Analyse des des publications des chercheurs UPEC et enrichissement des données

MENIER Thomas
M2 Informatique - Logiciels Sûrs
UPEC
Créteil, France
thomas.menier@etu.u-pec.fr

MOUNTHANYVONG Florian
M2 Informatique - Logiciels Sûrs
UPEC
Créteil, France
florian.mounthanyvong@etu.u-pec.fr

LEUNG Marc
M2 Informatique - Logiciels Sûrs
UPEC
Créteil, France
marc.leung@etu.u-pec.fr

***Abstract*—Ce rapport présente le développement d'un projet visant à classifier les publications scientifiques de la plateforme HAL selon les catégories ERC (European Research Council). Initialement conçu pour enrichir les métadonnées des publications en analysant les champs " références " et " mots sujet " à l'aide de techniques de big data, le projet a été recentré sur la classification ERC en raison de limitations liées à la qualité des données disponibles.**

Le projet actuel repose sur l'utilisation d'un jeu de données structuré, extrait de HAL, contenant des informations telles que les titres, résumés, domaines de recherche, et dates de publication. Après une phase de préparation, comprenant la normalisation et le nettoyage des données, un modèle d'apprentissage supervisé a été développé. Ce modèle exploite des algorithmes avancés, incluant Random Forest et des modèles basés sur Transformers, pour effectuer une classification précise selon les catégories ERC.

Les résultats de ce travail ont également été intégrés dans une application web. Celle-ci permet une recherche intuitive par langage naturel, un filtrage avancé et l'exportation des résultats. Des tests de performance ont démontré l'efficacité de la solution proposée, avec une scalabilité adaptée à des bases de données de grande taille et des temps de réponse optimisés grâce à des techniques de parallélisation.

Ce projet s'inscrit dans une démarche d'innovation en open science, en améliorant l'accessibilité et la valorisation des données scientifiques via une solution technologique robuste et performante.

***Index Terms*—Classification ERC, HAL, open science, big data, apprentissage supervisé, normalisation des données, application web, API HAL, gestion des données scientifiques.**

I. Introduction

A. Définition du projet

Le projet initial avait pour objectif d'enrichir les données des publications scientifiques disponibles sur la plateforme HAL en utilisant des techniques de big data. Plus précisément, il s'agissait d'analyser les champs "références" et "mots sujet " afin de construire un modèle robuste capable de classifier les publications selon des catégories thématiques définies par la classification ERC (European Research Council). Cette démarche incluait également la complétion et la correction des données manquantes.

B. Changement du projet

Cependant, au cours des premières étapes du projet, plusieurs problèmes ont été rencontrés avec les jeux de données utilisés. Ces derniers étaient insuffisamment documentés et nécessitaient des corrections manuelles substantielles. Ces limitations ont significativement réduit la faisabilité de l'analyse des champs " mots sujet " et " références ". Pour répondre à ces défis, le projet a été recentré exclusivement sur la classification ERC, permettant ainsi de simplifier l'approche et d'éviter la surcharge de travail liée à l'enrichissement manuel des données.

C. Projet actuel

Le projet, dans sa forme actuelle, vise à :

- Utiliser un jeu de données existant tiré de HAL pour classifier les publications selon les catégories ERC ;
- Développer un modèle d'apprentissage supervisé basé sur des informations clés issues du jeu de données, telles que le titre des publications, le domaine de recherche et les résumés ;
- Garantir que le modèle soit capable de traiter efficacement les problèmes spécifiques aux données scientifiques, tels que la variabilité des termes utilisés.

II. Jeu de données

A. Présentation de HAL

HAL est une plateforme d'archivage ouverte conçue pour permettre aux chercheurs de déposer leurs publications scientifiques. Elle offre un accès à une vaste collection de métadonnées, comprenant notamment les titres des publications, les résumés, les domaines de recherche, et les affiliations des auteurs.

