UNIVERSITÉ SAAD DAHLEB DE BLIDA1

Faculté des sciences

Département d'informatique

MEMOIRE DE MASTER

En Informatique

Option : Ingénierie Des Logiciels

THÈME :

Découverte des liens sémantiques à partir des données liées en se basant sur les fonctions de croyance.

Réalisé par
TAHRAOUI Meriem
ZEGUENDRI Mohammed Alaeddine

Encadré par
Dr. I. RIALI
Dr. M. FAREH

Nom. Président : Mme.MANCER
Nom. Examinateur : Mr.HAMMOUDA

Juillet 2023

Remerciements

Ce travail est le fruit de la combinaison d'efforts , de patience et du courage durant toutes les années de notre parcours.

Nous remercions tout d'abord ALLAH le tout puissant qui, par sa grâce nous a permis d'arriver au bout de nos efforts en nous donnant la santé, la force, la volonté et en nous faisant entourer des merveilleuses personnes dont nous tenons à remercier.

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont contribué à la réalisation de ce projet de fin d'études.

Nous souhaitons tout d'abord remercier nos promoteurs : ***Mme.FAREH*** *et* ***Mr.RIALI*** *pour leur guidance, leur soutien et leurs précieux conseils tout au long de ce parcours.*

Nous tenons à remercier chaleureusement nos collègues et amis qui nous ont apporté leur soutien et leur encouragement tout au long de ce projet. Leurs encouragements et leur présence ont été une source d'inspiration et de motivation.

Nous sommes également reconnaissants envers nos familles pour leur soutien inconditionnel et leur encouragement constant tout au long de nos études. Leur amour et leur confiance ont été nos plus grandes motivations.

Merci à tous ceux qui ont contribué de près ou de loin à la réussite de ce projet. Votre soutien a été inestimable et nous vous en sommes profondément reconnaissants.

Résumé

La publication croissante de données liées sur le web présente un défi majeur en raison de leur hétérogénéité et de leur volume en constante augmentation. La découverte de liens entre les ressources web consiste à identifier les correspondances sémantiques entre des éléments similaires dans les données du web. Cependant, avec la quantité toujours croissante de données disponibles sur le web, il devient essentiel de disposer d'outils automatiques pour la découverte de ces liens. Néanmoins, l'identification automatique de correspondances sémantiques entre les données reste très difficile, notamment en ce qui concerne la qualité des liens extraits.

Pour contribuer à résoudre ce problème, nous proposons une solution pour effectuer la découverte des liens entre deux ensembles de données liées en utilisant la théorie de la fonction de croyance. Après l'extraction des différentes ressources des ensembles de données, nous lançons le processus de découverte des liens pour trouver les ressources équivalentes. Nous utilisons un mécanisme de filtrage pour regrouper les données en catégories, réduisant ainsi l'ensemble de recherche des données similaires. Ensuite, nous combinons les mesures terminologiques, extensionnelles et structurelles en utilisant la théorie de la fonction de croyance, afin de définir une mesure de similarité globale sémantique. Cette mesure combinée est calculée en utilisant les degrés de croyance des différentes mesures de similarité. Enfin, nous validons les liens trouvés pour démontrer l'efficacité de notre système.

Mots clés :

Données liées, découverte de liens, Web sémantique, hétérogénéité des données, correspondance sémantique, filtrage, mesure de similarité, théorie de la fonction de croyance.

Abstract

The increasing publication of linked data on the web poses a major challenge due to their heterogeneity and continuously growing volume. The discovery of links between web resources involves identifying semantic correspondences among similar elements in web data. However, with the ever-increasing amount of data available on the web, it has become crucial to have automated tools for link discovery. Nevertheless, the automatic identification of semantic correspondences between data remains highly challenging, particularly regarding the quality of extracted links.

To address this issue, we propose a solution for discovering links between two sets of linked data using the theory of belief functions. After extracting various resources from the data sets, we initiate the link discovery process to find equivalent resources. We employ a filtering mechanism to categorize data, thereby reducing the search space for similar data. Subsequently, we combine the syntactic, lexical, extensional, and structural measures using the theory of belief functions to establish a comprehensive semantic similarity measure. This combined measure is computed using the degrees of belief from the different similarity measures. Finally, we validate the identified links to demonstrate the effectiveness of our system.

Key words :

Linked data, link discovery, Semantic Web, data heterogeneity, semantic correspondence, filtering, similarity measure, belief function theory.

ملخص

تواجه الزيادة المستمرة في نشر البيانات المرتبطة على الويب تحديًا كبيرًا بسبب تشتتها وزيادتها المستمرة في الحجم. اكتشاف الروابط بين موارد الويب يتمثل في تحديد التطابقات الدلالية بين العناصر المشابهة في بيانات الويب. ومع ذلك، مع الكمية المتزايدة دائمًا من البيانات المتاحة على الويب، أصبح من الضروري الحصول على أدوات تلقائية لاكتشاف هذه الروابط. ومع ذلك، فإن التعرف التلقائي على التطابقات الدلالية بين البيانات لا يزال صعبًا جدًا، خاصة فيما يتعلق بجودة

للمساهمة في حل هذه المشكلة، نقترح حلاً لإجراء اكتشاف الروابط بين الروابط المستخرجة. مجموعتين من البيانات المترابطة باستخدام نظرية وظيفة الاعتقاد. بعد استخراج الموارد المختلفة من مجموعات البيانات، نقوم ببدء عملية اكتشاف الروابط للعثور على الموارد المكافئة. نستخدم آلية تصفية لتجميع البيانات في فئات، مما يقلل من مجموعة البحث عن البيانات المماثلة. ثم نجمع بين القياسات المصطلحية والتمديدية والهيكلية باستخدام نظرية وظيفة الاعتقاد لتعريف قياس دلالة شامل للتشابه الدلالي. يتم حساب هذا القياس المجمع باستخدام درجات الاعتقاد للقياسات المختلفة للتشابه. أخيرًا، نتحقق من الروابط المكتشفة لإثبات فعالية نظامنا.

الكلمات الدالة:

البيانات المرتبطة، اكتشاف الروابط، الويب الدلالي، تشتت البيانات، التطابق الدلالي، التصفية، قياس التشابه، نظرية وظيفة الاعتقاد.

Table des matières

Table des figures

Introduction générale

Contexte

Ces dernières années, l'évolution du web a été marquée par une croissance significative du web sémantique et des données ouvertes et liées. Le web sémantique vise à promouvoir l'utilisation de formats de données qui facilitent le partage, la réutilisation et le traitement par les machines, permettant ainsi la création de nouvelles connaissances grâce au raisonnement. Les données liées, quant à elles représentent une méthode de publication des données qui met l'accent sur le traitement automatisé et l'établissement de liens avec d'autres sources de données[34].

Dans cette perspective, la théorie de la fonction de croyance joue un rôle crucial. Elle permet de combiner différentes mesures de similarité et de fusionner les informations provenant de sources hétérogènes. En utilisant la théorie de la fonction de croyance, il devient possible de définir une mesure de similarité globale sémantique, en prenant en compte la confiance accordée à chaque mesure individuelle[13].

Le concept du web de données repose sur l'idée de publier des données structurées et non structurées sur le web, non pas sous forme de silos de données isolés, mais en les interconnectant pour former un réseau global d'informations. Les données liées visent à partager et à interconnecter des données structurées sur le web selon les principes des données liées, en utilisant une représentation lisible par les machines, afin de former un espace de données global. L'intérêt de construire un ensemble de données liées réside dans leur capacité à établir des liens avec d'autres données, ce qui permet d'enrichir les descriptions et les relations entre les différents éléments.

Problématique

Dans le web de données, établir des liens pertinents entre les données provenant de différentes sources est essentiel. Cependant, les méthodes classiques de matching d'ontologies ne répondent pas toujours aux exigences de précision, nécessitant ainsi des solutions d'adapta-

tion ou d'amélioration. La théorie de fonction de croyance joue un rôle crucial en permettant de combiner différentes mesures de similarité et d'estimer la confiance accordée à chaque mesure.

En effet, l'efficacité des approches existantes doit être améliorée pour garantir des résultats précis et complets. Un outil de découverte de liens doit générer des mappings de haute qualité en produisant autant de liens que possible. En intégrant la théorie de fonction de croyance, il devient possible d'améliorer la précision et l'efficacité des méthodes d'établissement de liens en combinant de manière cohérente les mesures de similarité et en évaluant la confiance accordée à chaque mesure.

Objectif

L'objectif de ce projet est de proposer des solutions d'adaptation ou d'amélioration des méthodes classiques de matching d'ontologies dans le contexte du web de données. La notion de théorie de fonction de croyance va être intégré afin de combiner de manière cohérente différentes mesures de similarité et d'estimer la confiance accordée à chaque mesure.

Le principal but est d'améliorer la précision et l'efficacité des méthodes d'établissement de liens entre les données provenant de différentes sources, en répondant aux besoins spécifiques du contexte des données liées et en garantissant des résultats précis et complets.

Organisation du mémoire

Afin d'atteindre le but de notre travail, l'organisation de notre mémoire sera comme suit :

— **Chapitre 1 : Web de données** Dans ce chapitre, nous avons abordé une étude sur le web de données, nous avons parlé des données liées, par la suite nous avons entamé les différentes mesures de similarité utilisées et enfin la notion de théorie de fonction de croyance et ses différentes méthodes de combinaison.

— **Chapitre 2 : Conception du système** Dans ce chapitre, nous exposons notre approche novatrice pour résoudre le problème de découverte des liens, en mettant en avant la théorie de fonction de croyance (TFC). Nous décrivons en détail les différentes étapes de notre méthode de découverte des liens, visant à établir des correspondances sémantiques entre les données provenant de différentes sources.

— **Chapitre 3 : Implémentation du système** Dans ce dernier chapitre, qui est consacré à l'implémentation de notre solution proposée, nous présentons les différents outils utilisés, ensuite, nous abordons l'évaluation de notre système selon les paramètres de précision et rappel pour évaluer la qualité des liens trouvés.

Chapitre 1

WEB DE DONNÉES

1.1 Introduction

Les technologies du web sémantique sont cruciales pour les systèmes web intelligents et communicants car elles automatisent la réflexion sur les connaissances. Pour un web de données efficace, il est essentiel de rendre les liens entre les données disponibles. Les ensembles de données interdépendants sur le web, appelés données liées, nécessitent une gestion efficace. Pour découvrir automatiquement les liens sémantiques entre ces derniers, on utilise souvent des techniques basées sur des mesures de similarité pour aligner les données liées. Les expérimentations littéraires montrent qu'une seule mesure de similarité ne permet pas d'obtenir un alignement parfait. Pour certaines entités, les mesures utilisées peuvent différer, reflétant ainsi l'incertitude dans les mesures de similarité. Il est donc crucial de gérer cette différence en utilisant des fonctions de croyance.

1.2 Web de données

Dans cette partie nous allons élaborer l'évolution du web de données.

1.2.1 Définition

Le web de données est une initiative du W3C (Consortium World Wide Web) visant à favoriser la publication de données structurées sur le web, toute en les reliant entre elles pour constituer un réseau global d'informations. Il s'appuie sur les standards du web, tels que HTTP et URI et les technologies du web sémantique[2].

1.2.2 Historique

Le web de données visait initialement à partager des informations sans connaître leur contenu, but élargi et modifié au fil des années. Nous citons les quatres étapes de son évolution[15].

— **Web 1.0 :** Appelé également web traditionnel, est statique et se compose de sites web axés sur les produits et contenant principalement du texte et des éléments multimédias créés par des professionnels. Aucune intervention de l'utilisateur n'était nécessaire dans le web "read only" très consultatif et à sens unique. Le lecteur n'intervenait que très peu et ne pouvait pas contribuer en temps réel au contenu qu'il lisait. C'était un immense magazine en ligne.

— **Web 2.0 :** Aussi appelé web social (Vickery andWunsch-Vincent (2007)), change de perspective. Le web se dynamise en favorisant le partage et l'échange d'infos et de contenus. Le web actuel est illimité grâce à des utilisateurs pro et amateurs, et fonctionne désormais dans les deux sens. L'usager est à la fois consommateur et acteur, mais seule une minorité joue le rôle du producteur (read et write), tandis que la majorité se contente de lire et de partager (read et share). L'utilisateur peut consulter, réagir, partager et contribuer au contenu, devenant ainsi la source et pouvant modifier le contenu.

— **Web 3.0 :** Initié à l'époque par Berners-Lee et al. (2001), le Web sémantique de 2001 se concentre sur le savoir, rendre les ressources Web compréhensibles par les machines et les humains pour répondre aux besoins des utilisateurs mobiles sur diverses applications.

— **Web 4.0 :** Le web symbiotique ou intelligent. Un Web d'intelligence connectant individus et objets, avec des données en évolution vers des standards ouverts et un langage universel. Le Web symbiotique vise à innover avec des connexions intelligentes et immerger l'utilisateur dans un environnement Web prégnant. L'esprit humain et les machines vont interagir en symbiose, permettant à l'utilisateur de devenir un créateur constamment connecté à son environnement. C'est une suite du web 3.0, mais soulève des questions majeures sur la vie privée et le contrôle des données.

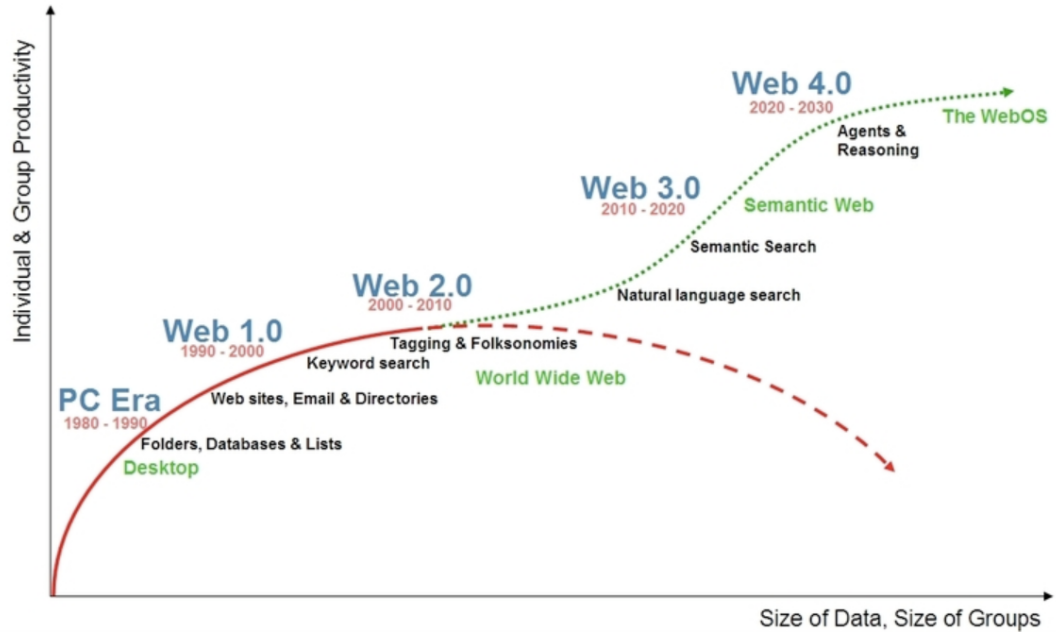

FIGURE 1.1 – L'évolution du Web [8]

La figure 1.1 montre les différentes étapes précitées de l'évolution du Web :

— En Web 1.0, l'information est limitée en fonction de la productivité et du nombre d'utilisateurs.

— Le Web 2.0 génère plus d'infos avec plus d'internautes.

