



COMPUTER SCIENCE AND STATISTICS
ENGINEER POLYTECH LILLE

Documentation

Data Extraction from Publications and Statistics of
the HAL Database

Anas Nay

School name and address:

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

School supervisor: Frédéric
Hoogstoel

Company name and address:

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

Company supervisor: Mihaly
Petreczky

Academic Year 2024-2025

Contents

1	Contexte et objectifs du projet	4
2	Modes d'utilisation de l'outil	5
2.1	Interface graphique interactive (fichier <code>app.py</code>)	5
2.2	Interface en ligne de commande (fichier <code>main.py</code>)	5
3	Présentation de l'application interactive	6
3.1	Lancement de l'application	6
3.2	Architecture générale	6
3.3	Module d'extraction de données	7
3.3.1	Chargement des données sources	7
3.3.2	Modes d'extraction disponibles	7
3.3.3	Exécution et monitoring	9
3.4	Module d'analyse de données	10
3.4.1	Génération automatique des visualisations	10
3.4.2	Consultation interactive	10
3.4.3	Production de rapports	11
3.5	Configuration de la sensibilité de correspondance	12
4	Présentation de l'interface ligne de commande	14
4.1	Fonctionnalités principales	14
4.2	Sélection du fichier d'entrée	14
4.3	Options de filtrage	14
4.4	Configuration de la sensibilité	15
4.5	Génération automatique de contenu	15
4.6	Commandes utilitaires	15
4.7	Exemples d'utilisation	15
4.7.1	Extraction simple	15
4.7.2	Extraction avec filtres	15
4.7.3	Extraction avec sensibilité personnalisée	16
4.7.4	Extraction avec génération automatique	16
4.7.5	Commandes d'information	16
4.8	Suivi de progression et résultats	16
4.9	Récapitulatif avant extraction	16
4.10	Organisation des fichiers de sortie	17
5	Module de détection des doublons et homonymes	18
5.1	Principe de fonctionnement	18
5.2	Accès au module depuis l'interface graphique	18
5.3	Configuration et lancement de l'analyse	19
5.3.1	Sélection des fichiers d'entrée	19
5.3.2	Fichier laboratoire optionnel	19
5.4	Interface d'analyse et résultats	19
5.4.1	Onglet Résumé	20
5.4.2	Onglets de résultats détaillés	20
5.5	Types de détection supportés	21

5.6	Traitement automatique des données	21
5.6.1	Options de traitement disponibles	21
5.6.2	Résultats du traitement	21
5.7	Exportation des résultats	22
5.7.1	Fichiers générés	22
5.8	Utilisation en ligne de commande	23
5.8.1	Exemple de session en ligne de commande	23
5.9	Limitations et considérations	24
5.9.1	Dépendance à l'API HAL	24
5.9.2	Cas particuliers	24
5.10	Conclusion	24
6	Annexe	25
6.1	Fichier CSV attendu pour l'extraction	25
6.2	Description du fichier CSV obtenu	25

Note d'avant garde

Cette documentation s'adresse aux utilisateurs possédant des compétences informatiques de base, incluant la capacité d'exécuter des commandes dans un terminal et de gérer l'exécution de fichiers Python. Une familiarité avec les systèmes de gestion de version, notamment GitHub, est également recommandée.

Avant d'utiliser les scripts de ce projet, il est **impératif** de configurer correctement votre environnement de travail. Les scripts, développés en Python, nécessitent l'installation préalable de dépendances spécifiques qui jouent un rôle essentiel dans le traitement des données, la génération de rapports et la visualisation des graphiques.

Installation et configuration

1. Clonage du dépôt

Ouvrez un terminal et exécutez les commandes suivantes :

```
git clone https://github.com/anasnay11/PROJET_HAL_.git
cd PROJET_HAL_
```

2. Création d'un environnement virtuel

Créez et activez un environnement virtuel Python :

```
python -m venv venv
```

Puis activez-le selon votre système d'exploitation :

- **Linux/macOS** : `source venv/bin/activate`
- **Windows** : `venv\Scripts\activate`

3. Installation des dépendances

Une fois l'environnement virtuel activé, installez les packages requis :

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

Vérifiez que tous les packages s'installent correctement avant de procéder à l'exécution des scripts.

Points importants à retenir

1. Les fichiers principaux `app.py` et `main.py` se trouvent dans le sous-dossier `python code`. Positionnez-vous dans ce répertoire avant de les exécuter.
2. Le fichier CSV d'entrée doit impérativement contenir les colonnes '`nom`' et '`prenom`' pour être utilisable par l'application.
3. Pour une visualisation optimale des captures d'écran de l'interface, agrandissez votre fenêtre d'application afin de reproduire fidèlement les images présentées dans cette documentation.

1 Contexte et objectifs du projet

Ce projet vise à développer un outil interactif et intuitif pour l'extraction, l'analyse et la visualisation de données scientifiques issues de la plateforme HAL (Hyper Articles en Ligne). HAL constitue l'archive ouverte nationale française qui centralise et diffuse les publications scientifiques produites par les institutions de recherche, laboratoires et chercheurs du territoire.

Face aux volumes croissants de données scientifiques et à la nécessité d'analyser efficacement la production de recherche, cet outil propose une solution complète articulée autour de trois fonctionnalités principales :

- **Extraction automatisée** : récupération ciblée des identifiants d'auteurs et métadonnées de publications selon des critères personnalisables (périodes temporelles, domaines scientifiques, types de documents).
- **Visualisation interactive** : génération automatique de graphiques dynamiques (histogrammes, séries temporelles, diagrammes en barres, nuages de mots) permettant une appréhension immédiate des tendances et patterns dans les données.
- **Reporting professionnel** : production de rapports structurés aux formats PDF et LaTeX, intégrant les visualisations pour une présentation claire et exploitable.

L'architecture de l'outil privilégie l'accessibilité et la simplicité d'usage. Une interface graphique permet à tout utilisateur, indépendamment de ses compétences techniques, d'analyser rapidement les travaux de recherches scientifiques d'un certain laboratoire ou d'un groupe de recherche. Le système requiert uniquement un fichier CSV structuré contenant à minima les noms et prénoms des auteurs d'intérêt.

