problèmes de collecte dû au réseau ou bien à la limitation imposée selon la région du monde où se trouve l’utilisateur ou bien les données.

Certaines données sont très volumineuses et imposent des contraintes au niveau de l’espace de stockage et la mémoire, nécessitant des ressources informatiques importantes et des optimisations de traitement.

### Spécification techniques relatifs à l’extraction des données

L’objectif de mon projet est de réaliser une Appli utilisant les données médicales du site : pubmed.ncbi.nlm.nih.gov afin de poser des questions techniques sur ces données et d’obtenir des réponses en langage naturel ainsi que les sources ayant servis à produire ces réponses.

#### Pubmed

PubMed (pubmed.ncbi.nlm.nih.gov) est une ressource utilisée pour la recherche biomédicale et scientifique offrant un accès gratuit à une vaste base de données de références et de résumés d'articles scientifiques. Gérée par la National Library of Medicine (NLM) des États-Unis, cette plateforme regroupe des millions de publications provenant de revues spécialisées, couvrant des domaines tels que la médecine, la biologie, la pharmacologie et bien plus encore.

Cette source de donnée est une des sources de données utilisées par l’entreprise dans laquelle j’effectue mon alternance : Data Observer, car elle contient les publications des professionnels et des groupes de professionnels de santé. Un des principaux objectifs de l’entreprise étant justement d’identifier ces professionnels et ces groupes de professionnels par rapport à leurs publications. Si deux professionnels par exemple on les mêmes noms et prénoms on doit pouvoir identifier qui à produit quoi et pour ça, on utiliser leurs affiliations (les établissements dans lesquels ils sont travaillés), les postes qu’ils ont occupés, les villes où ils ont travaillé, les personnes avec qui ils ont travaillés…

J’ai également choisi cette source de donnée car dans l’ensemble elle est bien structurée : les titres, abstracts, auteurs et affiliations… sont très souvent identifié par les mêmes balises html ce qui en facilitent grandement l’extraction. De plus, comme précisé dans le premier paragraphe, il s’agit de résumés ce qui simplifie le traitement des données lorsque l’on va vouloir les interroger.

#### Extraction des données

Pour extraire les données depuis le site PubMed, j’ai identifié trois outils :

* **Scrapy** qui est un Framework complet pour le scraping web.
* **Beautiful Soup** qui est une bibliothèque axée sur la manipulation du HTML
* **L’api PubMed** qui permet de récupérer les différentes données, d’effectuer des recherches basées sur des mots-clés, des termes MeSH (Medical Subject Headings), ou des identifiants uniques (comme les PMID). Les résultats sont donnés sous format JSON ou XML.

Connaissant déjà l’outil Beautiful Soup qui est également celui utilisé par mon entreprise et mon extraction étant simple, c’est celui que j’ai choisis.

J’ai choisi d’extraire les données de deux maladies : multiple sclérosis (une maladie qui revient souvent dans les données de l’entreprise) et herpès zoïster (une autre maladie pour alimenter la base de données mais dont le contenu était plus limité) sur la période 2024. Pour ce faire, j’ai scraper les liens de toutes les pages web traitant de ces deux maladies et j’ai scraper le contenu des articles associés à savoir :

* Le titre de la revue dans laquelle un article est publié
* La date de publication de l’article
* Le titre de l’article
* Le résumé de l’article
* L’id pubmed de l’article
* Le doi (Digital Object Identifier) permettant d’identifier et de localiser du contenu numérique.
* Le disclosure indiquant les éventuels conflits d’intérêts.
* Les termes MeSH (Medical Subject Headings) sont un système de vocabulaire contrôlé et hiérarchique développé par la **National Library of Medicine (NLM)** des États-Unis. Ils sont utilisés pour indexer, classer et rechercher des articles scientifiques dans des bases de données biomédicales, comme PubMed et MEDLINE.
* L’URL de l’article.
* Les auteurs de l’article et leurs affiliations.

Pour chaque extraction, on n’extrait les données que si le résumé est présent. Si une donnée est manquante elle prend la valeur None.

#### Insertion des données en base

Les scripts de scrapping et d’insertion en base de données s’écrivent dans business\_logic (logique métier) car il s’agit d’un script propre à se projet, les fonctions courantes sont stockées dans utils.py.

Afin de réaliser l’extraction et l’insertion en base de données, j’ai créé deux commandes dans le fichier commands.py: commands scrap\_article et commands article\_to\_database.

Une fois scrapés, les articles sont stockés dans un JSON, puis de là insérer dans une base de données.

Le stockage dans un JSON permet de séparer les étapes de scraping et d’insertion en base de données. Ce qui est pratique car s’il y a un problème lors de l’insertion des données en base, il faut reprendre le scraping depuis le début, de plus stocker les données à deux endroits différents permet d’éviter la perte des données si la base de données est effacée, cela permet également de pouvoir modifier la structure de la base de données tout en conservant les données ou encore de faire une insertion dans plusieurs bases de données…

La base de données choisit est Postgres, car c’est celle utilisée en entreprise.

Postgres est open-source, elle existe depuis 1986 donc il y a une vaste communauté, beaucoup de modules et de documentation. Elle supporte les bases relationnelles et les objets.

J’ai défini quatre classes :

* *Affiliations* pour stocker les noms des affiliations.
* *Authors* pour stocker les noms des auteurs.
* *Article* pour stocker toutes les informations basiques liées aux articles et la relation Many-to-Many permettant de relier un article avec ces auteurs et pour chaque auteur ses affiliations.
* *Authorship* qui est la table intermédiaire avec trois clés étrangères pour les articles, les auteurs et les affiliations.

La connexion avec la base de données se fait dans settings.py.

Pour créer la base de données on va créer dans models.py des classes dont chacune correspondra à une table de la base de données. On définit ici, le type de variable, si elle peut être nulle, le nom des colonnes... Ensuite on valide les migrations avec la commande makemigrations et on crée la base de données avec migrate. La base de données est créée.

#### CRUD

Maintenant que toutes les données ont été ajoutés à la base, il faut pouvoir mettre à jour la base.

On peut créer un article, lire l’ensemble des articles, mettre à jour un article ou le supprimer.

Le CRUD (Create, Read, Update, Delete) se fait dans