B. Description du jeu de données

Le jeu de données utilisé dans ce projet est issu de la plateforme HAL et se présente sous la forme d'un fichier CSV structuré avec les colonnes suivantes :

- **Nom du chercheur** : Nom de famille de l'auteur de la publication.
- **Prénom du chercheur** : Prénom de l'auteur.
- **Nom complet** : Concatenation du prénom et du nom.
- **Nom complet abrégé** : Contraction du prénom et du nom (forme abrégée).
- **Doc ID** : Identifiant unique de la publication.

- **Titre de la publication** : Titre complet de la publication scientifique.
- **Domaine de recherche** : Catégorie HAL associée à la publication.
- **Résumé** : Brève description du contenu de la publication.
- **Date** : Date de publication.

C. Analyse du jeu de données

Une inspection initiale des données a mis en évidence plusieurs limitations :

- Des champs incomplets ou mal formatés, notamment des noms abrégés inconsistants.
- Une grande variété de styles dans les titres et les résumés, reflétant la diversité des disciplines représentées.
- Des erreurs mineures, telles que des dates manquantes ou des résumés très courts ou absents.

Ces observations ont été prises en compte pour assurer une préparation adéquate des données avant l'entraînement du modèle.

D. Complétion du jeu de données

Pour améliorer la qualité et la cohérence des données, plusieurs actions ont été menées :

- **Normalisation des noms** : Conversion uniforme en majuscules pour les noms et prénoms.
- **Correction des dates** : Recherche manuelle pour combler les valeurs manquantes.
- **Nettoyage des champs de texte** : Suppression des caractères inutiles et correction orthographique basique des titres et résumés.
- **Filtrage des doublons** : Élimination des enregistrements dupliqués.
- **Ajout de domaines manquants** : Complétion des catégories ERC pour certaines publications, basée sur des informations existantes.

Ces étapes ont permis de constituer un jeu de données prêt à être utilisé pour l'entraînement d'un modèle de classification performant et robuste.

III. Étapes de développement

A. Description des travaux réalisés/à rendre

Le développement du projet s'articule autour de deux livrables principaux :

- **Application web** : Une plateforme interactive permettant aux utilisateurs de rechercher et de classifier les publications scientifiques.
- **IA de classification** : Un modèle d'intelligence artificielle conçu pour mapper les catégories HAL aux panels ERC correspondants.

B. Description du rendu

1) Application web:

L'application web offre les fonctionnalités suivantes :

- **Recherche par langage naturel** : Les utilisateurs peuvent formuler des requêtes textuelles afin d'obtenir des résultats pertinents.
- **Filtrage avancé** : Possibilité de filtrer les résultats par domaine ERC, date ou auteur.
- **Visualisation des publications** : Une interface intuitive permet de consulter les publications ainsi que leurs métadonnées associées.

2) IA de classification:

Le modèle d'intelligence artificielle est basé sur un apprentissage supervisé. Il exploite les champs suivants du jeu de données : titre , domaine de recherche et résumé .

a) Exemple d'entraînement:

- **Donnée d'entrée** :
  - *Titre* : " Modèles théoriques pour les systèmes complexes ".
  - *Domaine* : Mathématiques appliquées.
  - *Résumé* : " Cette étude présente des approches mathématiques pour modéliser les systèmes complexes ".
- **Prédiction** : Classification ERC PE1 (Mathématiques).

C. Modèle IA pour la classification ERC

Dans le cadre de ce projet, un modèle de machine learning a été développé pour prédire les panels ERC associés à un domaine scientifique. L'objectif est d'associer de manière automatique chaque domaine à son ou ses panels ERC respectifs à partir des données disponibles dans le fichier `domaine.csv`, qui contient des associations entre tous les domaines HAL et des panels ERC. Le fichier domaine.csv a été formatté à l'aide d'une requête sur l'api HAL permettant de récupérer tous les domaines dans un fichier JSON, puis les 397 domaines ont été enrichi "à la main" avec leurs panel ERC associés. L'association est faite arbitrairement, en se référant aux descriptions des domaines dans HAL et dans le panel ERC.

