Université de Montréal

IFT 3150

Rapport de projet

Par

Guillaume Genois, 20248507

guillaume.genois@umontreal.ca

Responsable du projet : Eugène Syriani

syriani@iro.umontreal.ca

Présenté à Benoit Baudry

dans le cadre du cours IFT 3150

Projet d'informatique

Automne 2024

23 décembre 2024

Résumé

Les revues littéraires systématiques (RLS) constituent une méthodologie clé pour synthétiser les connaissances scientifiques en suivant un processus rigoureux. Ce projet s'inscrit dans une recherche sur l'application des modèles d'apprentissage génératifs (large language models - LLM) pour automatiser des tâches critiques des RLS, notamment la sélection d'articles, souvent laborieuse et source de biais. L'objectif principal est de créer des ensembles de données annotés pour évaluer l'efficacité des LLM dans cette tâche. Les données utilisées proviennent d'une quinzaine de RLS publiées, représentant plus de 30 000 articles. Ces données incluent des métadonnées (titres, résumés, mots-clés) ainsi que des informations décisionnelles (critères d'inclusion/exclusion, décisions des réviseurs). Cependant, ces informations sont souvent incomplètes et nécessitent une normalisation pour garantir leur exploitabilité. Pour cela, plusieurs étapes ont été menées : analyse des méthodologies de RLS, extraction des métadonnées et compilation des données à partir de fichiers HTML et BibTeX issus des principaux moteurs de recherche (IEEE, ACM, Science Direct, etc.). Des algorithmes ont été développés pour automatiser la récupération, le nettoyage et l'alignement des données. Au total, 16 jeux de données standardisés ont été créés, regroupant environ 30 000 articles. Ce projet contribue au développement d'outils permettant d'assister les chercheurs dans la curation de données tout en offrant une opportunité d'apprentissage approfondi de la méthodologie des RLS et de leur application à des cas pratiques.

Mots-clés : revue littéraire systématique, Selenium, automatisation robotisée des processus, jeu de données étalons, génie logiciel, extraction de données du web

1. Introduction

1.1. Contexte

Une revue littéraire systématique (RLS) est une méthodologie de recherche visant à collecter, identifier et analyser de manière critique les études de recherche à travers une procédure systématique. L'objectif est d'examiner les points critiques des connaissances actuelles sur un sujet, en lien avec des questions de recherche, afin de suggérer des axes d'exploration future [1].

Les outils logiciels dédiés à la réalisation des RLS permettent de définir un processus systématique à suivre et d'automatiser le plus grand nombre possible de tâches pour les chercheurs. Avec les avancées des modèles d'apprentissage génératifs (large language models - LLM), nous explorons comment automatiser les tâches de RLS les plus ardues, en particulier la sélection d'articles qui est la tâche la plus longue et la source principale d'erreur et de biais. Les LLM peuvent donc aider à automatiser ou assister les chercheurs dans cette tâche en faisant une présélection ou un tri des articles pertinents. Pour utiliser un LLM de façon automatisée, il faut trouver la bonne requête à formuler (prompt engineering) avec la bonne structure, formuler les bons termes et identifier si des exemples ou explications sont nécessaires pour obtenir le meilleur résultat. Pour la sélection dans une RLS, le LLM doit décider si un article doit être inclus ou exclu de la RLS.

1.2. Problématique

Ce projet vise à construire des ensembles de données fiables et annotées avec la bonne décision de sélection afin d'évaluer l'efficacité des LLM dans la sélection d'articles. Les données proviennent de RLS déjà publiées et pour lesquelles le processus de sélection d'articles est disponible. Nous avons déjà identifié une 15aines de publications (30 000+ articles au total). Les données incluent les métadonnées des articles (ex. titre, résumé, mots-clés) et l'information sur le processus décisionnel (ex. décision de chaque réviseur, décisions conflictuelles, décision finale). Cependant, ces données sont souvent incomplètes et représentées dans des formats différents. De plus, les informations décisionnelles doivent souvent être inférées à partir de l'article publié et des données disponibles.

1.3. Contributions

Le projet consiste à faire la collecte des métadonnées des articles ciblés et l'analyse des informations à inférer pour construire des ensembles de données annotés. Pour ce faire, il est nécessaire d’effectuer le nettoyage, la définition et l'alignement de tous ces articles. Des algorithmes sont développés pour suivre une automatisation robotisée des processus de cette tâche afin de réduire le risque d'erreur. En parallèle, l'étudiant M.Sc. Gauransh Kumar développe l'outil pour évaluer l’efficacité des LLM dans la sélection d’articles pour les RLS avec cet ensemble de données. De plus, personnellement, l'analyse de RLS me permet d'apprendre la méthodologie de

ce type d'étude. Avec le développement d'outil de curation de données, j'applique les connaissances acquises dans les cours que j'ai suivis sur des cas réels.

1.4. Plan

Dans ce rapport, nous présenterons au chapitre 2 l'état de l'art des outils et technologies pertinents. Dans le chapitre 3, nous présenterons la méthodologie utilisée pour construire les jeux de données. Dans le chapitre 4, nous présenterons plus en détail le fonctionnement de l'application de recherche et d'extraction d'articles automatiques dans les principales bases de données. Dans le chapitre 5, nous présenterons ensuite en détail le paquet logiciel d'extraction des métadonnées des HTML et BibTeX. Dans le chapitre 6, nous présenterons les résultats obtenus, incluant la création et la validation des jeux de données. Nous conclurons, dans le chapitre 7, sur les points à améliorer et les perspectives futures du projet.

2. État de l'art

2.1. LLM appliqués aux RLS

Les RLS sont des méthodologies rigoureuses destinées à synthétiser les connaissances existantes sur un sujet spécifique, jouant un rôle essentiel dans l'orientation des recherches futures [1]. Cependant, le processus de tri des articles pertinents, basé sur leurs titres, résumés et critères préétablis, est notoirement long et sujet à des erreurs humaines. Avec l'émergence des LLM, une nouvelle voie s'ouvre pour automatiser cette étape, en combinant efficacité et précision [2].