Le développement de ce système s'est appuyé sur un cas d'usage réel : l'analyse des publications du groupe de recherche MACS¹. Cette approche garantit la pertinence fonctionnelle et la robustesse de l'outil dans des conditions opérationnelles authentiques.

L'ambition finale consiste à délivrer une solution polyvalente et évolutive, adaptée tant aux besoins académiques (évaluations de projets, bilans d'activité scientifique) qu'aux exigences institutionnelles (rapports d'évaluation, tableaux de bord pour directions de laboratoires, aide à la décision stratégique).

¹Le fichier de données de test est disponible en section 5.1

2 Modes d'utilisation de l'outil

L'outil propose deux approches complémentaires pour exploiter les fonctionnalités d'extraction et d'analyse des données scientifiques HAL, chacune adaptée à des profils d'utilisateurs et contextes d'usage distincts.

2.1 Interface graphique interactive (fichier `app.py`)

L'application, développée avec la bibliothèque Tkinter, offre une expérience utilisateur intuitive et accessible. Cette interface graphique structure le processus en deux modules fonctionnels :

- **Module d'extraction** : Configuration et lancement des requêtes de récupération selon des critères personnalisables (fenêtres temporelles, domaines scientifiques, typologies documentaires).
- **Module d'analyse** : Génération de visualisations interactives et export de rapports structurés aux formats PDF et LaTeX.

Cette approche privilégie l'accessibilité et convient particulièrement aux utilisateurs occasionnels ou peu familiers avec les environnements en ligne de commande.

2.2 Interface en ligne de commande (fichier `main.py`)

L'exécution via terminal propose une alternative robuste pour des utilisateurs avancés. Cette méthode permet :

- **Extraction paramétrable** : Définition précise des critères via arguments de ligne de commande, facilitant l'intégration dans des scripts et pipelines de traitement.
- **Génération batch** : Production automatisée de visualisations et rapports sans intervention manuelle, optimisant les traitements sur de gros volumes.
- **Automatisation complète** : Possibilité d'enchaîner extraction, analyse et reporting en une seule commande, idéale pour les analyses récurrentes.

Cette approche maximise l'efficacité et la reproductibilité, particulièrement adaptée aux environnements de recherche nécessitant des analyses systématiques.

Les deux modes garantissent une qualité d'analyse équivalente et accèdent aux mêmes fonctionnalités. Le choix entre interface graphique et ligne de commande dépend principalement des préférences utilisateur, du niveau technique et du contexte d'utilisation.

3 Présentation de l'application interactive

Le fichier `app.py` constitue le cœur de l'interface graphique du projet. Développée avec la bibliothèque `tkinter`, cette application offre une expérience utilisateur intuitive pour l'extraction, l'analyse et la génération de rapports basés sur les données de l'API HAL.

3.1 Lancement de l'application

Deux méthodes permettent d'exécuter l'application :

- **Depuis un IDE** : Exécution directe du fichier Python dans un environnement comme Spyder ou PyCharm
- **Depuis le terminal** : Navigation vers le dossier `PROJET_HAL_/python` code et exécution de la commande `python3 app.py`

3.2 Architecture générale

L'application s'articule autour de deux modules fonctionnels complémentaires :

- **Module Extraction** : Récupération et filtrage des données scientifiques depuis HAL
- **Module Analyse** : Génération de visualisations et production de rapports

Au lancement, l'utilisateur accède directement à la section Extraction via l'interface d'accueil :



Figure 1: Interface d'accueil - Section Extraction

3.3 Module d'extraction de données

Le processus d'extraction s'initialise par le chargement d'un fichier CSV structuré contenant les identités des auteurs cibles (colonnes **nom** et **prenom** obligatoires). Les données extraites enrichissent ce fichier initial avec les métadonnées d'auteurs (identifiants HAL, affiliations) et les informations détaillées de leurs publications (titres, types, domaines, etc.). Une description exhaustive de ces champs est disponible en section 5.2.

3.3.1 Chargement des données sources

L'interface propose un bouton central pour sélectionner le fichier CSV d'entrée. Une fois le fichier validé, un message de confirmation s'affiche :

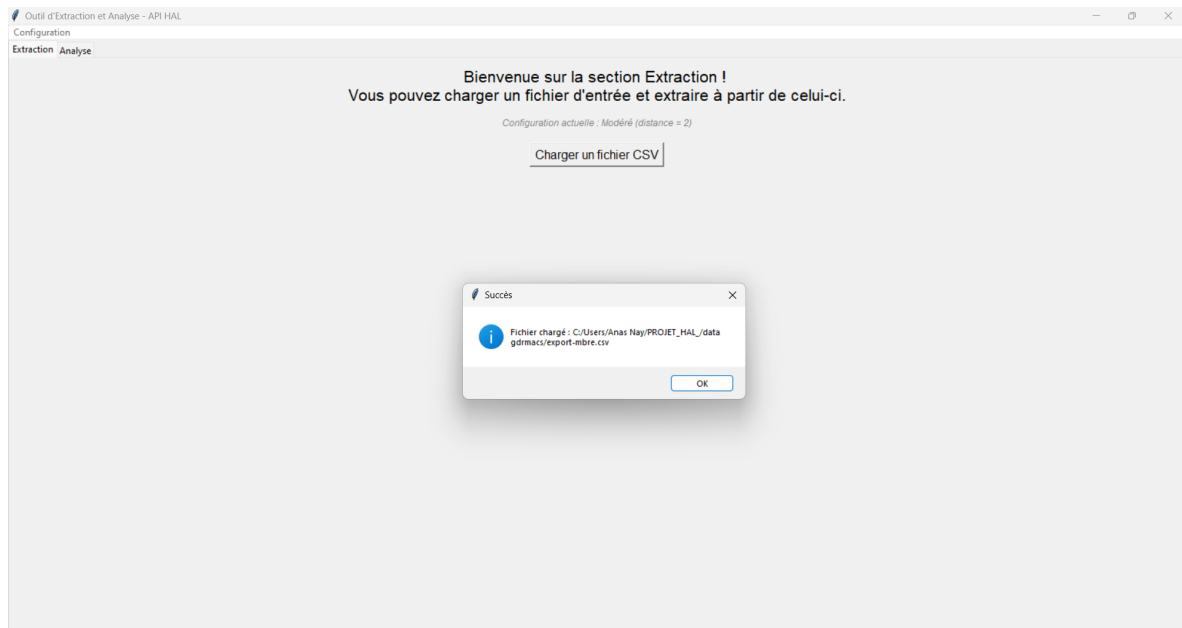


Figure 2: Confirmation du chargement réussi

3.3.2 Modes d'extraction disponibles

Deux approches d'extraction sont proposées selon les besoins analytiques :