Le premier modèle était un SVM sans pré-traitement des données, ce qui a posé des problèmes sur les prédictions. En effet, les domaines dans le fichier des publications sont formatés de la façon suivante [0.spi, 1.spi.nano], ou sont tout simplement non spécifiés et remplacés par la mention "Domaine non disponible". Le premier modèle n'utilisé pas non plus le titre et le résumé des publications pour faire ces prédictions.

Le second modèle pour lequel nous avons opté, réutilise les mêmes fichiers .csv, mais cette fois-ci, nous faisons un pré-traitement des domaines présents dans le fichier des publications. Nous combinons également des techniques de NLP avec la classification multi-labels.

Pour résoudre le problème des préfixes, on va simplement les supprimer et conserver ceux de "plus haut niveau", [0.spi, 1.spi.nano] devient donc [spi.nano]. Dans le cas où les domaines ne sont pas disponibles, on ignore simplement la publication. La différence avec le premier modèle se fait également aux niveaux du traitement des données d'entraînement, puisque on combine le domaine, titre et résumé de chaque publication en une seule colonne pour faciliter l'analyse. Encore une fois, il faut traiter les publications ayant des champs manquant, ici, en général, ce sont

les résumés. Cette fois-ci, on traite les champs manquant en ajoutant du “vide” (‘ ‘). On vectorise ensuite les données à l’aide de la méthode TF-IDF (Term Frequency-Inverse Document Frequency), c’est-à-dire les transformer en représentation numérique. On peut ensuite entraîner le modèle avec un classificateur multi-label, ici MultiOutputClassifier avec LogisticRegression, pour prédire les panels ERC. Chaque publication peut être associée à un ou plusieurs panels ERC. La suite est la prédiction des panels ERC, en fonction des caractéristiques textuelles. Les prédictions sont ensuite converties en noms de panels ERC à l’aide d’un MultiLabelBinazer. Le modèle produit ensuite un fichier CSV contenant les informations originales des publications, en ajoutant une nouvelle colonne contenant les panels ERC prédits.

Le deuxième modèle est plus performant, mais à tout de même des limites. Il dépend de la qualité des données d’entrée. Les panels ERC prédits sont basés sur les correspondances définies dans le fichier domaine.csv. Si des domaines HAL sont ajoutés, le modèle ne les prendra pas en compte. Il devra donc être ré-entraîné pour prendre en compte les nouveaux domaines. L’entraînement peut également être amélioré en faisant entrant comme données d’entraînements les publications avec leur panel ERC directement attribué. Les pistes d’amélioration sont multiples. On pourrait intégrer des modèles de langage plus avancés, comme BERT, pour mieux comprendre le contexte des titres et des résumés. Une autre piste a été commencée, et consiste à faire les associations des domaines, titres et résumés avec les panels ERC à la main, puis d’entraîner le modèle avec ces données. Le problème est que cette méthode, bien que le résultat sera plus satisfaisant, nécessite le traitement de chaque publication individuelle.

D. Fonctionnalités de l’application web

L’application web inclut les fonctionnalités suivantes :

- **Recherche par langage naturel** : Permettant aux utilisateurs d’interroger la base de données de manière intuitive.
- **Affichage des résultats de classification** : Présentation des publications avec les domaines de publications correspondants.
- **Exportation des résultats** : Possibilité d’exporter les résultats en format CSV pour une utilisation ultérieure.

IV. Tests et analyses

A. Extension de la base de données initiale

Suite à des considérations de taille et de représentativité, il a été décidé d’étendre la base de données initiale de HAL UPEC en incluant la globalité des publications disponibles sur HAL. Pour ce faire, la dernière version du backup RDF de HAL a été extraite à partir de l’archive `RDF_ARCHIVE_2024-11-01.ZIP` disponible sur https://data.hal.science/backup. Cette archive contient des données structurées au format RDF, qui ont été transformées et intégrées dans une base de données relationnelle pour faciliter leur exploitation.

