

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET  
POPULAIRE MINISTÈRE DE L'ENSEIGNEMENT  
SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE**

---

**UNIVERSITE SAAD DAHLEB DE BLIDA**

**Faculté des sciences**

Département d'informatique



**MEMOIRE DE MASTER**

**En Informatique**

Option : Ingénierie du Logiciel

---

**THÈME : Découverte sémantique des  
liens dans les données liées**

---

**Réalisé par**

BRAHAM Camelia

DAOUDI Ouahiba

**Promotrice** : Mme. FAREH

**Encadreur** : Mr. BOUDISSA

**Président** : Mme. LAHIANI

**Examineur** : Mr. HAMOUDA

**Soutenu le** : 07/07/2022

---

## Remerciements

*Nous tenons à remercier tout d'abord ALLAH le Tout-Puissant de nous avoir donné la santé, le courage et la volonté pour réaliser ce mémoire.*

*Nous remercions et témoignons notre reconnaissance à notre promotrice : Mme. FAREH et à notre encadreur : Mr. Djamel BOUDISSA, de leurs orientations, leurs précieux conseils, leurs soutiens constants et leurs aides qui nous ont permis de mener à bien ce travail, et sans oublier Mr. Oussama HAMEL, et Mr. Mahfoud BOUKERT qui nous a guidé dans notre travail et nous a aidé à trouver des solutions pour avancer.*

*Nous tenons également à exprimer une reconnaissance aux membres du jury pour avoir accepté d'examiner et de porter leur jugement sur notre travail.*

*Nos vifs remerciements à nos familles pour nous avoir aidées à surmonter tous les obstacles et forger sur dent à travers les difficultés vécues tout au long de cette période de travail.*

*Enfin nous tenons à remercier nos proches amis qui nous ont vivement soutenues et encouragées au cours de la réalisation de ce modeste travail.*

*Nos remerciements à toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.*

---

## Dédicaces

*Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance, c'est tout simplement que :*

*Je dédie ce mémoire à*

*A Ma tendre Mère Fahima : Tu représentes pour moi la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.*

*A Mon très cher Père Abderrahmane : Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien-être.*

*Ce travail et le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation le long de ces années.*

*A ma chère tante Nour malika BOUKERCHA.*

*A mon cher frère Nassim.*

*A mon oncle Ali Rahim.*

*A ma binôme Ouahiba.*

*A monsieur Djamel BOUDISSA : Cette humble dédicace ne saurait exprimer mon grand respect et ma profonde estime, que dieu vous procure bonne santé et long vie.*

*A monsieur Mahfoud BOUKERT : qui ne cessé pas de m'encourager et me conseillée. Cette humble dédicace ne saurait exprimer mon grand respect et mon profonde estime.*

*A tous mes enseignants depuis mes premières années d'études.*

**Camelia BRAHAM**

---

## Dédicaces

*Je dédie ce modeste travail plus particulièrement et chaleureusement dire MERCI à mon Grand-Père 'AMAR' qui a été ma force surtout durant tout mon parcours, merci pour tes prières et surtout merci pour ton amour, et à la femme la plus chère à mes yeux : ma grande mère que J'aime beaucoup, que dieu leur donne une longue vie.*

*A Qui a été avec moi dans les difficultés des temps, le temps de ma joie et de ma douleur. Grâce à elle et à ses encouragements je suis arrivée à ce stade. A ma chère Mère, la femme la plus adorable de ma vie.*

*À Mon très cher Père, pour tous ses conseils et pour toute la confiance qu'il a mise en moi et pour son dévouement pour mon bonheur.*

*A ma chère soeur : Mounia.*

*A mon cher frère : Oualid chames-dinne.*

*A toute ma famille, ainsi qu'à ma chère tante 'KARIMA'.*

*A ma binôme Camelia et sa famille surtout sa chère tante 'Malika'.*

*A monsieur Djamel BOUDISSA, et monsieur BOUKERT pour leur pas encouragements et leur conseillées. Cette humble dédicace ne saurait exprimer mon grand respect et ma profonde estime.*

*A toute personne chère pour moi qu'elle m'a donné le courage et le soutien de près ou de loin.*

**Ouahiba DAUDI**

---

## Résumé

Un nombre croissant de données liées sont publiées sur le web, les données du web sont connues par leur grande hétérogénéité et leur volume croissant. La découverte des liens entre les ressources du web consiste à découvrir la correspondance sémantique entre les éléments similaires du web de données. Cependant, le nombre de plus en plus croissant des données disponible sur le web, nécessite des outils automatiques de découverte des liens. Toutefois, l'identification automatique des correspondances sémantiques entre les données est très difficile en termes de qualité des liens extraits.

Pour contribuer à résoudre ce problème, nous proposons une solution pour effectuer la découverte des liens entre deux datasets des données liées. Après l'extraction des différentes ressources des dataset, le processus de découverte des liens est lancé pour trouver les ressources équivalentes. Un filtrage est réalisé pour construire des catégories des données, afin de réduire l'ensemble de recherche des données similaires. Par la suite, les mesures syntaxique, lexicale, extensionnelle et structurelle sont combinées, ceci afin de définir une mesure de similarité globale sémantique calculée en combinant ces mesures de similarités. Une validation est réalisée sur les liens trouvés pour montrer l'efficace du système.

### **Mots clés :**

RDF (Ressource Description Framework), web données, données liées, découverte des liens, mesure de similarité, sémantique.

---

## Abstract

A large amount of linked data is published on the web, web data is known by its great heterogeneity and increasing volume. The discovery of links between web resources consists in discovery of the semantic correspondence between similar elements of the web of data. Despite that, the increasing amount of data available on the web requires discovering the automatic link tools. However, the difficulty of automatic identification of semantic correspondences between data links in the quality of the extracted links.

To help solve this problem, we suggest a solution to perform link discovery between two datasets of linked data. After extracting the different resources from the datasets, the link discovery process is started to find the equivalent resources. Filtering is performed to build categories of the data, in order to reduce the search set of similar data. Then, the syntactic, lexical, extensional and structural measures are combined to define a global semantic similarity measure computed by combining these similarity measures. A validation is performed on the links found to show the efficiency of the system.

**Keywords :**

RDF (Resource Description Framework), web data, linked data, link discovery, similarity measurement, semantics.

## ملخص

يتم نشر كمية كبيرة من البيانات المرتبطة على الويب، وهذه البيانات معروفة بعدم تجانسها الكبير وزيادة حجمها، اكتشاف الروابط بين موارد الويب يكمن في اكتشاف المراسلات الدلالية بين العناصر المتشابهة لشبكة البيانات. بالرغم من ان الكمية المتزايدة من البيانات المتاحة على الويب تتطلب اكتشاف الارتباط التلقائي للأدوات. ومع ذلك، فان صعوبة التجديد التلقائي للمراسلات الدلالية في جودة الروابط المستخرجة. للمساعدة في حل هذه المشكلة، نقترح حلاً لإجراء اكتشاف الارتباط بين مجموعات البيانات المرتبطة. بعد استخراج الموارد المختلفة من مجموعات البيانات، نبدأ عملية الاكتشاف للعثور على الموارد المكافئة. يتم إجراء التصفية لبناء فئات البيانات المتشابهة، النحوية والمعجمية. يتم الجمع بين المقاييس البنائية والهيكلية لتحديد مقياس التشابه الدلالي العالمي محسوبة من خلال الجمع بين تدابير التشابه. يتم إجراء التحقق من صحة الروابط الموجودة لإظهار كفاءة النظام.