— Avec le Web 3.0, plus d'utilisateurs et de données. Le nombre de sites et d'utilisateurs de Facebook a considérablement augmenté, obligeant les moteurs de recherche traditionnels à s'adapter.

— Avec le Web 4.0, les agents intelligents sont présents. Vision future.

1.2.3 Défis

Afin d'être à la hauteur de ses promesses, le web doit faire face à toutes ces questions [22] :

— **Immensité :** Le web contient plusieurs milliards de pages, donc le web sémantique devrait gérer une grande quantité de données.

— **Imprécision :** Les termes flous tels que "grand" ou "jeune" résultent de demandes d'utilisateurs imprécises. La logique floue est la solution à ce défi.

— **Incertitude :** Des concepts précis peuvent avoir des valeurs incertaines, par exemple un patient présentant divers symptômes qui correspondent à plusieurs diagnostics possibles avec des probabilités différentes.

— **Incohérence :** Des contradictions logiques dans les ontologies provenant de sources diverses.

1.3 Données liées

L'évolution d'un web basé sur les documents a un web de données qui sont liées entre eux.

1.3.1 Définition

Linked data étend la lisibilité du web pour les machines. L'initiative a publié des milliards de morceaux de données, transformant le web traditionnel en un web de liens[21].

1.3.2 Architecture du web de données

La pyramide des langages de Tim Berner Lee est fondamentale pour l'architecture du web sémantique. Elle permet de représenter et de standardiser les connaissances sur le web de manière flexible et interopérable.Cette architecture en couches facilite l'acceptation progressive des standards du web sémantique, qui sont ouverts et établis par le W3C[35].

Cette architecture est présentée par la figure 1.2.

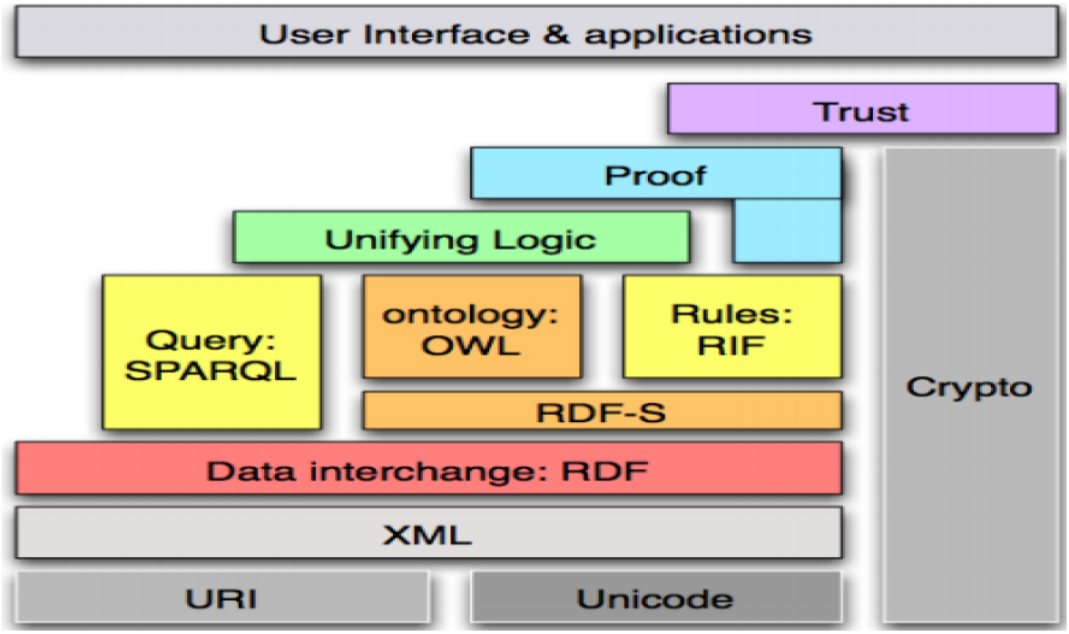

FIGURE 1.2 – L'architecture « Layer cake » proposée par Tim Berners Lee [28]

Pour lier les connaissances entre les différentes couches de manière souple et présenter les données de façon interopérable, le W3C utilise des technologies telles que RDF, RDFS, URI, OWL et SPARQL.
Ces technologies sont décrites ci-dessous :

— **URI :** Label numérique représentant l'adresse ou le nom d'une ressource sur un réseau. Une URL est un URI indiquant l'emplacement de la ressource tandis qu'un URN est un URI désignant son nom[25].

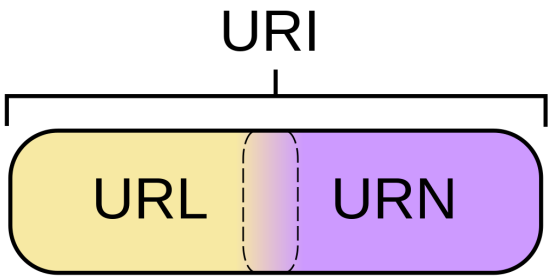

FIGURE 1.3 – Représentation de la relation entre URI, URN et URL [24]

— **XML :** À ce stade, XML ne permet que de structurer des données selon un format de message standard, sans leur donner de sens ou de sémantique. XML décrit la structure et manipule les documents avec des balises identifiées par l'espace de nommage. Le schéma XML permet de définir des vocabulaires pour des documents XML valides, mais

l'interopérabilité syntaxique ne suffit pas à comprendre et à manipuler les données de manière significative. L'interprétation de la sémantique de l'information par la machine reste un problème non résolu à ce niveau de l'architecture. XML et XML Schéma sont couramment utilisés dans les applications orientées Web, mais ils ont des limites en raison de l'absence de sémantique formelle[4].

— **RDF :** RDF est un modèle de données pour les ressources du web, représenté en XML, avec des sémantiques simples et des métadonnées via des URI. La structure fondamentale de RDF se compose d'un triplet (Ressource, Propriété, Valeur) appelé déclaration RDF. Ce triplet est similaire à la structure[26].
  — **Ressource :** Objet référencé par un URI, web ou non (page HTML, PDF, fichier multimédia, etc.). Ou pas (Personne, Région, etc.).
  — **Propriété :** Description de la ressource (titre, couleur, taille, auteur, etc.).
  — **Valeur :** C'est la valeur qui sera affectée à la propriété de la ressource.
— **RDFS :** Langage de description de vocabulaire de RDF permettant de définir la nature des concepts et propriétés d'un triplet, de les combiner de manière significative[10].
— **SPRQL :** Le SPARQL est un langage qui permet d'interroger des graphes RDF en filtrant les requêtes. Il est recommandé par le W3C depuis 2008 et permet d'accéder à des sources de données distribuées sur le Web.
— **OWL :** OWL est un langage d'ontologies pour le Web sémantique.
  1. **Description de OWL :**
     OWL est le standard proposé par W3C pour concevoir des ontologies. Il facilite l'interprétation et le traitement du contenu Web par rapport à XML, RDF et RDFS, grâce à un vocabulaire plus complet qui permet de décrire des ontologies complexes. Cela inclut la disjonction de classe, des types de propriété plus riches, la cardinalité, des caractéristiques de propriété comme la symétrie et la transitivité, l'égalité et les classes énumérées, ainsi qu'une sémantique plus formelle.
  2. **Famille de OWL :**
     OWL a trois sous-langages de plus en plus expressifs : OWL Lite,OWL DL et OWL Full [1].
     — **OWL Lite :** Ce langage est moins expressif que les deux autres car il utilise uniquement certaines fonctionnalités d'OWL. Les contraintes de classe sont limitées et définies seulement en termes de superclasses nommées. Les restrictions de classe sont également restreintes. Permet une faible cardinalité (0 ou 1).
     — **OWL DL :** DL est plus expressif que OWL Lite et a une séparation de type stricte. Les éléments du langage de OWL ne peuvent pas être restreints comme dans OWL Full.

1. : https ://www.w3.org/TR/owl-ref

— **OWL Full :** Ce langage offre une grande expressivité et une liberté syntaxique similaire à RDF, en regroupant les classes, propriétés, individus et valeurs dans les données. Toutefois, l'utilisation de toutes les fonctionnalités OWL peut entraîner une perte de garanties pour les systèmes de raisonnement, contrairement à OWL DL et OWL Lite.

3. **Composants d'ontologie OWL :**
   Les ontologies formalisent la connaissance en utilisant cinq types de composants : classes, relations, fonctions, axiomes et instances.

   — **Les concepts :** Les concepts, ou classes d'ontologie sont des abstractions pertinentes d'une réalité. Ils sont composés de trois éléments distincts : le terme, qui permet d'exprimer le concept en langue naturelle. Intention : sens du concept avec propriétés et contraintes. Extension : les objets manipulés.

   — **Relations :** Elles sont utilisées pour exprimer des relations entre deux concepts dans un domaine donné[19].

   — **Fonction :** Est un cas particulier de relation, où l'élément de la relation le nième est défini par N-1.

   — **Instances :** Ce sont des représentations spécifiques des éléments des classes, par exemple une classe "étudiant", chaque étudiant est une instance de cette classe.

   — **Axiomes :** Ce sont des représentations spécifiques des éléments des classes, par exemple une classe "étudiant", chaque étudiant est une instance de cette classe élément précèdent [23].

1.3.3 Principe données liées

Les principes de l'initiative de données liées sont les suivants [5] :

— Utilisation d'URI comme noms pour les choses pour étendre la portée du web de ressources en ligne, y compris pour le Web des objets.

— Fournir une sémantique utile pour les URI recherchés en utilisant des formats standardisés tels que RDF. Cette normalisation facilite l'interopérabilité et favorise l'évolutivité, comme dans le cas d'HTML sur le Web.

— Inclure des liens vers d'autres URI pour permettre aux utilisateurs d'accéder à davantage de données. Cela suit la notion d'hyperliens dans le Web classique.

1.3.4 Découverte des liens sémantiques

La découverte des liens sémantiques consiste à reconnaître les relations entre divers éléments de données tels que concepts, entités et propriétés[30].

1.3.4.1 Découverte des liens sémantiques dans un graphe RDF

Identifier les liens sémantiques dans un graphe RDF implique de reconnaître les relations entre les entités selon la sémantique du vocabulaire utilisée pour les définir.
Voici quelques méthodes pour découvrir des liens sémantiques dans un graphe RDF :

— **Raisonnement ontologique :**
Le raisonnement déduit de nouvelles infos des données existantes. Les ontologies offrent des règles pour inférer les entités et leurs liens. Le raisonnement permet d'identifier des liens sémantiques non indiqués dans le graphe, comme lorsque deux entités partagent des propriétés avec une troisième entité sans relation explicite entre elles [1].

— **Apprentissage automatique :**
Les algorithmes d'apprentissage automatique identifient des modèles dans le graphique pour repérer les liens sémantiques. Cette approche est pratique pour les liens complexes à détecter par des méthodes basées sur des règles.

1.3.4.2 Défis de découverte de liens sémantiques dans un graphe RDF

La découverte de liens sémantiques dans graphes RDF peut être difficile en raison de plusieurs défis. Certains incluent :

— **Échelle :** Les graphes RDF volumineux sont difficiles à analyser et déduire des liens, car cela demande beaucoup de calculs[29].

— **Hétérogénéité :** Les graphes RDF peuvent être hétérogènes, avec des données provenant de sources différentes et utilisant des vocabulaires distincts, rendant difficile la définition d'un vocabulaire commun et l'identification de modèles couvrant l'ensemble des sources [6].

— **Complexité :** Les graphes RDF sont souvent très complexes et abstraits, ce qui rend la définition d'un vocabulaire commun difficile et complique l'identification de modèles pertinents pour les données [17].

— **Interprétation :** L'interprétation des liens sémantiques découverts peut être difficile, nécessitant des connaissances spécifiques et compliquant leur utilisation pour les applications[17].

1.3.5 Types de liens des données liées

— **Liens d'identités :** Dans le web de données, plusieurs fournisseurs de données utilisent des URIs différents pour identifier une même entité, ce qui crée une multitude d'identifiants. Le web de données liées résout la duplication des entités de manière évolutive et distribuée en ajoutant une grande quantité de liens owl : sameAs au fil du temps. Cette méthode est utilisée couramment et des centaines de millions de liens owl : sameAs sont disponibles sur le web [20].

— **Liens relationnels :** Les liens relationnels relient les données à l'intérieur de plusieurs ensembles de données. Ils pointent vers des connaissances dans d'autres sources de données, créant un réseau de données potentiellement infini pour les applications clientes[20].

— **Liens de vocabulaires :** Le web de données facilite la découverte et l'intégration de sources de données via les liens RDF. Intégrer les données pour avoir une vue unifiée des schémas utilisés par différentes sources de données. Dans le contexte des données liées, un schéma est un mélange de termes de plusieurs vocabulaires RDF utilisés par une source de données sur le web [20].

1.3.6 Domaines d'applications des données liées :

Le web de données est actuellement utilisé dans différents domaines d'applications[18] :

— **E-commerce :** Il facilite l'exploitation des données essentielles par les moteurs de recherche en restituant leur contexte ; malgré les configurateurs d'échange divers et le manque de fiabilité sur l'internet, l'intégration à grande échelle est impossible. Il permet également une description structurée des produits, prix et informations d'entreprise.

— **Application médicale :** La médecine utilise le web sémantique et les systèmes experts pour partager de nombreuses informations complexes. Le web sémantique utilise des annotations pour décrire les ressources, ce qui est crucial en bio-informatique pour le partage des ressources génomiques.

— **Traitements des langages automatiques :** La sémantique pour le traitement automatique se concentre sur la modélisation de phénomènes sémantiques du langage humain, souvent limité à l'approche formelle de la phrase. L'auditeur décrypte le message de l'orateur en utilisant ses compétences linguistiques et ses connaissances de la situation. Il se crée ainsi une représentation pour choisir sa réponse.

1.3.7 Différentes techniques de découverte des liens dans le contexte des données liées

Cette section regroupe les définitions relatives aux méthodes d'alignement. Elle introduit aussi les différentes mesures de similarité ainsi que leurs caractéristiques.

1.3.7.1 L'alignement

Un alignement de données est un ensemble de correspondances entre les entités (classes,propriétés, prédicats, etc.) formant des données liées, et le processus d'alignement appelé Matching en anglais est l'action qui permet de retrouver ces correspondances qui sont des relations (équivalence,disjonction).

Processus d'alignement : Le processus d'alignement une tâche pendant laquelle est déterminé un alignement A entre deux ontologies O et O', cette tâche est réalisée en utilisant un certain nombre de techniques d'alignement.
En général, l'alignement regroupe trois dimensions :

1. **L'input :** Est constitué essentiellement des ontologies (décrites en OWL, RDFS . . . etc.) qui sont destinées à être aligner ou des instances d'une base de données ou d'une ontologie.
2. **Le processus d'alignement :** Comme le montre la figure 1.4, l'alignement peut être considéré comme une fonction f, tel que à partir de deux ontologies O et O', d'un ensemble des paramètres p et d'un ensemble de ressources externes on aboutit un alignement A'.
3. **L'output :** Est un ensemble d'alignement reliant les entités qui constituent les deux ontologies. Un alignement est d'écrit comme un ensemble de cinq éléments <id, e, e', r , n> tel que :
   — Id : un identifiant unique de l'alignement.
   — e : l'entité à aligner appartenant a O.
   — e' : l'entité à aligner appartenant à O'.
   — r : la relation qui relie e à e'.
   — n : la mesure de confiance de la relation r.