Extraction intégrale Ce mode récupère exhaustivement toutes les publications associées aux auteurs listés, sans restriction temporelle, thématique ou typologique. Une fenêtre récapitulative présente le nombre d'auteurs traités et le seuil de sensibilité appliqué (détaillé en section 3.5) :

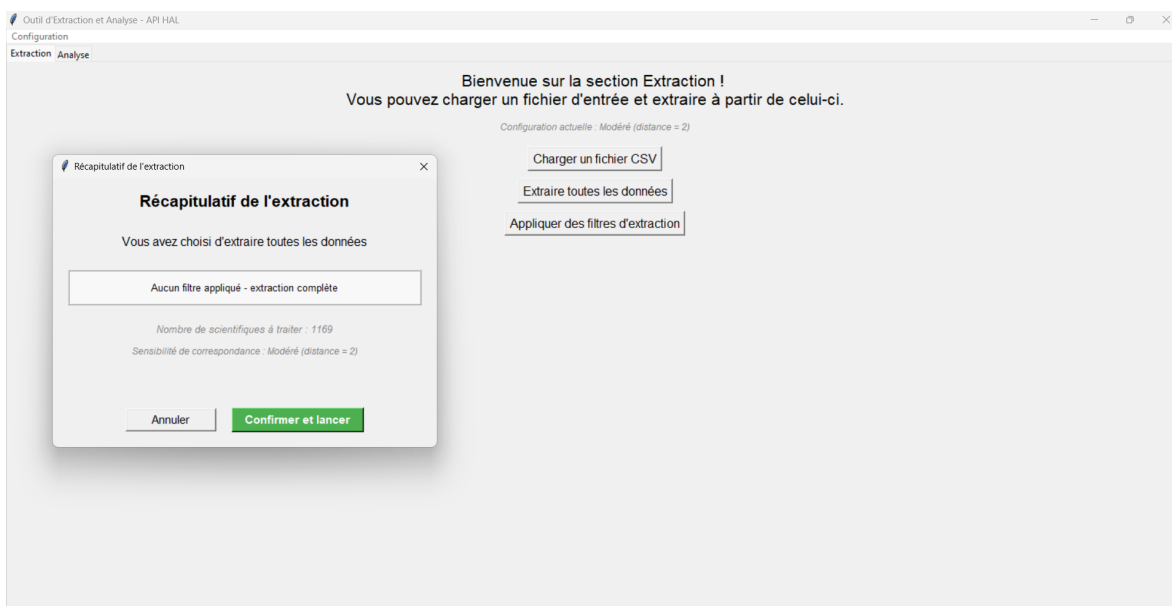


Figure 3: Récapitulatif de l'extraction intégrale

Cette option avancée permet de définir des critères de filtrage multiples :

- **Fenêtre temporelle** : Restriction à une période spécifique (ex. : 2019-2022)
- **Typologie documentaire** : Sélection ciblée (articles, thèses, communications, rapports)
- **Domaines scientifiques** : Filtrage disciplinaire (informatique, mathématiques, biologie, etc.)

L'interface de configuration centralise ces paramètres :

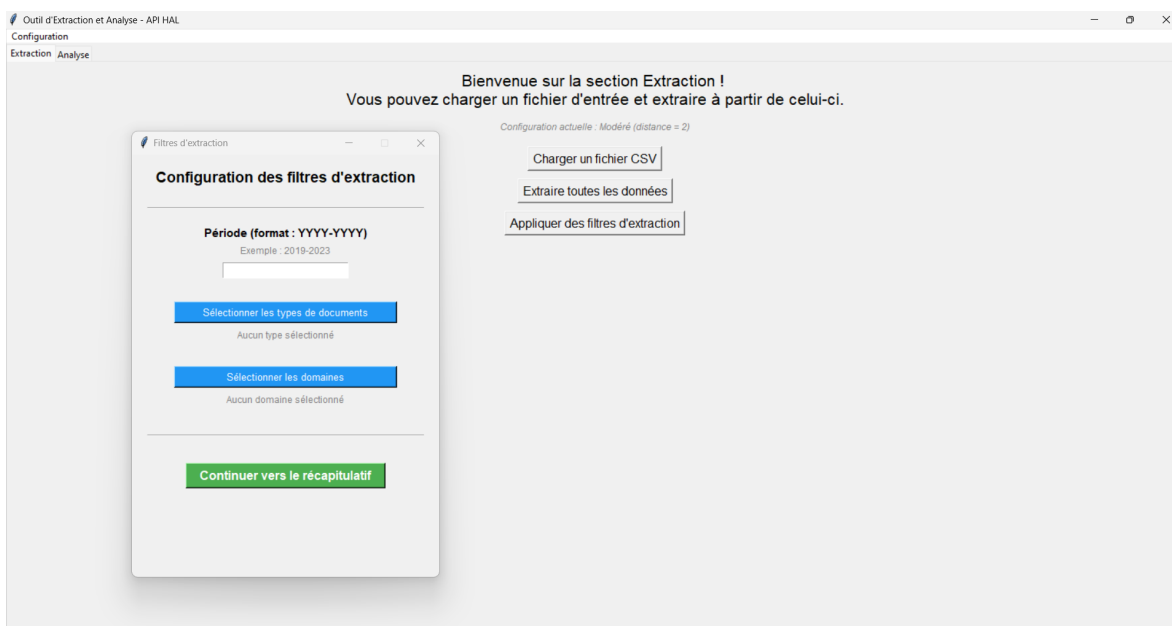


Figure 4: Interface de configuration des filtres

Une fenêtre récapitulative valide la configuration avant lancement. L'exemple ci-dessous illustre un filtrage sur les thèses et rapports en informatique et mathématiques pour la période 2010-2020 :