### الكلمات الدلالية

، اكتشاف الارتباط ، قياس التشابه ، الدلالات. ، بيانات الويب ، البيانات المرتبطة ، RDF (Resource Description Framework)

---

# Table des matières

---

<b>Table des figures</b>	<b>11</b>
<b>Liste des tableaux</b>	<b>13</b>
<b>Introduction Générale</b>	<b>15</b>
<b>1 WEB DE DONNÉES ET DONNÉES LIÉES</b>	<b>17</b>
1.1 Introduction . . . . .	17
1.2 Web de données . . . . .	17
1.2.1 Définition . . . . .	17
1.2.2 Historique . . . . .	17
1.2.3 Objectifs du web de données . . . . .	19
1.2.4 Différents types de données . . . . .	19
1.3 Données liées . . . . .	21
1.3.1 Définition . . . . .	21
1.3.2 Architecture du web . . . . .	21
1.3.3 Principes de données liées . . . . .	27
1.3.4 Types de liens des données liées . . . . .	28
1.3.5 Domaines d'applications des données liées . . . . .	28
1.4 Ontologie . . . . .	29
1.4.1 Définition . . . . .	29
1.4.2 Composants d'ontologie . . . . .	30
1.4.3 Découverte de correspondance . . . . .	31
1.4.4 Classification des conflits de découverte des liens de données . . . . .	33
1.5 Conclusion . . . . .	33
<b>2 MÉTHODES EXISTANTES POUR LA DÉCOUVERTE DE LIENS DANS LES DONNÉES LIÉES</b>	<b>35</b>
2.1 Introduction . . . . .	35



2.2	Découverte des liens . . . . .	35
2.3	La similarité . . . . .	36
2.4	Mesure de similarité . . . . .	36
2.4.1	Mesures simples . . . . .	37
2.4.2	Mesure combinée . . . . .	42
2.5	Les méthodes de découverte des liens dans le contexte du web de données . . .	42
2.5.1	RDF-AI . . . . .	42
2.5.2	SILK . . . . .	42
2.5.3	KNOFUSS . . . . .	43
2.5.4	LIMES . . . . .	43
2.5.5	Travail de [Yuliu et al, 2015] . . . . .	43
2.5.6	STS [John et al, 2018] . . . . .	44
2.5.7	Travail de [Armando et al, 2022] . . . . .	44
2.6	La comparaison des méthodes . . . . .	44
2.6.1	Critères de comparaison . . . . .	45
2.6.2	Analyse . . . . .	48
2.7	Conclusion . . . . .	48
<b>3</b>	<b>CONCEPTION DU SYSTÈME</b>	<b>49</b>
3.1	Introduction . . . . .	49
3.2	Caractéristiques de notre système . . . . .	49
3.3	Schéma global du système . . . . .	50
3.4	Description de la solution proposée . . . . .	52
3.4.1	Phase 1 : Pré-traitement . . . . .	53
3.4.1.1	Étape 1 : Téléchargement des données . . . . .	53
3.4.1.2	Étape 2 : Vérification de la validité du dataset . . . . .	55
3.4.1.3	Étape 3 : Nettoyage . . . . .	58
3.4.1.4	Étape 4 : Normalisation . . . . .	58
3.4.2	Phase 2 : Découverte sémantique . . . . .	59
3.4.2.1	Étape 1 : Catégorisation par domaine . . . . .	59
3.4.2.2	Étape 2 : Détermination de l'équivalence entre les domaines . . . . .	60
3.4.2.3	Étape 3 : Découverte de lien entre les triplets . . . . .	61
3.4.3	Phase 3 : Post-découverte . . . . .	65
3.4.3.1	Étape 1 : Combinaison des mesures . . . . .	65
3.4.3.2	Étape 2 : Filtrage . . . . .	65
3.4.3.3	Étape 3 : Génération du fichier des liens . . . . .	67
3.5	Conclusion . . . . .	67
<b>4</b>	<b>IMPLÉMENTATION ET TEST DU SYSTÈME</b>	<b>69</b>
4.1	Introduction . . . . .	69

4.2	Environnement de développement . . . . .	69
4.2.1	Matériel utilisé . . . . .	69
4.2.2	Langage utilisé . . . . .	70
4.2.2.1	Python . . . . .	70
4.2.2.2	Paquets utilisés dans Python . . . . .	70
4.3	Présentation de l'application . . . . .	71
4.4	Test du système . . . . .	75
4.4.1	Résultats expérimentaux et discussion . . . . .	75
4.4.2	Mesures d'évaluation utilisées . . . . .	77
4.4.3	Interprétation des résultats . . . . .	79
4.5	Conclusion . . . . .	79
	<b>Conclusion et perspectives</b>	<b>81</b>
	<b>Bibliographie</b>	<b>83</b>

---

# Table des figures

---

1.1	Les phases d'évolution de web [1]. . . . .	18
1.2	Les couches du Web [29]. . . . .	21
1.3	Couches du web de données [29]. . . . .	22
1.4	La structure d'un triplet RDF [23] . . . . .	23
1.5	Exemple RDF [23] . . . . .	23
1.6	Formats sérialisables RDF [2]. . . . .	24
1.7	Exemple RDFS [15]. . . . .	25
1.8	Exemple d'une requête SPARQL [36]. . . . .	26
1.9	Exemple OWL [3]. . . . .	27
1.10	Triangle sémantique [14]. . . . .	30
1.11	Processus d'alignement d'ontologie [37]. . . . .	31
1.12	Exemple d'alignement entre deux données [37]. . . . .	31
1.13	Les trois dimensions de l'alignement [12]. . . . .	32
2.1	Mesures de calcul de similarités [11]. . . . .	37
3.1	Schéma Global. . . . .	51
3.2	Exemple des triplets YAGO. . . . .	54
3.3	Exemple de triplets DBpedia. . . . .	54
3.4	Exemple de nombre d'éléments qui existe au niveau du fichier d'équivalence et dans YAGO . . . . .	57
3.5	Exemple de nombre d'éléments qui n'existent pas au niveau de fichier d'équivalence et dans DBLP . . . . .	57
3.6	Normalisation de casse. . . . .	58
3.7	Suppressions des mots vides. . . . .	58
3.8	Domaines équivalents. . . . .	61
3.9	Triplets équivalents. . . . .	61
3.10	Mesures de similarités. . . . .	62
3.11	Le fichier des résultats RDF. . . . .	66

3.12 Le fichier RDF. . . . .	67
4.1 L'interface d'accueil. . . . .	71
4.2 Chargement des datasets. . . . .	72
4.3 Normalisation. . . . .	72
4.4 Catégorisation par domaine. . . . .	73
4.5 Les mêmes domaines. . . . .	73
4.6 Résultats des mesures de similarités. . . . .	74
4.7 Évaluation. . . . .	74
4.8 Code de calcul de la similarité globale. . . . .	75
4.9 Mesure de performance . . . . .	79

---

# Liste des tableaux

---

2.1	Comparaison des techniques de découverte des liens . . . . .	47
3.1	Exemple des triplets du fichier de correspondance. . . . .	54
3.2	La catégorisation par domaines du dataset yago. . . . .	60
3.3	La catégorisation par domaines du dataset DBpedia. . . . .	60
4.1	Les résultats des similarités obtenus entre les propriétés des deux datasets. . . . .	76
4.2	Les résultats des similarités obtenus entre les objets des deux datasets. . . . .	76
4.3	Similarité globale. . . . .	77



---

# Introduction Générale

---

## Contexte de travail

Au niveau du web, un nouvel essor s'est récemment développé autour du web sémantique et des données ouvertes et liées. Le web sémantique tend à promouvoir l'utilisation de formats de données qui facilitent le partage, la réutilisation, le traitement par des machines et qui permettent de produire de nouvelles connaissances grâce au raisonnement. Les données liées sont une méthode de publication des données qui favorisent le traitement automatisé et l'établissement de relations vers d'autres sources de données.

