UNIVERSITÉ SULTAN MOULAY SLIMANE

ÉCOLE NATIONALE DES SCIENCES APPLIQUÉES – KHOURIBGA

Master IMSD

Développement d'un Outil d'Exploration et Visualisation des Connaissances à partir des Publications Scientifiques

Réalisé par :
M.KHEMMI Omar
M.TAFAHI Soufiane
M.HAFIDI Yassine
M.JARMOUNI Abdellah

Encadré par :
M.HAFIDI Imad

Année Universitaire : 2024/2025

Table des matières

graphicx float

Table des figures

1 Remerciements

Nous tenons à exprimer notre profonde gratitude à **Monsieur HAFIDI Imad** pour son encadrement rigoureux, sa disponibilité constante et son engagement tout au long de ce projet. Son expertise dans le domaine de l'analyse des données scientifiques et son approche pédagogique claire ont été des atouts précieux pour la réussite de notre travail.

Nous le remercions sincèrement pour la confiance qu'il nous a accordée et pour l'opportunité qu'il nous a offerte de travailler sur un projet aussi formateur et en phase avec les enjeux actuels de la recherche scientifique.

Nos remerciements s'adressent également à l'ensemble du corps enseignant pour la qualité des enseignements dispensés durant ce semestre, ainsi qu'aux camarades pour leur collaboration, leur soutien et les échanges constructifs qui ont enrichi cette expérience.

Enfin, nous souhaitons remercier toutes les personnes ayant contribué, de près ou de loin, à l'aboutissement de ce projet.

2 Introduction et Contexte de projet

2.1 Introduction

Dans un monde où la recherche scientifique progresse à un rythme sans précédent, la quantité d'informations disponibles dans les bases de données académiques ne cesse d'augmenter. Cette explosion de données présente à la fois une opportunité et un défi : comment exploiter efficacement cette masse d'informations pour en extraire des connaissances pertinentes et les organiser de manière compréhensible et accessible ?

C'est dans ce contexte que s'inscrit notre projet, qui vise à concevoir un système générique d'exploration, d'analyse et de visualisation des publications scientifiques. À travers une solution basée sur l'intelligence artificielle et l'analyse des données, nous proposons un outil permettant d'identifier automatiquement les entités clés (auteurs, concepts, institutions) et de visualiser les relations scientifiques sous forme de graphes interactifs.

2.2 Contexte et problématique de projet

2.2.1 Contexte

La transformation numérique a fortement impacté le monde de la recherche scientifique. De nombreuses plateformes telles que **arXiv** ou **PubMed** ou **Semantic Scholar** offrent un accès libre à des millions de publications. Bien que cette accessibilité soit une avancée majeure, elle entraîne une nouvelle problématique : comment naviguer dans cette masse d'informations pour en extraire les éléments les plus pertinents pour un sujet donné ?

Dans ce cadre, les outils d'exploration sémantique, d'extraction automatique d'informations et de visualisation des données deviennent indispensables pour aider les chercheurs à identifier rapidement les tendances, les collaborations potentielles ou les lacunes dans un domaine donné.

2.2.2 Problématique

Face à ce volume croissant d'informations scientifiques, il devient crucial de développer un système qui ne se limite pas à la recherche textuelle traditionnelle, mais qui soit capable de :

- **Collecter automatiquement** les publications scientifiques depuis des sources fiables et variées .
- **Analyser et extraire** les entités nommées (chercheurs, institutions, concepts, méthodes, résultats) à partir du texte brut .
- **Mettre en évidence les relations** entre ces entités (citations, collaborations, thématiques connexes) .
- **Représenter visuellement** ces liens à travers une interface interactive pour faciliter l'exploration et l'analyse.

2.2.3 Objectifs de projet

L'objectif global du projet est de mettre en place une solution complète et fonctionnelle permettant :

- **La collecte intelligente** de publications scientifiques via des APIs publiques (arXiv, PubMed, Semantic Scholar) .
- **L'extraction sémantique** des entités clés grâce aux techniques de traitement automatique du langage naturel (NLP).
- **L'analyse des relations** entre chercheurs, institutions et concepts .
- **La structuration et le stockage** des données dans une base orientée graphes (Neo4j) .
- **La visualisation interactive** des résultats via un tableau de bord développé avec Streamlit ou Dash.

Le systeme proposera les fonctionnalites suivantes :

- **Recherche avancee de publications :** Possibilite de ltrer les articles par mots-cles, auteur ou institution.
- **Extraction et classi cation automatique des concepts cles :**Detection des thematiques principales dun corpus.
- **Recherche avancee de publicationsCartographie des relations scientiques :**Visualisation des collaborations entre chercheurs et des liens entre concepts.
- **Analyse des tendances :**Suivi de levolution des sujets de recherche dans le temps.

3 Définition du cahier de charges et Analyse des besoins

3.1 Introduction

Cette section vise à détailler les spécifications fonctionnelles et non fonctionnelles nécessaires à la mise en œuvre du projet. Elle identifie également les besoins auxquels le système doit répondre pour atteindre les objectifs fixés.

3.2 Spécifications non techniques

Le système répond aux besoins suivants :

- **Simplicité d'utilisation :**L'interface devra être intuitive, accessible même à des utilisateurs non spécialistes.
- **Modularité :** Le système doit être structuré de manière modulaire afin de permettre son évolution future.
- **Performance :**Le système doit traiter efficacement de larges volumes de données, avec une réponse rapide lors des requêtes.
- **Fiabilité :**Les données extraites et visualisées doivent être cohérentes, justes et facilement traçables.
- **Portabilité :** Le système doit pouvoir être déployé facilement sur différentes plateformes.

3.3 Fonctionnalités techniques

Le système intégrera les modules suivants :

1. **Collecte des données :**
   - Utilisation des APIs publiques (arXiv, PubMed, Semantic Scholar) pour extraire les métadonnées et le texte intégral des articles.
   - Stockage initial dans une base SQL.
2. **Prétraitement des textes :**
   - Nettoyage des textes (suppression du bruit, des stop words).
   - Tokenization
   - lemmatisation
   - segmentation des phrases.
3. **Extraction des entités et des relations :**
   - Utilisation de modèles de reconnaissance d'entités nommées (NER).
   - Détection automatique des collaborations entre auteurs, des liens entre concepts et des résultats expérimentaux.
4. **Structuration des données :**
   - Insertion des entités et relations dans une base de données orientée graphes (Neo4j) pour faciliter la navigation sémantique.
5. **Visualisation et exploration :**
   - Conception d'un tableau de bord web permettant :
     - Une recherche avancée (par auteur, concept, mot-clé) .
     - Une cartographie dynamique des relations scientifiques .
     - Une analyse des tendances dans le temps.
     - Une fonctionnalité de recherche par cluster a .

4 Conception du système

4.1 Architecture générale du système

L'architecture du système repose sur un pipeline de traitement des articles scientifiques, entièrement automatisé et modulaire. Il est structuré selon cinq étapes principales :

1. **Collecte des articles scientifiques** à partir de arXiv.
2. **Prétraitement linguistique** des textes pour la normalisation.
3. **Extraction d'entités et de relations** pertinentes entre chercheurs, concepts et institutions.
4. **Stockage structuré** dans des bases de données relationnelles (PostgreSQL) et orientées graphe (Neo4j).
5. **Visualisation interactive** via un tableau de bord pour l'exploration des relations extraites.

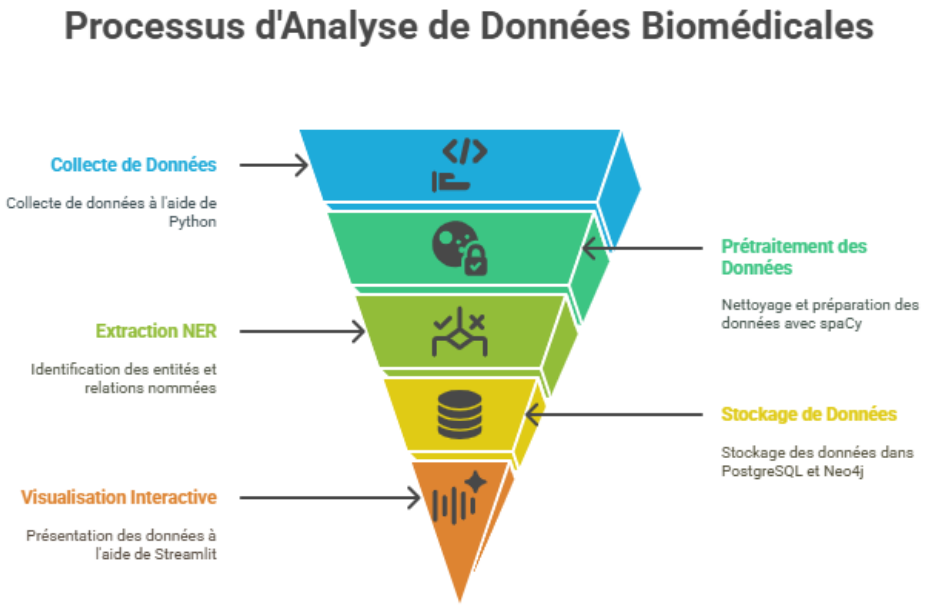

FIGURE 1 – Schéma global de l'architecture du système proposé

Comme illustré dans la figure 7, le pipeline est composé de plusieurs modules...

