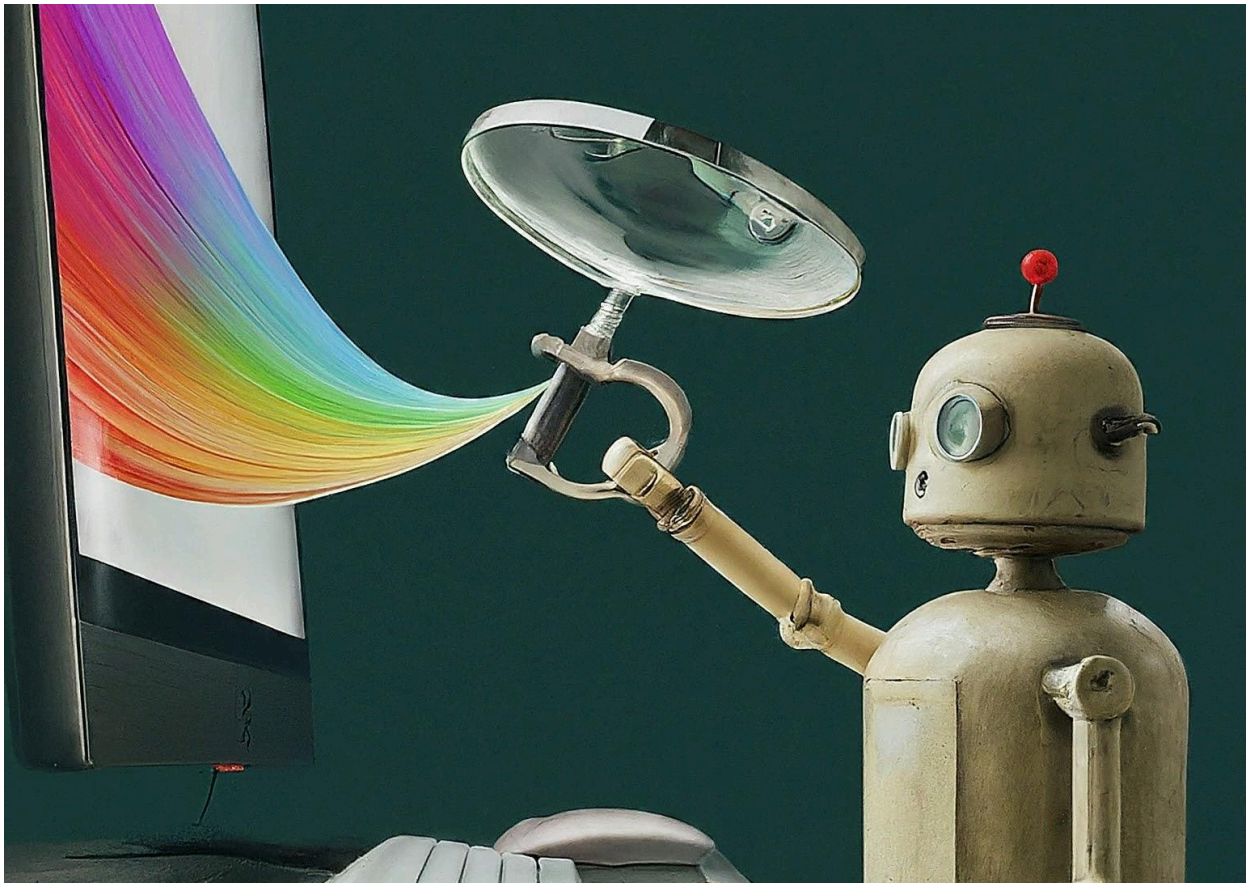
DATA SCRAPING CHALLENGE

Résumé



Amos Bationo

amosb.dev@gmail.com

Statistician - Mathematician - Data Scientist

Table des matières

INTRODUCTION

Ceci est une brève description du programme d'information automatique d'information dans le cadre de la compétition "Data Scraping Challenge" organisée par Open Burkina.

NOTE

Le système a été développé en langage Python 3.10 sous une distribution Linux. Bien que n'ayant pas été testé sur d'autres systèmes tels que Mac ou Windows, il devrait quand même fonctionner dans un environnement virtuel après installation des prérequis (fichier *requirements.txt*).