Le web de données permet de publier des données structurées et non structurées sur le web, non pas sous la forme de silos de données isolés les uns des autres, mais en les reliant pour constituer un réseau d'informations global. Les données liées visent à partager et à interconnecter des données structurées sur le web selon les principes des données liées, sous forme d'une représentation lisible par la machine pour former un seul espace de données global. L'intérêt de construire un jeu de données liées, c'est lorsqu'elles entretiennent des liens avec d'autres données, ce qui permet d'étoffer les descriptions.

## Problématique

Dans le contexte du web de données, il est important d'établir des liens pertinents entre les données de différentes sources. Le problème d'établissement des liens entre les ensembles de données est lié au problème de matching d'ontologies utilisées dans le web sémantique.

Le contexte des données liées, et la publication de grands ensembles de données liées au besoin de stratégies efficaces, Vu que les sources des données sont sémantiquement hétérogènes, en plus du volume important de ces sources. Dans ce contexte, le problème majeur est que les méthodes classiques de matching d'ontologies qui existent déjà ne répondent pas aux besoins de précision. Donc le besoin de trouver une solution d'adaptation ou d'amélioration de ces méthodes existe fortement.

En plus, les approches existantes ont besoin d'améliorer leurs efficacités. L'amélioration de l'efficacité permet aux outils de découverte des liens de générer des mappings de haute

qualité. Par conséquent, les résultats doivent être précis. Un outil de découverte des liens devrait également générer autant que possible des liens pour assurer la complétude.

## Objectif du travail

L'objectif de ce projet consiste à concevoir et réaliser un système de découverte des liens dans le contexte du web de données, en effet, la découverte des liens dans des ensembles de données liées est une tâche très importante pour la liaison des données, la recherche de connaissances et l'interrogation des sources de données liées ce qui rend l'exploitation de ces sources très efficace.

La découverte des liens doit intégrer l'aspect sémantique des ressources du web, en traitant les différents types d'hétérogénéité, afin d'avoir des liens de bonne qualité.

## Organisation du mémoire

Pour mener à bien notre mémoire, nous avons organisé notre travail en quatre chapitres.

- **Chapitre 1 : Web de données et données liées**

Dans ce chapitre, nous avons abordé une étude sur le web de données, son historique et les différents types de données. Nous avons parlé des données liées, en présentant son architecture et les principes de données liées, nous avons mentionné aussi la notion des ontologies, ses composés et le processus d'alignement.

- **Chapitre 2 : Méthodes existantes pour la découverte de liens dans les données liées**

Ce chapitre commence par la définition de la découverte des liens, nous présentons par la suite les différentes mesures de similarité utilisées, puis nous enchaînons avec un état de l'art sur les méthodes de découverte des liens et une comparaison entre ces dernières.

- **Chapitre 3 : Conception du système**

Dans ce chapitre, nous avons présenté notre solution proposée pour le problème de découverte des liens, avec une description détaillée des différentes phases de découverte des liens.

- **Chapitre 4 : Implémentation et test du système**

Dans ce dernier chapitre, qu'il 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.

La conclusion de ce mémoire synthétise les principales étapes de construction de notre système, et dégage quelques perspectives de ce travail.



## *Chapitre 1*

---

# **WEB DE DONNÉES ET DONNÉES LIÉES**

---

## **1.1 Introduction**

Les évolutions technologiques du web ont permis de passer d'un web de documents (navigation hypertextuelle) qui concentre sur la publication, la recherches, la navigation. Passons au web de données qui nous permet directement du publie et de liées des données sur le web. Ce passage a pu se faire grâce à l'avènement de différents standards issus du web sémantique comme RDF, OWL, SPARQL et URI.

Ce chapitre est organisé comme suit : nous débutons avec un petit survol de l'histoire du Web, sa définition et quelques concepts de base de ce dernier. Par la suite nous allons parler des données liées et les différentes couches du web de données, puis nous énonçons quelques principes fondamentaux des ontologies.

## **1.2 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, non pas sous forme de silos de données isolés les uns des autres, mais en les reliant entre elles pour constituer un réseau global d'informations [40].

### **1.2.2 Historique**

Le web est né au CERN, le centre européen de recherche nucléaire à la fin des années 1980 par **Tin Berners-Lee**.

D'après toutes ces années le web il a connu une évolution incroyable (Figure 1.1).

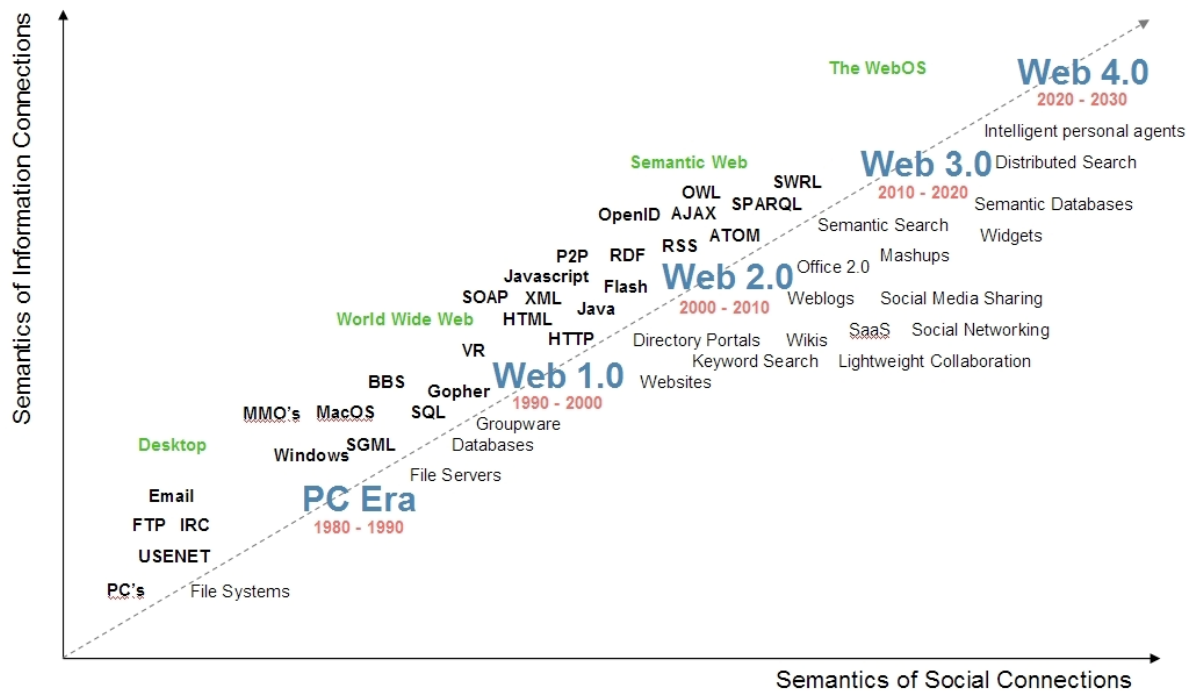


FIGURE 1.1 – Les phases d'évolution de web [1].

### 1. LE Web 1.0

Le web de documents est un réseau statique basé sur un ensemble de standards simples, URI (Uniform Resource Identifier) est comme un mécanisme d'identification unique au monde, HTTP (HyperText Transfer) comme mécanisme d'accès général, tel que HTML (HyperText Transfer Markup Language) comme format de contenu largement utilisé.

HTML est conçu pour écrire des pages Web riches en texte et créer des liens hypertexte entre des documents Web pouvant résider sur divers serveurs Web.

Le web documentaire permet le transfert d'informations de manière unidirectionnelle (c'est-à-dire du développeur vers l'internaute). Les applications les plus couramment utilisées dans le Web 1.0 sont : les portails de contenu, la messagerie électronique et certains sites de commerce électronique [1].

### 2. Le Web 2.0

Le web social est facilité le partage et l'échange d'informations et des contenus (textes, images, vidéos, etc.).