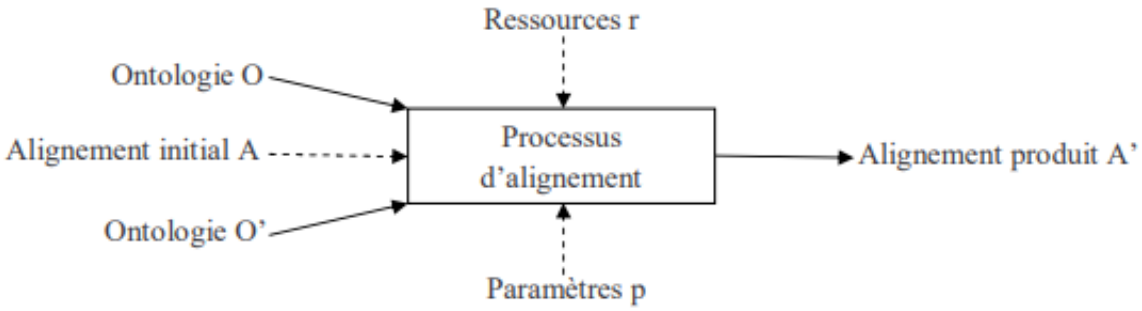

FIGURE 1.4 – Les trois dimensions de l'alignement

1.3.7.2 La similarité

On utilise la notion de similarité sémantique pour évaluer la ressemblance entre des documents, termes ou entités. Cette similarité est mesurée par une métrique qui se base sur leur contenu sémantique. La précision de cette fonction peut varier selon les approches et propriétés souhaitées. Cette fonction est souvent normalisée entre 0 et 1, permettant une interprétation probabiliste de la similarité. Les fonctions ont des propriétés communes telles que la positivité, la maximalité et la symétrie[3].

1.3.7.3 Mesure de similarité

Une mesure de similarité notée σ, permet de mesurer le degré de ressemblance entre deux entités [37]. Soit E un ensemble d'entités, la similarité entre les paires d'entités de cet ensemble est définie par la fonction $\sigma : E \times E \rightarrow \mathbb{R}$ tel $\forall$ x, y, z $\in E$

$$\begin{cases} \sigma(x,y) \geq 0 & \text{(positivité)} \\ \sigma(x,x) \geq \sigma(y,z) & \text{(maximalité)} \\ \sigma(x,y) = \sigma(y,x) & \text{(symétrie)} \\ \sigma(x,y) \leq \infty & \text{(finitude)} \end{cases} \tag{1.1}$$

1.3.7.4 Classification des mesures de similarité

Les différentes mesures de similarité utilisées dans le processus d'alignement sont organisées selon la classification suivante :

Mesures simples :
Ces techniques comprennent les méthodes terminologiques, structurelles, extensionnelles et sémantiques [14]. Elles seront passées en revue ci-après.

1. **Méthodes terminologiques :** On utilise ces méthodes pour mesurer la similarité entre entités textuelles en comparant les chaînes de caractères. Il existe deux approches : syntaxique et lexicale (ou linguistique).

   — **L'approche syntaxique :** Cette approche compare la structure des chaînes en prenant en compte l'ordre des caractères et le nombre d'apparitions de chaque lettre. Cela permet de mesurer la similarité des chaînes en fonction de leurs caractères communs, mais ne tient pas compte de leur signification. Ces méthodes nécessitent en général une normalisation préalable des chaînes à comparer avant leur traitement par les fonctions de similarité, dont on citera la distance de Jaccard :

   — **La distance de Jaccard :** La distance de Jaccard est une mesure de similarité utilisée pour évaluer la similarité entre deux ensembles en se basant sur la taille de leur intersection et de leur union, la distance de Jaccard quantifie la dissimilarité entre les ensembles, L'indice de Jaccard entre deux ensembles A et B est défini comme suit [7] :

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1.2}$$

— **L'approche lexicale :** Les méthodes lexicales avec ressources externes (dictionnaires, taxonomies, ...) Etc. Ces méthodes analysent la similarité entre deux termes représentant des entités. La similitude est évaluée en utilisant des liens sémantiques externes, en se basant sur les connaissances linguistiques et les dictionnaires.

2. **Méthodes structurelles :** Les méthodes structurelles mesurent les similitudes entre entités à partir des informations de leur structure, liées par des liens sémantiques ou syntaxiques. Ces liens créent une hiérarchie ou un graphe d'entités. Les méthodes structurelles sont divisées en internes et externes. L'une se contente des attributs d'entités, l'autre examine les relations entre elles[14].

   — **Méthodes structurelles internes :** Elles calculent la similarité entre deux concepts en exploitant les informations relatives à leur structure interne, dans la plupart des cas, ce sont des informations concernant des attributs de l'entité (restrictions et cardinalités sur les attributs, valeurs des instances...

   — **Méthodes structurelles externes :** Contrairement aux méthodes structurelles internes, les méthodes structurelles externes exploitent les relations existantes entre les entités elles-mêmes.

3. **Méthodes extensionnelles :** Elles comparent les concepts ou classes grâce à leurs ensembles d'instances représentés par des vecteurs. Les vecteurs sont comparés pour obtenir leur similarité. Si les ensembles d'instances sont en partie communs, les mesures extensionnelles telles que la distance de Hamming ou la mesure de Jaccard sont utilisées. Hamming compte les éléments différents, Jaccard mesure l'intersection et l'union des ensembles. Ces mesures peuvent être adaptées pour créer des mesures extensionnelles [45].

   — **Distance de Hamming :** La distance de Hamming entre deux ensembles S et T est définie comme le nombre d'éléments qui diffèrent entre les deux ensembles. Plus précisément, si S et T sont des ensembles de taille n, la distance de Hamming est calculée comme suit :

$$\text{distance\_hamming}(S,T) = \frac{\cdot|S \cup T| - |S \cap T|}{2} \tag{1.3}$$

4. **Méthodes sémantiques :** Les méthodes sémantiques se basent principalement sur deux approches. La première approche repose sur les modèles de la logique tandis que la deuxième approche regroupe les méthodes de déduction afin de déduire la similarité entre deux entités.

1.4 Théorie des fonctions de croyance

Dans cette partie nous allons aborder la notion de théorie de fonction de croyance ainsi que ses caractéristiques.

1.4.1 Définition

La théorie des fonctions de croyance de Dempster et Shafer (1967) permet de modéliser les données incertaines et imprécises et de fusionner différentes sources d'informations pour une décision plus fiable. Les fonctions de masse sont utilisées pour modéliser les données incertaines et imprécises et les croyances communes sont mises en évidence[40].

1.4.2 Principes de base

Soit un cadre de discernement $\Omega = \{\omega 1, \omega 2, \ldots, \omega n\}$ l'ensemble de toutes les hypothèses exclusives et exhaustives. Le cadre de discernement est aussi l'univers de discours d'un problème donné.

— **Fonction de masse :** Une fonction de masse est une fonction de 2^{Ω} vers l'intervalle [0, 1] qui affecte à chaque sous-ensemble une masse de croyance élémentaire et représente des connaissances imprécises et incertaines. Formellement, une fonction de masse, notée m^{Ω}, est définie comme suit :

$$m : 2^{\Omega} \rightarrow [0, 1] \tag{1.4}$$

$$\Omega_{A \subseteq \Omega} m(A) = 1 \tag{1.5}$$

Un sous-ensemble ayant une masse de croyance élémentaire non-nulle est un élément focal.

— **Fonction de masse combinée :** La fonction de masse combinée combine plusieurs fonctions de masse portant sur les mêmes variables aléatoires. Elle donne la masse de chaque ensemble plausible d'événements selon au moins l'une des fonctions initiales. Si m1 et m2 sont deux fonctions de masse sur un ensemble d'événements E, alors la fonction de masse combinée m est obtenue en utilisant la formule :

$$m_C(A) = 1 - (1 - m_1(A))(1 - m_2(A)) \tag{1.6}$$

où A est un sous-ensemble de E, et m(A) est la masse combinée de l'événement A.

— **Fonction de croyance :** La fonction de croyance mesure les degrés de croyance des hypothèses/propositions, quantifie l'incertitude et combine différentes sources d'information. La fonction de croyance donne un degré de croyance entre 0 et 1 à chaque hypothèse. 0 signifie qu'il n'y a aucune croyance, 1 signifie une totale croyance. Les degrés de croyance de 0 à 1 expriment l'incertitude partielle.

— **Fonction de plausibilité :** La fonction de plausibilité est une fonction qui mesure à quel point une hypothèse donnée est plausible ou vraisemblable, en fonction des connaissances

disponibles. Plus précisément, si m est une fonction de masse définie sur un espace des propositions H, la fonction de plausibilité P(A) pour une hypothèse A est définie comme :

$$p(A) = \sup_{\substack{B \subseteq H \\ \neg A \cap B \neq \emptyset}} \{m(B)\} \tag{1.7}$$

— **Fonction de normalisation :** La normalisation est utilisée en théorie des fonctions de croyance pour garantir que la somme des degrés de croyance ou de plausibilité de tous les événements possibles est égale à 1.
Sa formule est simple : diviser chaque degré par la somme totale. Cela assure la cohérence de la théorie des fonctions de croyance et la comparabilité des degrés d'événements.

— **La notion de conflit :** Gérer les conflits est essentiel en théorie des fonctions de croyance. Cela implique de résoudre les contradictions entre sources d'information ou hypothèses. En théorie des fonctions de croyance, des conflits surgissent lorsque différentes sources fournissent des degrés de croyance contradictoires pour un même ensemble d'hypothèses. Par exemple, une source peut avoir un degré de croyance élevé alors qu'une autre source a un degré de croyance faible dans le même ensemble d'hypothèses. La gestion de conflit vise à harmoniser les contradictions pour évaluer les croyances.

1.4.3 Méthodes de combinaison

La fusion de données imparfaites est résolue par la combinaison d'informations agrégées grâce à la théorie de la fonction de croyance. En effet, pour un même problème et cadre de discernement, une fonction de masse peut être obtenue en combinant des sources d'informations indépendantes.
Ci-dessous nous allons présenter trois modes de combinaison :

— **Combinaison conjonctive :** Ce mode de combinaison est utilisé pour deux sources distinctes et indépendantes [12] [38]. Cette formule utilise l'opérateur logique "ET" pour combiner les résultats des différentes mesures de similarité. Elle renvoie la similarité minimale parmi toutes les mesures considérées. tel que :

$$m(A) = \sum_{B \cap C = A} m_1(B) m_2(C) \tag{1.8}$$

— **Combinaison disjonctive :** La combinaison disjonctive est une approche utilisée pour fusionner les informations provenant de différentes sources ou experts[39]. Elle utilise le “OR“ logique et donne plus de poids à la mesure maximale pour indiquer la plus grande similarité entre les entités. Cela favorise la prise de décision en considérant la similarité la plus forte et en accordant plus de confiance à la mesure la plus significative

grâce à la combinaison disjonctive de différentes sources d'information. La formule de la combinaison disjonctive est la suivante :

$$m(A) = \sum_{B \cup C = A} m_1(B) m_2(C) \tag{1.9}$$

— **Combinaison de Dempster-Shafer :** La règle de combinaison de Dempster-Shafer permet de fusionner les fonctions de croyance provenant de sources différentes, tout en prenant en compte les incertitudes et les conflits. La formule de Dempster-Shafer permet d'estimer de manière cohérente les croyances relatives aux fonctions. La formule de combinaison de Dempster-Shafer est la suivante [11] :

$$CombinedSim = \frac{\sum_{B \cap C = A} m_1(B) \times m_2(C)}{1 - \sum_{B \cap C = A} m_1(B) \times m_2(C)} \tag{1.10}$$

1.4.4 Objectifs de la fonction de croyance

Les objectifs de la fonction se résument en :

— Représenter mathématiquement l'incertitude et la croyance en une proposition ou hypothèse.

— Combinaison d'éléments pour une évaluation précise et robuste.

— Offrir un cadre général pour raisonner avec incertitude, utile dans plusieurs domaines tel que l'intelligence artificielle.

1.5 Méthodes de découverte des liens dans le contexte du web de données

Dans cette partie nous allons présenter quelques travaux :

1.5.1 LIMES

— LIMES (Link Discovery Framework for Metric Spaces) est un outil open-source d'interconnexion pour le Web sémantique.

— Conçu pour découvrir des liens entre des entités dans des sources de données liées, il utilise une architecture extensible unique. Il peut détecter des liens sous forme de deux fichiers RDF liés.

— Le framework LIMES est fiable et supporte diverses techniques d'interconnexion appelées mesures de similarité, utiles dans de nombreux cas[31].

1.5.2 SILK

— Avec le Silk Link Spéciation Langage, l'utilisateur choisit les entités à lier et la mesure de similarité.

— Outil Web souple pour lier entités de sources de données variées, employant diverses techniques d'alignement et de mesure de similarité.

— Silk récupère des données RDF stockées dans un SPARQL [33].

1.5.3 MINTE

— Technique d'intégration utilisant les vocabulaires RDF et la similarité sémantique pour fusionner des graphes RDF équivalents.

— MINTE utilise une approche en deux étapes pour intégrer des graphes RDF : un algorithme d'appariement parfait est d'abord employé pour identifier les entités RDF équivalentes, puis différentes politiques de fusion sont appliquées pour rassembler les triplets correspondants.

— Les résultats expérimentaux suggèrent que MINTE peut intégrer des graphes RDF sémantiquement équivalents avec précision[9].

1.5.4 Knofuss

— Knofuss fusionne des ensembles de données et les alignent en termes d'ontologies hétérogènes. La découverte de liens de données est guidée par des ontologies spécialisées qui déterminent les ressources à comparer et la technique d'alignement appropriée. Les ressources sont choisies en utilisant une requête SPARQL. Cet outil fournit des algorithmes d'alignement de chaînes[32].

1.5.5 PARIS

— PARIS est un outil pour aligner des données hétérogènes en utilisant la probabilité pour mettre en correspondance les relations, instances et schémas similaires malgré l'utilisation de termes différents [41].

1.6 Comparaison des méthodes

Nous présenterons un tableau récapitulatif synthétisant notre analyse des méthodes de découverte de liens entre données. Les critères pris en compte sont : les entrées du système, l'automatisation, la mesure de similarité et la technique associée, le type de lien en sortie ainsi que le domaine concerné.

Travaux	Entrée	Automatisation	Similarité	Technique de similarité	Lien de sortie	Domaine
SILK[2020]	RDF,XML,SPQL	Semi-automatique	Syntaxique	Taxonomie	Owl :sameAs	Science de la vie
Knofuss[2012]	RDF,SPQL	Semi-automatique	Syntaxique	Alignement des mesures de chaine de caractère	Owl :sameAs	Plusieurs
LIMES[2021]	RDF,CSV	Automatique	Syntaxique, Sémantique	Taxonomie,Alignement des mesures de chaine de caractère	Owl :sameAs	Plusieurs
PARIS[2011]	RDF, CSV	Automatique	Syntaxique	Alignement des mesures de chaines de caractères	Owl :sameAs	Plusieurs
MINTE[2017]	RDF, XML, CSV	Semi-automatique	Syntaxique, Sémantique	Taxonomie, Alignement des mesures de chaine de caractère	Owl :sameAs	Plusieurs

TABLE 1.1 – Comparaison des travaux existants pour la d"couverte de liens.

1.6.1 Analyse

Après l'étude que nous avons fait dans le tableau précédant on a distingué ces différents critères :

— Toutes ces méthodes acceptent en entrée un large choix de fichiers ; cela tout dépend de la source de données qu'on souhaite aligner.

— Pratiquement toutes les approches cités ci-dessus sont semi-automatique ; ce qui nécessite l'intervention de l'expertise humaine dans le choix des méthodes utilisées pour calculer la similarité et a fin de garantir la qualité des résultats finaux, Sauf LIMES et PARIS qui sont completement automatique ; ce qui permet a l'utilisateur de se concentrer sur la validation des liens proposés.

