INGENIEUR POLYTECH LILLE
INFORMATIQUES ET STATISTIQUES

Documentation

Extraction de Données de Publications et Statistiques de la Base HAL

Chancella Litoko & Anas Nay

Nom et adresse de l'école :

POLYTECH Lille
Boulevard Paul Langevin
59655, VILLENEUVE D'ASCQ CEDEX
03-28-76-73-60

Tuteur école : **Frédéric Hoogstoel**

Nom et adresse de l'entreprise :

Centre de Recherche en Informatique, Signal et Automatique de Lille (CRISTAL)
Université de Lille, Sciences et technologies, Batiment Esprit, 59655 Villeneuve-d'Ascq

Tuteur entreprise : **Mihaly Petreczky**

Année 2024-2025

Contents

Note d'avant garde

Avant d'exécuter les scripts inclus pour ce projet, il est **impératif** de s'assurer que votre environnement de travail est correctement configuré. Les scripts ont été développés en Python et nécessitent l'installation préalable des packages suivants pour fonctionner correctement. Chacun de ces packages joue un rôle essentiel dans le traitement des données, la génération de rapports et la visualisation des graphiques.

Liste des Packages Requis :

- **pandas** : Utilisé pour le traitement et l'analyse des données.
- **requests** : Nécessaire pour effectuer des requêtes HTTP afin de récupérer des données depuis des API en ligne.
- **unidecode** : Utilisé pour normaliser les chaînes de caractères en supprimant les accents et autres signes diacritiques.
- **fpdf** : Permet la création de fichiers PDF pour les rapports générés.
- **plotly** : Utilisé pour la création de visualisations interactives et dynamiques.
- **kaleido** : Requis par Plotly pour exporter des graphiques en format statique comme PNG.
- **tqdm** : Fournit une barre de progression pour les processus itératifs.

Installation des Packages :

Pour installer ces packages, exécutez la commande suivante dans votre terminal ou invite de commande :

```
pip install - U pandas requests unidecode fpdf plotly kaleido tqdm
```

Assurez-vous que tous les packages sont installés correctement avant de procéder à l'exécution des scripts pour éviter tout dysfonctionnement ou erreur d'exécution.

1 Contexte et objectifs du projet

L'objectif principal de ce projet est de développer un outil interactif et intuitif permettant l'extraction, l'analyse et la visualisation de données scientifiques issues de l'API HAL. HAL est une plateforme d'archivage ouverte qui centralise des publications scientifiques provenant de différentes institutions et chercheurs.
Cet outil vise à faciliter l'exploration et l'exploitation des données de recherche en offrant des fonctionnalités clés, telles que :

- **Extraction de données :** Récupérer automatiquement les publications scientifiques selon des critères définis (périodes, domaines, types de documents...).
- **Visualisation graphique :** Générer des graphiques interactifs (histogrammes, graphiques en barres, tendances temporelles...) pour représenter visuellement les statistiques issues des données extraites.
- **Rapports automatisés :** Produire des rapports complets au format PDF ou LaTeX intégrant les graphiques générés pour une présentation claire et professionnelle.

L'interface graphique est pensée pour être simple d'utilisation et accessible à toutes les personnes souhaitant obtenir des statistiques claires sur les publications scientifiques.
L'objectif final est de fournir un outil polyvalent et évolutif pouvant être utilisé aussi bien pour des projets de recherche académique que pour des rapports institutionnels.

2 Modes d'utilisation de l'outil

Ce projet propose deux façons d'utiliser les fonctionnalités d'extraction et d'analyse des données scientifiques issues de l'API HAL :

2.1 Via l'application interactive (`app.py`)

Une interface utilisateur conviviale développée en Python avec la bibliothèque Tkinter permet d'interagir facilement avec l'outil. Cette application est divisée en deux sections principales :

- **Extraction de données :** Permet de récupérer des publications scientifiques selon des critères sélectionnés par l'utilisateur (période, domaine, type de documents).
- **Analyse graphique et génération de rapports :** Permet de générer des graphiques interactifs et d'exporter des rapports complets au format PDF ou LaTeX.