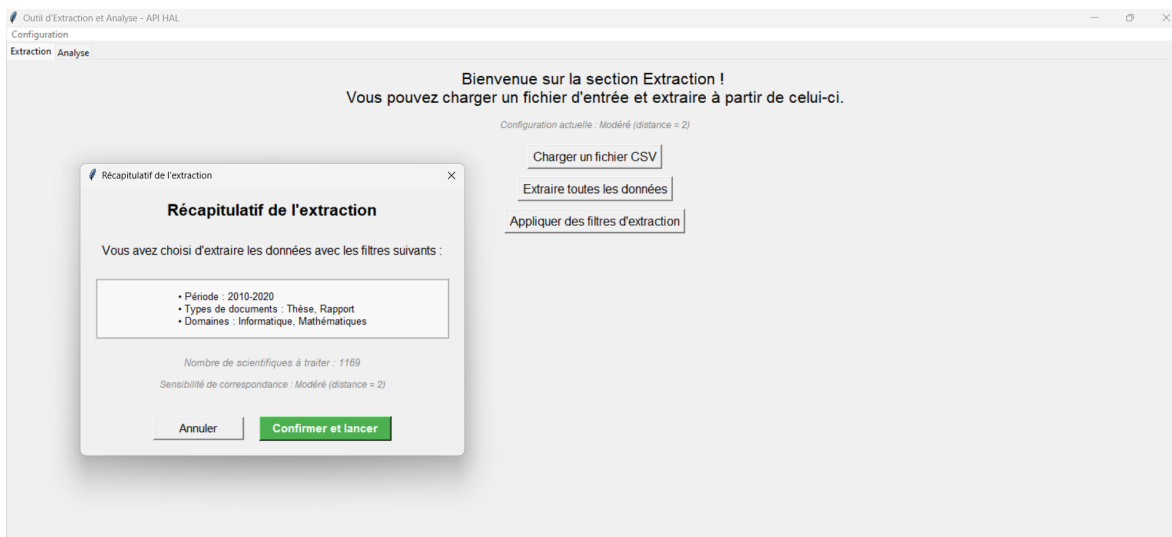


Figure 5: Récapitulatif d'extraction avec filtres appliqués

3.3.3 Exécution et monitoring

Le lancement de l'extraction active une barre de progression avec pourcentage d'avancement. Un bouton d'arrêt d'urgence permet l'interruption du processus si nécessaire. À l'issue de l'extraction, les résultats sont automatiquement sauvegardés dans le dossier **extraction** selon une nomenclature standardisée :

- Extraction intégrale : `all_data.csv`
- Extraction filtrée : `all_data_{Domaine}_{Période}_{Type}.csv`

3.4 Module d'analyse de données

Le module d'analyse exploite les fichiers CSV générés lors de l'extraction pour produire des visualisations interactives et des rapports structurés.



Figure 6: Interface du module d'analyse

3.4.1 Génération automatique des visualisations

Le simple chargement d'un fichier CSV d'extraction déclenche la génération automatique de l'ensemble des graphiques. Le système produit simultanément :

- Des versions interactives HTML (dossier `html`)
- Des exports statiques PNG (dossier `png`)

3.4.2 Consultation interactive

Le bouton **Afficher les graphiques** lance automatiquement un tableau de bord HTML dans le navigateur par défaut. Ce dashboard, développé avec `plotly`, centralise toutes les visualisations avec des fonctionnalités interactives avancées (zoom, survol, filtrage).



Figure 7: Interface de consultation et export

3.4.3 Production de rapports

Le bouton **Générer un rapport** initie la création d'un document compilé intégrant l'ensemble des visualisations. L'utilisateur sélectionne le format de sortie souhaité :

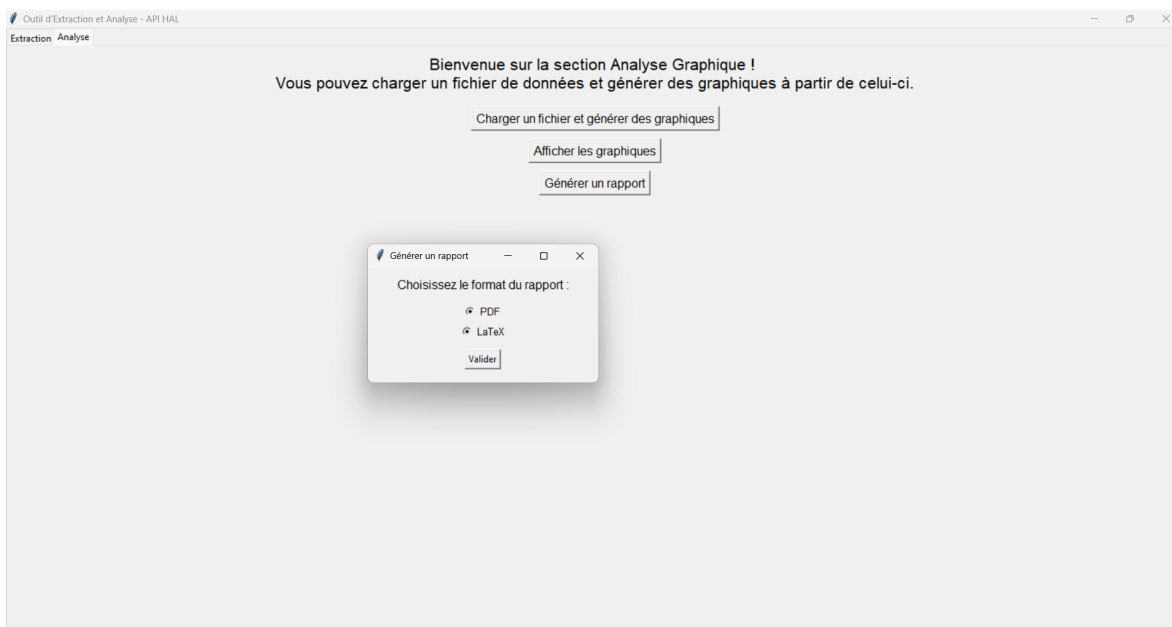


Figure 8: Sélection du format de rapport

Les rapports générés sont automatiquement stockés dans le dossier **rapports** avec les images PNG intégrées pour une consultation offline optimale.