— Pour l'utilisation de la similarité entre les données liées, la même technique de calcul qui est souvent basée sur la mesure de similarité syntaxique.

— L'aspect sémantique de MINTE réside dans sa capacité à évaluer la similarité des concepts et des relations entre les entités.

— Ces framework génèrent en sortie des liens owl : sameAs. Le framework SILK Peut générer d'autres types de liens qui doivent être spécifiés par l'utilisateur.

1.7 Conclusion

Dans ce premier chapitre, nous avons donné une vision générale sur le web de données et les données liées, leurs caractéristiques et leurs langages de représentation, nous avons également abordé la notion de théorie de fonctions de croyance.

Dans le chapitre suivant, nous présentons la conception de notre système, nous allons présenter en premier lieu une architecture générale pour le fonctionnement de notre application ainsi que les méthodes utilisées pour le calcul des similarités entre les triples de données.

Chapitre 2

CONCEPTION DU SYSTÈME

2.1 Introduction

La conception d'un système est une étape importante pour l'évaluation des besoins d'un utilisateur, dans ce chapitre nous proposons une solution à fin de résoudre le problème de découverte de liens.
D'abord, nous allons commencer par la description des caractéristiques de notre système par rapport aux systèmes existants présentés dans le chapitre précédent.
Ensuite, nous allons entammer la présentation du schéma global de notre système et son fonctionnement, avec une explication détaillée de chaque étape.

2.2 Caractéristiques de notre système

La plupart des travaux examinés précédemment combinent quelques mesures de similarité, mais cela diminue la granularité sémantique. De plus, la plupart des tâches requièrent une intervention humaine. Notre objectif est de proposer un système de découverte de liens en combinant plusieurs mesures de similarité avec des fonctions de croyance pour obtenir une mesure globale de haute qualité.

Notre système se caractérise par :

— La conception d'un systeme purement automatique.

— L'utilisation du standard RDF comme un langage de représentation des ressources.

— L'utilisation de plusieurs mesures de similarités pour avoir la similarité sémantique (terminologique,structurelle, extensionnelle).

— L'utilisation des fonctions de croyances pour la combinaison des mesures de similarité.

2.3 Modélisation des données liées en utilisant la combinaison de Dempster Shafer

Une valeur de similarité mesure la similarité entre deux objets, concepts ou entités. Dans les fonctions de croyance, cela peut représenter la confiance accordée à une hypothèse. La similarité est généralement numérique et mesure la distance entre deux éléments. Elle sert à calculer la fonction de masse de croyance et plus elle est élevée, plus la confiance dans l'hypothèse est grande.

Pour la combinaison de ces mesures de similarité, il n'y a pas de méthode universelle, car la meilleure technique de combinaison dépend des résultats de mesures de similarité et du degrés de conflits entre eux.

On a choisi de démontrer la méthode de Dempster-Shafer par l'exemple ci-dessous [11] : Supposons que nous disposions de trois mesures de similarité, S1, S2 et S3 utilisées pour comparer deux objets A et B, nous procédons comme ceci :

- **Définition de propositions :**
  P : A et B sont semblables.
  q : A et B ne sont pas semblables.
- **Source d'information :**
  Afin de construire le bba (fonction de masse), il faut identifier la source d'information donnée par une source. Par exemple, faire correspondre les deux ressources A et B avec un degrés de 0.4 par la similarité terminologique est une information.
- **Affectation de croyance de base :**
  Selon la théorie de croyance, cette similarité peut-être interprétée comme un degré de croyance d'une mesure de similarité. Ce degré est lié au fait si les deux ressources sont proches ou éloignées l'une de l'autre.
  Donc, on considère par exemple que la valeur 0.4 n'est autre que le degré de croyance de similarité terminologique. En plus de cela, les valeurs de similarité sont dans l'intervalle $[0, 1]$, nous n'avons donc pas a convertir ces valeurs mais plutôt de les interpréter comme masse.

Le tableau suivant présente les valeurs de similarités terminologique, structurelle et extensionnelle entre les deux ressources.

Mesure de similarité	Valeur donnée
$S_{term}(A, B)$	0.4
$S_{Int}(A, B)$	0.2
$S_{ext}(A, B)$	0.7

TABLE 2.1 – Résultats de similarité des deux ressources

D'aprés ce tableau, on pourra construire les fonctions de masses :

Mesure de similarité	Fonction de masse
$S1_{term}$	$m1_{term}(p) = 0.4; m1_{term}(q) = 0.6$
$S2_{Int}$	$m2_{Int}(p) = 0.2; m2_{Int}(q) = 0.8$
$S3_{ext}$	$m3_{ext}(p) = 0.7; m3_{ext}(q) = 0.3$

TABLE 2.2 – Construction des fonctions de masse

Une fois que nous avons obtenus toutes les correspondances, on pourra combiner avec Dempster Shafer selon la formule ci-dessous :

$$m_{1\oplus 2}(A) = \frac{\sum_{B\cap C=A} m_1(B) \times m_2(c)}{1 - \sum_{B\cap C=\oslash} m_1(B) \times m_2(c)} \tag{2.1}$$

Où :
$\sum_{B\cap C=A} m_1(B) \times m_2(c)$,représente le conflit global et la règle est normalisée via :
$1 - K = 1 - \sum_{B\cap C=\oslash} m_1(B) \times m_2(c)$, cette normalisation est utilisée pour masquer le conflit.

Ainsi pour appliquer cette formule sur notre cas, on va procéder par paire, les résultats sont démontrés dans le tableau suivant :

Masses	$m1_{term}(p)$	$m1_{term}(q)$
$m2_{Int}(p)$	$0.4 \times 0.2 = 0.08$	$0.6 \times 0.2 = 0.12$
$m2_{Int}(q)$	$0.4 \times 0.8 = 0.32$	$0.6 \times 0.8 = 0.48$

TABLE 2.3 – Combinaison de m1 et m2

Donc :

$$K = \sum_{B \cap C = \varnothing} m_1(B) \times m_2(c) =$$

$$(m1_{term}(p) \times m2_{Int}(q)) + (m2_{Int}(p) \times m1_{term}(q)) = 0.32 + 0.12 = 0.44$$

ie :

$$m_{1\oplus 2}(p) = \frac{0.08}{1 - 0.44} = 0.14$$

$$m_{1\oplus 2}(q) = \frac{0.48}{1 - 0.44} = 0.86$$

On va procéder de la meme façon en combinant ces deux masses résultats avec la 3ème mesure de similarité, les résultats sont représentés par le tableau ci-dessous :

Masses	$m_{1\oplus 2}(p)$	$m_{1\oplus 2}(q)$
$m3_{ext}(p)$	$0.14 \times 0.7 = 0.099$	$0.86 \times 0.7 = 0.6$
$m3_{ext}(q)$	$0.14 \times 0.3 = 0.04$	$0.86 \times 0.3 = 0.26$

TABLE 2.4 – Combinaison avec m3

Où :

$$K = (m_{1\oplus 2}(q) \times m3_{ext}(p)) + (m_{1\oplus 2}(p) \times m3_{ext}(q)) = 0.6 + 0.04 = 0.64$$

Donc la fonction de croyance globale est :

$$Bel(p) = \frac{0.099}{1 - 0.64} = 0.28$$

Ce résultat indique que les deux ressources sont similaires avec dempster shafer en combinant les trois mesures de similarité avec un degré de croyance de 0.28

2.4 Schéma global du système

Dans ce qui suit nous présentons le schéma global de notre solution, ensuite nous détaillons les différentes étapes et algorithmes utilisés.

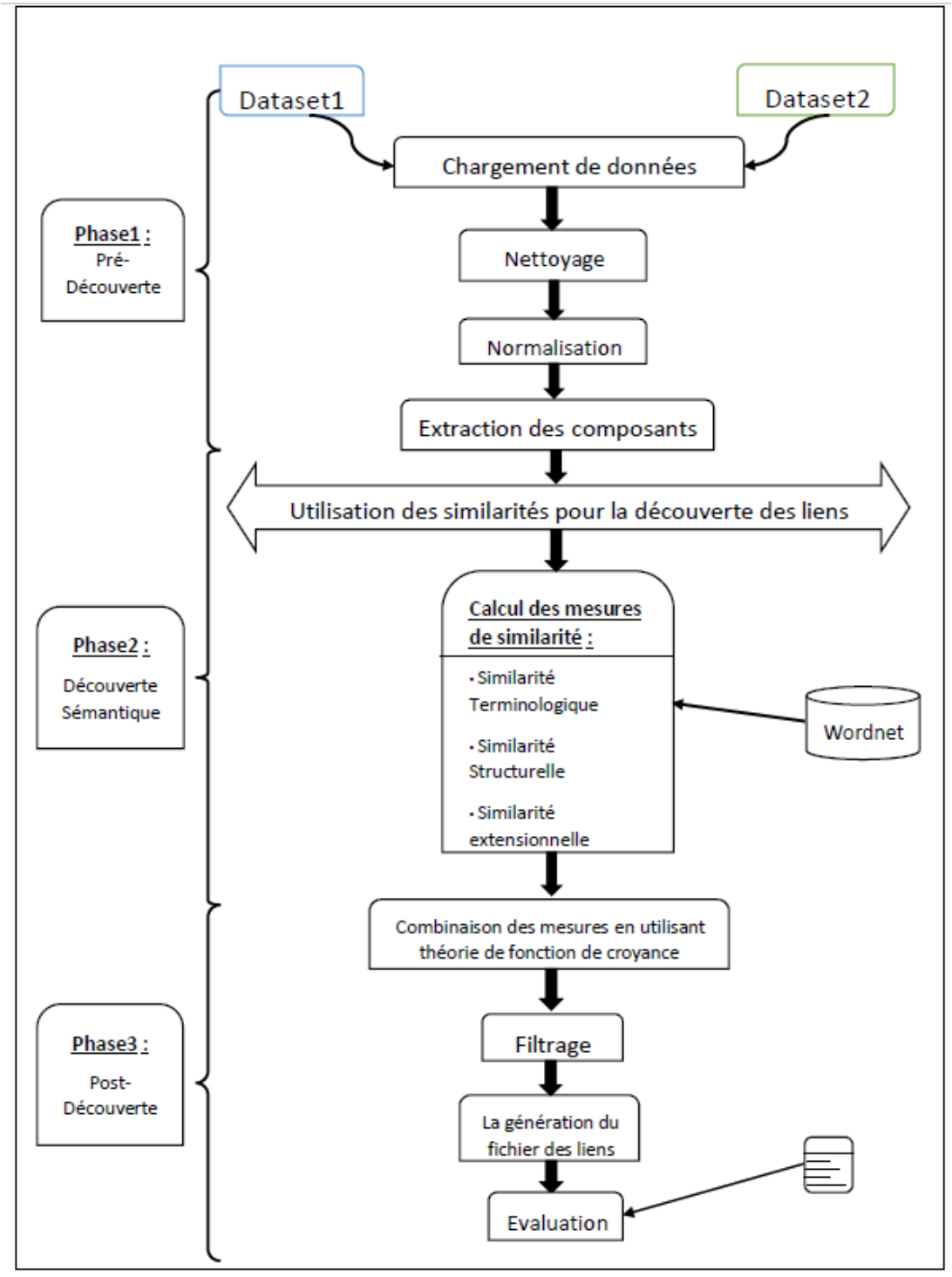

FIGURE 2.1 – Schéma global

2.5 Description du schema

La solution proposée consiste en :

Phase 1 : Pré-découverte

La pré-découverte représente la première phase de notre solution. Tout d'abord nous avons téléchargé les deux dataset.
Après une étape de nettoyage qui consiste à éliminer tous les triplets inutiles. Ensuite, nous avons normalisé les données et enfin l'extraction des composants.

Cette phase comporte quatre étapes principales :

— **Etape 1 :** Chargement de données.

— **Etape 2 :** Nettoyage.

— **Etpe 3 :** Normalisation.

— **Etpe 4 :** Extraction des composants.

Phase 2 : Découverte sémantique

Cette phase répond au besoin de la découverte de liens de données. Elle comporte trois étapes essentielles :

— **Etape 1 :** Calcul de la similarité terminologique.

— **Etape 2 :** Calcul de la similarité structurelle.

— **Etpe 3 :** Calcul de la similarité extensionnelle.

Phase 3 : Post-découverte

Il s'agit de la dernière phase de notre solution. Elle permet de faire une combinaison des mesures de similarités utilisées a fin de répondre a l'aspect sémantique, ainsi engendrer des liens entre les triplets a l'aide de cette combinaison.

Elle comporte trois étapes essentielles :

— **Etape 1 :** Combinaison des mesures de similarité.

— **Etape 2 :** Filtrage.

— **Etape 3 :** Générer un fichier de liens.

— **Etape 4 :** Evaluation.

2.5.1 Pré-découverte

La pré-découverte est la première phase de notre solution. Il faudra télécharger nos deux datasets, ainsi que passer par l'étape de nettoyage et de normalisation et enfin extraire les composants.

2.5.1.1 Etape 1 : Chargement de données

Cette étape consiste à extraire tous les triplets (Sujet,Prédicat,Objet) de chaque dataset d'une manière automatique.

Les datasets utilisés sont des datasets OAEI (Ontology Alignment Evaluation Initiative) qui sont des ensembles de données utilisés pour évaluer et comparer les performances des méthodes d'alignement d'ontologies. Chaque dataset OAEI se compose généralement de paires d'ontologies, où chaque paire représente deux ontologies distinctes qui doivent être alignées. Ces paires d'ontologies sont accompagnées d'un ensemble de correspondances de référence, également appelé gold standard, qui spécifie les correspondances correctes entre les ontologies.

Les datasets OAEI sont créés de manière à couvrir différents aspects et défis de l'alignement d'ontologies, tels que l'hétérogénéité sémantique, la similarité des concepts, la structure des ontologies, etc. Cela permet d'évaluer la capacité des méthodes d'alignement à traiter des scénarios variés et complexes.

2.5.1.2 Etape 2 : Nettoyage

Cette partie consiste à éliminer tous les triplets dont les sujets ne possèdent aucun prédicats et objets dans les deux datasets, pour faciliter les traitements dans les étapes suivantes.

2.5.1.3 Etape 3 : Normalisation

Afin d'améliorer les résultats de comparaison entre les chaînes de caractères, nous avons opté pour l'utilisation des méthodes suivantes. Celle-ci consistent à normaliser chaque triplet <Sujet, Prédicat, Objet>.

1. **Normalisation de caractères spéciaux :** Convertir tout parenthèses, tirets... etc en un espace blanc.

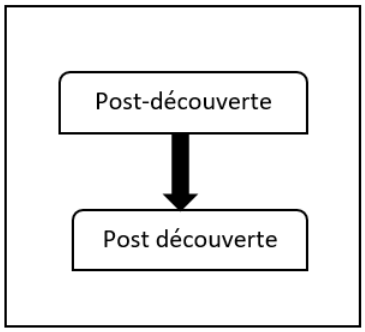

FIGURE 2.2 – Normalisation de caractères spéciaux

2. **Suppression des ponctuations :** Dans cette phase nous allons supprimer tout point d'exclamation, interrogation...etc.

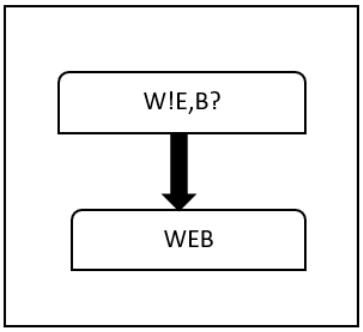

FIGURE 2.3 – Suppression des ponctuations

3. **La normalisation de la structure :** Chaque élément du dataset est sous forme "Ressource,Predicat1, Objet1, Predicat2, Objet2, etc."