4.2 Workflow détaillé du projet

4.2.1 Collecte des données

- **Source :**API de **arXiv**.
- **Méthode :**Script Python utilisant Entrez (biopython) pour interroger arXiv et récupérer les métadonnées.
- **Volume total :**20 000 articles scientifiques collectés
- **Données extraites :**titre, résumé, auteurs, affiliations, date, mots-clés MeSH

4.2.2 Prétraitement NLP

- **outils :** spaCy
- **Étapes :**
  - **Nettoyage :** Suppression des caractères spéciaux, chiffres inutiles, espaces multiples, etc.
  - **Suppression des Stopwords :** Mots vides (ex : "the", "and", "of"...).
  - **Tokenisation :** Découpage du texte en mots.
  - **Lemmatisation :** Réduction des mots à leur forme de base.
  - **Segmentation des phrases :** Séparation en phrases .
- **But :** préparer les données pour l'extraction d'entités nommées

4.2.3 Extraction d'entités et de relations

- **Méthodes :**
  - Modèles de reconnaissance d'entités nommées (NER) via spaCy
  - Détection manuelle de patrons de texte pour relations simple
- **Entités extraites :** AUTEUR, INSTITUTION, CONCEPT, RÉSULTAT
- **Relations identifiées :** COLLABORE-AVEC, APPARTIENT-À, UTILISE-CONCEPT

4.2.4 Structuration et stockage

- **Base relationnelle : PostgreSQL**→ stockage des métadonnées brutes (titres, résumés, auteurs, mots-clés)
- **Base graphes : Neo4j** → stockage des entités extraites et relations scientifiques
- **Avantage :**
  - PostgreSQL : requêtes complexes sur les attributs
  - Neo4j : visualisation et navigation intuitive des liens entre entités

4.2.5 Visualisation interactive

- **Outil utilisé :** Streamlit
- **Fonctionnalités :**
  - **Recherche avancee de publications**
  - **Extraction et classi cation automatique des concepts cles**
  - **Cartographie des relations scienti ques**
  - **Analyse des tendances**

4.3 Diagrammes UML

Dans cette section, nous présentons les différents diagrammes UML modélisant les interactions entre l'utilisateur et le système, ainsi que les flux de traitement associés aux principales fonctionnalités.

4.3.1 Diagramme de cas d'utilisation

Ce diagramme offre une vue d'ensemble des fonctionnalités principales accessibles à l'utilisateur final. Il illustre comment celui-ci peut interagir avec le système pour :

- rechercher des publications,
- cartographier les relations scientifiques,
- extraire et classifier les concepts clés,
- analyser les tendances.

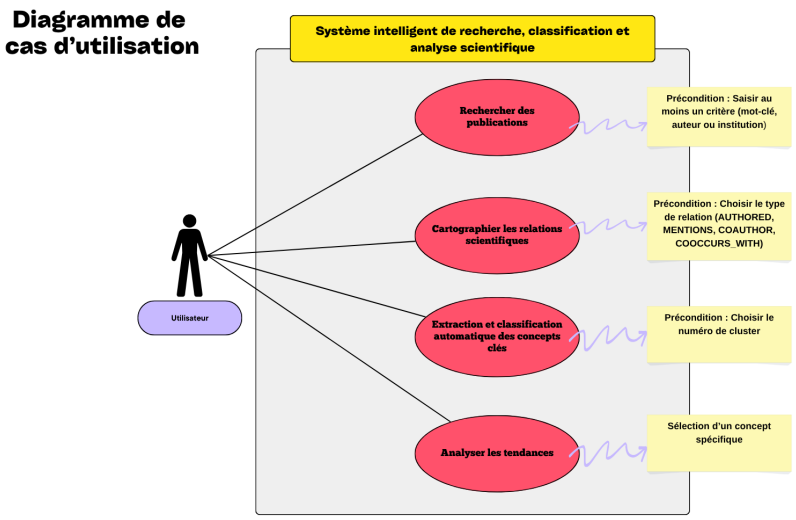

FIGURE 2 – Diagramme de cas d'utilisation du système

4.3.2 Diagrammes de séquence

Ces diagrammes décrivent en détail les échanges entre les composants du système (interface, contrôleur, base de données) lors de l'exécution de chaque fonctionnalité.

1. Rechercher des publications Ce diagramme montre le processus par lequel l'utilisateur saisit ses critères de recherche, et obtient les publications correspondantes.

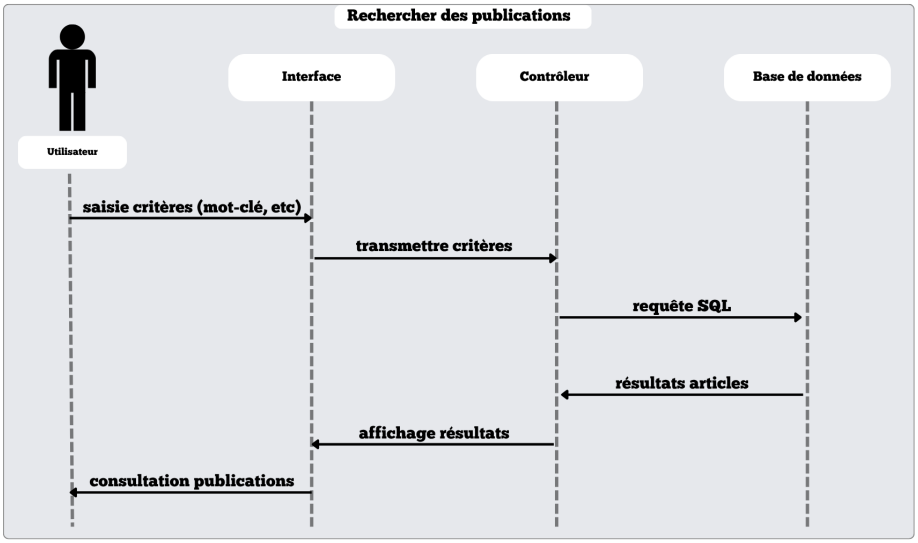

FIGURE 3 – Diagramme de séquence – Rechercher des publications

2. Extraction et classification des concepts clés Ce diagramme décrit comment les concepts sont extraits à partir d'un cluster spécifique, puis affichés à l'utilisateur.

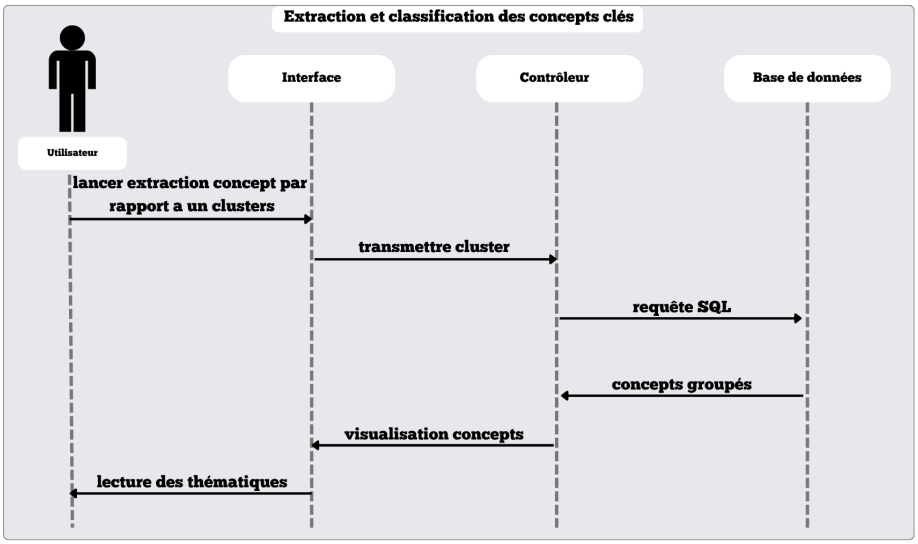

FIGURE 4 – Diagramme de séquence – Extraction des concepts clés

3. Cartographie des relations scientifiques Ce diagramme illustre l'interaction avec Neo4j pour visualiser un graphe relationnel interactif.