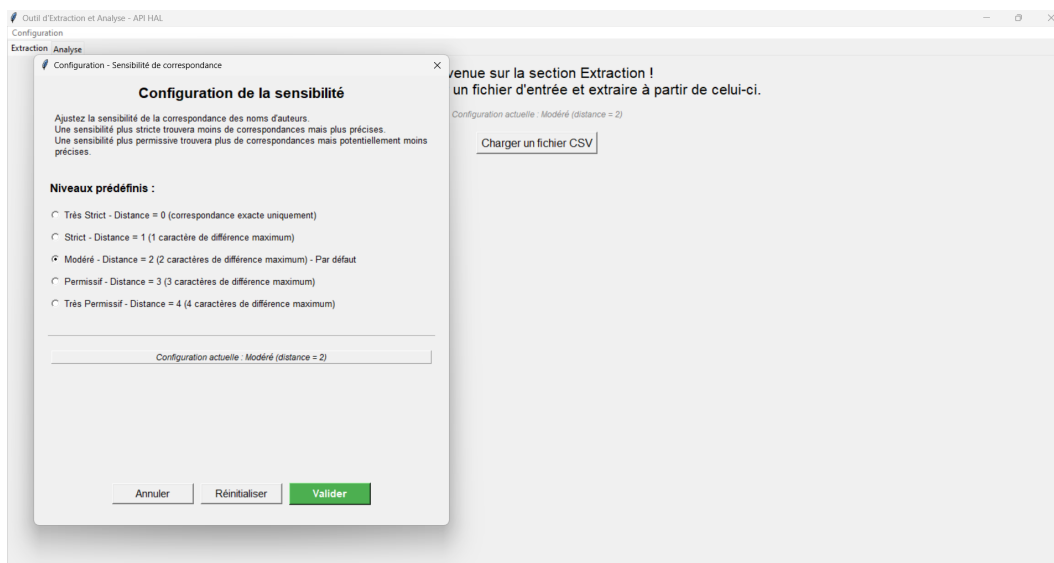
3.5 Configuration de la sensibilité de correspondance

L'application intègre une fonctionnalité avancée permettant de personnaliser le niveau de sensibilité utilisé lors du processus de correspondance des noms et prénoms d'auteurs. Cette configuration est utile pour optimiser la précision de l'extraction en fonction de la qualité et de la cohérence des données d'entrée.

L'accès à ces paramètres s'effectue depuis la barre de menu principale : **Configuration** > **Sensibilité de correspondance**....

La correspondance entre les noms du fichier CSV d'entrée et ceux de la base HAL repose sur la distance de Levenshtein, un algorithme qui mesure la similarité entre deux chaînes de caractères en calculant le nombre minimum d'opérations (insertions, suppressions ou substitutions) nécessaires pour les rendre identiques. Plus cette distance est faible, plus les noms sont similaires. Ce mécanisme gère efficacement les variations d'orthographe, erreurs de frappe, différences d'accentuation ou abréviations courantes.

La fenêtre de configuration présente plusieurs niveaux de sensibilité prédéfinis, chacun correspondant à une valeur de distance spécifique :



Cette interface propose cinq niveaux prédéfinis :

- **Très Strict** (Distance = 0) : Correspondance exacte uniquement, aucune tolérance aux variations
- **Strict** (Distance = 1) : Tolérance d'un seul caractère de différence maximum
- **Modéré** (Distance = 2) : Niveau par défaut, autorise jusqu'à 2 caractères de différence
- **Permissif** (Distance = 3) : Tolérance élargie pour 3 caractères de différence maximum
- **Très Permissif** (Distance = 4) : Niveau le plus tolérant, accepte jusqu'à 4 caractères de différence

Exemple d'application : Pour un nom "Müller" dans le fichier CSV :

- Niveau *Très Strict* : seul "Müller" exact sera trouvé
- Niveau *Modéré* : "Muller", "Müller" seront détectés
- Niveau *Permissif* : inclura également "Miller", "Moller", etc.

Un encadré informatif affiche la configuration actuelle en cours d'utilisation.

Dans l'image précédente, le niveau "Modéré" est sélectionné par défaut avec une distance de 2.

L'interface propose également trois boutons d'action : **Annuler** pour fermer sans sauvegarder, **Réinitialiser** pour revenir aux paramètres par défaut, et **Valider** pour appliquer la nouvelle configuration avant d'extraire les données. Une fois validée, la nouvelle configuration sera confirmée par un message informatif et s'appliquera à toutes les extractions ultérieures.

4 Présentation de l'interface ligne de commande

Le fichier `main.py` constitue une interface en ligne de commande complète pour l'extraction et l'analyse de données scientifiques depuis l'API HAL. Cette approche offre une alternative puissante à l'interface graphique, particulièrement adaptée aux utilisateurs avancés.

4.1 Fonctionnalités principales

Le script propose un ensemble complet d'options permettant de personnaliser finement l'extraction et l'analyse des données :

- **Extraction filtrée** : possibilité de cibler les publications selon des critères spécifiques
- **Gestion de la sensibilité** : configuration du seuil de correspondance des noms d'auteurs
- **Génération automatique** : création de graphiques et rapports en une seule commande
- **Interface interactive** : sélection guidée des fichiers d'entrée

4.2 Sélection du fichier d'entrée

Contrairement aux versions précédentes nécessitant une modification manuelle du chemin de fichier, la version actuelle propose une interface interactive pour sélectionner le fichier CSV contenant les noms des auteurs.