2.5.1.4 Etape 4 : Extraction des composants

Les triplets extraits constituent essentiellement les ressources destinées à être alignées qui seront stockées dans deux dictionnaires différents ; un dictionnaire est une structure de données qui permet d'associer des clés à des valeurs correspondantes. Il s'agit d'une collection d'éléments où chaque élément est composé d'une paire clé-valeur.
Les triplets extraits du dataset1 seront stockés dans un dictionnaire **"Dict1"** ,les triplets du dataset2 seront stockés dans un dictionnaire **"Dict2".**
Dans cette étape nous avons développé deux algorithmes, le premier pour l'extraction de prédicats et d'objets pour chaque ressource, le deuxième pour afficher ces derniers.

Algorithme 1 : Algorithme d'extraction de propriété et d'objet pour chaque ressource

```
Input : Graph
Result : Dictionnaire
for triplet in graph do
    ressource,predicat,objet = triplet for ressource in ressources do
        dictionnaire[ressource] = ressource
    if ressource in dictionnaire then
        dictionnaire[ressource].ajouter([predicat,objet])
    dictionnaire[ressource] = ([predicat,objet]);
```

Le deuxième algorithme pour l'affichage présenté ci-dessous :

Algorithme 2 : Algorithme d'affichage du dictionnaire

```
Input : Graph
Result : Dictionnaire
for ressource in dictionnaire do
    Ressource = ressource;
    for predicat, objet in dictionnaire[Ressource] do
        predicat = tab[0];
        objet = tab[1]
```

2.5.2 Découverte sémantique

Nous avons commencé par le calcul des mesures de similarité entre les deux datasets, on a pris en considération le coté terminologique, structurelle et extensionnelle.
Dans tout les aspect on a choisi de travailler avec la mesure de jaccard, de cosinus pour atteindre le coté syntaxique et la mesure de wordnet pour le coté lexicale.

Pour les mesures terminologiques le traitement se fait sur la ressource avec des mesures syntaxiques et lexicales, pour les mesures intensionnelles le traitement se fait avec wordnet sur les predicats et en dernier un traitement simple sur les objets, dans le but de répondre a l'aspect sémantique, comme le montre la figure ci dessous :

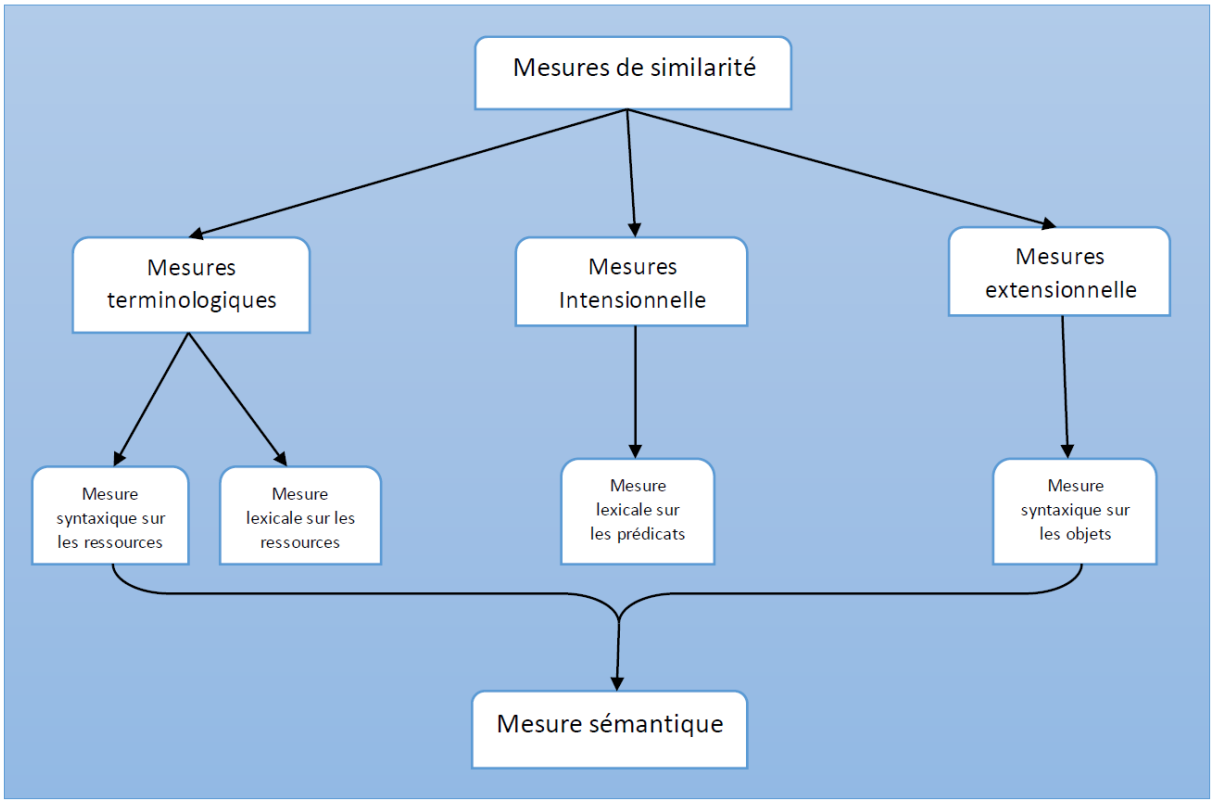

FIGURE 2.4 – Mesures de similarité

2.5.2.1 Mesures terminologiques :

Dans cette étape, nous avons appliqué la similarité terminologique, afin de chercher les correspondances entre les triplets des deux datasets.
Elle est décomposée en deux types de similarité, la similarité syntaxique et la similarité lexicale.

1. **La similarité syntaxique :** Pour évaluer la correspondance entre les triplets sans comprendre leur sens. Pour cette tâche, nous avons choisi d'utiliser ces deux mesures :
   - **La mesure de Jaccard :** La mesure de « Jaccard » qui calcule la similarité entre deux paires de chaînes de caractères.
     Jaccard est le rapport entre la cardinalité de l'intersection des ensembles de comparaison et les cardinalités de l'union des ensembles, le résultat de cette mesure est compris entre 0 et 1

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.2}$$

     On considère A comme étant une ressource de dataset1 et B une ressource de dataset2.
     La similarité entre A et B est calculée en traitant chaque caractère.

Exemple : la similarité de Jaccard entre "Austen" et "Austin".

"Austen" : est composé de 6 caractères

"Austin " : est composé 6 caractères

La cardinalité de l'union est de 7 et la cardinalité de l'intersection est de 5. L'indice de Jaccard pour ces deux chaînes est de 0.71.

$$J(\text{"}Austen\text{"}, \text{"}Austin\text{"}) = \frac{|\text{"}Austen\text{"} \cap \text{"}Austin\text{"}|}{|\text{"}Austen\text{"} \cup \text{"}Austin\text{"}|} = \frac{|5|}{|7|} = 0.71$$

— **La mesure de Cosinus :** La mesure de similarité cosinus est une mesure couramment utilisée pour calculer la similarité entre deux vecteurs dans un espace vectoriel. Cependant, il n'est pas directement applicable aux chaînes de caractères telles quelles. Pour utiliser la similarité cosinus, nous devons d'abord représenter les chaînes de caractères sous forme de vecteurs.

Pour cela nous avons opté pour la fonction de CountVectorizer, qui est une technique de vectorisation de texte qui convertit un ensemble de chaine de caractères en une représentation vectorielle selon le nombre d'occurence de la lettre.

Algorithme 3 : Cosine Similarity for Strings

Input : str1 : Chaîne de caractères, str2 : Chaîne de caractères

Output : Similarité cosinus

```
corpus ← [str1, str2];
vectorizer ← CountVectorizer().fit_transform(corpus);
vectors ← vectorizer.toarray();
similarity ← cosine_similarity(vectors [0].reshape(1, -1), vectors [1].reshape(1, -1));
return similarity [0][0];
```

Cet algorithme prend en entrée deux chaines de caractères et renvoie la similarité entre eux.

Exemple : la similarité de cosinus entre "Austen" et "Austin".

1. Création d'un ensemble de tous les caractères uniques :
{'a','u','s','t','e','n','i'}.

2. Création des vecteurs binaires pour chaque chaîne de caractères :

"Austen" : [1, 1, 1, 1, 1, 1, 0]

"Austin " : [1, 1, 1, 1, 0, 1, 1]

3. Calcul du produit scalaire entre les deux vecteurs :
$1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 1 = 5.$

4- Calcul de la norme de chaque vecteur :

"Austen" : $\sqrt{(1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 0^2)} = \sqrt{5}$

"Austin " : $\sqrt{(1^2 + 1^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2)} = \sqrt{6}$

5- Le résultat est : $5/(\sqrt{5} * \sqrt{6}) = 0.91$

$$similarity[Austen][Austin] = 0.91$$

2. **La similarité lexicale :** La similarité WordNet mesure la similarité sémantique entre des concepts en utilisant la structure hiérarchique du réseau sémantique WordNet. Pour le calcul de la similarité lexicale entre les deux concepts. Nous avons utilisé la mesure WUP est l'une des mesures de similarité les plus couramment utilisées dans WordNet, elle est basée sur le principe que deux synsets ayant un ancêtre commun dans la hiérarchie de WordNet sont plus similaires. Elle parcourt chaque paire de synsets, calcule la similarité WUP entre eux et stocke les valeurs de similarité numériques dans une liste. Ensuite, la fonction retourne la valeur maximale de similarité parmi toutes les paires de synsets. Si aucune similarité numérique n'est trouvée, la fonction retourne 0 [44].

$$WUPSimilarity(S_1, S_2) = \frac{2 \cdot Depthof LowestCommonSubsumer(S_1, S_2)}{Depthof(S_1) + Depthof(S_2)} \quad (2.3)$$

Où :

— S_1 est le premier synset.

— S_2 est le deuxième synset.

— $Depthof LowestCommon(S_1, S_2)$ est la profondeur du plus bas ancêtre commun.

— $DepthS_1$ est la profondeur du synsetS_1.

— $DepthS_2$ est la profondeur du synsetS_2.

La profondeur d'un synset représente la distance entre le synset et la racine de la hiérarchie de WordNet. Le plus bas ancêtre commun est le synset le plus proche de la racine qui est un ancêtre commun des deux synsets.

Aprés avoir calculé les trois mesures de similarité, nous allons les combiner avec la méthode de Dempster-Shafer pour avoir un seul résultat.

2.5.2.2 Mesures structurelles

Pour avoir la mesure structurelle, il nous faut d'abord calculer la similarité structurelle entre les deux prédicats, on a choisi WordNet entre deux listes de synsets en utilisant la mesure de similarité WUP (Wu-Palmer). Elle parcourt chaque paire de synsets, calcule la similarité WUP entre eux et stocke les valeurs de similarité numériques dans une liste. Ensuite, la fonction retourne la valeur maximale de similarité parmi toutes les paires de synsets. Si aucune similarité numérique n'est trouvée, la fonction retourne 0.

Algorithme 4 : Algorithme de calcul de similarité structurelle

```
Input  : ListePrédicat1, ListePrédicat2
Output : Similarité Structurelle
sim ← [];
foreach predicat1 ∈ ListePredicat1 do
    synset1= wordnet.synsets(predicat1)
    foreach predicat2 ∈ ListePredicat2 do
        synset2= wordnet.synsets(predicat2)
        similarity ← synset1.wup_similarity(synset2);
        if type(similarity) == int or type(similarity) == float then
            sim.append(similarity);
if len(sim) != 0 then
    return max(sim);
else
    return 0;
```

Cet algorithme est utilisé pour calculer la mesure structurelle entre deux listes de prédicats et retourne la similarité maximale.

2.5.2.3 Mesures extensionnelles

Pour calculer la mesure de similarité extensionnelle sur les objets, la valeur de l'objet doit être une valeur littérale, on a opté pour ces deux mesures :

— **La mesure de Jaccard :** La mesure de « Jaccard » qui calcule la similarité entre deux paires de chaînes de caractères.
Jaccard est le rapport entre la cardinalité de l'intersection des ensembles de comparaison et les cardinalités de l'union des ensembles.

$$J(O1, O2) = \frac{|O1 \cap O2|}{|O1 \cup O2|} \tag{2.4}$$

Exemple : la similarité de Jaccard entre "Paris" et "Lisbon"
"Paris" : est composé 5 caractères
"Lisbon" : est composé de 6 caractères
La cardinalité de l'union est de 9 et la cardinalité de l'intersection est de 2.L'indice de Jaccard pour ces deux chaînes est de 0.18

$$J(" Paris", " Lisbon") = \frac{|" Paris" \cap " Lisbon"|}{|" Paris" \cup " Lisbon"|} = \frac{|2|}{|9|} = 0.22$$

— **La mesure de Cosinus :** Nous avons importé la fonction de CountVectotizer pour calculer la mesure de cosinus.
Exemple : la similarité entre "Paris" et "Lisbon".
1. Création d'un ensemble de tous les caractères uniques :
{'p', 'a', 'r', 'i', 's', 'l', 'b', 'o', 'n'.}
2. Création des vecteurs binaires pour chaque chaîne de caractères :
"Paris" : [1, 1, 1, 1, 1, 0, 0, 0, 0]
"Lisbon" : [0, 0, 0, 1, 1, 1, 1, 1, 1]
3. Calcul du produit scalaire entre les deux vecteurs :
$1 \times 0 + 1 \times 0 + 1 \times 0 + 1 \times 1 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 2$
4. Calcul de la norme de chaque vecteur :
"Paris" : $\sqrt{(1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2 + 0^2} = \sqrt{5}$
"Lisbon" : $\sqrt{(0^2 + 0^2 + 0^2 + +1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2)} = \sqrt{6}$.
5. Le résultat est : $2/(\sqrt{5} * \sqrt{6}) = 0.37$

$$similarity[Paris][Lisbon] = 0.37$$

Aprés le calcul des mesures de Jaccard et Cosinus, on va utilisé la méthode de Dempster-Shafer pour combiner les deux résultats.

2.5.3 Post-découverte

Après le calcul de la similarité entre les différents triplets provenant des deux datasets, vient la dernière phase de notre solution.

2.5.3.1 Etape 1 : Combinaison des mesures

La similarité sémantique évalue le lien sémantique de deux triplets. Nous avons utilisé la moyenne pour estimer cette similarité. Ensuite, nous avons abordé la théorie de la fonction de croyance en utilisant la combinaison de Dempster-Shafer pour des évaluations plus précises de la similarité sémantique des triplets.
Dont on définera les formules ci-dessous :

$$Moy(R1, R2) = \frac{M_{term}(R1, R2) + M_{Int}(R1, R2) + M_{ext}(R1, R2)}{3} \tag{2.5}$$

$$m_{1 \oplus 2}(A) = \frac{\sum_{B \cap C = A} m_1(B) \times m_2(C)}{1 - \sum_{B \cap C = \oslash} m_1(B) \times m_2(C)} \tag{2.6}$$

2.5.3.2 Etape 2 : Filtrage

Le filtrage fixe un seuil pour conserver les triplets dont la similarité dépasse cette valeur et supprimer les autres.
Nous obtenons donc un fichier des ressources les plus similaires.