B. Schéma de la base de données

Le schéma de la table principale `articles` a été conçu pour accueillir les métadonnées des publications scientifiques. Voici la structure de la table :

```
CREATE TABLE IF NOT EXISTS articles (
id INTEGER PRIMARY KEY AUTOINCREMENT,
rdf_about TEXT UNIQUE,
title TEXT,
abstract TEXT,
arxiv_id TEXT,
language TEXT,
volume TEXT,
page_start TEXT,
page_end TEXT,
publication_date TEXT,
created_date TEXT,
modified_date TEXT,
bibliographic_citation TEXT,
type TEXT,
subjects TEXT,
topic TEXT,
is_part_of TEXT,
contributors TEXT,
creators TEXT,
same_as TEXT
);
```

Ce schéma permet de stocker des informations détaillées sur chaque publication, y compris son titre, son résumé, ses auteurs, ses sujets, et ses dates de publication et de modification.

C. Extraction et échantillonnage des données

Afin de tester la performance du système et de valider les traitements, un échantillon de 434 453 publications a été extrait de l’ensemble des 4 014 620 publications disponibles dans HAL, représentant environ 10,8% de la base de données totale. Cet échantillon a été inséré dans une base de données SQLite pour permettre des tests de performances sur la taille de la base de données.

D. Comparatif de l’efficacité des requêtes et du nombre de coeurs utilisés

Pour évaluer les performances des requêtes, plusieurs paramètres ont été analysés. Les tests se sont concentrés sur trois aspects principaux : la taille de la base de données, le nombre de cœurs utilisés et les types d’opérations effectuées.

1) Taille de la base de données:
La taille de la base de données a été augmentée artificiellement en générant des copies des enregistrements existants avec des variations minimes. Cette approche a permis de simuler des bases de données de différentes tailles, allant de **10 000 à 400000 entrées**.

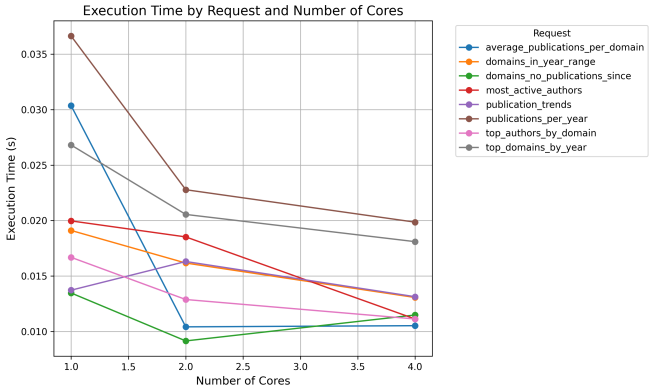

Fig. 1. Schéma représentant le temps d'execution en focntion du nombre de coeur et la complexité de la requête

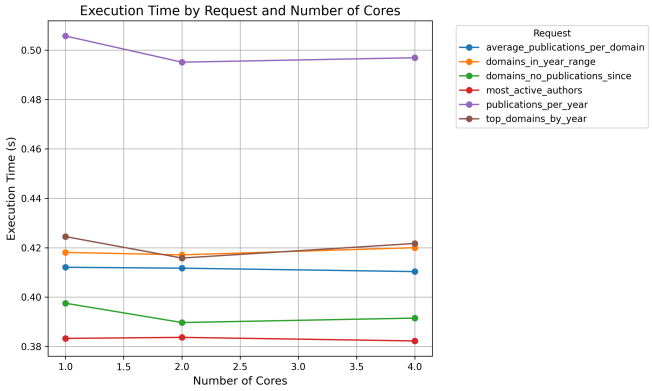

Fig. 2. Schéma représentant le temps d'execution en fonction du nombre de coeur et la complexité de la requête pour une base de données de 434 453 publications (10x plus grand)

2) Nombre de cœurs utilisés:
Des tests ont été effectués en utilisant un, deux et quatre cœurs, en exploitant des outils de parallélisation. Ces tests ont permis d'évaluer l'impact du parallélisme sur le temps d'exécution des requêtes.