2.2 Via l'exécution directe en terminal (`main.py`)

Une méthode plus directe est également disponible pour les utilisateurs avancés préférant travailler depuis un terminal. En exécutant le fichier `'main.py'`, l'utilisateur peut également :

- Extraire des données selon des critères spécifiques.
- Générer des graphiques basés sur les données extraites.
- Produire des rapports automatisés aux formats PDF et LaTeX.

Les deux méthodes garantissent la même qualité d'analyse et d'extraction, offrant ainsi flexibilité et accessibilité selon les besoins de l'utilisateur.

3 Présentation de l'application `app.py`

Le fichier `app.py` est l'élément central de l'application graphique de ce projet. Il a été développé en utilisant la bibliothèque `tkinter` et permet d'interagir de manière conviviale avec l'utilisateur pour effectuer l'extraction, l'analyse et la génération de rapports basés sur des données issues de l'API HAL.

Le lancement de l'application s'effectue en exécutant le fichier python, depuis un environnement d'édition comme Spyder, ou bien depuis un terminal à l'aide de la commande `python app.py`.

Une fois l'application lancée, vous arrivez sur cette page d'accueil :

Figure 1: Page d'accueil de l'application - Section Extraction

Il s'agit de la page dédiée à l'extraction de données.

L'application est structurée en deux sections principales : **Extraction** et **Analyse**.

La section **Extraction** constitue la première étape. Elle permet de charger un fichier CSV contenant les noms et prénoms des auteurs pour lesquels des informations doivent être extraites. Ces informations incluent des données sur les auteurs eux-mêmes, comme leur identifiant HAL et leur laboratoire de recherche rattaché, ainsi que des informations sur leurs publications, telles que le titre, l'identifiant, le type de document, les mots-clés associés et les domaines de recherche.
La section **Analyse** s'appuie sur les données extraites dans la section précédente pour offrir des fonctionnalités d'analyse graphique. Elle permet de charger un fichier CSV contenant ces données, de générer des visualisations graphiques, et de les afficher dans un tableau de bord interactif au format HTML. En outre, ces graphiques peuvent être intégrés dans des rapports au format PDF ou LaTeX, générés directement depuis l'application, offrant ainsi une présentation claire et complète des analyses réalisées.

3.1 Extraction de données

Dans cette page, vous devez commencer par charger un fichier CSV contenant les données de noms et prénoms des auteurs dont vous voulez extraire les données de publications. Vous pouvez trouver un exemple du type de fichier attendu en annexe.

Une fois le fichier CSV chargé, un message de succès s'affichera :