Les LLM ont démontré leur capacité à évaluer la pertinence des publications grâce à des approches automatisées basées sur des invites structurées et des systèmes de classification. Par exemple, ces outils atteignent des niveaux de sensibilité élevés tout en maintenant une spécificité notable, bien que variant selon les modèles et jeux de données [3]. Intégrés au travail des experts humains, les LLM peuvent réduire considérablement le temps de sélection tout en améliorant la cohérence [4]. Dans le cadre du projet, la curation des métadonnées des articles issus des revues systématiques s'aligne directement avec cette innovation : elle offre une base structurée et normalisée pour faciliter la recherche à l'intégration et l'utilisation des LLM, accélérant ainsi les processus d'analyse tout en assurant une robustesse face à la variabilité des sources.

2.2. Automatisation robotisée des processus

L'automatisation robotisée des processus (ARP), ou Robotic Process Automation (RPA), propose une approche non intrusive pour l'automatisation des flux de travail en définissant et en opérationnalisant des règles d'automatisation via les interfaces graphiques des outils d'ingénierie et de gestion. Grâce à son cycle de développement rapide, l'ARP est devenue un élément central dans de nombreuses initiatives actuelles de transformation numérique [5]. Celle-ci est une technologie qui permet d'automatiser des tâches répétitives et basées sur des règles en utilisant des robots logiciels capables d'interagir avec les systèmes informatiques de la même manière qu'un humain. Cette approche vise à améliorer l'efficacité opérationnelle, réduire les erreurs et libérer les employés de tâches monotones pour qu'ils puissent se concentrer sur des activités à plus forte valeur ajoutée [6].

Dans le cadre de ce projet, l'ARP est particulièrement pertinente pour automatiser le processus de récupération des articles utilisé dans les RLS. Cette tâche, traditionnellement effectuée manuellement, est fastidieuse et sujette à des erreurs. En appliquant des techniques d'ARP, il est possible de développer des robots capables de récupérer automatique les articles pertinents, améliorant ainsi l'efficacité et la fiabilité du processus de création de jeu de données [7]. Dans ce projet, le robot viendra répliquer l'utilisateur qui recherche chaque article, un à la fois, dans chaque base de données, et enregistrera sa page HTML et son BibTeX.

2.3. Selenium

Selenium est une bibliothèque Python populaire et puissante permettant l'automatisation des navigateurs web [8]. Initialement conçue pour effectuer des tests d'applications web, elle est devenue un outil polyvalent utilisé dans divers domaines, tels que le « web scraping », l'automatisation de flux de travail et la simulation d'interactions utilisateur. Selenium fonctionne avec plusieurs navigateurs (comme Chrome, Firefox, et Edge) et offre une interface simple pour automatiser des tâches complexes, comme remplir des formulaires, cliquer sur des boutons, ou naviguer entre des pages [9].

Aujourd'hui, Selenium est largement utilisé en raison de sa flexibilité et de son extensibilité. Avec l'intégration de WebDriver, un composant clé de Selenium, il est possible d'émuler de manière réaliste le comportement d'un utilisateur sur un navigateur. Cela inclut la gestion des

témoins, des scripts JavaScript, et même des interactions avec des éléments dynamiques générés par des cadres logiciels modernes comme React ou Angular. Cette capacité à interagir avec des sites complexes en temps réel dépasse les simples méthodes de scraping basées sur les requêtes HTTP et rend Selenium indispensable dans des contextes où une navigation complète est nécessaire.

Dans le cadre de ce projet, Selenium joue un rôle clé pour automatiser la récupération d'informations pertinentes à partir des principaux moteurs de recherche scientifique tels qu'IEEE [10], ACM [11] ou Springer Link [13]. Par exemple, il peut être utilisé pour naviguer automatiquement sur ces plateformes, effectuer des recherches selon les titres des articles, extraire leurs métadonnées et enregistrer les pages HTML et BibTeX associées. Cette approche réduit considérablement la charge manuelle tout en garantissant un processus reproductible et efficace. De plus, son intégration dans des flux automatisés, couplée à d'autres outils, renforce la capacité à construire des ensembles de données cohérents et exploitables pour la sélection des articles dans une revue systématique. Il s'agit donc d'une technologie d'ARP intéressante à notre problématique.

3. Méthodologie

Toute la programmation effectuée dans ce projet est en Python.

3.1. Analyse de la méthodologie des RLS

Chaque RLS a des jeux de données avec leur propre méthodologie. Ces données sources des RLS ont été récupérées en communiquant directement avec les auteurs de celles-ci qui nous ont envoyé leurs données disponibles. Un travail d'analyse pour chacune d'entre elles est donc nécessaire pour extraire les informations utiles de leurs données sources disponibles et venir les standardiser. Généralement, ce travail d'analyse consiste à identifier où les données qui nous intéressent se trouvent, car quelquefois elles sont toutes sur une feuille Excel, parfois elles sont sur plusieurs feuilles. Ensuite, pour trouver quels articles ont été sélectionnés lors des phases de tris, il faut parfois simplement regarder une colonne booléenne qui l'indique ou parfois calculer la différence entre deux feuilles différentes. Chaque RLS est donc du cas par cas.

3.2. Extraction des informations utiles du jeu de données fourni

Une fois l'analyse complétée, le but est d'extraire toutes les informations nécessaires à la reproductivité du processus de sélection d'articles et de les standardiser dans le jeu de données créé. Pour chaque RLS, la logique est implémentée dans un objet différent dont chacun hérite de la même classe parente. Les informations sont compilées dans un Pandas Dataframe pour chaque RLS. Ainsi, chaque RLS peut être standardisée selon ses manipulations nécessaires et être ensuite utilisée dans l'application principale de la même façon.