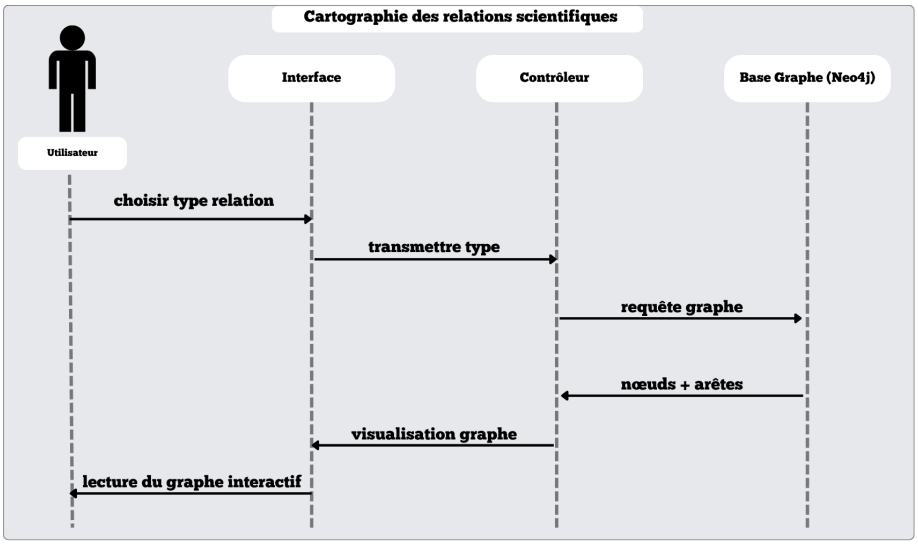

FIGURE 5 – Diagramme de séquence – Cartographie des relations scientifiques

4. Analyser les tendances Ce diagramme détaille l'analyse d'un concept spécifique et l'affichage de visualisations correspondantes.

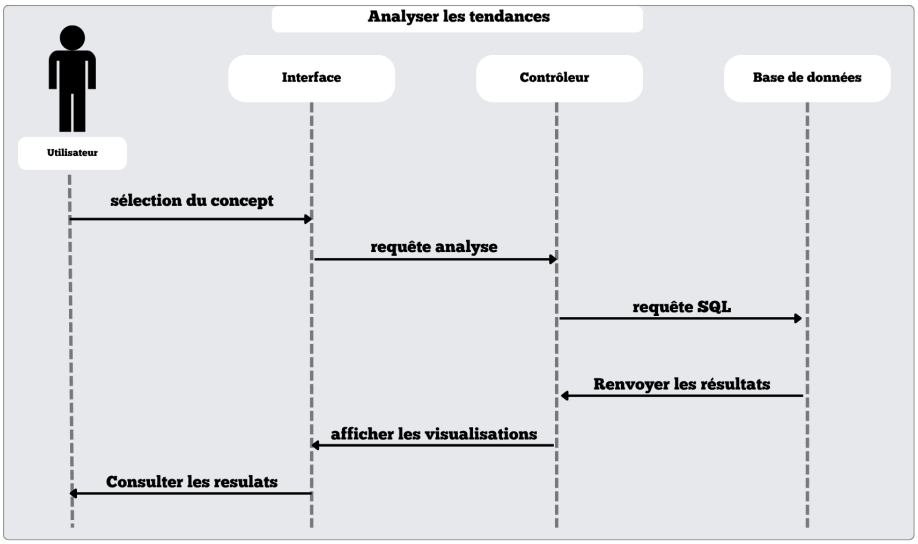

FIGURE 6 – Diagramme de séquence – Analyse des tendances

4.4 Choix des technologies

Module	**Technologie**	**Justification**
Collecte de données	Python + Entrez API (arXiv)	Accès structuré à une large base de publications biomédicales
Prétraitement NLP	spaCy,en$_{c}ore_{s}ci_{s}m$	Outils performants pour le nettoyage, la tokenisation et la lemmatisation du texte scientifique
Stockage relationnel	PostgreSQL	Base relationnelle robuste pour stocker les métadonnées des publications
Stockage graphe	Neo4j	Base orientée graphe idéale pour représenter les relations entre auteurs, concepts et institutions
Visualisation interactive	Streamlit	Framework léger et interactif pour le développement d'un tableau de bord de visualisation
Langage principal	Python	Large écosystème pour le traitement de texte, l'analyse de données et l'intégration de bases

TABLE 1 – Technologies utilisées et leurs justifications

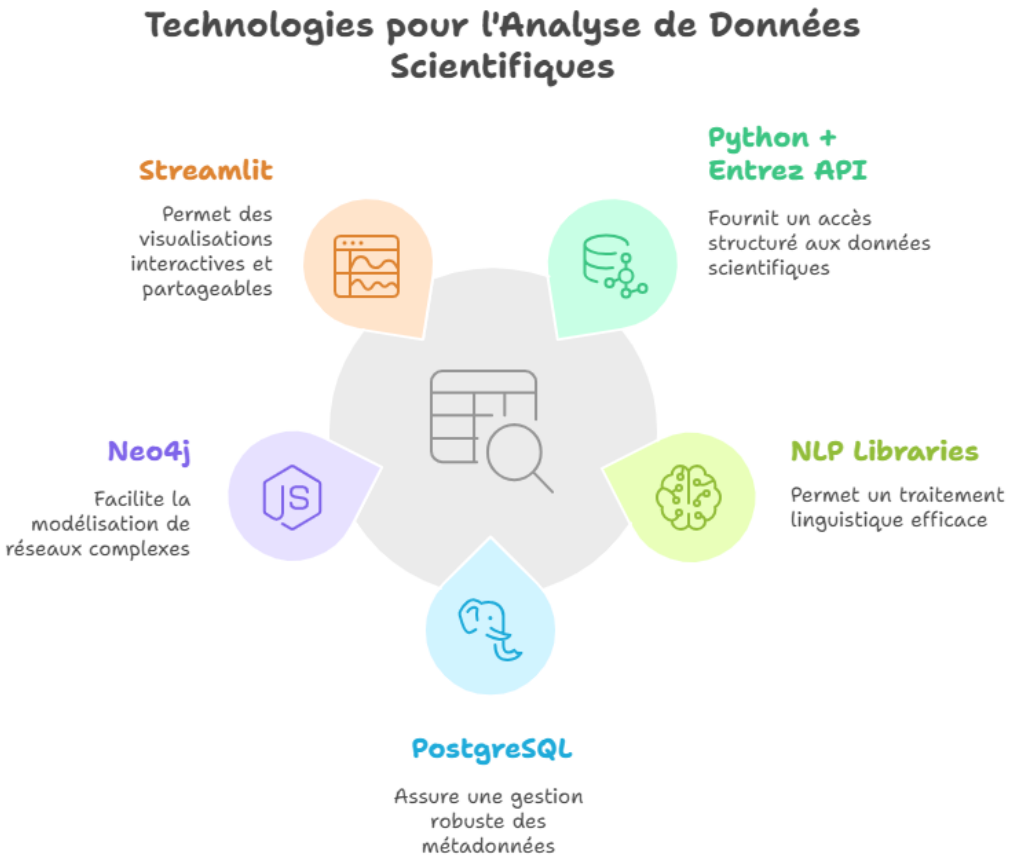

FIGURE 7 – Schéma sur les technologies utilise

5 Implémentation du système

5.1 Présentation générale

L'implémentation du système a été réalisée en suivant le pipeline décrit dans le cahier des charges : extraction, prétraitement, structuration, et visualisation des articles scientifiques. Cette section détaille les aspects techniques liés à la mise en œuvre de chaque composant du système à partir de la source `arXiv`, en se focalisant sur le domaine de l'informatique.