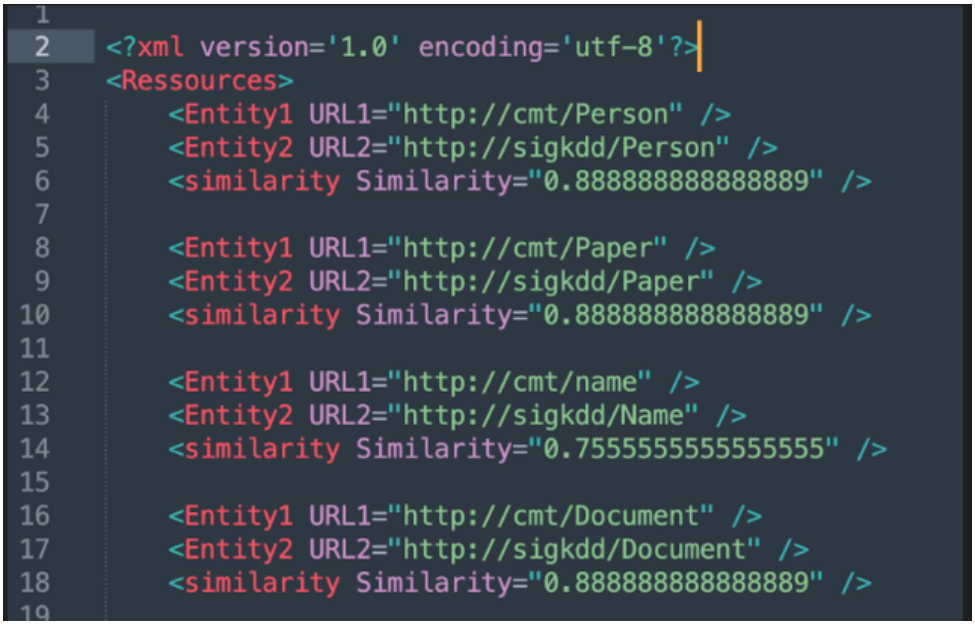

FIGURE 2.5 – Un aperçu du fichier des résultats RDF

2.5.3.3 Etape 3 : Génération du fichier des liens

Le fichier généré par notre système est un fichier de format RDF, qui représente les liens entre les ressources équivalentes.

FIGURE 2.6 – Un aperçu du fichier RDF

2.5.3.4 Etape 4 : Evaluation

Les mesures utilisées pour évaluer la qualité des correspondances produites entre les ressources des dataset sont principalement les mesures de pertinence en recherche d'information, telles que la précision, le rappel et F-mesure dont on expliquera en détails dans le prochain chapitre.

2.6 Conclusion

Dans ce chapitre, nous avons présenté les étapes suivies pour la construction de notre système. En détaillant les trois phases : pré-découverte, découverte sémantique, ainsi que les mesures des similarités utilisées et enfin poste découverte.

Dans le chapitre suivant, nous allons implémenter ce que nous avons proposé dans la partie conception et tester notre système et son fonctionnement.

Chapitre 3

IMPLÉMENTATION DU SYSTÈME

3.1 Introduction

Après avoir réaliser la phase de conception de notre système, nous nous intéressons maintenant à l'implémentation du système, nous allons tout d'abord présenter l'environnement de développement de ce travail ainsi que les différents outils utilisés, puis nous décrirons de façon visuelle notre implémentation via des captures d'écran des différentes interfaces de notre système et à la fin nous passerons au test et validation du système.

3.2 Environnement de développement

La réalisation de notre système nécessite les outils suivants :

3.2.1 Python

Pour l'implémentation de notre système nous avons utilisé comme langage de programmation le langage « python », Python est un langage de programmation de haut niveau interprété pour la programmation à usage général. Créé par Guido van Rossum, et publié pour la première fois en 1991.
Python repose sur une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces significatifs. Il fournit des constructions permettant une programmation claire à petite et grande échelle.

Python propose un système de typage dynamique et une gestion automatique de la mémoire.Il prend en charge plusieurs paradigmes de programmation, notamment orienté objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque standard étendue et complète[43].

L'interpréteur Python est facilement étendu avec de nouvelles fonctions et de nouveaux types de données implémentés en C ou C ++ (ou d'autres langages pouvant être appelés à partir de C).

Il convient également comme langage d'extension pour les applications personnalisables. Le langage étant très complet et ne nécessitant aucune phase de compilation. Le choix de ce langage présente les avantages suivants :

— Python est entièrement gratuit.

— C'est un langage complet et puissant dans de nombreux domaines.

— Il est orienté objet mais n'impose pas ce type de programmation.

— Sa syntaxe reste très simple et le code peut être très lisible.

— Raccourcit le cycle de développement par rapport aux langages compilés et permet un prototypage rapide des projets.

3.2.2 NLTK

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'Université de Pennsylvanie. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API).

3.2.3 NumPy

NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

3.2.4 WordNet

WordNet est une base de données lexicale pour la langue anglaise, créée par Princeton et faisant partie du corpus NLTK.
WordNet peut être utilisé avec le module NLTK pour trouver la signification des mots, des synonymes, des antonymes, etc[16].

3.2.5 Google Colab

Google Colab appelé aussi Colaboratory est un service proposé par Google gratuitement.
Il est basé sur l'environnement Jupyter Notebook et est destiné à la formation et à la recherche en apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud sans avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.

3.2.6 Pandas

Pandas est une bibliothèque très utile pour la manipulation et l'analyse de données tabulaires en Python, offrant une syntaxe concise et puissante pour effectuer des opérations courantes sur les données[27].

3.2.7 Bootstrap

Bootstrap est un framework web front-end open-source, qui simplifie la création d'interfaces utilisateur attractives grâce à une série d'outils préconçus. Bootstrap facilite la conception de sites Web adaptables aux mobiles en utilisant HTML, CSS et JavaScript, avec une grille flexible pour des mises en page qui s'adaptent à toutes tailles d'écran. Il propose divers composants prêts à l'emploi pour créer rapidement des interfaces utilisateur professionnelles[42].

3.2.8 Scikit-learn

Scikit-learn est une bibliothèque d'apprentissage automatique pour le langage de programmation Python. Il comporte divers algorithmes de classification, de régression et de catégorisation, notamment des support-vector machines, random forests, MDS, k-means and DBSCAN, et est conçu pour interagir avec les bibliothèques numériques et scientifiques Python NumPy et SciPy[36].

3.2.9 RDFLib

RDFLib est une bibliothèque Python pour travailler avec RDF, un langage simple mais puissant pour représenter la connaissance. Cette bibliothèque contient des parseurs/sérialiseurs pour presque toutes les sérialisations RDF connues, telles que RDF/XML, Turtle, N-Triples et JSON-LD.

3.2.10 Matplotlib

Matplotlib est une bibliothèque de visualisation de données en Python. Elle offre des fonctionnalités permettant de créer une grande variété de graphiques, de diagrammes et de visualisations interactives. Matplotlib est flexible et hautement personnalisable, offrant un contrôle précis sur chaque élément graphique

3.2.11 Visual Studio Code

Est un éditeur de code open-source créé par Microsoft pour les systèmes d'exploitation Windows et Linux. Il offre une grande flexibilité en termes de langages de programmation pris en charge, notamment Java, Javascript, Python et C++. Grâce à ses extensions, il permet de développer des applications Web.

3.3 Présentation de l'application

Après avoir présenté tous les outils et l'environnement que nous avons utilisé pour développer notre système. Nous passons à une vue plus proche et concrète.
Dans cette section, nous présentons l'interface de notre système.
La figure 3.1 représente l'interface d'accueil de notre système.

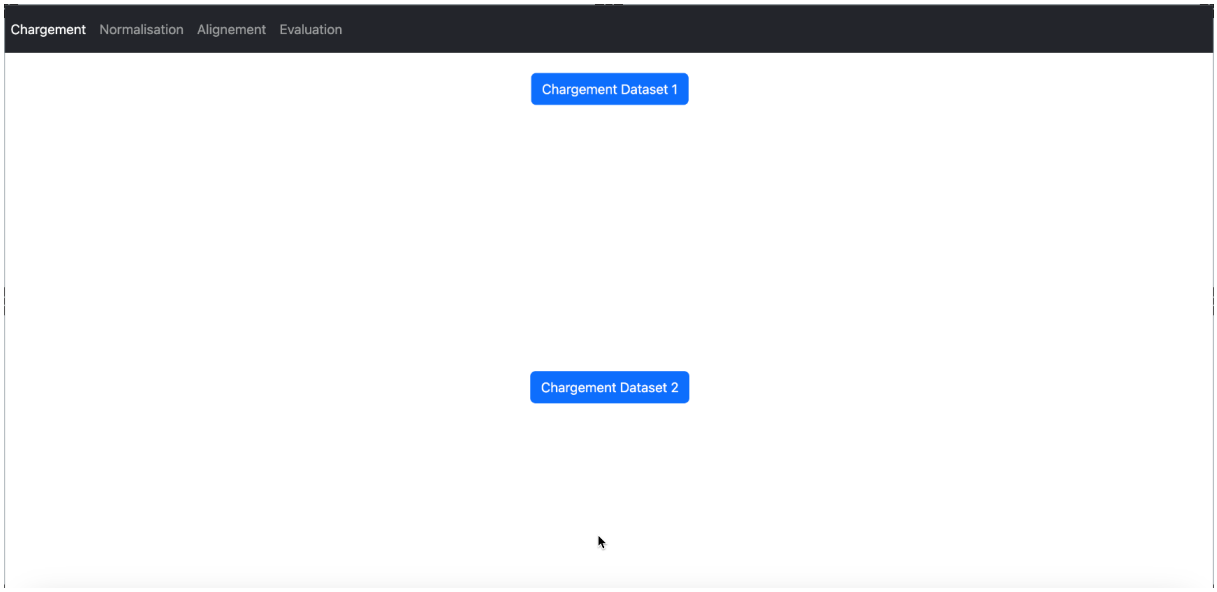

FIGURE 3.1 – Interface d'accueil

En cliquant sur les deux boutons (Chargement Dataset 1) et (Chargement Dataset 2), le processus de chargement se lance et son résultat s'affiche illustré par la figure 3.2.

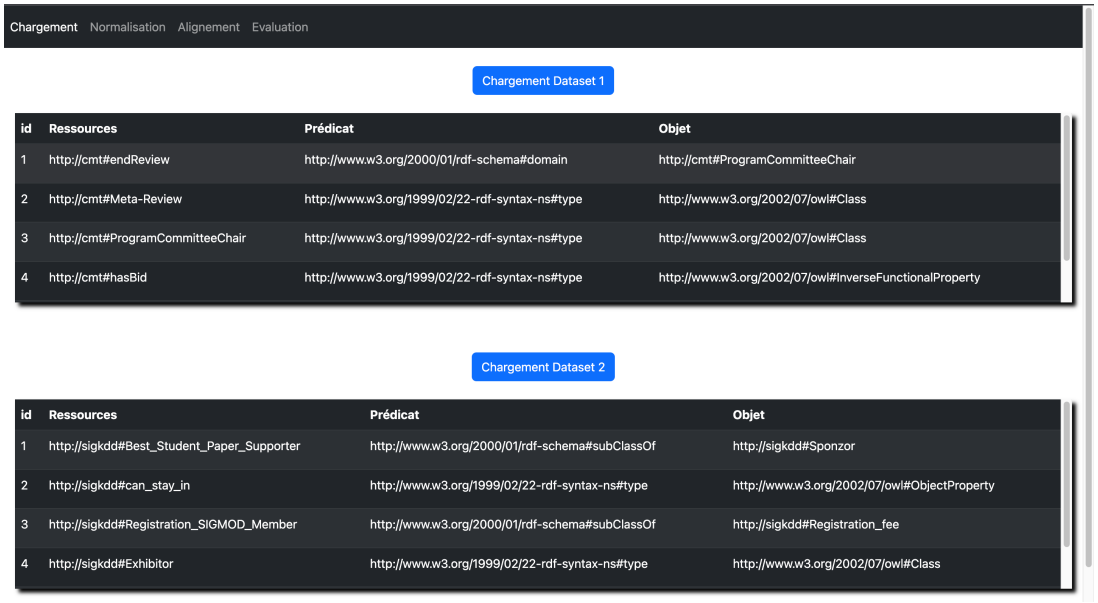

FIGURE 3.2 – Chargement des datasets

Pour obtenir les résultats de la normalisation, il est nécessaire d'appuyer sur les deux boutons de l'interface de normalisation : "Normaliser DS1" et "Normaliser DS2" (figure3.3).

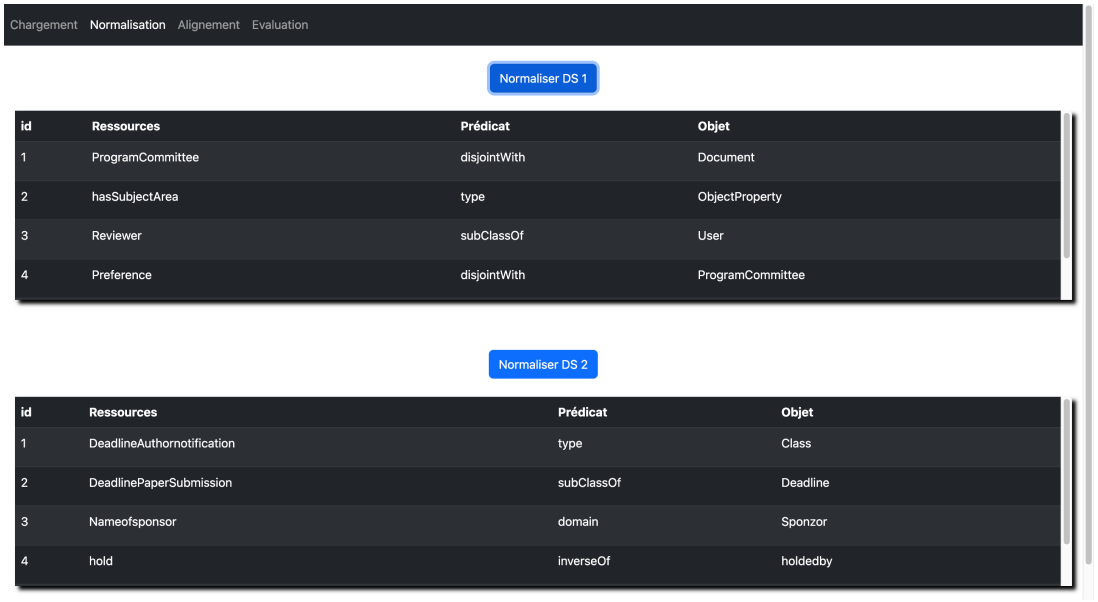

FIGURE 3.3 – Normalisation

La troisième page de l'interface où on trouve les trois boutons de calcul de mesures de similarité (figure3.4).