APPROCHE

Le système se compose de deux programmes (ou tâches) principaux indépendants implémentés en packages: ***scraper***, en charge de la collection et la compilation des journaux; et ***extractor***, nomme ***'association'*** en charge de l'extraction des journaux collectés et leur enregistrement sous format CSV.

Scraper

Le package *scraper* utilise le package Selenium pour pour accomplir sa tâche de de collecte et compilation des textes dans les journaux. L'opération est effectuée de la façon suivante:

1. **Collecte les liens des journaux**: Le programme commence par collecter les liens de l'ensemble des publications des journaux
2. **Assemblage des pages**: Pour chaque publication ou journal, collecter et rassembler toutes les pages du journal en un seul fichier
3. **Enregistrement de fichiers**: Chaque fichier contenant les pages d'une même publication (ou journal) est ensuite enregistré sous format text (.txt)

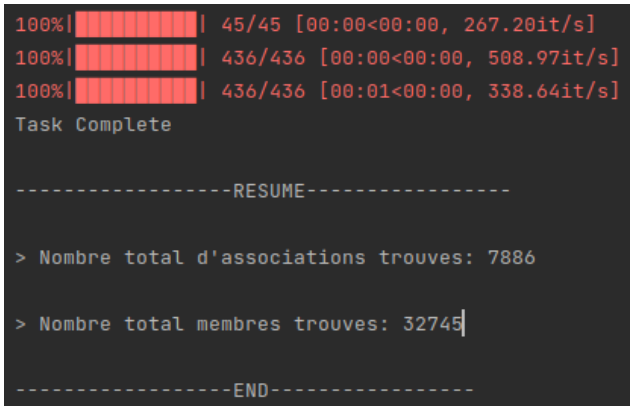
Association(extractor)

Le package ***association*** contient la class ***extractor*** qui est chargée de l'extraction des informations se trouvant dans les fichiers collectées par la tâche de collecte effectuée par le package ***scraper***. Le processus se déroule comme suit:

1. **Identification des fichiers**: Le programme commence par identifier l'ensemble des fichiers textes contenant les publications existant sur le disque
2. **Identifications de des sections clés**: Pour chaque journal, une méthode parcourt le contenu à l'aide d'une expression régulière pour extraire les sections faisant mention d'une déclaration d'association
3. **Extraction d'information**: Pour chaque section identifiées, le programme extrait deux types d'information qui sont les détails des associations et la liste de leurs membres et leurs rôles dans l'association. Toutes les extractions d'informations sont effectuées en utilisant uniquement les expressions régulières.
4. **Enregistrement des informations d'associations et membres**: Pour chaque publication, une fois que les associations et leurs membres extraient du journal, ils sont enregistrés sous format json dans deux dossiers temporaires l'un pour les associations et l'autres pour les membre
5. **Création de fichier CSV**: Le programme s'achève par la création de deux fichiers csv. L'ensemble des fichiers json se trouvant dans les dossiers temporaires association et membres sont collectés séparément, transformés en data frame pandas puis exportés en deux fichiers csv

LE TEMPS

Il est important de noter que faire dérouler le programme sur l'ensemble des journaux du site web prendra plusieurs heures, voire jours, suivant la performance de l'ordinateur et de la connexion internet utilisés. Le programme se termine par un résultat similaire a celui dans l'image ci-dessous.



APPROCHE ALTERNATIVE: IA Générative

Une autre approche serait d'inclure un modèle d'intelligence artificielle générale dans le stage d'extraction d'information tout juste après l'identification des sections relatives aux déclarations d'existence d'associations. Il faudrait sûrement écrire un prompta avec une température zéro afin d'avoir des résultats plus deterministics tout en évitant tout élans de créativité de la part du modèle.

Llama 2, un modèle open source, donc gratuite d'utilisation pourra potentiellement convenir vu que la tâche est relativement peu complexe. Même si Llama 2 donne de bons résultats, Chatgpt reste le leader en termes de performance. Son API sous version premium permet une intégration fluide dans ce type de programme.

Mais pourquoi n'avons-nous pas utilisé ces modèles IA pour le projet?