Au lancement du script, une liste numérotée des fichiers CSV disponibles s'affiche :

```
PS C:\Users\Anas Nay\TEST_PROJET_HAL\python code> python .\main.py
Sensitivity threshold used: 2 (moderate)
Available CSV files:
1. export-mbre-test - export-mbre.csv.csv
2. export-mbre.csv

Enter the number of the file you want to use: |
```

L'utilisateur sélectionne simplement le numéro correspondant au fichier souhaité. Cette amélioration rend l'outil plus flexible et élimine la dépendance à la configuration locale.

4.3 Options de filtrage

Le script accepte plusieurs arguments optionnels pour filtrer les résultats d'extraction :

- - - **year** : spécifier une plage d'années (format YYYY-YYYY)
- - - **type** : filtrer par type de document
- - - **domain** : filtrer par domaine scientifique
- - - **threshold** : configurer la sensibilité de correspondance des noms (0-4)

4.4 Configuration de la sensibilité

La nouvelle version intègre la gestion de la sensibilité de correspondance directement en ligne de commande via l’argument `--threshold`. Les niveaux disponibles sont :

- **0** : Très strict (correspondance exacte uniquement)
- **1** : Strict (1 caractère de différence maximum)
- **2** : Modéré (2 caractères de différence maximum) - *Par défaut*
- **3** : Permissif (3 caractères de différence maximum)
- **4** : Très permissif (4 caractères de différence maximum)

4.5 Génération automatique de contenu

Trois nouvelles options permettent la génération automatique de visualisations et rapports :

- `--graphs` : génération automatique de graphiques et ouverture du tableau de bord
- `--reportpdf` : création automatique d’un rapport PDF
- `--reportlatex` : génération automatique d’un rapport LaTeX

4.6 Commandes utilitaires

Le script propose également des commandes d’information :

- `--list-domains` : afficher tous les domaines scientifiques disponibles
- `--list-types` : lister tous les types de documents supportés
- `--list-sensitivity` : détailler les niveaux de sensibilité

4.7 Exemples d’utilisation

4.7.1 Extraction simple

```
python main.py
```

Extraction de toutes les données sans filtres.

4.7.2 Extraction avec filtres

```
python main.py --year 2019–2024 --domain "Mathematics" --type "Theses"
```

Extraction des thèses en mathématiques publiées entre 2019 et 2024.

4.7.3 Extraction avec sensibilité personnalisée

```
python main.py --threshold 1 --domain "Informatique"
```

Extraction avec correspondance stricte des noms pour le domaine informatique.

4.7.4 Extraction avec génération automatique

```
python main.py --graphs --reportpdf --reportlatex
```

Extraction complète avec génération automatique de graphiques et rapports.

4.7.5 Commandes d'information

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

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

python main.py --list-sensitivity

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

4.8 Suivi de progression et résultats

Le script affiche une barre de progression native pendant l'extraction, incluant :

- Pourcentage de completion
- Nombre d'éléments traités
- Estimation du temps restant (ETA)
- Barre visuelle de progression

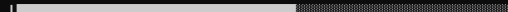
4.9 Récapitulatif avant extraction

Avant de commencer l'extraction, le programme affiche un récapitulatif formaté des paramètres sélectionnés :

```
PS C:\Users\Anas Nay\TEST_PROJET_HAL\python code> python .\main.py
Sensitivity threshold used: 2 (moderate)
Available CSV files:
1. export-mbre-test - export-mbre.csv.csv
2. export-mbre.csv

Enter the number of the file you want to use: 2
You selected the file: export-mbre.csv

=====
EXTRACTION SUMMARY
=====
• Extraction: all data (no filters)
• Matching: sensitivity moderate (distance = 2)
• Outputs: no additional output
=====

Starting extraction...
Extraction in progress:  | 57.0% (666/1169) ETA: 00:13
```

4.10 Organisation des fichiers de sortie

Les résultats sont organisés automatiquement :

- **Fichiers CSV d'extraction** : dossier `extraction/`
- **Graphiques HTML** : dossier `html/`
- **Images PNG** : dossier `png/`
- **Rapports** : dossier `rapports/`

5 Module de détection des doublons et homonymes

Ce module constitue une fonctionnalité avancée de l'outil permettant d'identifier et de traiter automatiquement les publications en double et les cas d'homonymie dans les données extraites de HAL. Cette fonctionnalité s'appuie sur une méthode utilisant les identifiants d'auteur dans la base HAL (`authIdPerson_i`) pour garantir une précision maximale dans la détection.

5.1 Principe de fonctionnement

Le système de détection repose sur un algorithme en plusieurs étapes :

1. **Groupement initial** : Les publications sont regroupées par couple (nom, prénom) d'auteur
2. **Enrichissement via API HAL** : Pour chaque publication, une requête est effectuée vers l'API HAL pour récupérer les métadonnées complètes, notamment les identifiants `authIdPerson_i`
3. **Analyse comparative** : Les publications d'un même auteur sont comparées selon plusieurs critères :
 - Similarité des titres (seuil par défaut : 0.8)
 - Écart temporel entre publications (seuil par défaut : 2 ans)
 - Identifiants HAL officiels
 - Position de l'auteur dans la liste des co-auteurs
4. **Classification automatique** : Les cas détectés sont classifiés en différentes catégories

5.2 Accès au module depuis l'interface graphique

Le module de détection est accessible via l'onglet "Détection Doublons & Homonymes" de l'interface principale :