Ce web a commencé avec l'apparition des scripts, qui permettent aux personnes qui ne comprenant aucun langage de programmation de gérer les sites. Ces scripts offrent la possibilité d'insérer des modules dans la page (heure et date, listes de diffusion etc.) [1].

### 3. Le Web 3.0

Web sémantique est un espace de données unique mondialement distribué, il a été conçu pour être interprétable par les humains et les machines grâce à la mise en place des

liaisons sémantiques entre les données qui se trouvent à l'intérieur des documents. Le Web sémantique permet d'organiser et de donner du sens aux données en fonction du contexte et des besoins des utilisateurs.

De plus, il ajoute des informations cachées (métadonnées) destinées à être utilisées par des moteurs de recherches, des applications, etc. Ces informations sont présentées à l'aide du formalisme RDF (basé sur le langage XML) qui permet de structurer les données sous forme de triplets < sujet, prédicat, objet > [1].

### 4. Le Web 4.0

Web intelligent vise à immerger l'individu dans un environnement web très solide qu'il fonctionnera grâce à des agents intelligents. C'est un terrain d'expérimentation qu'il n'est pas encore exploré [1].

### 1.2.3 Objectifs du web de données

Les principes objectifs du web de données est de permettre aux utilisateurs d'utiliser la totalité du potentiel du web. Ainsi, ils pourront trouver, partager et combiner des informations plus facilement. Aujourd'hui tout le monde est capable d'utiliser des forums, d'utiliser des réseaux sociaux, de faire des recherches ou même d'acheter différents produits. Néanmoins, il serait mieux que la machine fasse tout ceci à la place de l'homme, car actuellement, les machines ont besoin de l'homme pour effectuer ces tâches. La raison principale est que les pages web actuelles sont conçues pour être lisibles par des être humaines et non par des machines. Le web de données a donc comme principal objectif que ces mêmes machines puissent réaliser seules toutes les tâches fastidieuses comme la recherche ou l'association d'informations et d'agir sur le web lui-même [8].

### 1.2.4 Différents types de données

Il existe plusieurs types de donnée, parmi ces types on trouve [9] :

#### 1. Données de recherches :

Les données de recherche font référence aux données générées à l'intérieur d'un projet de recherche, en milieu académique, gouvernemental. Les données de recherches sont des mégas données. Cependant, elles peuvent aussi être des petites données.

On peut donc y retrouver des ensembles comme les suivants :

- Données statistiques générées automatiquement par des appareils de mesure ou par ordinateur.
- Ensemble de réponses courtes ou fixes à des questions.

## **2. Données ouvertes :**

Le terme "données ouvertes" fait référence à des données qu'un organisme met à la disposition de tous sous forme de fichiers numériques afin de permettre leur réutilisation. Les données sont ouvertes (Open data en anglais) quand elles sont non seulement disponibles, mais aussi en formats informatiques qui peuvent être traités aussi par ordinateur que par les humains.

Quelque exemple de données ouvertes :

- Statistiques de fréquentations d'un évènement.
- Données météorologiques nationales pour une année entière.

## 1.3 Données liées

### 1.3.1 Définition

"Les données liées sont un ensemble de principe de conception pour le partage de données lisibles par machine sur le web pour une utilisation par les administration publiques, les entreprises et les citoyens " [17].

### 1.3.2 Architecture du web

#### 1. Les couches du web

Les standards du web de données sont plusieurs ordres. La première partie de la pile garantit une **identification** cohérente des données (URI). La deuxième partie de la pile concerne la **représentation** des données, cette représentation elle va faire appel à un standard qu'on appelle RDF. Une fois les données sont publiées on va les interroger en utilisant des **requêtes** SPARQL. Les autres couches se chargent de valider les résultats au sein d'interface utilisateurs.

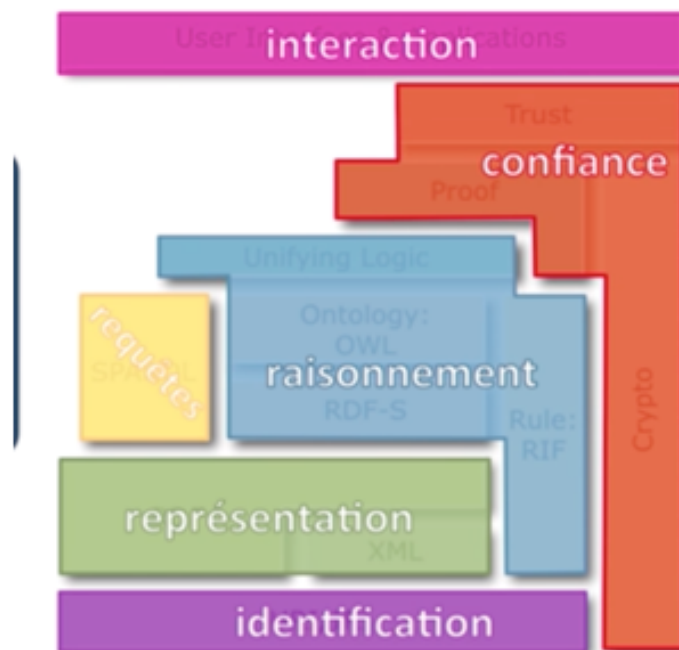


FIGURE 1.2 – Les couches du Web [29].

## 2. Les couche du web de données

La figure (1.3) représente la formalisation graphique des différentes couches technologiques composant le Web de données.

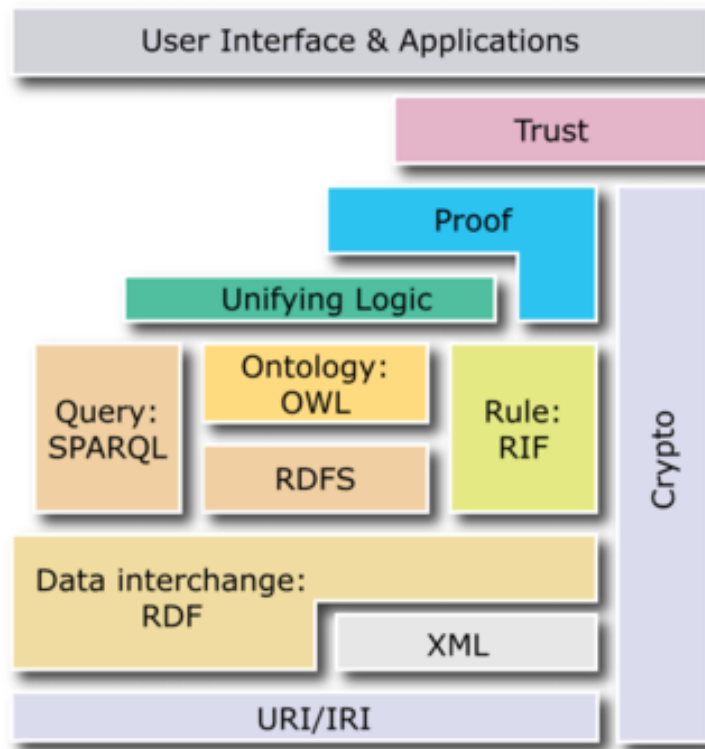


FIGURE 1.3 – Couches du web de données [29].

### 1. URI/IRI

URI est un protocole simple qui permet d'identifier une ressource abstraite ou physique sur le web d'une manière unique et globale, dans tous les hyperliens sur le web exprimé sous forme d'URI.

Dans le cas du web de données, URI est une séquence de caractères avec une syntaxe qui permet d'identifier toute ressource utilisée dans le cadre d'une application web de données [28].

### 2. XML

C'est un métalangage proposé par le W3C pour permettre la représentation des documents texte, c'est un langage de balisage comme HTML. XML se contente d'imposer une syntaxe destinée à mettre en évidence la structure de l'information inscrit dans un fichier.