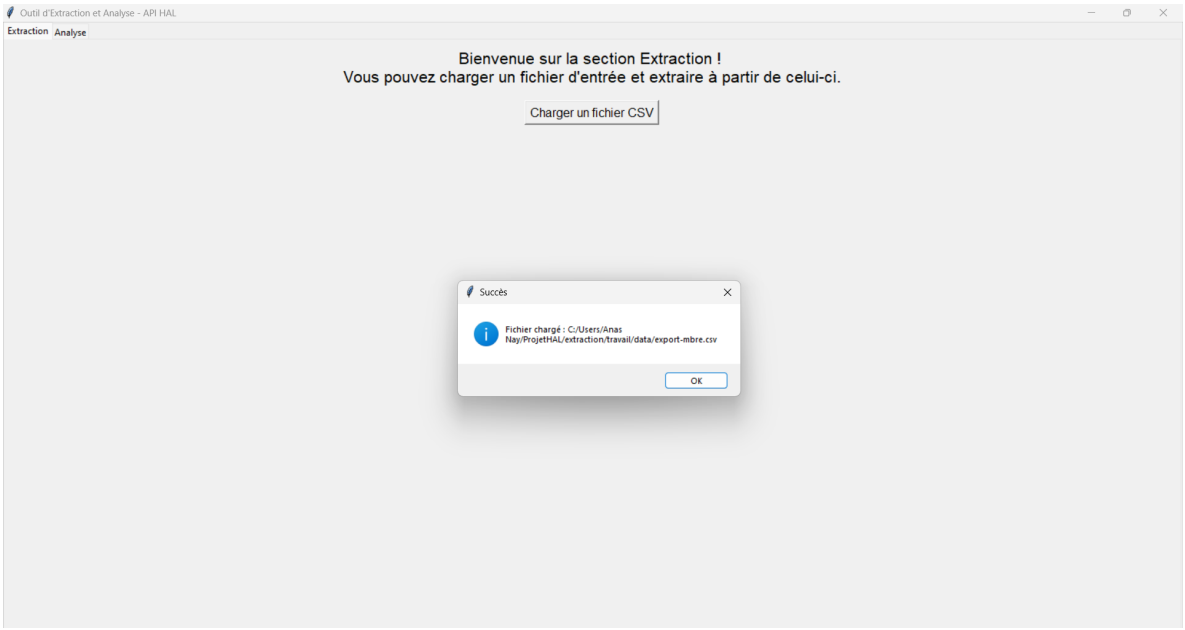

Figure 2: Message de succès du chargement du fichier CSV

Une fois le fichier contenant les noms et prenoms des auteurs qui nous intéressent est bien chargé, l'extraction peut commencer. Deux modes d'extraction sont disponibles.

Le premier permet d'extraire l'intégralité des données sans appliquer de filtres, tels que la période de publication ou le type de document. Le fichier résultant contiendra alors l'ensemble des publications de tous les auteurs présents dans le fichier CSV chargé préalablement. L'extraction démarre en cliquant sur le bouton `"Extraire toutes les données"`.

Le second mode d'extraction offre à l'utilisateur la possibilité de filtrer les publications avant leur extraction afin de cibler des résultats plus pertinents. Pour accéder à cette fonctionnalité, il suffit de cliquer sur le bouton `"Appliquer des filtres d'extraction"` dans l'interface de l'application. Trois critères de filtrage sont proposés :

- **La période de parution :** L'utilisateur peut spécifier une plage d'années sous la forme d'un intervalle (par exemple : 2019-2022) afin de n'extraire que les publications parues durant cette période.
- **Le type de document :** Divers types de publications scientifiques sont disponibles, tels que les articles de revue, les thèses, les actes de conférences, ou encore les synthèses. L'utilisateur peut sélectionner un ou plusieurs types à inclure dans l'extraction.
- **Le domaine des publications :** Il est également possible de filtrer les résultats en fonction des domaines scientifiques. L'utilisateur peut choisir un ou plusieurs domaines tels que l'informatique, la biologie ou la physique, selon ses besoins spécifiques.

Pour ces deux modes d'extraction, une barre de progression s'affichera pour indiquer l'état d'avancement du processus d'extraction (voir image ci-dessous).