Les métadonnées récupérées à cette phase-ci sont donc les informations sur le mode de récupération de l'article (nouvelle sélection ou boule de neige), leur critère d'exclusion et/ou d'inclusion, leur décision après les premières phases de sélection par le résumé, leur décision finale et le nombre d'évaluateurs. Parfois, dans les données sources, certaines de ces informations, dont régulièrement les critères d'exclusion ou d'inclusion sur chacun des articles, sont manquants. Pour ces informations, il nous est impossible d'aller les combler. Parfois, certaines métadonnées des articles sont aussi présentes dans les données sources, mais aucune RLS ne comporte toutes les métadonnées nécessaires, alors il a fallu procéder à du « web scraping » pour récupérer les métadonnées manquantes.

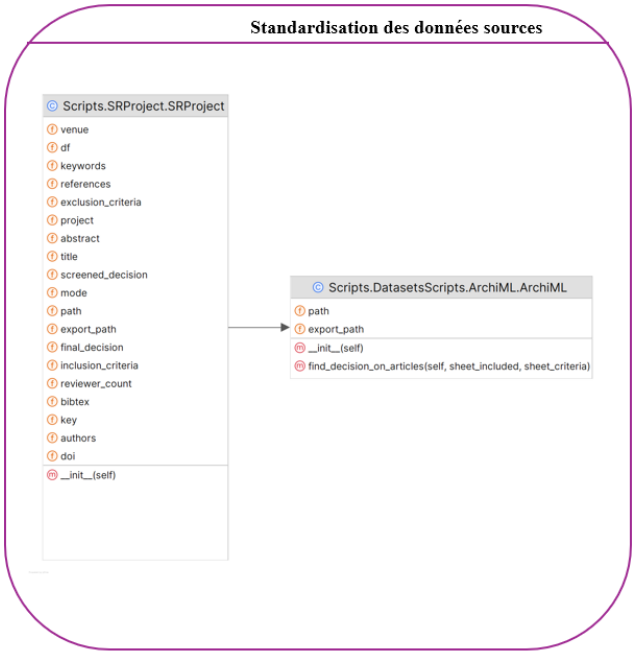

Figure 1 - Diagramme de classe de ArchiML [33] (une RLS) héritant de SRProject

3.3. Recherche automatique dans les principales bases de données

Afin de récupérer les articles présents sur les principaux moteurs de recherche, un système automatisé, exploitant Selenium, vient lancer des requêtes sur les sites web des moteurs de recherche. Sur chaque requête, le système vérifie si un des articles trouvés est l'article recherché en comparant les titres. Une fois un article trouvé, le système enregistre la page HTML courante et télécharge le BibTeX offert par la base de données. Ce système informatique viend s'inscrire comme une ARP. De plus, chaque article extrait de cette façon est identifié par un marqueur à la fin de son nom de fichier (_XX.html) afin de retrouver facilement de quelle source ce HTML ou BibTeX provient. Celui-ci est ensuite enregistré dans une banque de données.

3.4. Extraction des métadonnées des HTML (parser)

Un système d'extraction des métadonnées voulues devra venir extraire les informations des HTML des articles présents sur les principaux moteurs de recherche, soient IEEE [10], ACM [11], Science Direct [12], Springer Link [13], Web of Science [14], Scopus [15], PubMed Central [16] et arXiv [17]. Ce système est indépendant des autres et pourrait être utilisé dans le futur sans le système de recherche automatique sur le web. Il ne demande qu'un fichier HTML en paramètre avec sa base de données de provenance afin de retourner les métadonnées de l'article. Les métadonnées que cet outil vient récupérer d'un article sont le titre, le résumé, les mots-clés, les auteurs, le lieu de publication, le DOI, les références, les pages, la source, l'année, le lien vers l'article et l'éditeur.

3.5. Nettoyage et compilation des données de chaque jeu de données

Avec les HTML et les BibTeX extraits et enregistrés, pour chaque article présent dans un jeu de données, tous ces fichiers lui étant associés, provenant de plusieurs ou non moteurs de recherche, sont envoyés au système d'extraction de métadonnées. Les informations sont donc compilées en essayant de récupérer le maximum d'informations à travers les sources extraites. Un jeu de données intermédiaires est ainsi créé avec certaines informations supplémentaires telles que le titre original dans les données sources et quelles métadonnées ont été manquées lors de l'extraction automatique. Ces colonnes permettent de valider les résultats extraits et d'essayer de déceler des problèmes dans le jeu de données. Avec ces problèmes identifiés, je repasse sur ceux-ci en les corrigeant manuellement et en allant extraire manuellement les articles manquants. Généralement, les articles manquants ne sont pas trouvés par l'application d'extraction automatique, car ceux-ci

se retrouvent sur d'autres journaux que ceux principaux où les recherches sont faites. Finalement, après les dernières corrections, les deux colonnes pour valider sont supprimées et la version finale est envoyée.