Ce langage a été développé pour faciliter l'échange, le partage et la publication de données à travers l'ensemble du réseau.

XML est aujourd'hui un format d'échange de données et de documents et beaucoup répondu au contexte du web sémantique [7].

## 3. RDF

Resource Description Framework est une recommandation du W3C comme un modèle standard d'échange de donnée sur le web et un langage d'expression de graphe de données dirigé sous forme de triples < sujet, prédicat, objet > respectivement < ressource, propriété, valeur >.

Ce modèle est associé à une grammaire à base triple écrite en XML [23] :

- a) **Ressource (sujet) :** une ressource qui représente une information en utilisant une URI. Une ressource est une chose abstraite ou physique, elle représente une personne, une ville, un concept, etc.
- b) **Propriété (prédicat) :** est une URI qui représente le type de la relation qui existe entre le sujet et l'objet. Par exemple, le nom ou la date de naissance (dans le cas d'un littéral), ou le lieu de la naissance d'une personne (dans le cas d'une autre ressource).
- c) **Valeur (objet) :** l'objet peut être soit un littéral (rdf :literal) (comme une chaîne de caractères, une date ou un nombre ) ou une URI d'une autre ressource qui est reliée avec le sujet.

Un triplet RDF est un ensemble de trois informations liées < sujet, prédicat, objet >, comme l'indique la figure (1.4).

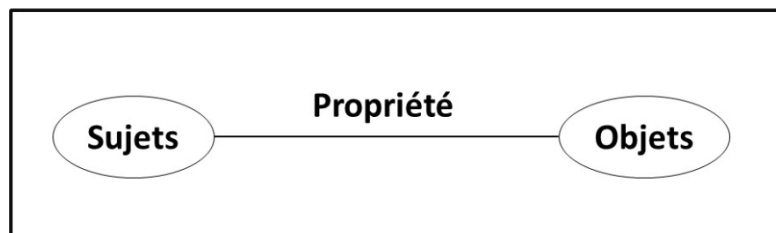


FIGURE 1.4 – La structure d'un triplet RDF [23]

Voici un exemple RDF :

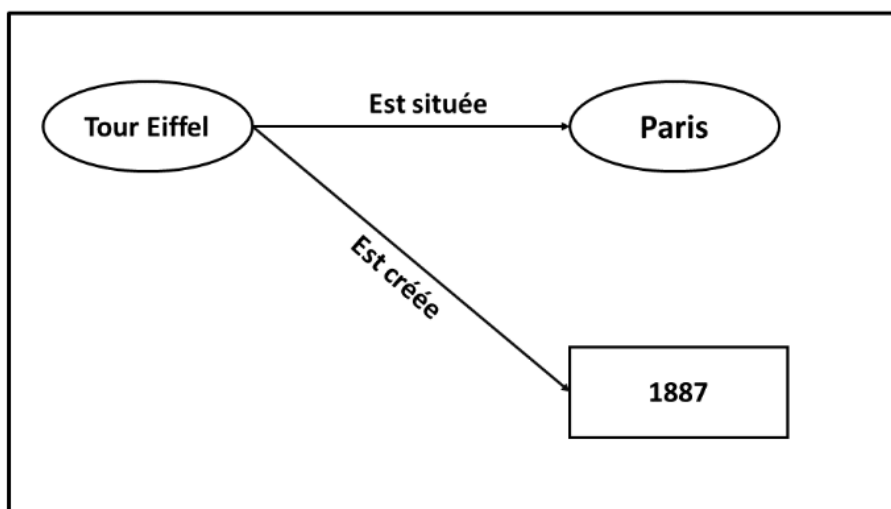


FIGURE 1.5 – Exemple RDF [23]

Un document RDF est un document qui encode un graphe RDF ou un ensemble de données RDF dans une syntaxe RDF concrète qui peut être soit en une sérialisation RDF/XML, RDFa, N-Triples, Turtle, JSON-LD, TriG, N-Quadsge.

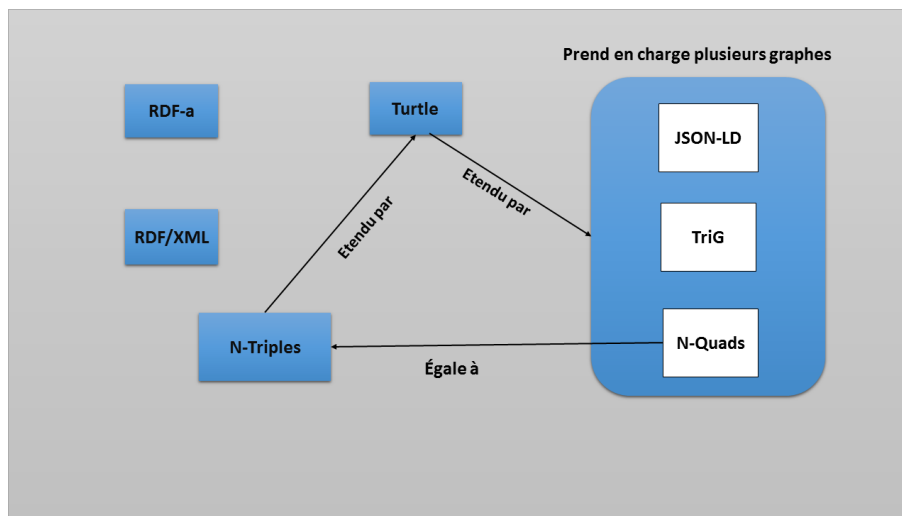


FIGURE 1.6 – Formats sérialisables RDF [2].

- **Bases de données RDF :**

- a) **RedLand**

Est un ensemble de bibliothèques de logiciels "C" gratuits qui fournissent un support pour RDF, c'est une bibliothèque modulaire, basées sur des objets et API pour manipuler des graphes RDF, triples, URI et Literals.

Elle prend en charge des syntaxes multiples pour la lecture et l'écriture de RDF sous forme de syntaxes RDF/XML, N-Triples et Turtle, RSS et Atom via la bibliothèque Raptor RDF Syntax.

Elle permet la consultation avec SPARQL et RDQL à l'aide de la bibliothèque de requêtes RDQ de Rasqal [2].

- b) **AllegroGraph**

Est une base de données Triple Store qui est conçu pour stocker des triples RDF moderne, à haute performance, il utilise efficacement la mémoire en combinaison avec le stockage sur disque. AllegroGraph prend en charge SPARQL, RDFS++ et le raisonnement Prolog de nombreuses applications clientes.

AllegroGraph est actuellement utilisé dans des projets commerciaux et un projet du département américain de la défense [2].

- c) **Neo4j**

Est une base de données qui permet de stocker des graphes, elle est extrêmement performante pour traiter les relations, elle possède un langage de requête puissant, qui permet d'interroger un graphe pour obtenir toutes sortes d'informations sur les noeuds, leurs liens et le contenu de ces derniers [2].



#### 4. RDFS

RDFS est un méta modèle recommandé par le W3C, permettant d'écrire la définition de schéma et modèle du domaine sémantique de la déclaration RDF, RDFS fournit un système de types pour les instructions RDF.

RDF Schéma définit des entités tel que (rdfs :class), (rdfs :subclass), (rdfs :subproperty), (rdfs :domain), et (rdfs :range), permettant de modéliser des classes et des propriétés avec une restriction du domaine, cependant il ne permet pas d'exprimer l'exclusion et la négation, et il limite l'axiomatisation aux restrictions, pour cela il est considéré comme un langage ontologique simple, la combinaison de RDF et RDF schéma est appelée RDF(S) [43].

Les principales caractéristiques de RDFS :

- a) **Rdfs : class** Permet de déclarer une ressource RDF comme une classe pour d'autres ressources, la définition de rdfs : class est récursive.
- b) **Rdfs : subclassof** Permet de définir des hiérarchies de classes.