3) Types d'opérations différentes:
Les performances ont été mesurées pour diverses opérations, notamment :

- **Recherche par mot-clé** : Identifier des publications contenant des termes spécifiques.
- **Filtrage par date** : Limiter les résultats à des plages temporelles définies.
- **Tri des résultats** : Ordonner les publications selon des critères tels que la date ou l'auteur.

E. Analyse des performances

- **Impact de la taille de la base de données** : L'augmentation de la taille de la base de données entraîne un impact linéaire sur le temps de réponse pour des recherches variées. Cela démontre que le système est scalable pour des bases de données de grande taille. En effet, on observe une augmentation de seulement de 0.50 ms pour une augmentation de la taille de la base de données d'un facteur 10.
- **Effet du parallélisme** : L'utilisation de plusieurs cœurs réduit significativement le temps d'exécution, en particulier pour des opérations complexes telles que le tri ou la recherche par mot-clé. Cependant le parallellisme a eu moins d'effet sur la base de données plus grandes.
- **Combinaison de critères multiples** : Les requêtes combinant plusieurs critères (par exemple, recherche par mot-clé et filtrage par date) présentent une latence plus élevée. Ces résultats mettent en évidence la nécessité d'optimiser les algorithmes sous-jacents afin de maintenir des performances acceptables.

V. Problèmes rencontrés

Nous avons rencontré de nombreux problèmes sur ce projet :

- Une personne n'a pas forcément d'identifiant HAL
- Une publication peut ne pas avoir de domaine ou de résumé
- Une publication peut avoir dans son titre des symboles non pris en charge par le format UTF-8
- Les métadonnées d'une publication ne contiennent pas l'intégralité des données
- Les métadonnées sont regroupées dans un seul champs ce qui le rend très compliqué à parser car il contient le titre et le résumé.

L'un des plus gros problèmes que nous avons rencontré est la variabilité des noms des chercheurs dans la base de données HAL. Un même auteur peut apparaître sous différentes formes. Ce manque d'uniformité entraîne des doublons et des erreurs d'identification, compliquant la recherche d'auteurs, la consolidation des publications et les statistiques scientifiques. Ce problème est un enjeu fondamental en gestion des données scientifiques et en traitement des identités en bases de données. Il est particulièrement aigu dans des systèmes comme HAL où les chercheurs peuvent être enregistrés sous différentes variantes de leur nom. Voici une analyse approfondie du problème, son impact, et les solutions existantes ou envisageables.

A. Pourquoi ce problème existe-t-il ?

L'ambiguïté des noms dans les bases de données scientifiques est due à plusieurs facteurs. Le principal est la variabilité naturelle des écritures des noms. Un même chercheur peut être enregistré sous différentes formes, soit par habitude personnelle, soit selon les conventions de l'institution ou du système d'enregistrement :

- **Ordre prénom-nom vs. nom-prénom** :
  - Jean Dupont (Prénom + Nom)
  - Dupont Jean (Nom + Prénom)
  - Dupont, J. (Format bibliographique anglo-saxon)
  - J. Dupont (Abréviation prénom)
- **Utilisation d'initiales** :
  - Jean Dupont → J. Dupont
  - Jean-Claude Martin → J.-C. Martin ou J.C. Martin