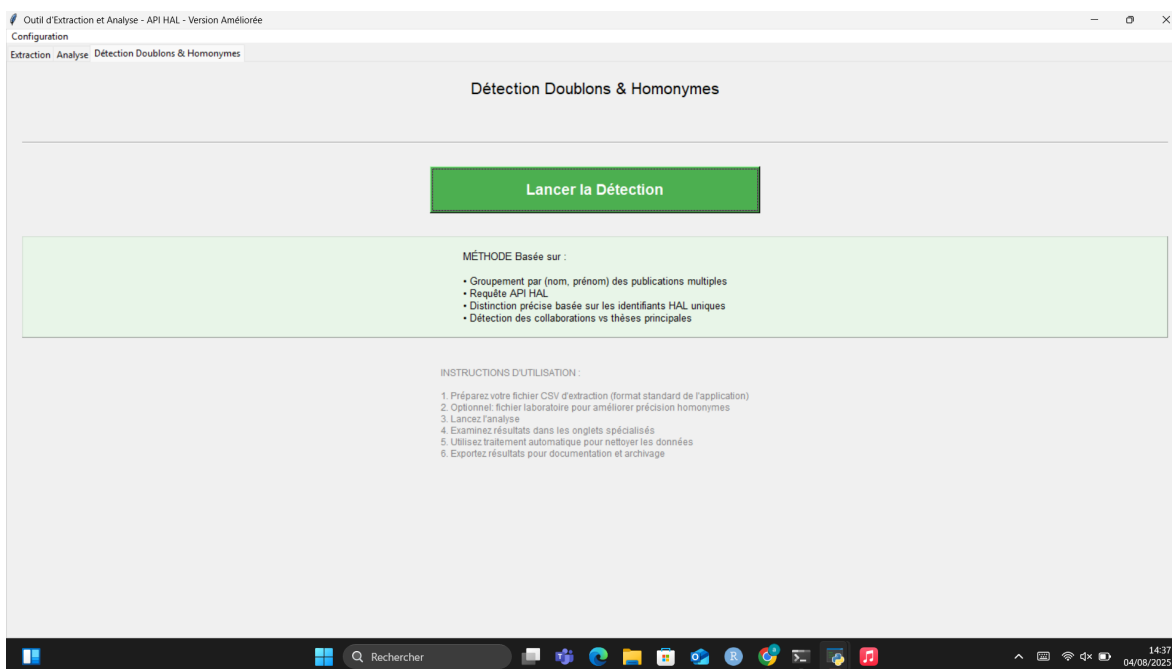


Figure 9: Interface d'accueil du module de détection

L'interface présente les caractéristiques principales de la méthode utilisée et propose un bouton central pour lancer l'analyse. Le processus nécessite un fichier CSV d'extraction préalablement généré par l'outil.

5.3 Configuration et lancement de l'analyse

5.3.1 Sélection des fichiers d'entrée

Le processus d'analyse débute par la sélection du fichier CSV à analyser. Le système propose automatiquement les fichiers du dossier **extraction** et demande si l'utilisateur souhaite utiliser un fichier laboratoire optionnel.

5.3.2 Fichier laboratoire optionnel

L'utilisateur peut optionnellement fournir un fichier contenant les informations de laboratoire des auteurs. Ce fichier doit contenir les colonnes **nom**, **prenom** et **unite_de_recherche**. Cette information supplémentaire améliore significativement la précision de détection des homonymes (en se référant au laboratoire d'affiliation des auteurs, ce qui permet de mieux les différencier).

5.4 Interface d'analyse et résultats

Une fois les fichiers sélectionnés, l'analyse démarre automatiquement dans une interface dédiée organisée en onglets thématiques.

5.4.1 Onglet Résumé

Les résultats sont dans un premier temps présentés dans un onglet résumant l'analyse. En voici un exemple :

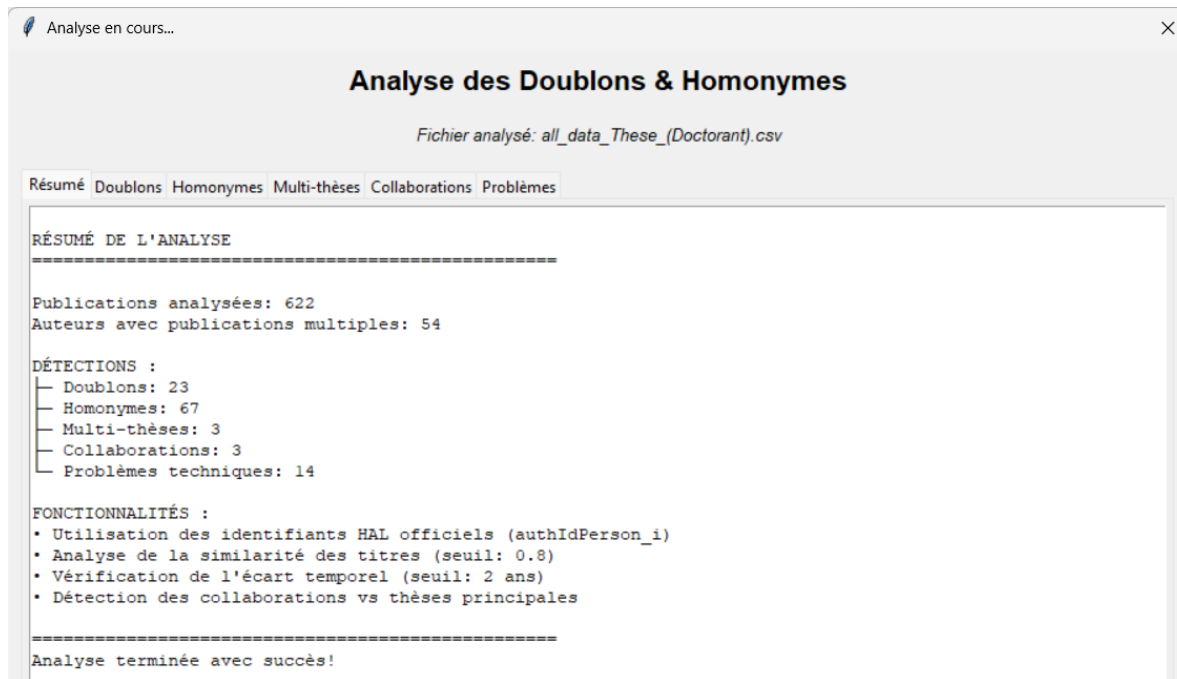


Figure 10: Résumé Détection Doublons et Homonymes

Cet onglet présente une vue d'ensemble des résultats d'analyse avec les statistiques globales incluant :