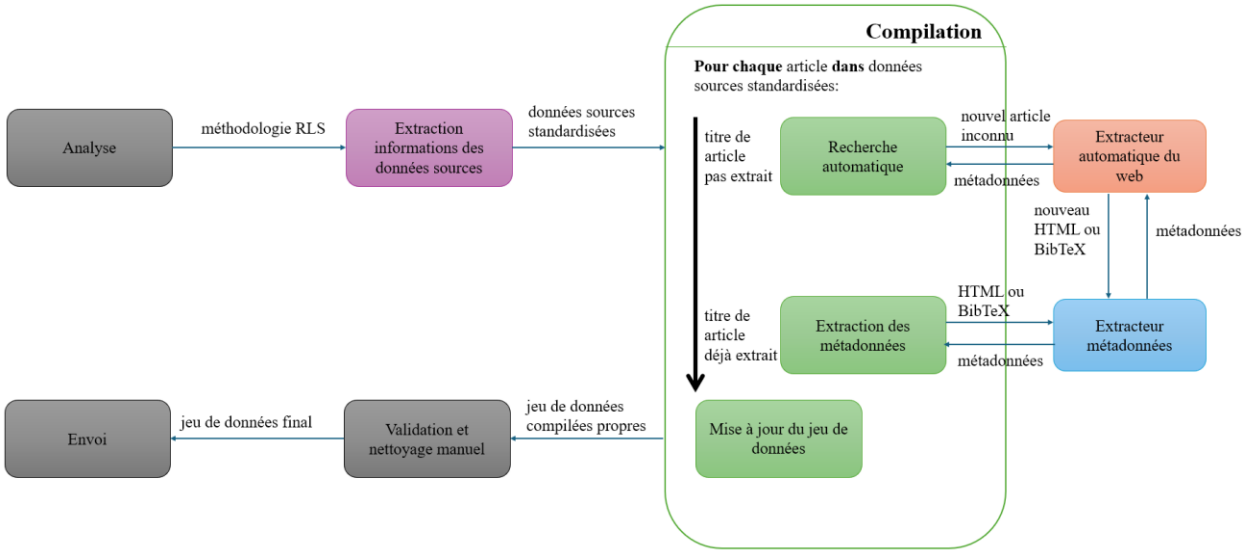

Figure 2 - Diagramme de flux simplifié du pipeline global

4. Recherche automatique dans les principales bases de données

4.1. Fonctionnement

Le système de recherche automatique dans les principales bases de données d'articles scientifiques est utilisé lors de la compilation des articles des données sources standardisées de la RLS courante. Lors de la compilation, une instance du système est créée et les articles des données sources manquants sont recherchés un par un. À noter, cependant, qu'il est très bien possible de lancer en parallèle le système plus d'une fois afin d'accélérer les recherches sur plus d'une base de données par exemple. L'instance du système de recherche se crée ensuite une instance d'un objet contenant les méthodes adaptées aux différentes bases de données. L'instance de recherche permet de gérer l'identification de la source du nouvel article manquant à rechercher et permet ensuite de rediriger vers la bonne fonction de l'instance avec les méthodes adaptées. Elle permet aussi de gérer

lorsqu'un article a un lien, ou n'en a pas, dans les données sources. L'instance de recherche gère aussi l'instance du pilote Selenium.

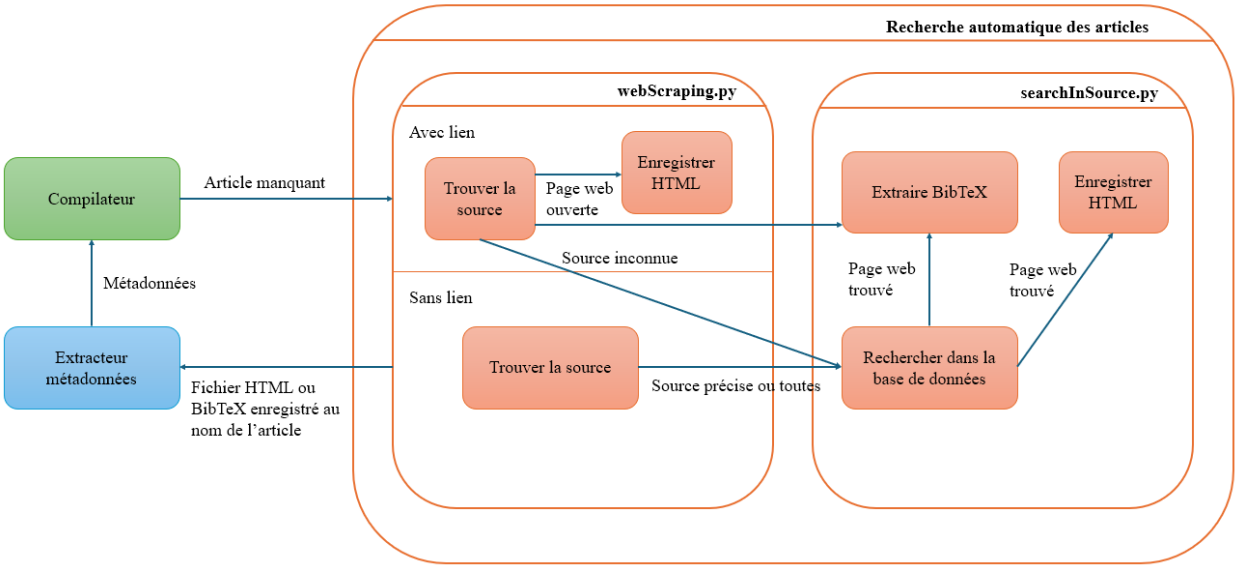

Figure 3 - Diagramme de flux simplifié de la recherche automatique

4.2. Gestion d'un lien donné dans les données sources et extraction

Le processus d'extraction des métadonnées suit les étapes suivantes : si un lien est fourni, identifier automatiquement la source et ouvrir le lien. Si un DOI est disponible, localiser la source correspondante, bien que cela puisse être plus complexe, et ouvrir le lien si la source est connue. En l'absence de lien, effectuer une recherche par titre, prioritairement dans une source spécifiée, ou élargir la recherche à toutes les sources disponibles si aucune n'est précisée. Une fois la page web de l'article ouvert, celle-ci est enregistrée en HTML dans une banque de données avec toutes les autres pages HTML au nom de la date suivi du titre de l'article et d'un identifiant à deux chiffres de la source (date_titre_XX.html).

Il est facile d'identifier la source lorsqu'un lien tel que « https://link.springer.com/article/... » est donné. Lorsqu'il s'agit d'un DOI, le système tente d'identifier la source à travers l'API de Crossref [18] et sinon ouvre le lien directement pour regarder sur quelle source le système est tombé. Si aucune source n'est encore trouvée ou qu'aucun lien n'était donné, alors le système procède à des recherches dans les bases de données. Il est important d'avoir la source pour extraire l'article, car autrement nous ne saurons pas comment lire

le HTML et comment extraire le BibTeX. En effet, le BibTeX est extrait lorsque possible, car celui-ci est une métadonnée ajoutée au jeu de donnée finale, et celui-ci permet aussi une extraction des métadonnées plus simples. Régulièrement, certaines informations ne sont pas affichées sur la page web, mais le sont dans le BibTeX, alors celui-ci aide à compléter les informations.