5.2 Extraction des articles depuis arXiv

5.2.1 Objectif

Extraire 20 000 articles scientifiques provenant exclusivement d'arXiv, dans différentes catégories du domaine informatique : `cs.AI`, `cs.LG`, `cs.CR`, `cs.NI`, `cs.CV`, `cs.SE`, `cs.DS`, `cs.RO`.

Code arXiv	Domaine
cs.AI	Intelligence Artificielle
cs.LG	Machine Learning
cs.CR	Cryptographie / Sécurité
cs.NI	Réseaux Informatiques
cs.CV	Vision par Ordinateur
cs.SE	Génie Logiciel
cs.DS	Structures de Données
cs.RO	Robotique

FIGURE 8 – Exemples de Catégories arXiv (Informatique)

Stratégie d'extraction

1. On va faire des **requêtes par catégorie.**
2. Pour chaque catégorie, on extrait par exemple **3000 à 2-20000 articles.**
3. On regroupe le tout pour atteindre **20 000 articles uniques.**

5.3 Technologies utilisées

- **Langage** : Python
- **API** : arXiv (via la bibliothèque `requests`)
- **Base de données** : PostgreSQL (`db_articles`)

5.4 Structure des données

- Table `articles` : *external_id, title, abstract, publication_date, url* (environ 20 300 entrées)

	id [PK] integer	source character varying (10)	external_id character varying (255)	title text
1	1	arXiv	cs/9308101v1	Dynamic Backtracking
2	2	arXiv	cs/9308102v1	A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems
3	3	arXiv	cs/9309101v1	An Empirical Analysis of Search in GSAT
4	4	arXiv	cs/9311101v1	The Difficulties of Learning Logic Programs with Cut
5	5	arXiv	cs/9311102v1	Software Agents: Completing Patterns and Constructing User Interfaces
6	6	arXiv	cs/9312101v1	Decidable Reasoning in Terminological Knowledge Representation Systems
7	7	arXiv	cs/9401101v1	Teleo-Reactive Programs for Agent Control
8	8	arXiv	cs/9402101v1	Learning the Past Tense of English Verbs: The Symbolic Pattern Associator vs. Connectionist Models
9	9	arXiv	cs/9402102v1	Substructure Discovery Using Minimum Description Length and Background Knowledge
10	10	arXiv	cs/9402103v1	Bias-Driven Revision of Logical Domain Theories
11	11	arXiv	cs/9403101v1	Exploring the Decision Forest: An Empirical Investigation of Occam's Razor in Decision Tree Induction
12	12	arXiv	cs/9406101v1	A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic
13	13	arXiv	cs/9406102v1	Applying GSAT to Non-Clausal Formulas
14	14	arXiv	cs/9408101v1	Random Worlds and Maximum Entropy

FIGURE 9 – table - articles

- Table `articles-authors` : contient 63284 relations.

	article_id [PK] integer	author_id [PK] integer
1	1	1
2	2	2
3	3	3
4	3	4
5	4	5
6	4	6
7	4	7
8	5	8
9	5	9
10	6	10
11	6	11
12	6	12
13	7	13
14	8	14

FIGURE 10 – table articles-authors

- Table `authors` : table authors : 37903 auteurs

	id [PK] integer	name character varying (255)
1	1	M. L. Ginsberg
2	2	M. P. Wellman
3	3	I. P. Gent
4	4	T. Walsh
5	5	F. Bergadano
6	6	D. Gunetti
7	7	U. Trinchero
8	8	J. C. Schlimmer
9	9	L. A. Hermens
10	10	M. Buchheit
11	11	F. M. Donini
12	12	A. Schaerf
13	13	N. Nilsson
14	14	C. X. Ling

FIGURE 11 – table authors

5.5 Prétraitement des textes

Étapes réalisées

Étapes du Prétraitement

1. **Nettoyage** : Suppression des caractères spéciaux, des chiffres inutiles, des espaces multiples, etc.
2. **Suppression des Stopwords** : Élimination des mots vides (ex. : « the », « and », « of »...).
3. **Tokenisation** : Découpage du texte en unités lexicales (mots).
4. **Lemmatisation** : Réduction des mots à leur forme de base (ex. : « running » devient « run »).
5. **Segmentation des phrases** : Découpage du texte en phrases complètes.

Après avoir appliqué les étapes de prétraitement décrites ci-dessus, nous avons obtenu une table contenant les phrases nettoyées et segmentées issues des résumés des articles scientifiques. Cette table est présentée ci-dessous :

	id [PK] integer	article_id integer	sentence_number integer	clean_sentence text
1	1	1	1	occasional need return shallow point search tree exist backtrack method erase meaningful progress solve search problem
2	2	1	2	paper present method backtrack point move deep search space avoid difficulty
3	3	1	3	technique develop variant backtracking use polynomial space provide useful control information retain completeness guarantee provide early approach
4	4	2	1	market price system constitute class mechanism certain condition provide effective decentralization decision make minimal communication overhead
5	5	2	2	programming approach distribute problem solving derive activity resource allocation set computational agent compute competitive equilibrium artificial economy
6	6	2	3	walras provide basic construct define computational market structure protocol derive corresponding price equilibria
7	7	2	4	particular realization approach form multicommodity flow problem careful construction decision process accord economic principle lead efficient distribute resource allocati
8	8	3	1	describe extensive study search gsat approximation procedure propositional satisfiability
9	9	3	2	gsat perform greedy number satisfied clause truth assignment
10	10	3	3	experiment provide complete picture gsat search previous account
11	11	3	4	describe detail phase search rapid follow long plateau search
12	12	3	5	demonstrate apply randomly generate problem simple scaling problem size mean number satisfied clause mean branching rate

FIGURE 12 – table preprocessed-sentences 134 391 ligne

5.6 Technologie

- **Librairie** : SpaCy
- **Modèle** : `en_core_sci_sm` (SciSpacy)

Résultat : `preprocessed_sentences` contient environ 134 391 phrases.

5.7 Extraction d'informations (NER et Relations)

5.7.1 Entités extraites

- **Concepts** (technologies, méthodes, etc.)
- **Institutions** (mot-clé : `university`, `institute`, etc.)
- **Résultats** (présence de pourcentages ou mots-clés : `accuracy`, `achieve`...)
- (Les auteurs sont déjà récupérés via arXiv).

5.7.2 stratégie pour la Classification des Entités :

Cible	Règle d'identification
INSTITUTION	Contient : `university`, `institute`, `college`, `lab`...
RESULT	Contient des chiffres, le symbole %, ou des mots-clés comme : `achieve`, `accuracy`, `improve`...
CONCEPT	Tout le reste (par défaut)

TABLE 2 – Règles d'identification des entités nommées

Résultats après l'étape d'extraction d'informations (NER et relations)

Après l'application des techniques de reconnaissance d'entités nommées et d'extraction de relations, deux structures de données principales ont été produites :

1. **Stockage des entités nommées** : Les entités extraites ont été enregistrées dans la table `named_entities`, contenant un total de 486 803 entités.

	id [PK] integer	article_id integer	entity text	label character varying (50)
89	89	5	automatically construct	CONCEPT
90	90	5	machine learning	CONCEPT
91	91	5	semantic	CONCEPT
92	92	5	user information	CONCEPT
93	93	5	completion string	CONCEPT
94	94	5	user interface	CONCEPT

FIGURE 13 – Extrait de la table `named_entities`

Enregistrement dans la table `named_entities` : 486 803 entités.

2. **Création des relations** : Les relations entre entités ont été établies et stockées dans la table `relations`, avec un total de 4 832 064 relations, incluant notamment des cooccurrences et des collaborations entre auteurs.

	id [PK] integer	article_id integer	entity1 text	entity2 text	relation_type character varying (50)
2999167	2999167	12763	distribution	stationary linear discrimin...	COOCCURRENCE
2999168	2999168	12763	distribution	time memory	COOCCURRENCE
2999169	2999169	12763	distribution	train revisit negative set	COOCCURRENCE
2999170	2999170	12763	estimation	identical	COOCCURRENCE

FIGURE 14 – Extrait de la table `relations` (cooccurrences)

5.8 Relations extraites

Objectif : Identifier les collaborations :

- **COAUTHOR** :Générer des relations COAUTHOR pour chaque article où il y a plusieurs auteurs.(Auteur A a collaboré avec Auteur B sur l'article X)
- **COOCCURRENCE** : concepts apparaissant dans un même abstract(Les Relations entre Entités)

Enregistrement dans la table `relations` : environ 4 832 064 relations.