La réponse se subdivise comme suit:

1. **Ressources matérielles**: Dans le cas Llama 2, bien qu'il soit de le faire fonctionner en local, il nécessite une bonne carte graphique d'accélération. A défaut, cela le temps d'exécution se verra rallongé. Sur du CPU, le temps pour l'extraction d'information d'une seule association sera équivalent au temps mis par une expression régulière d'en extraire pour plusieurs dizaines d'associations
2. **Ressources financières et open source mind**: L'avantage avec Chatgpt est qu'il n'utilise pas les ressources locales pour faire ses inference mais plutot les serveurs de OpenAI. Ceci offre une bonne adaptabilité mais engendre des coûts. Aussi, ces coûts mettent en mal l'esprit d'open source du projet.

RÉSULTATS

Indicateurs

Les résultats du programme sont consignés dans le tableau ci-dessous.

Resultats de l'execution du programme

Indicateurs	Valeurs
Nombre de journaux collectés	473
Journaux contenant au moins une déclaration d'existence	436
Nombre de total de déclarations d'associations	7886
Nombre de total de membres d'associations	32745

Data

La description des deux bases de données des deux fichiers csv se présente comme suite:

Table des associations

Colonnes	Descriptions	Note
number	Numéro de déclaration d'existence de l'association. Peut être vide si la structure de la déclaration est inconsistante	Cas vide: 4/7886
date	Date de la déclaration d'existence de l'association	
dénomination	Le nom de l'association	
abbreviation	L'acronyme de l'association	
siege	Le siège de l'association	

id	Unique identifiant de l'association obtenue en appliquant un cryptage MD5 sur la colonne "number"	
journal_link	Le lien ve le journal en ligne d'où vient la déclaration d'existence	Clé primaire

Table des membres des associations

Colonnes	Descriptions	Note
association_id	Identifiant de l'association à laquelle la personne fait partie	Clé étrangère > association.id
role	Le poste qu'occupe la personne au sein de l'association	
name	Nom complet du membre de l'association. Suivant les cas elle contient aussi les adresses et numéros de téléphone	

COMMENT LANCER LE PROGRAMME

Suivez le étapes suivantes afin de pouvoir lancer le programme:

- Étape 1 (pas obligatoire mais recommandée): Créer une
- Étape 2: Assurez- vous que vous possédez le code source du projet. Vous pouvez le télécharger manuellement à ce lien () ou le cloner à l'aide de votre terminal (ou invite de commande) en tapant la commande git ci-dessous:

  ```
  git clone git@github.com:bationoA/data-challenge-journaux-bf.git
  ```

- Étape 3: Installer packages requis se trouvant dans le fichier *requirements.txt*

  ```
  pip install -r requirements.txt
  ```

- Étape 4: A présent vous êtes prêt à lancer le programme avec la commande

  ```
  python main.py <parametre 1> <valeur 1> <parametre 2> <valeur 2> …
  ```

Les différent paramètres et leurs valeurs sont listés dans le tableau suivant:

Tableau des paramètres

Paramètres	Descriptions	Options
-t : Task	Signifie que l'on veut définir la ou les tâches à exécuter	• **s**: Scraping. Collecter uniquement les journaux en ligne. S'arrête une fois cette tâche accomplie • **e**: Extract. Extract et uniquement les informations des journaux collectés. Ignore la phase de téléchargement. • **b**: Both. Exécute les deux tâches dans l'ordre collecte puis extraction
-st : Scraping threads	Nombre maximal de téléchargement simultanés. Dépend du processeur	Doit être supérieur à 0 et inférieure ou égale au nombre maximal de threads du processeur. Défaut = 1
-et : Extraction threads	Nombre maximal de fichiers journal à traiter simultanément	Doit être supérieur à 0 et inférieure ou égale au nombre maximal de threads du processeur. Défaut = 1

Exemples de commandes

La commande suivante lance **uniquement le téléchargement des journaux**. Tout au long de l'exécution de la tâche, le programmes téléchargera **3 journaux maximum** de façon simultanée:

```
python main.py -t s -st 3
```

La commande suivante lance **uniquement l'extraction des détails** des associations et leurs membres se trouvant dans les journaux. Tout au long de l'exécution de la tâche, le programmes examinera **5 journaux maximum** de façon simultanée:

```
python main.py -t e -et 5
```

La commande suivante lance le **téléchargement des journaux puis l'extraction des détails** des associations et leurs membres se trouvant dans les journaux. Tout au long de l'exécution de la tâche, le programme téléchargera **2 journaux maximum** de façon simultanée puis examinera **4 journaux maximum** de façon simultanée.

```
python main.py -t b -st 2 -et 4
```