4.3. Recherche dans les bases de données

Le système de recherche comporte des fonctions adaptées à chaque base de données. Ces fonctions effectuent une recherche d'article sur une base de données en utilisant un titre. Elles chargent la page de recherche, réinitialisent les recherches précédentes si nécessaire, sélectionnent une recherche par titre pour améliorer la pertinence, saisissent le titre nettoyé, lancent la recherche, trient les résultats par pertinence, et ouvrent le premier document. Ensuite, elles vérifient si le titre correspond, enregistrent le contenu HTML, le BibTeX et le lien de l'article, puis retournent les métadonnées extraites. Elles incluent des mécanismes de gestion d'erreurs et de réessaie pour ajouter de la flexibilité. Lors de l'extraction des BibTeX, un processus de nettoyage et de standardisation des identifiants est effectué afin d'assurer la compatibilité et d'éviter les erreurs potentielles avec les librairies utilisées.

4.4. Validation des titres

Cette étape est cruciale pour assurer l'exactitude des titres d'articles et leur correspondance dans les bases de données lors de recherches automatisées tout en permettant une flexibilité en cas de différence minime (erreur humaine de retranscription, titre incomplet, titre différent selon la base de données). Une fonction commence par nettoyer et standardiser les titres d'articles. Elle supprime les caractères illégaux ou indésirables, remplace certains symboles par des espaces, transforme le texte en minuscules, enlève les accents, et élimine les mots très courts. Son objectif est de produire une version simplifiée et normalisée du titre, facilitant les comparaisons textuelles. Une deuxième fonction vérifie ensuite si les titres nettoyés sont les mêmes, en permettant une distance de Levenshtein minimale entre les deux et des différences mineures dans le nombre de mots et la longueur.

4.5. Composante aide au manuel

Pour faciliter l'extraction manuelle des métadonnées, une petite composante logicielle a été développée. Celle-ci permet de saisir le titre, le lien vers l'article et la source du lien afin d'extraire et d'ajouter à la banque de données automatiquement le HTML et le BibTeX de cet article afin de le compiler par après. Cet outil est particulièrement utile dans les cas où les mécanismes de recherche automatique échouent à récupérer les articles nécessaires.

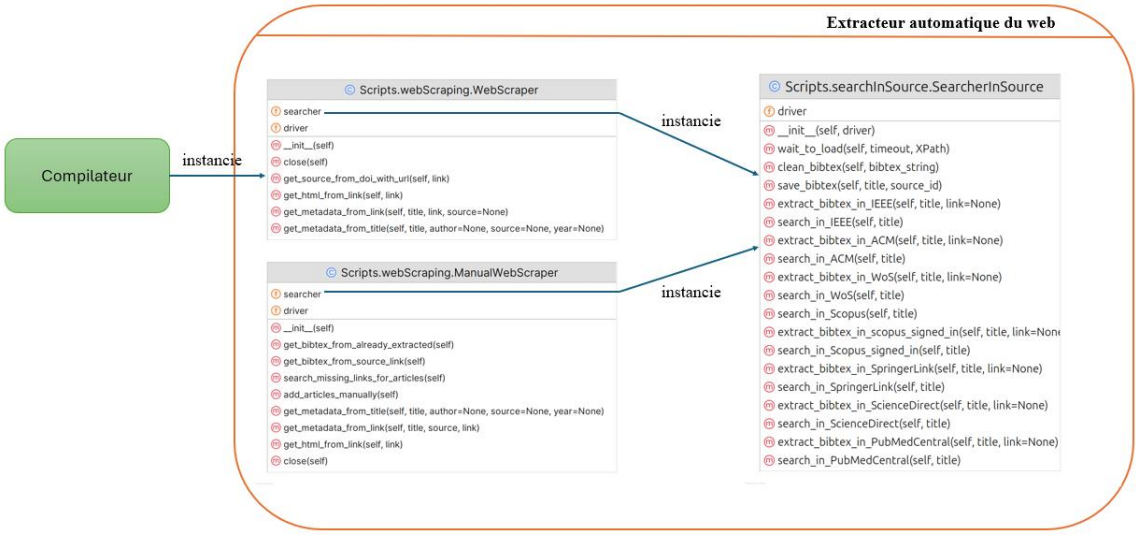

Figure 4 - Diagramme de classe du système d'extraction automatique des articles

5. Extraction des métadonnées des HTML et BibTeX

5.1. Fonctionnement global

Il s'agit d'un paquet logiciel qui fournit des fonctions adaptées en fonction de la source des données, qu'il s'agisse de fichiers HTML ou BibTeX. Conçu pour fonctionner de manière autonome, il présente une sensibilité réduite par rapport à l'extracteur automatique présenté précédemment. Il procède à l'extraction des informations en s'appuyant sur les balises clés, suivies d'un processus de nettoyage et de standardisation des métadonnées, lesquelles sont organisées dans une structure de type dictionnaire.