Technologie

- Utilisation de **SciSpacy** pour la reconnaissance d'entités nommées (NER) dans un contexte scientifique.
- Détection automatique des entités scientifiques dans les résumés des articles.

5.9 Clustering thématique Automatique avec KMeans

5.9.1 Objectif

- **Regrouper les articles** selon leurs concepts clés détectés dans la table `named_entities`.
- Utiliser une approche NLP + Machine Learning :
  1. **Construire une représentation vectorielle** des articles à partir des concepts extraits, en utilisant la pondération `TF-IDF`.

2. **Appliquer l'algorithme KMeans** pour identifier automatiquement des groupes thématiques dans le corpus.
3. **Associer chaque article à un cluster** afin de caractériser son appartenance à une thématique spécifique.

Après l'application de l'algorithme de clustering thématique automatique (KMeans), les articles ont été répartis en six groupes thématiques distincts. La table `article_clusters`, contenant un total de 20 290 lignes, résume les résultats de cette classification. Elle associe chaque article à un cluster identifié.

	article_id [PK] integer	cluster integer
1	1	1
2	2	1
3	3	1
4	4	5
5	5	0
6	6	1
7	7	1
8	8	5
9	9	1
10	10	1

FIGURE 15 – Extrait de la table `named_entities`

5.10 Méthodologie

- Vecteurs : calcul TF-IDF à partir des concepts
- Algorithme : KMeans
- Résultat : 6 clusters enregistrés dans la table `article_clusters` (20 290 articles classés)

5.11 Visualisation avec Neo4j

5.11.1 Modélisation

Pour permettre :

- **Une visualisation graphique** des collaborations, concepts, et cooccurrences.
- Des analyses avancées via des requêtes **Cypher** (langage de Neo4j).

Types de nœuds :

- `Article` (id, titre, URL...)
- `Author` (name)
- `Concept` (name)
- `Result` (description)
- `Institution` (name)

Types de relations :

Relation	Source → Cible	Signification
`:AUTHORED`	Author → Article	A écrit l'article
`:MENTIONS`	Article →Concept/Result/Institution	Mentionne un concept ou résultat
`:COAUTHOR`	Author → Author	Collaboration entre auteurs
`:COOCCURS_WITH`	Concept/Result ↔ Concept/Result	Cooccurrence dans un abstract

5.12 Plan d'Action Technique

Connexion et insertion dans Neo4j

Les données traitées ont été intégrées dans la base de données graphe **Neo4j** en suivant les étapes suivantes :

1. **Connexion à Neo4j** via Python, à l'aide du `neo4j-driver`.
2. **Insertion des nœuds** : les entités suivantes ont été insérées en tant que nœuds du graphe :
   - Articles
   - Auteurs
   - Concepts
   - Résultats
   - Institutions
3. **Insertion des relations** : les relations ont été construites à partir des données extraites et des tables PostgreSQL suivantes :
   - `authors`
   - `article_authors`
   - `named_entities`
   - `relations`

Relation `AUTHORED`

La relation `AUTHORED` relie un nœud `Author` à un nœud `Article`, indiquant que l'auteur a rédigé l'article en question.

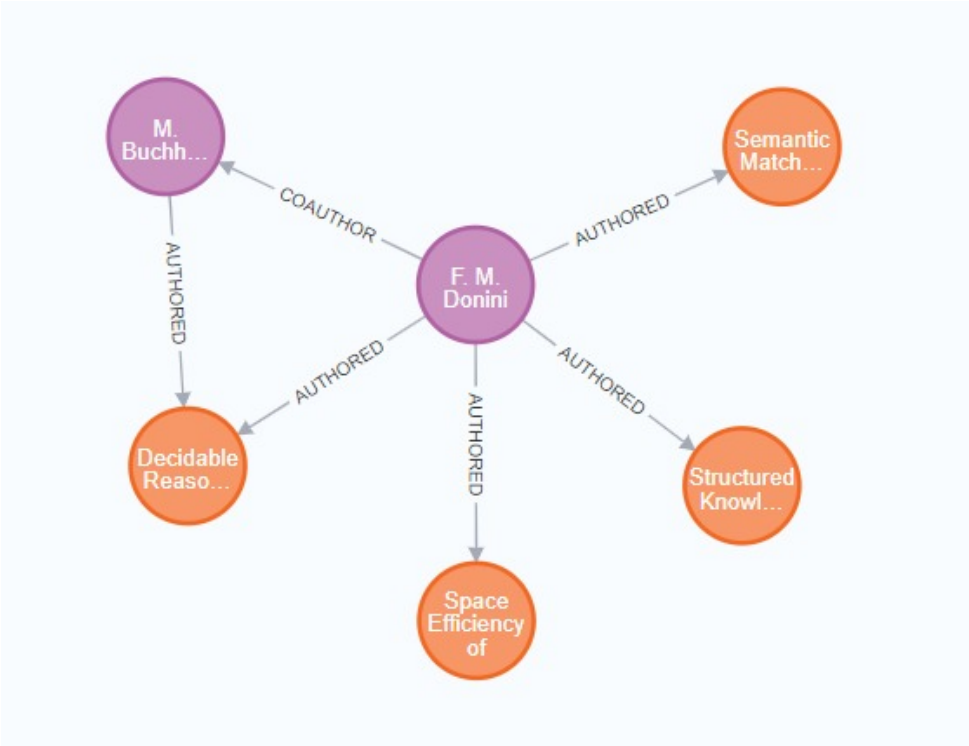

FIGURE 16 – Exemple de relation `AUTHORED` dans Neo4j

Relation `MENTIONS`

La relation `MENTIONS` relie un nœud `Article` à une entité (par exemple : `Concept`, `Result`, ou `Institution`). Elle indique que l'article mentionne cette entité dans son résumé. Ces relations sont dérivées du contenu de la table `named_entities`.

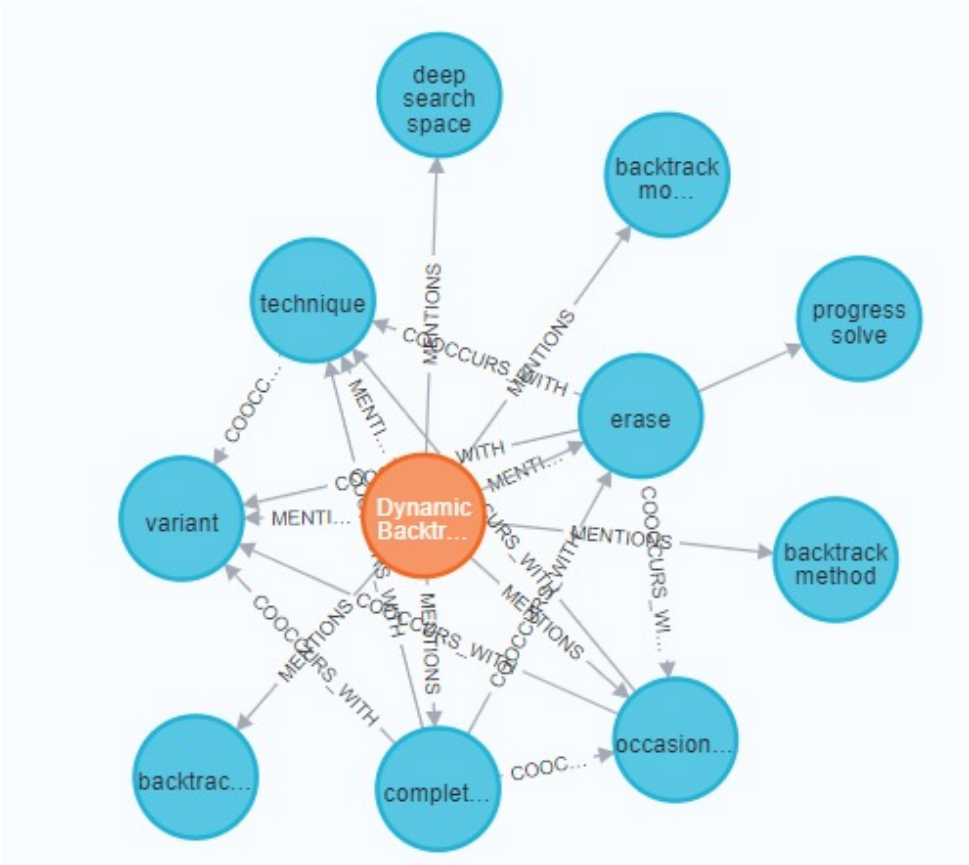

FIGURE 17 – Exemple de relation `MENTIONS` dans Neo4j

Relation `COAUTHOR`

La relation `COAUTHOR` connecte deux nœuds `Author` qui ont collaboré sur un même article. Elle permet de représenter les liens de collaboration entre chercheurs. Ces relations sont générées automatiquement à partir des co-auteurs d'un même article dans la table `article_authors`.