- Nombre total de publications analysées
- Nombre d'auteurs avec publications multiples
- Détections par catégorie (doublons, homonymes, multi-thèses, collaborations)
- Informations techniques sur la méthode utilisée

5.4.2 Onglets de résultats détaillés

Les différents onglets présentent les cas détectés organisés par catégorie :

- **Onglet Doublons** : Publications identifiées comme doublons potentiels avec scores de similarité, titres comparés et années de publication
- **Onglet Homonymes** : Cas d'homonymie avec critères de différenciation (identifiants HAL différents, domaines scientifiques distincts, laboratoires différents)
- **Onglet Collaborations** : Distinction entre thèses principales et collaborations d'un même auteur
- **Onglet Multi-thèses** : Cas rares d'auteurs avec multiples thèses

- **Onglet Problèmes** : Publications sans `authIdPerson_i` ou métadonnées incomplètes

5.5 Types de détection supportés

Le module identifie plusieurs catégories de problèmes :

Table 1: Types de détection et critères

Type	Critères de détection
Doublons	Même <code>authIdPerson_i</code> , similarité des titres > 0.8 , écart temporel < 2 ans
Homonymes	<code>authIdPerson_i</code> différents, même (nom, prénom), domaines/laboratoires distincts
Multi-thèses	Même <code>authIdPerson_i</code> , écart temporel > 3 ans, faible similarité titres
Collaborations	Position non-principale dans la liste d'auteurs, domaines différents
Problèmes techniques	Absence d' <code>authIdPerson_i</code> , métadonnées HAL incomplètes

5.6 Traitement automatique des données

Le module propose des fonctionnalités de traitement automatique pour nettoyer les données détectées via une interface dédiée accessible après l'analyse.

5.6.1 Options de traitement disponibles

Plusieurs niveaux de traitement sont proposés :

- **Suppression des doublons** : Élimination automatique des publications dupliquées (conservation de la première occurrence)
- **Marquage des homonymes** : Ajout d'une colonne `Homonyme_Potentiel` pour signaler les cas ambigus
- **Suppression des collaborations** : Élimination des collaborations en conservant uniquement les thèses principales
- **Signalement des multi-thèses** : Marquage des cas rares de multiples thèses par un même auteur

5.6.2 Résultats du traitement

Après traitement, le système génère :

- Un fichier CSV nettoyé (suffixe `_nettoye.csv`)
- Un rapport détaillé des actions effectuées
- Des statistiques comparatives avant/après traitement

5.7 Exportation des résultats

Le module offre des fonctionnalités d'exportation complètes pour documenter et archiver les résultats d'analyse. L'utilisateur peut choisir le dossier de destination et le système génère automatiquement plusieurs fichiers spécialisés.

5.7.1 Fichiers générés

L'exportation produit plusieurs fichiers spécialisés :

- `*_doublons_detecte.csv` : Liste détaillée des doublons avec scores de similarité
- `*_homonymes_detecte.csv` : Cas d'homonymie avec informations de différenciation
- `*_multi_theses.csv` : Cas rares de multiples thèses par auteur
- `*_collaborations.csv` : Distinction thèses principales/collaborations
- `*_resume_detecte.txt` : Rapport complet de l'analyse avec méthodologie

5.8 Utilisation en ligne de commande

Le module de détection est également accessible via l'interface en ligne de commande avec l'argument `--analyse` :

```
python main.py --analyse
```

Cette approche propose les mêmes fonctionnalités dans un environnement textuel interactif :

1. Sélection interactive du fichier CSV à analyser
2. Configuration optionnelle du fichier laboratoire
3. Analyse avec affichage des résultats en temps réel
4. Options de traitement et d'exportation intégrées

5.8.1 Exemple de session en ligne de commande

```
# Lancement de l'analyse
python main.py --analyse

# Selection du fichier (interface interactive)
Fichiers disponibles dans 'extraction/':
1. all_data_These_Doctorant.csv
2. publications_2020-2024.csv
Selectionnez un fichier (1-2): 1

# Configuration du fichier laboratoire
Utiliser un fichier laboratoire ? (o/n): o
Fichier laboratoire selectionne: laboratoires.csv

# Analyse en cours avec barre de progression
Analyse en cours... (patience requise - interrogation API HAL)
Analyse de Dupont Jean (3 publications) - 15/120...

# Affichage des resultats details
```

```
RESULTATS DE L'ANALYSE
```

```
Publications-analysees:-450
Auteurs-avec-publications-multiples:-120
Doublons-detectes:-23
Homonymes-detectes:-8
Collaborations-detectees:-15
```


5.9 Limitations et considérations

5.9.1 Dépendance à l'API HAL

La précision de la détection dépend de la qualité des métadonnées HAL :

- Publications sans `authIdPerson_i` : traitement par similarité de titres uniquement
- Métadonnées incomplètes : classification dans les "problèmes techniques"
- Évolution des données HAL : possibles variations dans le temps

5.9.2 Cas particuliers

Certains cas nécessitent une validation manuelle :

- Changements de nom d'auteur (mariage, etc.)
- Collaborations internationales avec variations d'orthographe
- Publications avec erreurs de métadonnées côté HAL

5.10 Conclusion

Le module de détection des doublons et homonymes constitue un outil puissant pour améliorer la qualité des données d'analyse bibliométrique. En combinant l'utilisation des identifiants officiels HAL avec des heuristiques avancées, il permet une détection automatique fiable de la majorité des cas problématiques tout en proposant des solutions de traitement adaptées.

L'intégration de ce module dans l'outil global garantit une chaîne de traitement complète, depuis l'extraction initiale jusqu'à l'analyse finale, en passant par une phase de nettoyage et de validation des données qui améliore significativement la pertinence des résultats obtenus, et donc des statistiques générées par la suite.

6 Annexe

6.1 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](#)

6.2 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 2: 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.