RDFS précise la notion de propriété définie par RDF en permettant de donner un type ou une classe au sujet et à l'objet des triplets. Pour cela, RDFS ajoute les notions de "domain" et "range".

- a) **Rdfs : domain** Définit la classe des sujets liés à une propriété.
- b) **Rdfs : range** Définit la classe ou le type de données des valeurs de la propriété.

La figure (1.7) montre un exemple sur RDFS :

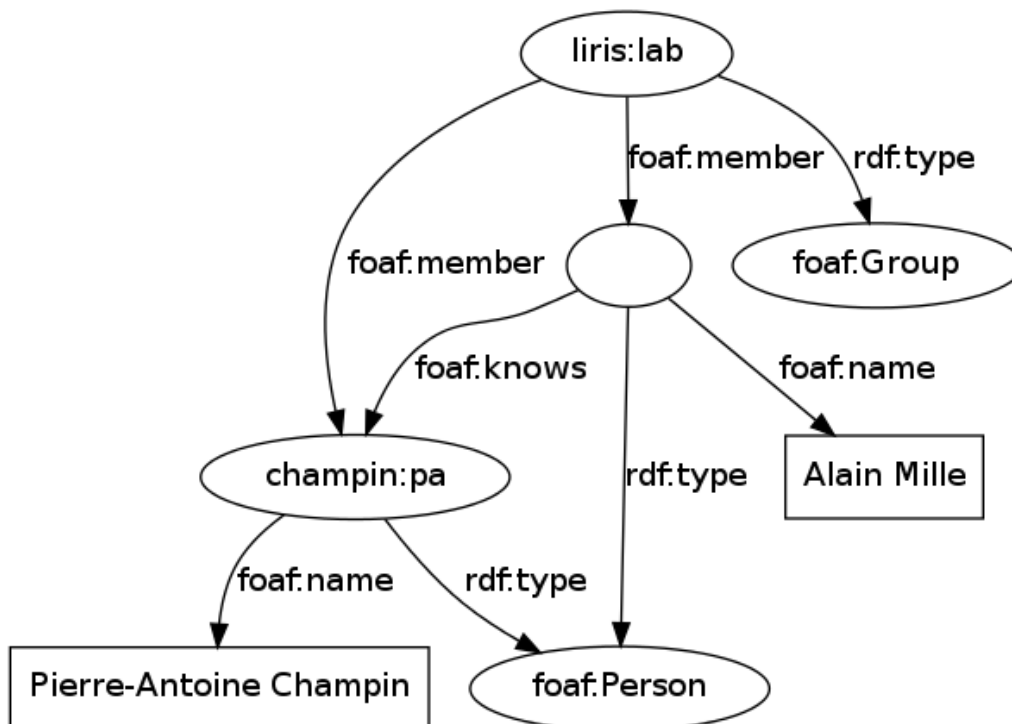


FIGURE 1.7 – Exemple RDFS [15].

## 5. SPARQL

En 2008, le groupe de travail DAWG (RDF data Acces Working Group) du W3C (Consortium World Wide Web) proposant le langage d'interrogation SPARQL qui est largement utilisé pour l'interrogation des données RDF.

SPARQL est un langage de requête et un protocole qui permet de recherche, d'ajouter de modifier ou de supprimer des données RDF.

Dans une requête interrogative, nous utilisons le mot clé SELECT pour extraire un sous graphe (ensemble de ressources) à partir d'un graphe RDF à vérifiant les contraintes qui sont définies dans la clause WHERE [36].

Le pseudo-code ci-dessous présente un exemple d'une requête SPARQL.

**Requête SPARQL**

```

PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntaxns#
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX foaf: <http://purl.org/dc/elements/1.1/>

SELECT DISTINCT ?nom ?image ?description.
WHERE {?personne rdf: type foaf: Person.
?personne foaf: name ?nom.
?image rdf: type foaf: image.
?personne foaf: img ?image.
?image dc: description ?description}

```

FIGURE 1.8 – Exemple d'une requête SPARQL [36].

## 6. OWL

OWL (Web Ontology Language) étend RDFS et mettent l'accent sur le soutien d'une inférence logique plus riche, c'est un successeur de deux autres langages OIL et DAML+OIL. OWL est disponible en plusieurs variantes avec une expressivité croissante (OWL Lite, OWL DL, OWL FULL) [16].

Avec OWL, nous pouvons définir :

- La subsomption entre classes et relations.
- Définition des concepts par énumération des instances.
- Définition des concepts par la combinaison des opérations : union, intersection et complément.
- Définition des contraintes en logique du premier ordre.

Ce nouveau langage est divisé en trois sous-langages [16] :

— **OWL Lite**

C'est le moins expressive des sous-langages d'OWL. Il est destiné aux cas de modélisation se limitant à une classification simple de concept et de contrainte.

L'avantage de ce langage est d'avoir une complexité formelle faible par rapport aux deux autres sous-langages d'OWL.

— **OWL DL**

C'est le sous-langage basé sur la logique de description d'où son nom OWL-DL (DL : Description Logic), qui offre une expressivité maximale en garantissant la complétude des raisonnements (calculabilité des inférences) et leur décidabilité (leur calcul se fait en une durée finie).

— **OWL Full**

Il est adressé aux utilisateurs qui cherchent un maximum d'expressivité, cependant, il ne possède pas les propriétés de complétude et de décidabilité.

Voici un exemple sur OWL :

```
<owl: Class rdf: about='#course'>
  <rdfs: subclassOf>
    <owl: Restriction>
      <owl:onProperty rdf: resource='#is TaughtBy' />
        <owl: minCardinality rdf: datatype='&xsd;nonNegativeInteger'>
          1
        </owl:minCardinality>
      </owl: Restriction>
    </rdfs: subclassOf>
  </owl: Class>
```

FIGURE 1.9 – Exemple OWL [3].

### 1.3.3 Principes de données liées

Les données liées se réfèrent à un ensemble de bonne pratique à mettre en oeuvre pour publier et lier les données structurées sur le web.

Parmi les principes des données liées sont [18] :

- Nommer des éléments des URI pour identifier non seulement des documents mais aussi des objets du monde réel cela peut être comme une extension des principes de web pour comprendre tous les objets.
- Fournir des informations nécessaires sous forme de standards (RDF, SPARQL) lors d'une recherche d'URI, ce principe est conseillé donne un modèle unique pour publier des données structurées.

- Inclus dans ces données des liens vers d'autres données du web qui permettant de découvrir d'autre élément, d'après l'utilisation des hyperliens afin de connecter toutes sortes d'élément et pas seulement les documents Web.

### 1.3.4 Types de liens des données liées

#### 1. Liens relationnels

Les liens relationnels permettent de relier des données à l'intérieur d'un ou de plusieurs ensembles de données liées, ce type de liens pointe vers des choses connexes à une connaissance dans d'autres sources de données.

Cela peut construire un réseau de données potentiellement infini qui peut être utilisé par des applications clientes [18].

#### 2. Liens d'identité

Dans le web de données, plusieurs fournisseurs de données parlent des mêmes entités. Comme ils utilisent leurs propres URIs pour désigner une personne ou un lieu, le résultat sera plusieurs et différentes URIs identifiant la même entité.

Le web de données liées repose sur la résolution du problème de la duplication des entités d'une façon évolutive (une quantité énorme de liens (owl : sameAs) peut être ajoutée au cours du temps) et distribuée (comme les liens (owl : sameAs) sont publiés par différents fournisseurs de données, l'effort global pour la création de ces liens peut être partagé entre les différentes parties).

Aujourd'hui, owl : sameAs est largement utilisé dans le contexte de données liées et des centaines de millions de liens owl : sameAs sont publiés sur le web [18].

#### 3. Liens de vocabulaires

Le web de données permet aux applications clientes de découvrir de nouvelles sources de données en suivant les liens RDF.