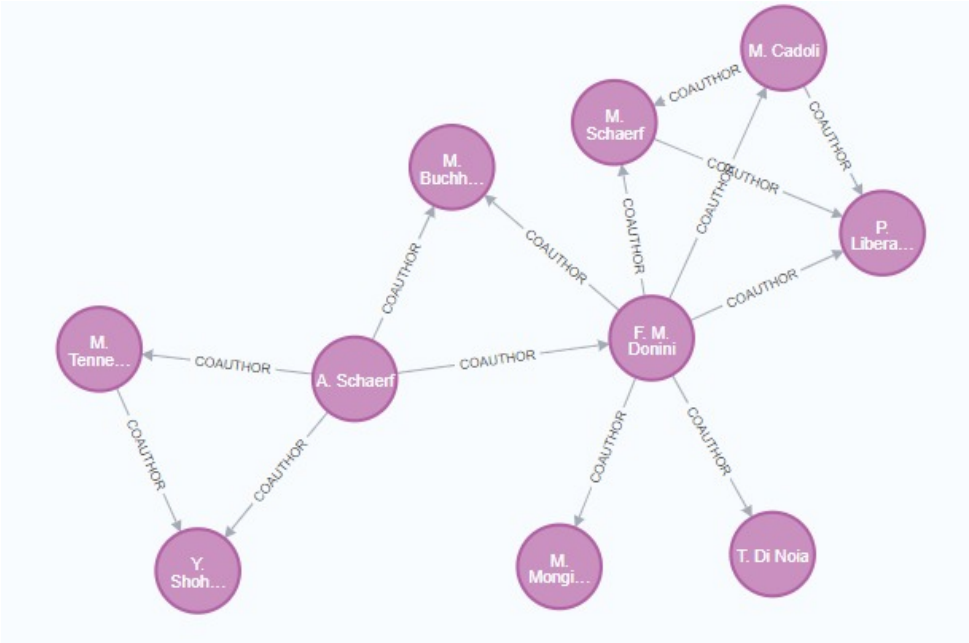

FIGURE 18 – Exemple de relation `MENTIONS` dans Neo4j

Relation `COOCCURS_WITH`

La relation `COOCCURS_WITH` connecte deux entités (par exemple deux concepts, ou un concept et un résultat) qui apparaissent ensemble dans le même résumé. Cette cooccurrence permet de visualiser les liens thématiques entre concepts scientifiques. Ces relations sont issues de la table `relations`.

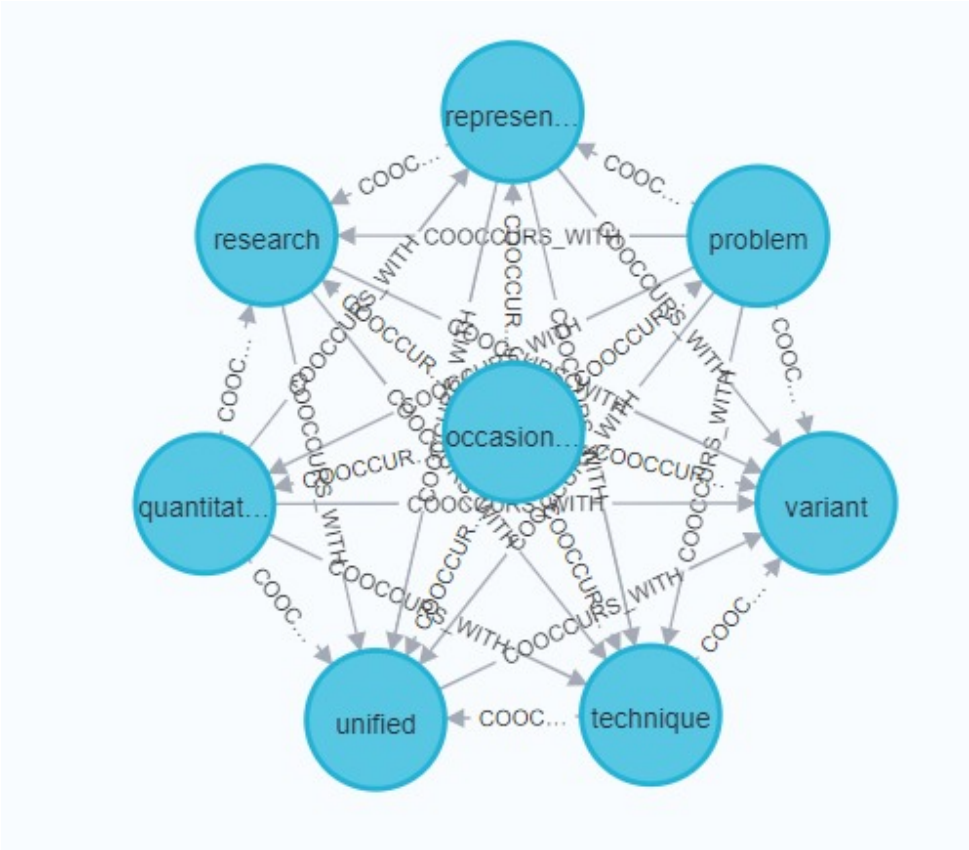

FIGURE 19 – Exemple de relation `MENTIONS` dans Neo4j

5.13 Méthodologie

5.13.1 Difficultés rencontrées

- Limites de requêtes de l'API arXiv
- Ambiguïté dans l'identification des entités
- Temps de traitement élevé (134 000 phrases)

5.13.2 Résumé technique

Étape	Données générées	Technologie
Extraction	20 300 articles	Python + API arXiv
Prétraitement	134 391 phrases	SpaCy / SciSpacy
NER	486 803 entités	SciSpacy + règles
Relations	4 832 064 relations	Python + pandas
Clustering	6 groupes	TF-IDF + KMeans
Visualisation	Graphe interactif	Neo4j + Cypher

6 Résultats et validation

Ce chapitre présente les résultats fonctionnels de l'application développée, illustrés par des captures d'écran de l'interface. L'objectif est de démontrer la capacité du système

à extraire, organiser et visualiser les connaissances à partir de publications scientifiques issues de arXiv.

—

6.1 Interface d'accueil et navigation

L'interface générale de l'application est minimaliste et intuitive. Elle offre une navigation par menu vertical, permettant d'accéder facilement aux quatre principales fonctionnalités : mot-cle, Nom d'auteur Institution, .

FIGURE 20 – Interface d'accueil et navigation dans le tableau de bord

6.2 Recherche avancée de publications

La première fonctionnalité de l'application est la recherche avancée de publications. L'utilisateur peut saisir un mot-clé, un nom d'auteur ou une institution pour filtrer dynamiquement les résultats.

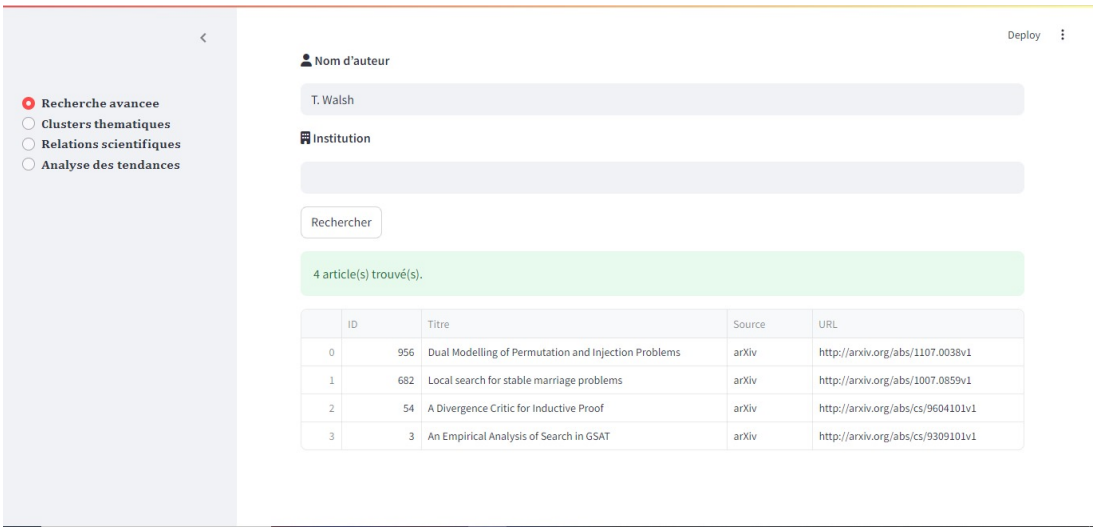

FIGURE 21 – Résultats de recherche avancée par auteur – liste d'articles récupérés depuis arXiv

6.3 Concepts fréquents par cluster

Pour chaque cluster, l'application affiche les concepts les plus représentatifs avec leur fréquence d'apparition. Cela permet d'interpréter la spécialisation sémantique de chaque groupe de publications.