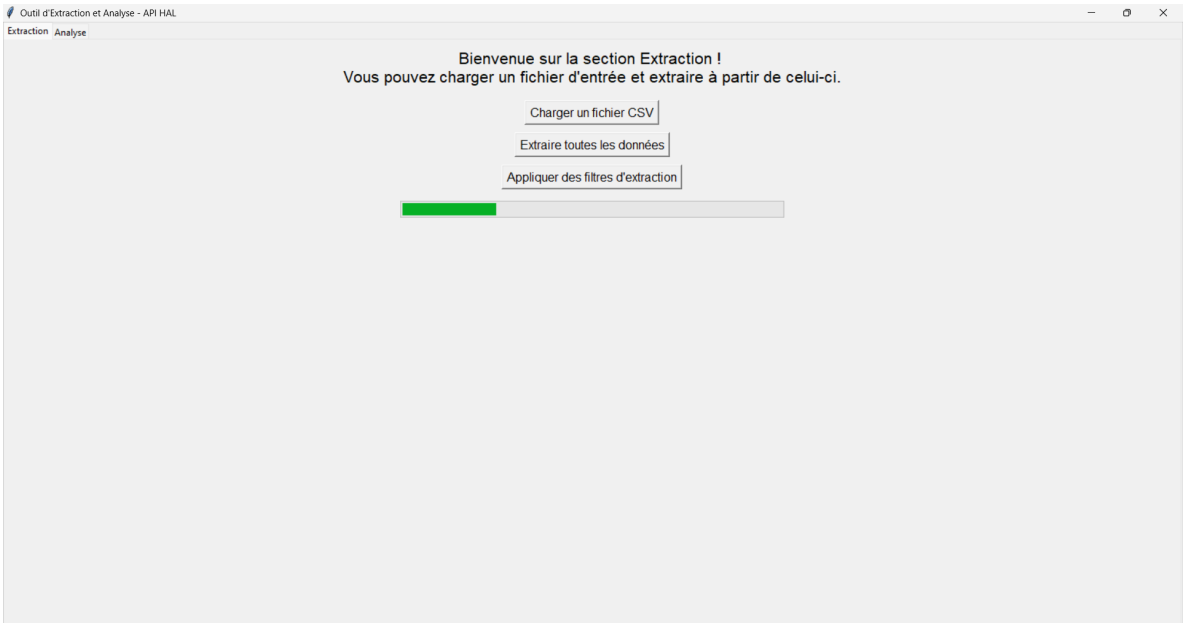

Figure 3: Barre de progression de l'extraction.

Une fois l'extraction terminée, un message de succès s'affichera et vous indiquera dans quel fichier CSV les résultats sont enregistrés. Dans le cas où toutes les données sont extraites (premier mode), le fichier aura le nom `all_data.csv`. Pour le second mode, le nom du fichier résultant sera : `all_data_`{*Domaine*}`_`{*Période*}`_`{*Type*}. Par exemple, pour les publications du domaine *Statistique*, parues entre 2018 et 2022 et de type *Synthèse*, le nom sera : `all_data_Statistique_2018-2022_Synthese.csv`. Tous ces fichiers csv seront rangés dans un dossier 'extraction' à la racine du fichier `app.py`.
Voici un exemple pour une extraction de toutes les données :

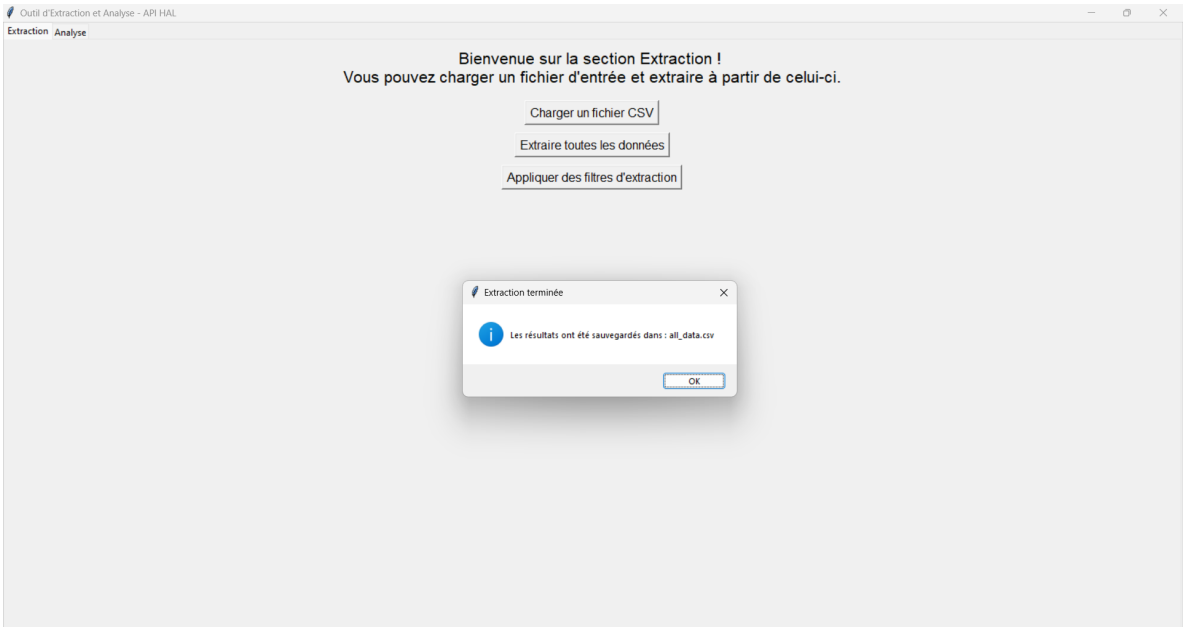

Figure 4: Message de succès de l'extraction.