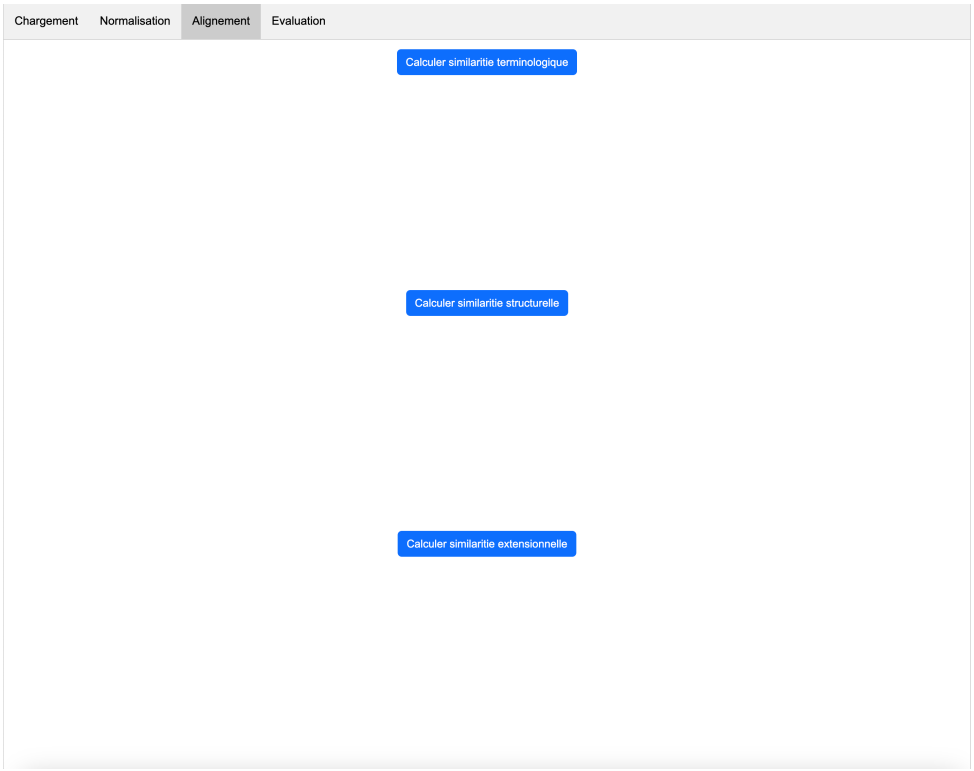

FIGURE 3.4 – Interface d'alignement

La figure3.5 montre la dernière page de l'interface qui permet d'evaluer les résultats en calculant des mesures appropriées .

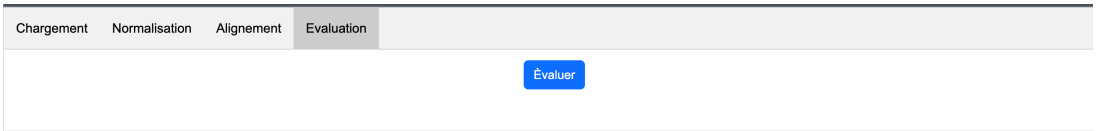

FIGURE 3.5 – Interface d'évaluation

3.3.1 Test du système

Pour évaluer la fiabilité et la validité de notre système, nous avons utilisé deux datasets. Nous avons comparé les correspondances obtenues avec un fichier d'équivalence qui représente les correspondances établies par un expert du domaine.

3.3.1.1 Résultats expérimentaux et discussion

Dans cette section nous présentons les résultats obtenus par les différentes mesures de similarité que nous avons testé et utilisé pour la découverte de liens entre les différentes ressources. Nous avons appliqué la mesure syntaxique et lexicale ensuite nous avons utilisé les différentes méthodes de combinaison. Par la suite nous avons appliqué la mesure extensionnelle sur les objets, les résultats obtenus sont mentionnés dans les figures ci-dessous.

1. La similarité entre les ressources des triplets des deux datasets :

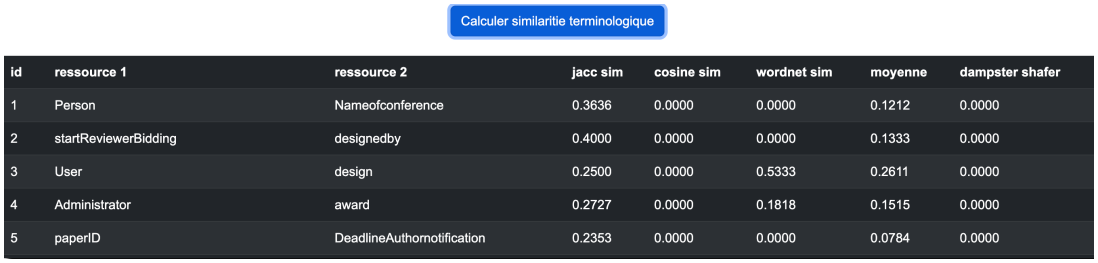

FIGURE 3.6 – Les résultats obtenus entre les ressources des deux datasets.

— Concernant la mesure syntaxique, nous avons utilisé "Jaccard" et "Cosinus", la mesure lexicale "WordNet".

— Nous avons obtenus les résultats des deux combinaisons en combinant les trois mesures de similarité comme le montre la figure3.6.

2. La similarité entre les prédicats des triplets des deux datasets :

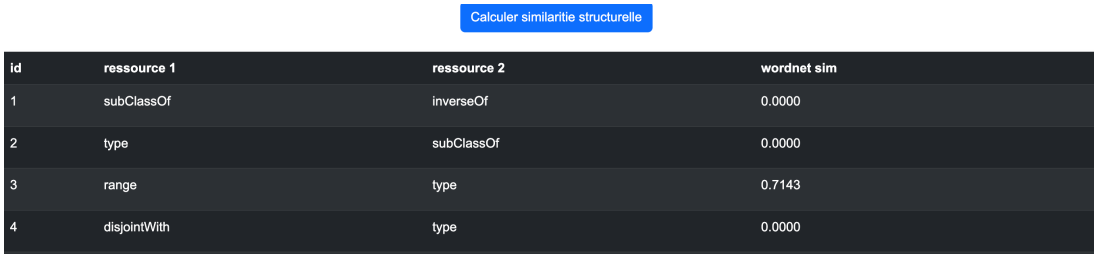

FIGURE 3.7 – Les résultats des similarités structurelles.

— Concernant la mesure structurelle, nous avons utilisé "Wordnet" pour comparer entre les deux prédicats (figure3.7).

3. La similarité entre les objets des triplets des deux datasets :

FIGURE 3.8 – Les résultats des similarités extensionnelles.

— Concernant cette partie, nous avons utilisé des mesures syntaxiques "Jaccard" et "Cosinus".

— Nous avons utilisé la moyenne et Dempster Shafer pour combiner les résultats obtenus (figure 3.8).

3.3.2 Mesures d'évaluation utilisées

Les mesures utilisées pour évaluer la qualité des correspondances produites entre les ressources des dataset sont principalement les mesures de pertinence en recherche d'information, telles que la précision, le rappel et F-mesure.

Le calcul de ces mesures est basé sur la comparaison entre les correspondances produites par un système automatique qu'on appellera **S** et un ensemble de correspondances de référence produit par un humain qu'on notera **H**.

— Les correspondances correctes trouvées par un système sont appelées **(the true positives (TP))** et sont calculées ainsi :

$$TP = S \cap H \tag{3.1}$$

— Les correspondances incorrectes trouvées par le système sont appelées **(the false positives (FP))** et sont calculées ainsi :

$$FP = S - S \cap H \tag{3.2}$$

— Les correspondances correctes omises par le système sont appelées **(the false negatives (FN))** et sont calculées ainsi :

$$FN = H - S \cap H \tag{3.3}$$

— La précision est une mesure d'exactitude, elle varie entre [0,1] elle est calculée de la manière suivante :

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

— Le rappel est une mesure de perfection, elle varie entre [0,1] elle est calculée de la manière suivante :

$$Rappel = \frac{TP}{TP + FN} \tag{3.5}$$

— Les résultats obtenus en calculant les correspondances entre les datasets ne peuvent pas être comparés uniquement à l'aide de la précision et du rappel. Le rappel peut être élevé au détriment de la précision en retournant toutes les correspondances possibles, tandis que la précision peut être élevée au détriment du rappel en ne retournant que les correspondances correctes mais en nombre limité.
C'est pourquoi il est préférable de prendre en compte les deux mesures simultanément en utilisant une mesure qui combine le rappel et la précision, comme la F-mesure. La F-mesure est calculée de la manière suivante :

$$F - mesure = \frac{(2 * (Precision * Rappel))}{(Precision * Rappel)} \tag{3.6}$$

— La F-mesure est une mesure globale de la qualité des correspondances produites, elle varie entre [0.1]. Cette mesure alloue la même importance à la précision et au rappel.

On a généré plusieurs fichiers avec des différents seuils et méthodes de combinaison pour les trois mesures ainsi on a calculé les mesures d'évaluation pour chaque expérience, les résultats sont affichés ci-dessous :

Première expérience :

	tp	fp	fn	cal Precision	cal rappel	fmesure
seuil_Moyenne=0.5	11	0	0	1.0	1.000000	1.0
seuil_Dampster_Shafer=0.5	9	0	2	1.0	0.818182	0.9
seuil_Moyenne=0.7	9	0	2	1.0	0.818182	0.9
seuil_Dampster_Shafer=0.7	9	0	2	1.0	0.818182	0.9

FIGURE 3.9 – Résultat de combinaison globale

Les résultats obtenus sont interprété par un histogramme 3.10, où on trouve les métriques d'évaluation pour chaque seuil défini.

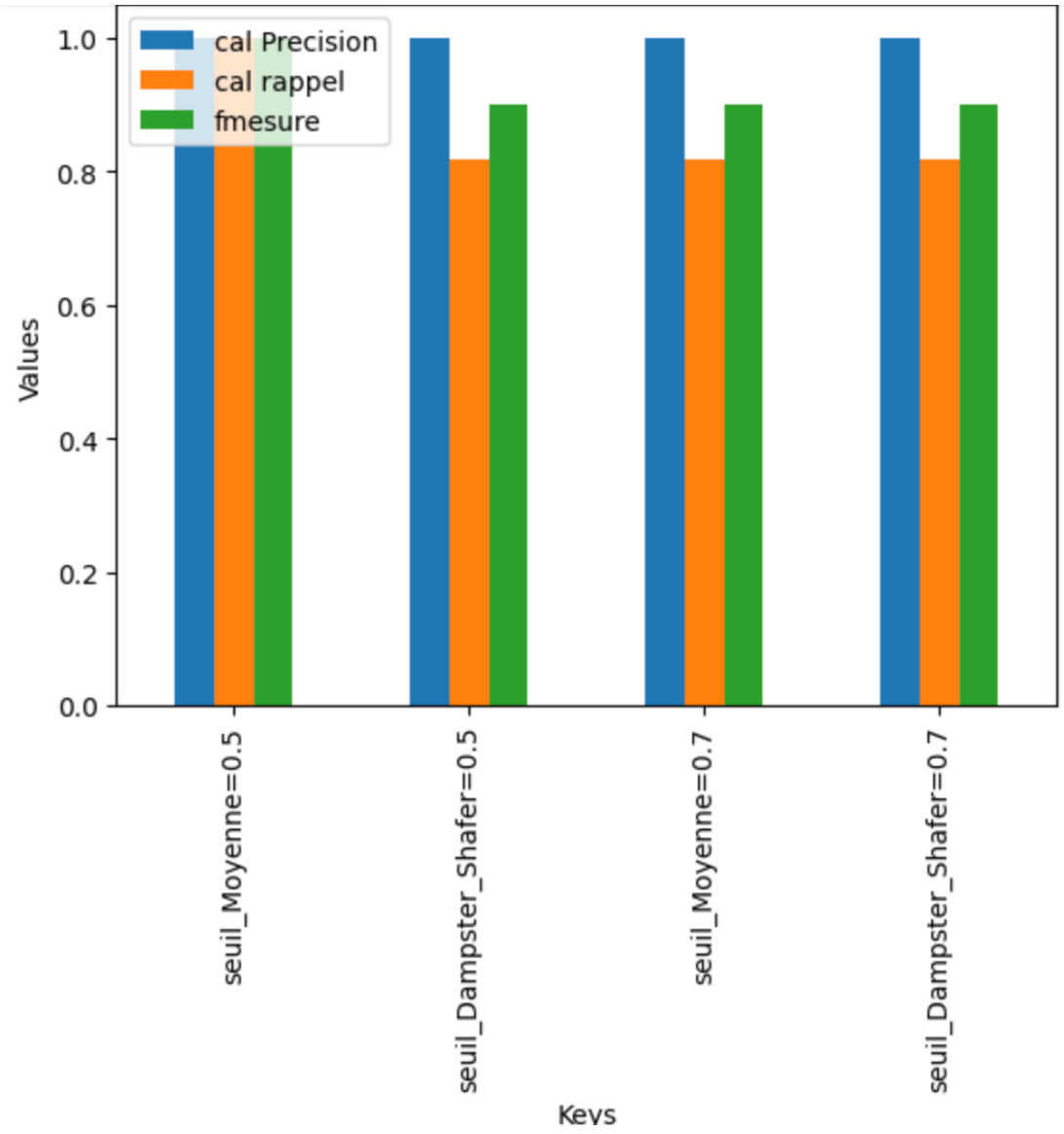

FIGURE 3.10 – Histogramme de résultats

Interprétation de résultats :

A travers les tableaux et l'histogramme présentés précédemment, nous pouvons clairement constater :

— La valeur de rappel obtenu par la combinaison Dempster-Shafer restitue un meilleur résultat avec un score de 0.81, ce qui signifie que les résultats sont à 81% complets.

— En terme de précision, notre système confirme son niveau de performance obtenant une valeur de 1, pour les deux méthodes de combinaisons et les deux seuils définis.

— Concernant la F-mesure elle obtient un résultat de 0.90 signifie que 90% des réponses pertinents, le taux de F-mesure est élevé grâce aux taux élevés de Précision et Rappel.

Deuxième expérience :

	tp	fp	fn	cal Precision	cal rappel	fmesure
seuil_Moyenne=0.5	5	1	3	0.833333	0.625	0.71
seuil_Dampster_Shafer=0.5	7	1	1	0.875000	0.875	0.88
seuil_Moyenne=0.75	4	0	4	1.000000	0.500	0.67
seuil_Dampster_Shafer=0.75	7	1	1	0.875000	0.875	0.88

FIGURE 3.11 – Résultat de combinaison globale

Les résultats sont représentés graphiquement sous forme d'un histogramme3.12, affichant les métriques d'évaluation correspondantes pour chaque seuil défini.

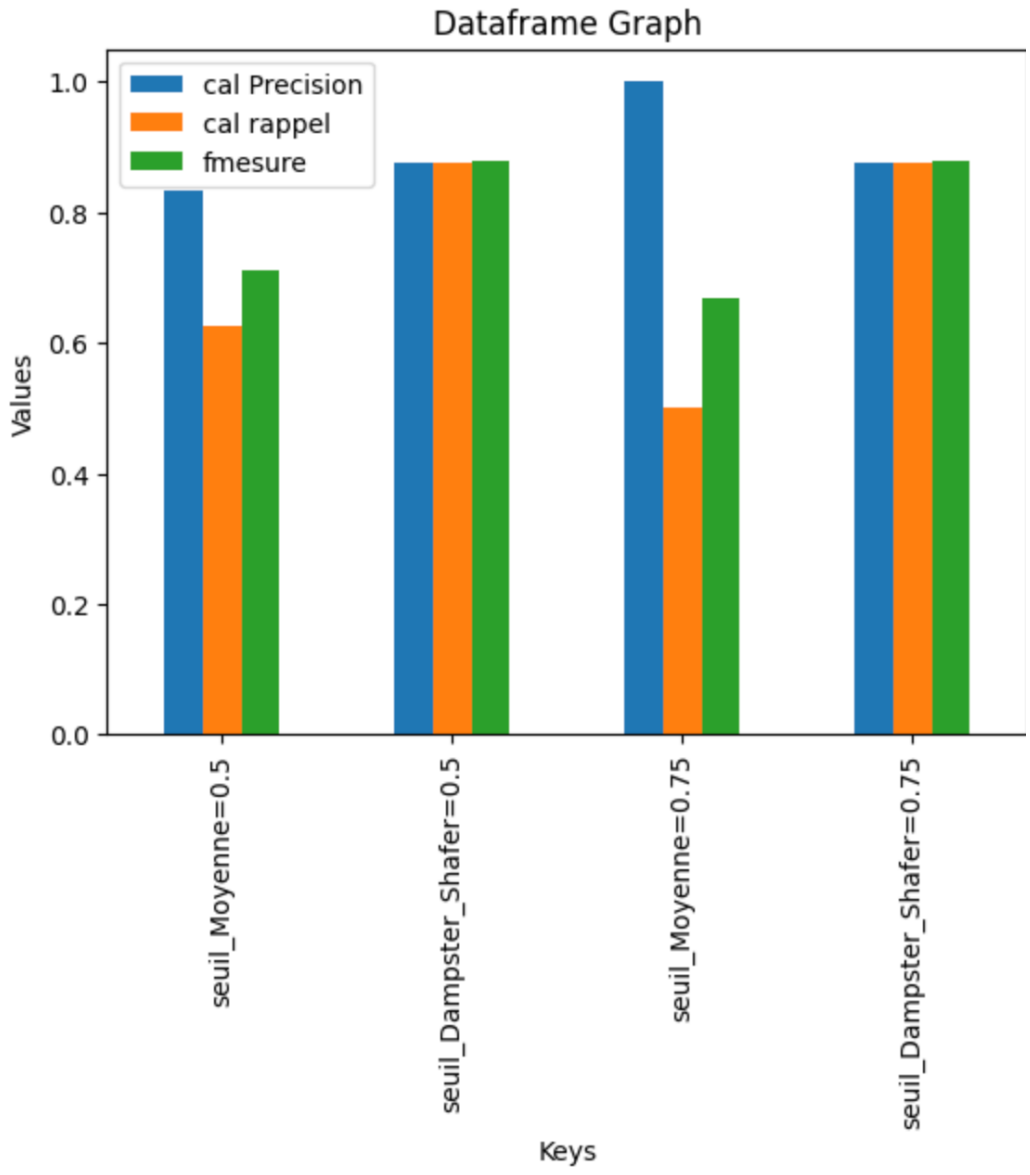

FIGURE 3.12 – Histogramme de résultats

Interprétation de résultats :

A travers les tableaux et l'histogramme présentés précédemment, nous pouvons clairement constater :

— La valeur de rappel obtenu par notre système qui combine en utilisant Dempster-Shafer est de 0.87, ce qui signifie que les résultats sont à 87% complets.