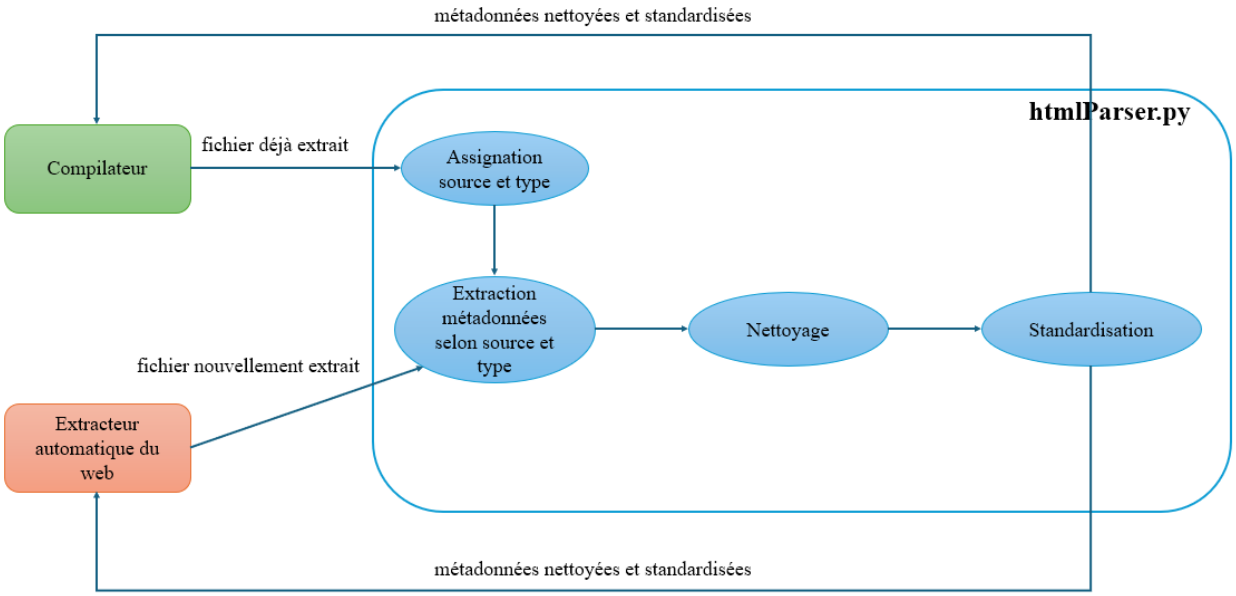

Figure 5 - Diagramme de flux simplifié de l'extraction des métadonnées des HTML et BibTeX

De plus, une fonction est présente aussi pour prendre en entrée un fichier de la banque de données d'articles extraits et déterminer où envoyer son contenu pour extraire ses métadonnées. Un BibTeX et un HTML sont différenciés par leur extension, mais la source (IEEE, ACM, …) est déterminée via l'identificateur apposé au nom du fichier lors de sa sauvegarde par le système d'extraction automatique. Ainsi, la fonction redirige vers la bonne fonction auxiliaire selon la source qui fournit le contenu du fichier et renvoie ses métadonnées.

5.2. Extraction des métadonnées des HTML

L'approche repose sur l'utilisation de la bibliothèque BeautifulSoup [19], qui offre une représentation en arbre du code HTML. Chaque base de données (telles qu'IEEE, ACM, etc.) dispose de fonctions spécifiques adaptées à ses balises HTML pour extraire les informations nécessaires. BeautifulSoup présente une flexibilité accrue par rapport à Selenium dans la manipulation des balises HTML, car elle permet de parcourir dynamiquement l'arbre pour localiser les informations, tandis que Selenium requiert des chemins absolus, souvent sujets à des modifications. Cette capacité à naviguer dans l'arbre HTML confère une robustesse accrue, notamment grâce à l'ajout de conditions permettant de capturer les métadonnées de différentes façons, quel que soit le niveau de profondeur où elles se trouvent. Une fois les métadonnées

extraites, elles sont transmises à une fonction de standardisation qui les convertit en un format structuré sous forme de dictionnaire et en profite pour les nettoyer par le fait même.

5.3. Extraction des métadonnées des BibTeX

Pour les BibTeX, seulement une fonction unique permet d'extraire et nettoyer les métadonnées du fichier. Ensuite, celles-ci sont structurées de manière uniforme dans un dictionnaire. À partir des entrées du fichier, elle récupère, si possible, les métadonnées comme le titre, le journal, les auteurs, le résumé, les mots-clés, les références, les pages, l'année, et d'autres attributs tels que le DOI ou le lien URL. Pour chaque champ, la fonction vérifie sa présence dans les données sources avant de l'extraire. Elle applique également les fonctions de nettoyage explicitées par après. Enfin, chaque valeur extraite est convertie en caractères ASCII pour garantir une compatibilité optimale. L'utilisation de la librairie Pybtex [20] permet de généraliser la lecture des fichiers BibTeX en réduisant les erreurs dues aux variations de format ou aux encodages spécifiques.

5.4. Nettoyage

Les fonctions de nettoyage visent à standardiser et purifier les métadonnées extraites afin de les rendre cohérentes et exploitables. De base, chaque métadonnée a ses caractères spéciaux convertis en ASCII pour éviter les problèmes d'encodage et d'accent. Une fonction s'occupe de nettoyer les résumés en supprimant les préfixes tels que « Abstract: » et en éliminant les mentions de droits d'auteur (par exemple, « Copyright »).

Une autre fonction traite les noms d'auteurs en retirant les caractères parasites tels que « , », « ; » et les chiffres. Elle filtre également les mentions inutiles comme « ORCID » et reformate les noms en supprimant les mots superflus tels que « and ». Elle veille à ce que chaque auteur soit présenté de manière uniforme et claire.

Enfin, une dernière fonction nettoie les noms des éditeurs en éliminant les mentions génériques comme « All rights reserved. » ainsi que les chaînes numériques inutiles avant ou après le texte principal. Elle garantit une version standardisée des noms d'éditeurs, facilitant ainsi leur utilisation dans les analyses ultérieures. Les autres informations telles que les auteurs et les lieux de publication sont préalablement nettoyées dans les fonctions distinctes aux sources, car celles-ci sont du cas par cas.

6. Résultats

Au total, 16 jeux de données ont été créés, représentant 32 614 articles au total. Chacun de ces jeux de données d'une RLS a fait l'objet d'une analyse de la méthodologie utilisée ainsi que d'une extraction des informations accessibles adaptée. Parmi cet ensemble, environ 900 articles ont été extraits manuellement. Au total, environ 99% des articles présents dans les données sources ont été récupérés et le système d'extraction automatique a extrait environ 97% des articles récupérés.