3.2 Analyse des données

Dans cette page, vous devrez également commencer par charger un fichier csv contenant les données extraites dans la section `Extraction`. Une fois ces données chargées, l'outil générera automatiquement des graphiques. Ces graphiques sont (pour l'instant) au nombre de sept : un histogramme du nombre de publications par années, un diagramme en cammembert indiquant la répartition des types de documents, un histogramme du top 10 des mots-clés les plus fréquents, un histogramme du top 10 des domaines les plus fréquents, un histogramme du top 10 des auteurs les plus prolifiques, un histogramme montrant le nombre de publications par structures et par années et enfin un graphique montrant l'évolution des publications par années.

Voici à quoi ressemble la section `Analyse` de l'application :

Figure 5: Section Analyse

Votre seule option est de charger un fichier csv contenant les données extraction dans la section `Extraction`. Lorsque vous sélectionnerez votre fichier, l'application générera automatiquement les graphiques cités plus tôt.
Un message indiquant que les graphiques ont bien été générés s'affichera.

Deux nouveaux boutons s'afficheront alors.
Le premier, intitulé `Afficher les graphiques`, permet d'ouvrir directement depuis votre navigateur un tableau de bord HTML. Ce dashboard, développé en Python en utilisant le module `plotly`, regroupe l'ensemble des graphiques générés. Chaque graphique intégré dans ce tableau de bord est interactif, offrant la possibilité de survoler les différents éléments graphiques pour en afficher des détails spécifiques.
Le second bouton, intitulé `Générer un rapport`, vous permettra de créer un document détaillé qui compile les graphique générés. Ce rapport est disponible en formats PDF ou LaTeX selon votre choix.

Les graphiques affiché dans le rapport sont en fait un fichier PNG enregistré lors de la génération du graphiques. Vous pourrez retrouver tous les graphiques sous format HTML et PNG à la racine du fichier `app.py`, dans des dossiers eponymes.

Figure 6: Boutons pour l'analyse graphique

Après avoir cliqué sur le bouton pour générer le rapport, une fenêtre s'ouvrira vous proposant le choix d'un rapport PDF ou LaTeX :

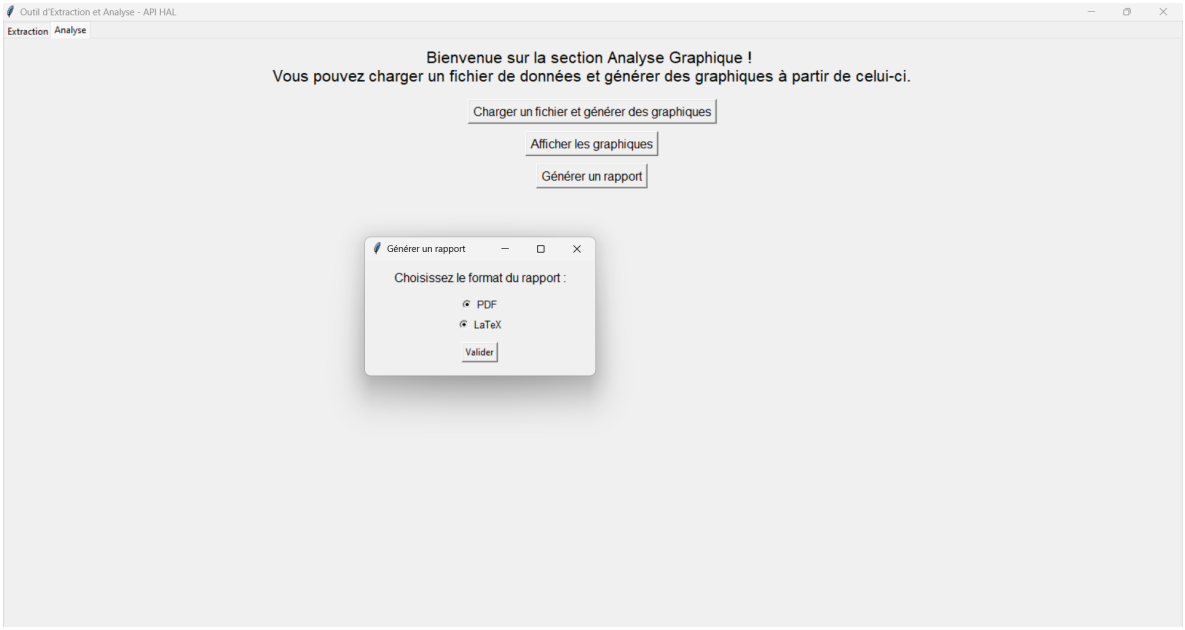

Figure 7: Choix du format du rapport

Une fois le rapport créé, un message de succès s'affichera et vous pourrez retrouver le rapport dans le dossier `rapports` à la racine du fichier `app.py`.