— En terme de précision, notre système confirme son niveau de performance obtenant une valeur de 0.87, pour les deux méthodes de combinaisons et les deux seuils définis.

— Concernant la F-mesure elle obtient un résultat de 0.88 signifie que 88% des réponses pertinents, le taux de F-mesure est élevé grâce aux taux élevés de Précision et Rappel.

Troisième expérience :

	tp	fp	fn	cal Precision	cal rappel	fmesure
seuil_Moyenne=0.6	9	1	2	0.900000	0.818182	0.86
seuil_Dampster_Shafer=0.6	10	1	1	0.909091	0.909091	0.91
seuil_Moyenne=0.7	6	1	5	0.857143	0.545455	0.67
seuil_Dampster_Shafer=0.7	10	1	1	0.909091	0.909091	0.91

FIGURE 3.13 – Résultat de combinaison globale

Les résultats obtenus sont interprétés dans l'histogramme suivant (figure3.14), affichant les métriques d'évaluation correspondantes pour chaque seuil défini .

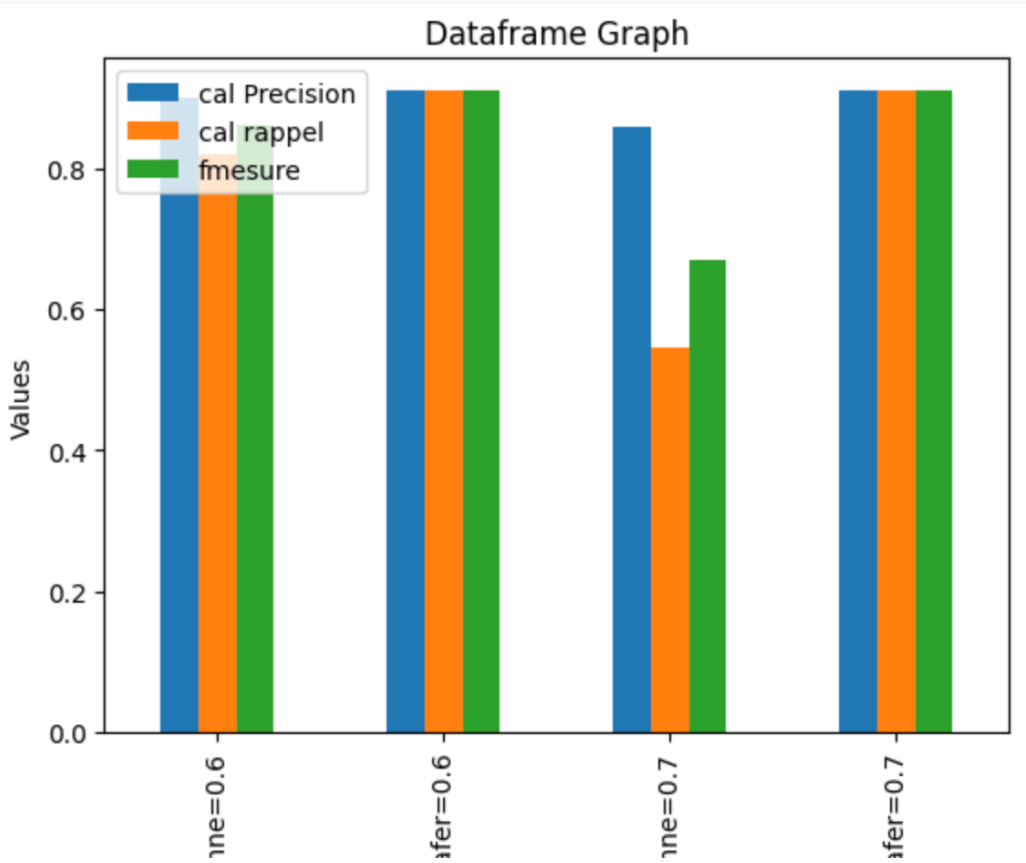

FIGURE 3.14 – Histogramme de résultats

Interprétation de résultats :

A travers les tableaux et l'histogramme présentés précédemment, nous pouvons clairement constater :

— La valeur de rappel obtenu par la combinaison de Dempster-Shafer est de 0.90, ce qui signifie que les résultats sont à 90% complets.

— En terme de précision, notre système confirme son niveau de performance obtenant une valeur de 0.90.

— Concernant la F-mesure elle obtient un résultat de 0.91 signifie que 91% des réponses pertinents, le taux de F-mesure est élevé grâce aux taux élevés de Précision et Rappel.

Discussion des résultats globaux :

A travers les éxperiences présentées précédemment on pourra conclure que la théorie de Dempster-Shafer est une approche plus avancée que la simple moyenne pour combiner des sources d'information incertaines.

Elle prend en compte l'incertitude et les degrés de croyance associés à chaque source d'information, ainsi que les poids qui indiquent leur crédibilité respective. Contrairement à la moyenne qui attribue le même poids à toutes les sources. Cette méthode permet d'obtenir des résultats plus fiables et précis.

Donc, la théorie de Dempster-Shafer offre une approche plus sophistiquée pour combiner les informations, ce qui en fait un choix préférable par rapport à la simple moyenne.

3.4 Conclusion

Dans ce chapitre, nous avons exposé les outils et l'environnement utilisés lors du développement de notre système, ainsi que l'interface graphique mise en place. Nous avons également introduit les diverses mesures d'évaluation de notre système, telles que le Rappel, la Précision et la F-mesure, et conclu le chapitre par une analyse interprétative des résultats obtenus.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Le Web de données permet de rendre disponibles des données structurées et non structurées sur Internet, en évitant de les isoler dans des silos de données indépendants. Au contraire, il favorise leur interconnexion pour former un réseau d'informations global. Les données liées sont conçues pour être partagées et interconnectées sur le Web en utilisant les principes des données liées. Elles sont représentées de manière lisible par les machines, ce qui permet de créer un espace de données unifié et cohérent.

Ce mémoire nous a offert une opportunité précieuse d'approfondir nos connaissances dans le domaine de l'ingénierie des connaissances, en se concentrant plus particulièrement sur le Web de données liées.

Dans le premier chapitre de ce travail, nous avons exploré les concepts du Web de données et des données liées, en mettant l'accent sur leur caractéristiques, ensuite nous avons abordé les différentes techniques de processus de découverte de liens. Ce processus représente l'objectif principal de notre travail de recherche. Nous avons étudié différentes techniques de similarité a fin de les combiner par les méthodes de théorie de fonction de croyance.

Dans le deuxième chapitre, nous avons proposé une solution innovante pour répondre à notre objectif de découverte des liens entre différentes données. Notre approche s'appuie sur la combinaison de la théorie de fonction de croyance avec des mesures de similarité syntaxiques, structurelles et extensionnelles. Cette approche combinatoire nous a permis d'obtenir des résultats prometteurs dans la découverte des liens, en prenant en compte les différentes caractéristiques des données.

Sur la base de cette approche, nous avons mis en œuvre notre système de découverte des liens afin d'évaluer notre travail et de démontrer l'efficacité de notre solution. Nous avons effectué plusieurs tests sur divers ensembles de données pour évaluer la qualité de notre solution. Ces tests ont été réalisé en utilisant des mesures d'évaluation telles que le "Rappel", la "Précision" et la "F-mesure". Les résultats obtenus nous ont permis de qualifier notre travail et de démontrer l'efficacité de notre solution dans la production de correspondances de haute qualité.

Enfin, on a eu quelques perspectives tel que l'utilisation d'autres mesures de similarité afin d'améliorer la précision du système et tester sur des datasets volumineux.

Bibliographie

[1] OWL 2 web ontology language : RDF-Based semantics. In M. Schneider, editor, *W3C Recommendation*. 2012.

[2] Y. Atig. *Découverte et réparation des alignements d'ontologies dans le web des données liées*.

[3] T. L. Bach. *Construction d'un web semantique multi-points de vues*. Ecole des Mines a Sophia Antipolis, France, 2006.

[4] J. Baget, E. Canaud, J. Euzenat, and M.-S. Hacid. *Les langages du Web Sémantique, série de la Revue Information - Interaction - Intelligence (I3)*, volume 4. Cépaduès, Toulouse, 2004.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semant. Web Inf. Syst.*, 5(3) :1–22, July 2009.

[6] A. Bosca, M. Casu, M. Dragoni, and C. Di Francescomarino. Using semantic and domain-based information in clir systems. In *The Semantic Web : Trends and Challenges : 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings 11*, pages 240–254. Springer, 2014.

[7] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing ontologies. *Web Semant.*, 1(4) :325–343, Oct. 2004.

[8] camillejourdain. Schémas : L'évolution du web.

[9] D. Collarana, M. Galkin, I. Traverso-Ribón, M.-E. Vidal, C. Lange, and S. Auer. Minte : semantically integrating rdf graphs. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*, pages 1–11, 2017.

[10] R. G. Dan Brickley.

[11] A. P. Dempster. *Dempster : Upper and lower probabilities induced by multivalued mapping*, volume 38. AMS, 1967.

[12] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 57–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[13] D. Dubois and H. Prade. Belief functions in information theory : An overview. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 30–43. Springer, 1990.

[14] J. Euzenat, P. Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

[15] S. A. Farsani. *EVOLUTION OF THE WORLD WIDE WEB : FROM*. 2012.

[16] C. Fellbaum. *Theory and applications of ontology : computer applications*.

[17] A. L. Gentile, D. Gruhl, P. Ristoski, and S. Welch. Information extraction in editorial setting. a tale of pdfs. In *The Semantic Web : ESWC 2019 Satellite Events : ESWC 2019 Satellite Events, Portorož, Slovenia, June 2–6, 2019, Revised Selected Papers 16*, pages 69–74. Springer, 2019.

[18] B. C. Grau, Z. Dragisic, K. Eckert, J. Euzenat, A. Ferrara, R. Granada, V. Ivanova, E. Jimenez-Ruiz, A. O. Kempf, and P. Lambrix. Results of the ontology alignment evaluation initiative 2013. In *OM : Ontology Matching*, pages 61–100. 2013.

[19] G. P. Guarino, N. *Ontologies and Knowledge Bases*. 1995.

[20] T. Heath and C. Bizer. *Linked data : Evolving the web into a global data space*. 2011.

[21] T. Heath and C. Bizer. *Linked Data : Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web*. Morgan & Claypool Publishers, 2011.

[22] Jridi. Proposition d'une nouvelle methode dalignement dontologies distribue, 2014.

[23] B. Khalid. Sélection des services web à base d'algorithmes génétiques multiobjectifs, 2012.

[24] P. Kra. A venn diagram of uniform resource identifier scheme categories.

[25] X. Lacot. *Introduction à owl, un langage xml d'ontologies web*. 2005.

[26] L. Lassila and R. R. Swick. Resource description framework (rdf) model and syntax specification. 3, 1999.

[27] McKinney, Wes et al. Pandas : Powerful data analysis toolkit, 2022.

[28] H. Melhem. *Usages et applications du web sémantique en bibliothèques numériques*. Doctor's thesis, université Grénoble Alpes.

[29] M. Mountantonakis and Y. Tzitzikas. Large-scale semantic integration of linked data : A survey. *ACM Computing Surveys (CSUR)*, 52(5) :1–40, 2019.

[30] A.-C. N. Ngomo and S. Auer. Limes—a time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[31] A.-C. Ngonga Ngomo, M. A. Sherif, K. Georgala, M. M. Hassan, K. Dreßler, K. Lyko, D. Obraczka, and T. Soru. Limes : a framework for link discovery on the semantic web. *KI-Künstliche Intelligenz*, pages 1–11, 2021.

[32] A. Nikolov, M. d'Aquin, and E. Motta. Unsupervised learning of link discovery configuration. In *ESWC*, pages 119–133, 2012.

[33] I. Nishanbaev, E. Champion, and D. A. McMeekin. A comparative evaluation of geospatial semantic web frameworks for cultural heritage. *Heritage*, 3(3) :875–890, 2020.

[34] K. O'Hara, T. Berners-Lee, W. Hall, and N. Shadbolt. 4.3 use of the semantic web in e-research. In *World Wide Research*, pages 130–134. The MIT Press, May 2010.

[35] K. O'Hara, T. Berners-Lee, W. Hall, and N. Shadbolt. 4.3 use of the semantic web in e-research. In *World Wide Research*, pages 130–134. The MIT Press, May 2010.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn : Machine learning in python. Jan. 2012.

[37] E. L. Rissland. Ai and similarity. *IEEE Intelligent Systems*, 21(03) :39–49, 2006.

[38] G. Shafer. *A mathematical theory of evidence A mathematical theory of evidence*. Princeton University Press, Princeton, NJ, June 2020.

[39] P. Smets. The combination of evidence in the transferable belief model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5) :447–458, May 1990.

[40] P. Smets. Belief functions : The disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 24(2-3) :167–234, 2000.

[41] F. M. Suchanek, S. Abiteboul, and P. Senellart. Probabilistic alignment of relations, instances, and schema. *arXiv preprint arXiv :1111.7164*, 2011. Paris.

[42] Twitter, Inc., Otto, Mark, Maddox, Jacob, Wenzel, Jacob, Strobel, Alan, and Thiessen, Steve. *Bootstrap : The most popular HTML, CSS, and JS framework*. Bootstrap, 2022.

[43] G. van Rossum and F. L. Drake Jr. *Python tutorial*, volume 620. Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, 1995.

[44] Z. Wu and M. Palmer. Verb semantics and lexical selection". In *Proceedings of the 32nd Annual In Proceedings of the 32nd Annual meeting of the Associations for Computational Linguistics*.

[45] M. Ziani, D. Boulanger, and G. Talens. Systeme d'aide a l'alignement d'ontologies metier-application au domaine geotechnique. In *Congres INFORSID 2010*, pages p345–360, 2010.