RLS	**Nombre d'articles récupérés automatiquement**	**Nombre d'articles récupérés manuellement**	**Nombre d'articles récupérés au total**	**Nombre d'articles dans les données sources**
TestNN [21]	168	7	175	175
ESM_2 [22]	114	0	114	114
DTCPS [23]	402	0	402	454
TrustSE [24]	475	78	553	556
Behave [25]	564	26	590	601
GameSE - abstract[1] [26]	1 080	53	1 133	1 135
OODP [27]	587	98	685	685
ESPLE [28]	963	0	963	963
SecSelfAdapt [29]	1 433	0	1 433	1 433
SmellReprod [30]	1 722	9	1 731	1 736
ModelGuidance [31]	1 771	6	1 777	1 777
ModelingAssist[2] [32]	*2 250*	*100*	*2 300*	2 350
ArchiML[2] [33]	*2 600*	*100*	*2 700*	2 766
GameSE - title[1] [26]	3 432	57	3 489	3 495
CodeCompr[2] [34]	*3 775*	*125*	*3 900*	3 999
CodeClone[2] [35]	*10 050*	*250*	*10 300*	10 454
Total	31 386	909	32 614	32 693

Tableau 1 - Résumé des jeux de données créés

[1] Deux jeux de données ont été créés pour GameSE, un pour la sélection d'articles selon leur titre et un selon leur résumé.

[2] Ces quatre jeux de données ne sont pas encore complétés. Les chiffres présents sont des estimations.

Les articles récupérés manuellement sont des articles qui ne se retrouvent pas dans les bases de données utilisées. Pour les articles non récupérés, régulièrement, certains « articles » dans les données sources étaient en fait des conférences et non un article en soi, d'autres étaient tout simplement introuvables. De plus, pour chacun de ces 15 RLS, une analyse de leur méthodologie employée a été faite afin de le traduire en code et extraire les informations utiles des données sources de ces RLS. Pour chacun d'entre elles, un objet a été créé représentant ses données sources standardisées. Par la suite, je suis allé récupérer leurs métadonnées afin d'aider à l'ingénierie de requête envoyée aux LLM faite par l'étudiant Gauransh. Les métadonnées récupérées sur ces RLS consitent surtout au sujet principal, aux définitions importantes dans la revue, aux questions de recherche utilisées et aux critères d'inclusion et d'exclusion utilisées pour sélectionner les articles. Ceci est dans le but de donner un meilleur contexte aux LLM.

Bien sûr, un résultat important de ce projet est l'application développée. Celle-ci comporte un paquet logiciel permettant l'extraction des métadonnées présentes dans les Bibtex des articles scientifiques ou des pages HTML de 8 importantes bases de données, soient IEEE, ACM, Science Direct, Springer Link, Scopus, Web of Science, PubMed Central et arXiv. Une autre partie de l'application développée est le système d'extraction automatique qui procède à des recherches automatiques dans 7 des 8 bases de données ci-mentionnées[3]. Ce système a permis l'extraction de de plus de 31 000 articles automatiquement.

Aussi, ce système est où se sont retrouvées le plus d'embûches au projet. En effet, comme mentionné précédemment, les bases de données sur le web ont changé plusieurs fois la disposition de leur contenu sur le site web. Ceci m'a beaucoup ralenti en me forçant de mettre à jour régulièrement les chemins utilisés. Une autre embûche posée par les sites web est que ceux-ci ont des systèmes de protection face aux attaques. J'ai donc dû travailler à contourner ces systèmes de protection, car les simples requêtes directes se faisaient bloquer. Des solutions que j'ai intégrées sont d'importer mon profil d'utilisateur personnel afin de camoufler ma signature de robot, de me connecter automatiquement à Science Direct afin que Scopus ne me bloque pas et de fermer et recommencer automatiquement Selenium afin que les bases de données oublient ma signature web. Une dernière embûche que je mentionnerai est la rigidité des moteurs de recherche des bases de

[3] arXiv a été omis, car il est préférable de récupérer les articles publiés et revus par les pairs. Il a été inclus dans l'extraction des métadonnées des HTML pour accélérer le processus manuel de récupération d'articles.

données. En effet, ceux-ci sont très rigides et ne réussissent pas à identifier des titres d'article mals écrits ou avec des symboles différents ce qui est arrivé régulièrement avec les données sources.

Finalement, pour valider les données, j'ai procédé à une vérification en comparant les titres, les auteurs, les sources, les lieux de publication et les dates, selon les informations disponibles. Une méthode simple dans l'application m'a permis d'identifier rapidement les articles présentant des divergences dans les titres, après quoi j'ai examiné chacun d'eux pour en confirmer la validité. Je me suis aussi assuré que le jeu de données était complet et exploitable. Par la suite, les jeux de données ont été transmis à l'étudiant Gauransh, qui a également effectué une validation indépendante de son côté.

7. Conclusion

Ce projet a permis de construire des ensembles de données annotés pour évaluer l'efficacité des LLM dans la sélection d'articles pour les RLS. En automatisant des tâches complexes comme l'extraction, le nettoyage et l'alignement des données, il a réduit les risques d'erreurs tout en garantissant la qualité des jeux de données. L'intégration d'outils comme Selenium et des algorithmes sur mesure a permis de standardiser le processus.

La compréhension des processus et des critères liés aux RLS s'est déroulée efficacement, et les apprentissages techniques réalisés ont été particulièrement enrichissants. Toutefois, certains aspects ont été plus problématiques, comme les restrictions imposées par certaines bases de données qui limitaient l'extraction automatique des articles, le manque de planification au début du projet, et une rigueur et une discipline parfois insuffisantes dans l'organisation des tâches. Cependant, ces défis m'ont permis de mieux comprendre l'importance de la planification et de la gestion du temps, notamment pour éviter les périodes d'attente prolongées lors de l'exécution des outils de collecte d'articles. J'ai également appris à anticiper les obstacles techniques et organisationnels pour améliorer l'efficacité du travail.

À l'avenir, des améliorations intéressantes seraient notamment la réduction de la sensibilité des outils aux variations des pages HTML, l'optimisation des stratégies de recherche pour contourner les limitations des bases de données, et le développement de méthodes plus généralisables pour capter et exploiter un plus large éventail de métadonnées.

8. Références

[1] Angela Carrera-Rivera, William Ochoa, Felix Larrinaga et Ganix Lasa, "How-to conduct a systematic literature review: A quick guide for computer science research." *MethodsX*, 2022.