4 Présentation du fichier `main.py`

Le fichier `main.py` est conçu pour extraire les données de l'API HAL à partir d'une liste d'auteurs spécifiée par l'utilisateur au début du programme. Ce script est utile pour les utilisateurs souhaitant obtenir des données scientifiques détaillées et générer des visualisations ainsi que des rapports basés sur ces données.

Il est impératif de modifier le chemin du fichier contenant les noms des membres auteurs au début du script pour correspondre à l'emplacement de votre fichier sur votre machine. Cette exigence constitue un point à améliorer pour rendre l'outil plus flexible et moins dépendant de la configuration locale.

Le script accepte plusieurs options de ligne de commande pour filtrer les résultats :

- –**year** pour spécifier une plage d'années (format YYYY-YYYY).
- –**type** pour filtrer par type de document.
- –**domain** pour filtrer par domaine scientifique.

Cette fonctionnalité est semblable à la section `Appliquer des filtres d'extraction` de l'application.

Par exemple, pour extraire les publications d'une période spécifique, utilisez la commande suivante:

```
python main.py -- --year 2010-2020
```

Cette commande par exemple permettra d'extraire les données parues entre 2010 et 2020.

De même, pour filtrer par type de document et/ou domaine scientifique:

```
python main.py -- --type journal -- --domain 'Computer Science'
```

Cette commande permettra d'extraire les publications de type `'journal'` et de domaine `'Computer Science'`.

Exécuter simplement la commande `python main.py` sans options extraira les données de toutes les publications de tous les auteurs, similaire à l'action du bouton 'Extraire toutes les données' sur l'application.

Pour voir la liste complète des domaines ou des types, vous pouvez exécuter les commandes :

```
python main.py -- --list-domains
python main.py -- --list-types
```

Pour obtenir de l'aide sur toutes les options disponibles, tapez :

```
python main.py -h
```

Après l'extraction des données, le programme demande à l'utilisateur s'il souhaite réaliser les fonctionnalités suivantes, nécessitant une réponse par 'oui' ou 'non' :

- Visualisation des données à travers des graphiques interactifs sur votre navigateur par defaut.
- Génération de rapports au format PDF ou LaTeX depuis le terminal.

Les fichiers d'extraction CSV sont enregistrés dans le dossier `extraction` situé à la racine du projet, tandis que les rapports générés sont stockés dans le dossier `rapports`.

5 Annexe

5.1 Fichier CSV attendu pour l'extraction

Pour télécharger et visualiser le type de fichier CSV requis pour l'extraction, veuillez utiliser le lien suivant:

Téléchargez le fichier CSV ici

5.2 Description du fichier CSV obtenu

Le fichier CSV obtenu après l'extraction des données, que ce soit via l'interface de l'application ou en exécutant le fichier `main.py`, présente une structure uniforme. Le fichier contiendra les mêmes informations peu importe la méthode utilisée pour l'extraction. Chaque ligne du fichier CSV représente une publication. Voici une description des colonnes typiques que l'on trouve dans ce fichier :

Table 1: Description des colonnes du fichier CSV

Colonne	**Description**
Nom	Le nom de famille de l'auteur.
Prénom	Le prénom de l'auteur.
IdHAL de l'Auteur	Identifiant HAL de l'auteur.
IdHAL des auteurs de la publication	Identifiant HAL des auteurs de la publication.
Titre	Le titre de la publication.
Docid	L'identifiant de la publication.
Année de Publication	L'année de publication.
Type de Document	Le type de document, par exemple article, thèse, etc.
Domaine	Le domaine scientifique de la publication.
Mots-clés	Les mots-clés associés à la publication.
Laboratoire de Recherche	Le laboratoire ou le centre de recherche associé à l'auteur de la publication.