  - Marie-Hélène Robert → M.-H. Robert ou M.H. Robert
- **Noms composés incomplets ou tronqués** :
  - Jean-Baptiste Lambert → Jean Lambert ou J.-B. Lambert
  - Marie-Thérèse Lemoine → Marie Lemoine
  - Perte d'un élément dans les bases qui ne gèrent pas bien les prénoms multiples.
- **Changement de nom** :
  - Ajout du nom de jeune fille après le mariage
  - Par exemple, Marie Dupont peut devenir Marie Dupont-Martin
  - Inversion possible (selon les pays) Marie Dupont — Marie Martin-Dupont
- **Problèmes d'accents et de caractères spéciaux** :
  - José García → Jose Garcia (perte de l'accent)
  - François Lévesque → Francois Levesque (normalisation ASCII)

B. Impact sur la Recherche Scientifique

Ce problème a des conséquences importantes dans plusieurs domaines :

1) Mauvaise indexation des publications:
Un même chercheur peut être vu comme plusieurs individus différents dans les bases bibliographiques comme HAL. Cela fausse les statistiques de publications, le calcul des indices, et la reconnaissance des contributions d'un chercheur.

2) Problèmes dans les collaborations internationales:
Les chercheurs travaillant avec des institutions utilisant différentes conventions de nommage peuvent voir leur identité fragmentée. Par exemple, Dupont, Jean en France peut devenir Jean Dupont dans une publication anglo-saxonne, puis J. Dupont dans une citation ultérieure.

3) Doublons et erreurs dans les bases de données:
L'absence de normalisation entraîne la duplication des enregistrements. Un même individu peut apparaître plusieurs fois, compliquant les efforts de consolidation des données.

4) Difficulté dans la recherche automatique et l'intelligence artificielle:
Les systèmes d'indexation et d'extraction de données (comme les moteurs de recherche) peuvent rater des articles associés à un auteur en raison des variations de nom. Une recherche de Marie Dupont peut ne pas retrouver Marie Dupont-Martin ou M. Dupont.

C. Comment résoudre ce problème ?

1) Utilisation d'identifiants uniques (ORCID, ResearcherID):
ORCID (Open Researcher and Contributor ID) permet d'attribuer un identifiant unique à chaque chercheur, indépendamment des variations de son nom. Scopus et Web of Science utilisent également ResearcherID et Scopus Author ID pour résoudre ce problème. Une solution serait d'encourager les chercheu rs à lier leur ORCID à leur profil HAL pour centraliser toutes leurs publications sous un identifiant unique.

2) Algorithmes de rapprochement (Name Disambiguation):
On peut utiliser des algorithmes de similarité de noms pour regrouper différentes variantes d'un même auteur. Exemples d'algorithmes : Jaro-Winkler (similitude entre chaînes de caractères) Levenshtein Distance (nombre de modifications nécessaires pour passer d'un nom à un autre) Soundex (normalisation phonétique pour les noms qui sonnent similaires) TF-IDF + Cosine Similarity (comparaison de noms en corpus)

HAL pourrait intégrer ces méthodes pour proposer des regroupements automatiques de publications par auteur.

3) Normalisation des données à l'importation:
À chaque nouvelle publication dans HAL, on pourrait appliquer des règles automatiques :

- Convertir tous les noms en majuscules ou minuscules : Jean Dupont deviendrait JEAN DUPONT
- Supprimer les accents : José García serait Jose Garcia
- Créer plusieurs variantes automatiquement : Jean Dupont, J. Dupont et Dupont Jean

L'objectif est de lier ces variantes à un même auteur.

4) Ajout d'un mécanisme de validation par les chercheurs:
HAL pourrait proposer aux auteurs une interface pour fusionner leurs propres alias sous une seule identité. Un chercheur pourrait voir toutes les variantes de son nom et confirmer lesquelles lui appartiennent.

D. Conclusion

Le problème des variations de noms dans HAL est ancien et structurel, mais il peut être atténué par : L'utilisation d'ORCID et d'identifiants uniques L'intégration d'algorithmes de rapprochement de noms La normalisation automatique des données Une interface de validation pour les chercheurs En combinant ces méthodes, on pourrait considérablement améliorer la qualité des données et éviter la fragmentation des profils de chercheurs.

ACKNOWLEDGMENT

REFERENCES