[2] Syriani, E., David, I., et Kumar, G., "Screening articles for systematic reviews with ChatGPT". *Journal of Computer Languages*, 2024.

[3] Li, M., Sun, J., et Tan, X., "Evaluating the effectiveness of large language models in abstract screening: a comparative analysis", *Systematic Reviews*, 2024.

[4] Dennstädt, F., Zink, J., Putora, P. M., Hastings, J., et Cihoric, N., "Title and abstract screening for literature reviews using large language models: an exploratory study in the biomedical domain", *Systematic Reviews*, 2024.

[5] David, I., Sousa, V., et Syriani, E., "Opportunities in Robotic Process Automation by and for Model-Driven Software Engineering", *Enterprise, Business-Process and Information Systems Modeling*, 2023.

[6] Ribeiro, J., Lima, R., Eckhardt, T., et Paiva, S., "Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review", *Procedia Computer Science*, 2021.

[7] El-Gharib, N. M., et Amyot, D., "Robotic process automation using process mining — A systematic literature review", *Data & Knowledge Engineering*, 2023.

[8] The Selenium Project, "Selenium", *Selenium WebDriver Documentation*. Available at: https://www.selenium.dev/documentation/webdriver/.

[9] Gojare, S., Joshi, R., et Gaigaware, D., "Analysis and Design of Selenium WebDriver Automation Testing Framework", *Procedia Computer Science*, 2015.

[10] IEEE, "IEEE Xplore Digital Library", *IEEE*, disponible au https://ieeexplore.ieee.org/.

[11] ACM, "ACM Digital Library", *Association for Computing Machinery (ACM)*, disponible au https://dl.acm.org/.

[12] Science Direct, "ScienceDirect", *Elsevier*, disponible au https://www.sciencedirect.com/.

[13] Springer Link, "SpringerLink", *Springer Nature*, disponible au https://link.springer.com/.

[14] Web of Science, "Web of Science," *Clarivate Analytics*, disponible au https://www.webofscience.com/.

[15] Scopus, "Scopus", *Elsevier*, disponible au https://www.scopus.com/.

[16] PubMed Central, "PubMed Central (PMC)", *U.S. National Library of Medicine*, disponible au https://www.ncbi.nlm.nih.gov/pmc.

[17] arXiv, "arXiv e-Print Archive", *Cornell University*, disponible au https://arxiv.org/.

[18] Swagger, "Crossref Metadata API," *Swagger docs*, disponible au https://api.crossref.org.

[19] Richardson, Leonard, "Beautiful Soup," Beautiful Soup Documentation. Available at: https://www.crummy.com/software/BeautifulSoup/bs4/doc/#.

[20] The Pybtex Team, "Pybtex," *Pybtex Documentation*, disponible au https://pybtex.org/.

[21] Jin Zhang, et Jingyue Li, "Testing and verification of neural-network-based safety-critical control software: A systematic literature review," *Information and Software Technology*, 2020.

[22] Wang, Wei, Hourieh Khalajzadeh, John Grundy, Anuradha Madugalla, Jennifer McIntosh, et Humphrey O. Obie. "Adaptive user interfaces in systems targeting chronic disease: a systematic literature review," *User Modeling and User-Adapted Interaction*, 2024.

[23] Richard J. Somers, James A. Douthwaite, David J. Wagg, Neil Walkinshaw, et Robert M. Hierons, "Digital-twin-based testing for cyber–physical systems: A systematic literature review," *Information and Software Technology*, 2023.

[24] Hou, Fang, et Slinger Jansen. "A systematic literature review on trust in the software ecosystem," *Empirical Software Engineering*, 2022.

[25] Leonard Peter Binamungu, et Salome Maro, "Behaviour driven development: A systematic mapping study," *Journal of Systems and Software*, 2023.

[26] Jorge Chueca, Javier Verón, Jaime Font, Francisca Pérez, et Carlos Cetina, "The consolidation of game software engineering: A systematic literature review of software engineering for industry-scale computer games," *Information and Software Technology*, 2024.

[27] William Flageol, Éloi Menaud, Yann-Gaël Guéhéneuc, Mourad Badri, et Stefan Monnier, "A mapping study of language features improving object-oriented design patterns," *Information and Software Technology*, 2023.

[28] Ana Eva Chacón-Luna, Antonio Manuel Gutiérrez, José A. Galindo, et David Benavides, "Empirical software product line engineering: A systematic literature review," *Information and Software Technology*, 2020.

[29] Irdin Pekaric, Raffaela Groner, Thomas Witte, Jubril Gbolahan Adigun, Alexander Raschke, Michael Felderer, et Matthias Tichy, "A systematic review on security and safety of self-adaptive systems," *Journal of Systems and Software*, 2023.

[30] Tomasz Lewowski, et Lech Madeyski, "How far are we from reproducible research on code smell detection? A systematic literature review," *Information and Software Technology*, 2022.

[31] Chakraborty, Shalini, et Grischa Liebel. "Modelling guidance in software engineering: a systematic literature review," *Software and Systems Modeling*, 2024.

[32] David Mosquera, Marcela Ruiz, Oscar Pastor, et Jürgen Spielberger, "Understanding the landscape of software modelling assistants for MDSE tools: A systematic mapping," *Information and Software Technology*, 2024.

[33] Roger Nazir, Alessio Bucaioni, et Patrizio Pelliccione, "Architecting ML-enabled systems: Challenges, best practices, and design decisions," *Journal of Systems and Software*, 2024.

[34] Delano Oliveira, Reydne Santos, Fernanda Madeiral, Hidehiko Masuhara, et Fernando Castor, "A systematic literature review on the impact of formatting elements on code legibility", *Journal of Systems and Software*, 2023.

[35] Morteza Zakeri-Nasrabadi, Saeed Parsa, Mohammad Ramezani, Chanchal Roy, et Masoud Ekhtiarzadeh, "A systematic literature review on source code similarity measurement and clone detection: Techniques, applications, and challenges," *Journal of Systems and Software*, 2023.