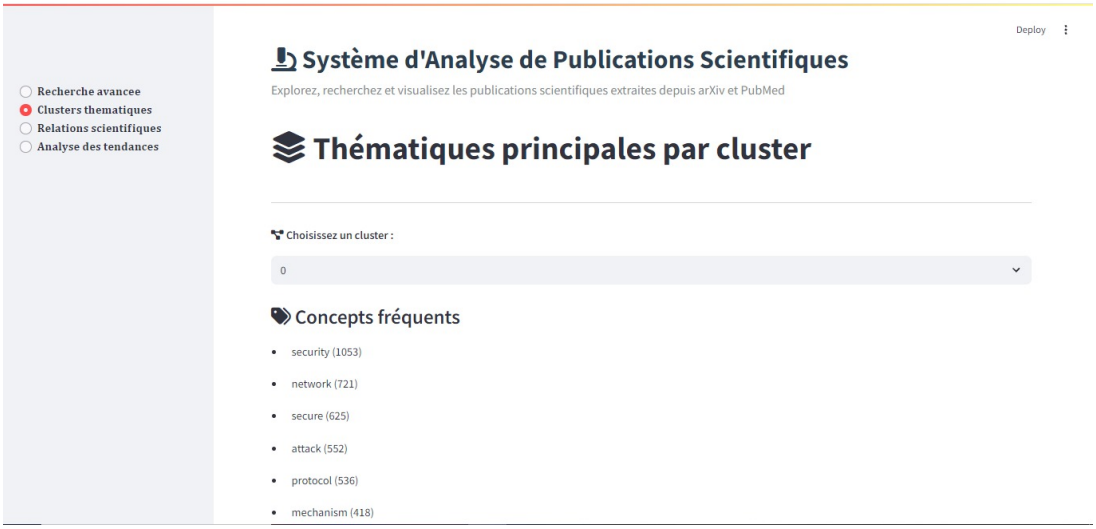

FIGURE 22 – Concepts dominants identifiés dans un cluster thématique

6.4 Nuage de mots des concepts

Cette section génère un nuage de mots à partir des concepts (mots-clés MeSH ou termes dominants) extraits des publications scientifiques. Plus un mot est fréquent, plus il est représenté en grand.

FIGURE 23 – Nuage de mots représentant les concepts dominants dans le corpus

6.5 Exploration des clusters thématiques

Le système regroupe les publications en clusters thématiques, chacun étant caractérisé par un ensemble de concepts fréquents. L'utilisateur peut visualiser les articles associés à un cluster donné.

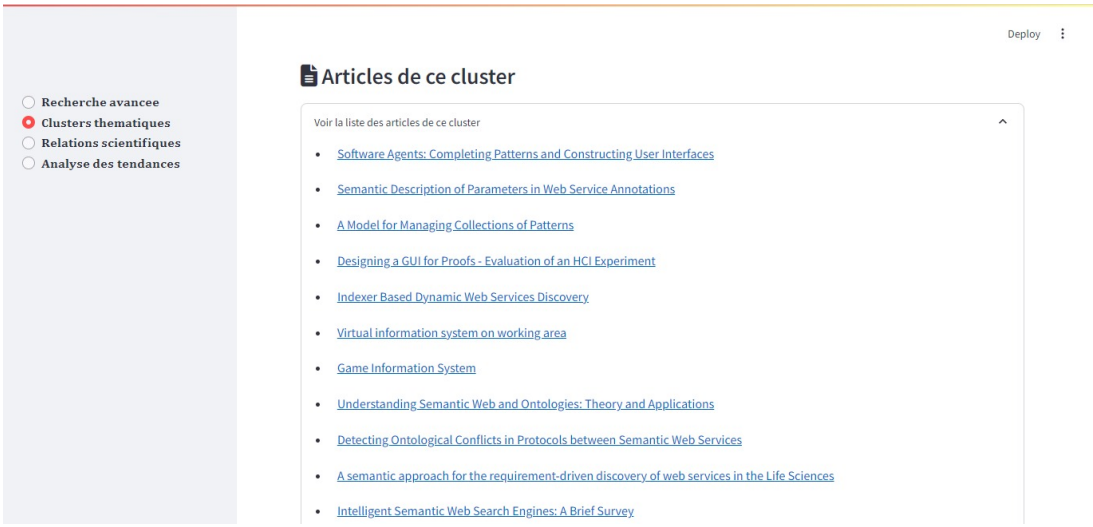

FIGURE 24 – Liste des articles appartenant à un cluster thématique sélectionné

6.6 Cartographie des relations scientifiques

L'utilisateur peut explorer visuellement les relations scientifiques identifiées dans les données : co-auteurs, mentions, collaborations, etc. Le graphe est généré à partir de la base Neo4j.

FIGURE 25 – Paramétrage de la visualisation des relations scientifiques (type, nombre de liens)

6.7 Visualisation des graphes relationnels

L'application affiche un graphe interactif où chaque nœud représente une entité (auteur, concept, etc.), et chaque lien, une relation (co-occurrence, mention, etc.).

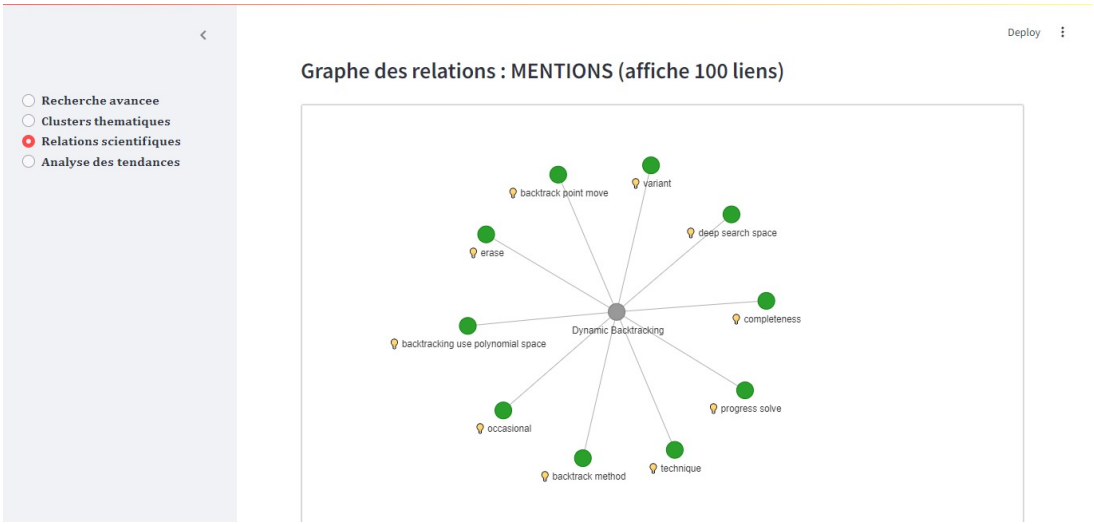

FIGURE 26 – Graphe représentant les relations scientifiques autour du concept sélectionné

6.8 Analyse des tendances dans le temps

Le système propose une visualisation des tendances de concepts dans le temps. Cela permet d'analyser l'évolution de l'intérêt scientifique pour un sujet donné (ex. : Deep Learning).