En outre, il aide aussi à intégrer les données provenant de ces sources. L'intégration des données vise à fournir une vue unifiée des schémas qui sont utilisés par différentes sources de données pour publier leurs données.

Dans le contexte des données liées, le terme schéma signifie le mélange des termes distincts de plusieurs vocabulaires RDF qui sont utilisés par une source de données pour publier des données sur le web [18].

### 1.3.5 Domaines d'applications des données liées

Le web de données est actuellement utilisé dans différents domaines d'applications [13] :

#### 1. E-commerce

Il permet aux moteurs de recherches de mieux exploiter ces données essentielles pour les restituer dans leur contexte de recherche, la vérité d'entreprise et les solutions de commerce

électroniques déployées faisant usages de configurant d'échange très diversifiées associée au manque de fiabilités et de sécurité sur l'internet rendent impossible le passage l'échelle par l'intégration , et aussi permet de décrire de manière structure les produit, les prix, et les informations relatives à l'entreprise.

### 2. Application médicale

La médecine est un des domaines d'applications privilégiés du web sémantique comme elle l'été, des techniques de l'intelligence artificielle, en particulier les systèmes experts. C'est en effet un domaine complexe où les informations à partager sont nombreuses. Un des principaux mécanismes du web sémantique qui est la description de ressources via des annotations est de la plus grande importance en bio-informatique, plus particulièrement autour des questions de partage des ressources génomiques.

### 3. Traitements des langages automatiques

La sémantique pour le traitement automatique s'intéresse à la modélisation des phénomènes sémantiques intervenant dans le langage humain (anaphore, ellipses, comparatif, références temporelles, attitudes, verbes, etc.). Traditionnellement, les approches formelles se sont situées au niveau de la phrase. Quand un auditeur reçoit un message d'un orateur, il essaie de comprendre ce que et pourquoi ce locuteur a produit ce message en faisant appel à ses compétences linguistiques, sa connaissance en général et en particulier celles de la situation d'énonciation, ses croyances, etc. L'auditeur construit donc une représentation (très probablement sémantique) de ce qu'il comprend de la proposition du locuteur, afin de sélectionner une réaction en retour.

## 1.4 Ontologie

### 1.4.1 Définition

Plusieurs définitions existent, en se basant sur [38] une ontologie consiste à définir de manière formelle et explicite (claire et précise) un ensemble de classes (concepts), propriétés (attributs), types de relations et entités (individus, instances) lisible par machine exprimant une vision partagée entre plusieurs parties (consensus).

Représentés par des prédicats unaires et binaires, une ontologie est basée sur une hiérarchie de concepts de généralisation/spécialisation, c'est-à-dire d'une taxonomie.

Les ontologies ont été utilisées avec succès afin de résoudre des problèmes tels que l'interopérabilité et l'hétérogénéité, dérivant de la gestion des connaissances partagées et distribuées et l'intégration efficace de l'information dans les applications.

## 1.4.2 Composants d'ontologie

L'ontologie utilise principalement cinq formalisations de type de composant : concept (ou classe), relation (ou attribut), Fonctions, axiomes (ou règles) et instances (ou individus).

### 1. Les concepts

Également appelés termes ou classes d'ontologie, correspondent à une abstraction pertinente d'une réalité (domaine problématique).

Un concept peut définir comme une entité composée de trois éléments distincts [14] :

**Le terme** : est un élément lexical qui permet d'exprimer le concept en langue naturelle.

**L'intention** : contient la sémantique du concept, exprimé en termes de propriétés et attributs, contraintes.

**L'extension** : regroupe les objets manipulés à travers le concept.

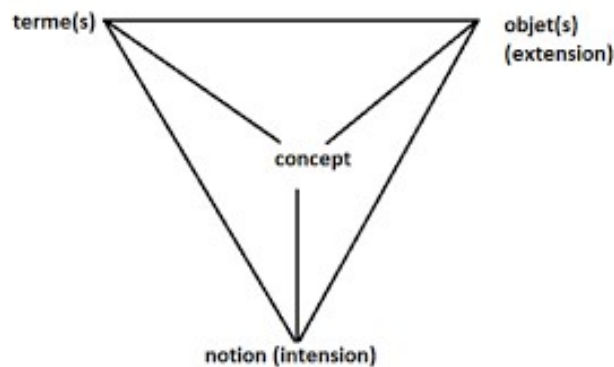


FIGURE 1.10 – Triangle sémantique [14].

### 2. Relation

Elles sont utilisées pour exprimer des relations entre deux concepts dans un domaine donné. Plus précisément, une relation décrit le lien entre le premier concept, représenté dans le domaine, et le second, représenté dans une portée.

### 3. Fonction

Est un cas particulier de relation, où l'élément de la relation le  $n$ ème est défini par  $N-1$  élément précédent.

### 4. Instance

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

### 5. Axiomes

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

### 1.4.3 Découverte de correspondance

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 par exemple d'équivalence, de subsumption plus générale, plus spécifique - et- de - disjonction.

Le schéma suivant illustre le processus d'alignement d'ontologies en générale :

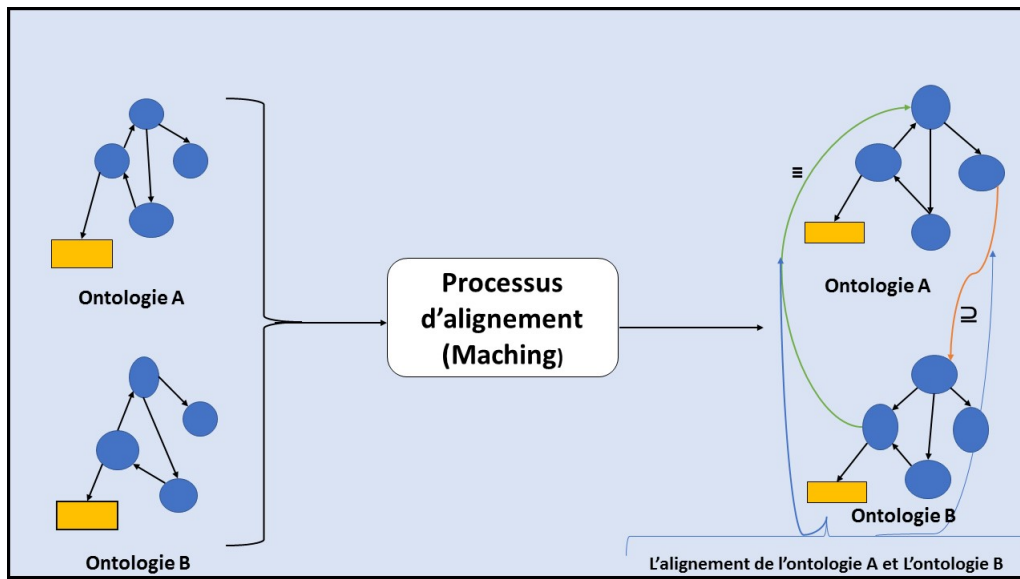


FIGURE 1.11 – Processus d'alignement d'ontologie [37].