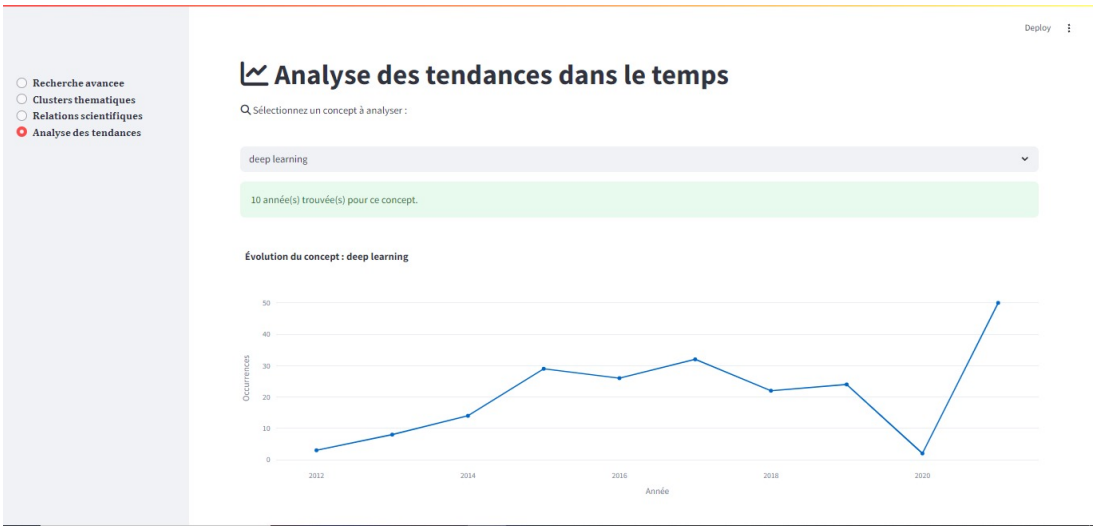

FIGURE 27 – Évolution temporelle de l'occurrence du concept « Deep Learning »

—

6.9 Synthèse des résultats

Les différentes visualisations montrent que le système développé permet de :

- filtrer efficacement les publications scientifiques ;
- identifier les thématiques dominantes dans un corpus ;
- explorer des graphes complexes d'interactions scientifiques ;
- suivre l'évolution des concepts dans le temps.

Ces résultats valident la pertinence du pipeline conçu et ouvrent des perspectives pour une exploitation plus large des publications biomédicales.

7 Difficultés rencontrées

Tout au long du projet, nous avons été confrontés à plusieurs défis techniques et fonctionnels. Ce chapitre présente les principales difficultés rencontrées, les solutions adoptées, ainsi que le moment précis du projet où ces ajustements ont été réalisés.

7.1 Difficultés liées à la collecte de données

Problème 1 : Limitations de l'API arXiv
Lors de la phase de collecte des 20 000 articles via l'API Entrez, nous avons constaté une limitation de fréquence imposée par arXiv, entraînant des blocages.

Quand ? Durant la première semaine de développement du script de collecte.
Solution : Nous avons implémenté une temporisation automatique entre les requêtes, couplée à un journal des ID téléchargés pour garantir la reprise après interruption.

Problème 2 : Structure irrégulière des résultats JSON
Certains enregistrements manquaient d'auteurs ou de résumés, ce qui posait problème pour le traitement.

Quand ? Dès les premiers tests de parsing JSON.
Solution : Ajout de conditions ‘try/except‘ et vérification de la complétude des champs avant insertion en base.

7.2 Difficultés liées au traitement linguistique

Problème 3 : Bruit lexical dans les textes
Les textes contenaient beaucoup de termes techniques inutiles ou de ponctuation erronée.
Quand ? Pendant les premiers tests de tokenisation.
Solution : Renforcement du nettoyage avec ‘re‘ (expressions régulières) et ajout de stop-words personnalisés.

Problème 4 : Faible précision des modèles NER génériques
Les modèles spaCy standard ne reconnaissaient pas bien les entités biomédicales.
Quand ? Lors des premiers essais d’extraction d’entités.
Solution : Enrichissement avec des listes MeSH et entraînement léger sur des extraits spécifiques.

7.3 Difficultés liées au stockage des données

Problème 5 : Choix entre PostgreSQL et Neo4j
Il n’était pas évident au départ de répartir correctement les données entre les deux bases.
Quand ? Lors de la modélisation de la base de données.
Solution : Nous avons défini une séparation claire :

- PostgreSQL pour les métadonnées (titre, résumé, etc.)
- Neo4j pour les entités et leurs relations

Problème 6 : Complexité des relations multiples (ex. co-auteurs)
Les graphes devenaient illisibles quand trop de relations étaient affichées.
Quand ? Durant la phase de test des graphes Neo4j.
Solution : Limitation du nombre de relations affichées + ajout de filtres par type de relation.

7.4 Difficultés liées à la visualisation

Problème 7 : Lenteur de Streamlit pour gros graphes
Les graphes lourds faisaient planter ou ralentir le tableau de bord.
Quand ? Lors de l’intégration de la visualisation interactive.
Solution : Ajout de contrôles : nombre de liens à afficher, profondeur, type de relation.

Problème 8 : Rendu inégal selon les navigateurs
L’affichage graphique ne s’adaptait pas bien sur tous les navigateurs.
Quand ? En phase de test utilisateur.
Solution : Utilisation de composants Streamlit standards + tests multi-navigateurs (Chrome, Firefox).

7.5 Conclusion

Ces difficultés, bien que nombreuses, nous ont permis d'améliorer la robustesse du système, de mieux comprendre les limites des outils utilisés, et d'affiner la qualité de l'expérience utilisateur. Elles constituent une réelle valeur ajoutée pédagogique et technique à notre projet.

8 Conclusion et perspectives

Conclusion générale

Ce projet avait pour objectif de développer une plateforme numérique permettant l'extraction, l'analyse et la visualisation de connaissances scientifiques à partir de publications issues de arXiv.

Grâce à l'intégration d'outils modernes en traitement automatique du langage naturel (spaCy), en bases de données relationnelles (PostgreSQL) et orientées graphe (Neo4j), ainsi qu'en visualisation interactive (Streamlit), nous avons pu concevoir un système complet, modulaire et fonctionnel.

Les principales réalisations incluent :

- La collecte automatisée de 20 000 articles scientifiques via l'API arXiv ;
- Le prétraitement linguistique du corpus et l'extraction des entités nommées ;
- La structuration des données sous forme de graphe sémantique ;
- La création d'une interface utilisateur simple et interactive ;
- La visualisation des collaborations scientifiques, des concepts dominants et de leur évolution dans le temps.

Ce travail a également permis de renforcer nos compétences techniques (Python, bases de données, visualisation) et analytiques (modélisation, structuration des connaissances, exploration sémantique).

Perspectives d'amélioration

Bien que le système soit opérationnel, plusieurs améliorations peuvent être envisagées pour enrichir la solution :

- **Extension des sources de données** : intégrer d'autres plateformes telles que PubMed, Semantic Scholar ou Springer pour élargir le corpus analysé.
- **Amélioration des modèles linguistiques** : utiliser des modèles pré-entraînés sur des corpus biomédicaux comme `BioBERT` pour améliorer la précision des entités extraites.
- **Personnalisation des analyses** : permettre à l'utilisateur de sauvegarder ses filtres, recherches ou visualisations sous forme de sessions.
- **Analyse plus avancée** : ajouter des modules de classification automatique, clustering thématique ou détection d'anomalies dans les tendances scientifiques.
- **Déploiement web sécurisé** : héberger le tableau de bord sur une plateforme en ligne avec une base de données distante, pour un accès public.

Conclusion personnelle

Ce projet a été l'occasion de mobiliser nos acquis en data science, base de données, extraction de texte et visualisation, tout en développant une solution concrète à fort potentiel d'exploitation. Il nous a également permis de mieux comprendre les enjeux liés à la structuration de la connaissance dans le domaine scientifique et les limites des outils existants.

Nous espérons que ce travail pourra servir de base à des projets futurs dans les domaines de la veille scientifique, de l'aide à la recherche ou de la visualisation intelligente de publications académiques.

9 Bibliographie

Références

[1] Cornell University Library. *arXiv.org e-Print archive*. [En ligne]. Disponible sur : `https://arxiv.org`

[2] ExplosionAI. *spaCy : Industrial-strength Natural Language Processing in Python*. [En ligne]. Disponible sur : `https://spacy.io`

[3] Neo4j. *Graph Database Platform*. [En ligne]. Disponible sur : `https://neo4j.com`

[4] Streamlit Inc. *Streamlit — The fastest way to build and share data apps*. [En ligne]. Disponible sur : `https://streamlit.io`

[5] Bird, S., Klein, E., Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[6] OpenAI. *ChatGPT : Assistant conversationnel basé sur l'intelligence artificielle*. [En ligne]. Disponible sur : `https://chat.openai.com`