La figure suivante montre un exemple d'alignement de deux données :

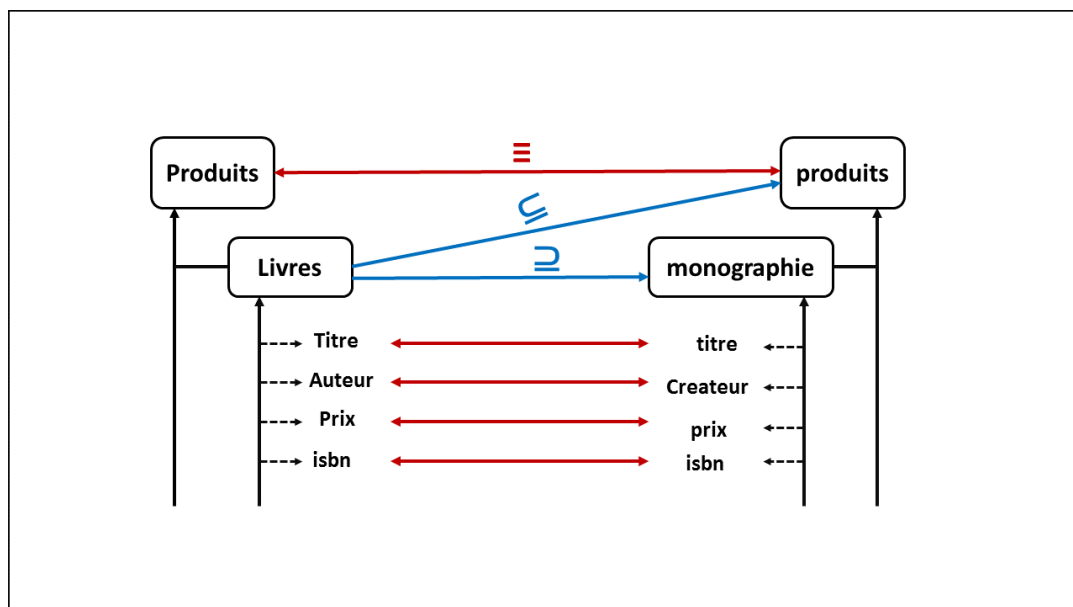


FIGURE 1.12 – Exemple d'alignement entre deux données [37].

### 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 [12] :

#### 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 en dessous, 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'.

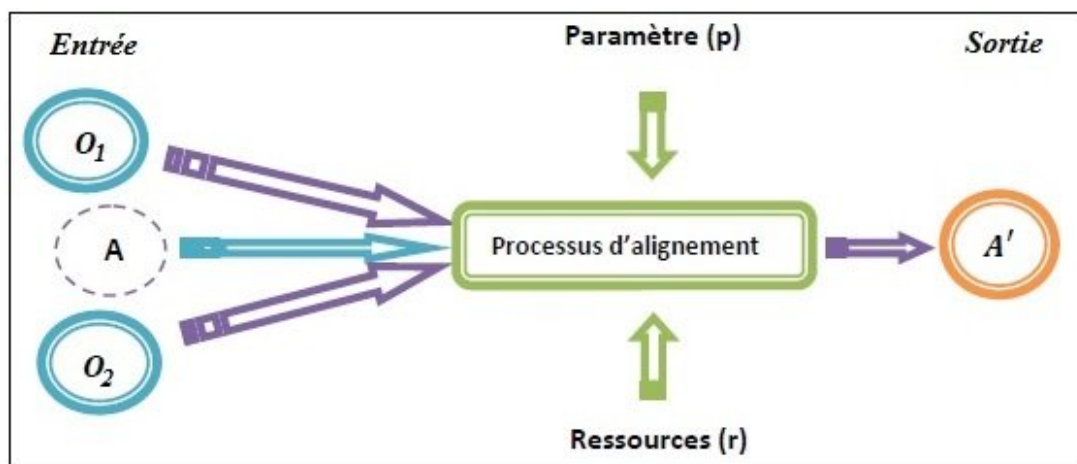


FIGURE 1.13 – Les trois dimensions de l'alignement [12].

#### 3. L'output

Est un ensemble d'alignement reliant les entités qui constituent les deux ontologies. Un alignement est décrit comme un ensemble de cinq éléments  $\langle id, e, e', r, n \rangle$  telle que :

- Id : un identifiant de l'alignement
- e : l'entité à aligner qui appartient à O
- e' : l'entité à aligner qui appartient à O'
- r : représente la relation qui permet de lier à e'
- n : la mesure de confiance de la relation r



### 1.4.4 Classification des conflits de découverte des liens de données

Les sources de données utilisent différentes représentations de la même information, classifie les conflits en deux types principaux [6] :

#### 1. Des conflits d'identités

Comportent les conflits qui existent au niveau d'instances qui identifient plusieurs représentations du même objet du monde réel.

#### 2. Des conflits de données

Un conflit de données est présenté dans un triplet RDF, si les ressources liées à ses composants sont sémantiquement équivalentes mais on réalité sont différentes.

## 1.5 Conclusion

Dans ce premier chapitre, nous avons donné une vision générale sur web de données et les données liées, leurs caractéristiques et leurs langages de représentation, nous avons également abordé quelques notions des ontologies.

Dans le chapitre suivant, nous décrivons les différentes méthodes existantes pour la découverte des liens.



## *Chapitre 2*

---

# **MÉTHODES EXISTANTES POUR LA DÉCOUVERTE DE LIENS DANS LES DONNÉES LIÉES**

---

## **2.1 Introduction**

Les données liées sont un ensemble de principes de concepts pour le partage des données lisibles par machine sur le Web pour une utilisation par les administrations publiques, les entreprises et les citoyens.

L'objectif du mouvement de données liées est d'étendre le réseau avec un espace de données mondial en publiant des ensembles de données selon un ensemble de bonnes pratiques et en établissant des liens RDF entre les sources de données.

Dans ce chapitre nous commençons par une explication de la découverte des liens ensuite nous présentons les différentes mesures de similarité, puis les différentes approches et méthodes qui existent pour la découverte de liens. A la fin nous comparons et analysons ces approches.

## **2.2 Découverte des liens**

Actuellement, le web de données liées connaît une croissance explosive de source de données, les liens entre les ensembles de données jouent un rôle important tel que l'intégration des données et l'interrogation des données.

Au cours de ces dernières années, plusieurs approches ont été développées pour découvrir des liens typés entre les différents ensembles de données. Ces approches jouent un rôle crucial dans la construction des liens sémantiques entre les données, sont généralement basées sur le calcul d'une mesure de similarité de paires de concepts [31].

## 2.3 La similarité

L'identification de la similarité dans les ressources est un concept fondamental qui a longtemps été reconnu comme un concept clé en intelligence artificielle, elle est au coeur du paradigme qui énonce qu'un transfert de connaissances d'un cas connu vers un cas inconnu est possible dans la mesure où ils sont suffisamment similaires.

Dans notre contexte, on se base sur la notion de similarité sémantique, également appelé proximité sémantique, elle est déterminée en associant des documents, des termes ou des entités à des mesures de similarité basées sur leur sens ou leur contenu sémantique [34].

La similarité est une fonction d'une paire d'entités à un nombre réel, représentant la similarité entre ces deux entités.

La similarité est définie comme suit [34] :

$$\forall a, b \in 0, s(a, b) \geq 0 \quad (2.1)$$

$$\forall a, b, c \in 0, s(a, a) \geq s(b, c) = s(a, b) \implies a = b \quad (2.2)$$

$$\forall a, b \in 0, s(a, b) = s(b, a) \quad (2.3)$$

$$\forall a, b, c \in 0, s(a, b) = s(b, c) \implies s(a, b) = s(a, c) \quad (2.4)$$

$$\forall a, b \in 0, s(a, b) \leq \infty \quad (2.5)$$

## 2.4 Mesure de similarité

La plupart des opérations effectuées sur les données, telles que l'alignement, la fusion, la traduction, la coordination, etc. Sont conçues et développées sur la base de la phase de découverte des correspondances. Ce dernier vise à identifier les classes similaires entre les données.

Plusieurs travaux proposant différentes techniques pour découvrir les correspondances qui existent entre les données qui existent dans la littérature. Selon le type de données analysées dans ces techniques.

## 2. MÉTHODES EXISTANTES POUR LA DÉCOUVERTE DE LIENS DANS LES DONNÉES LIÉES

La figure (2.1) résume les différentes mesures de similarités, catégorisées selon les techniques utilisées.

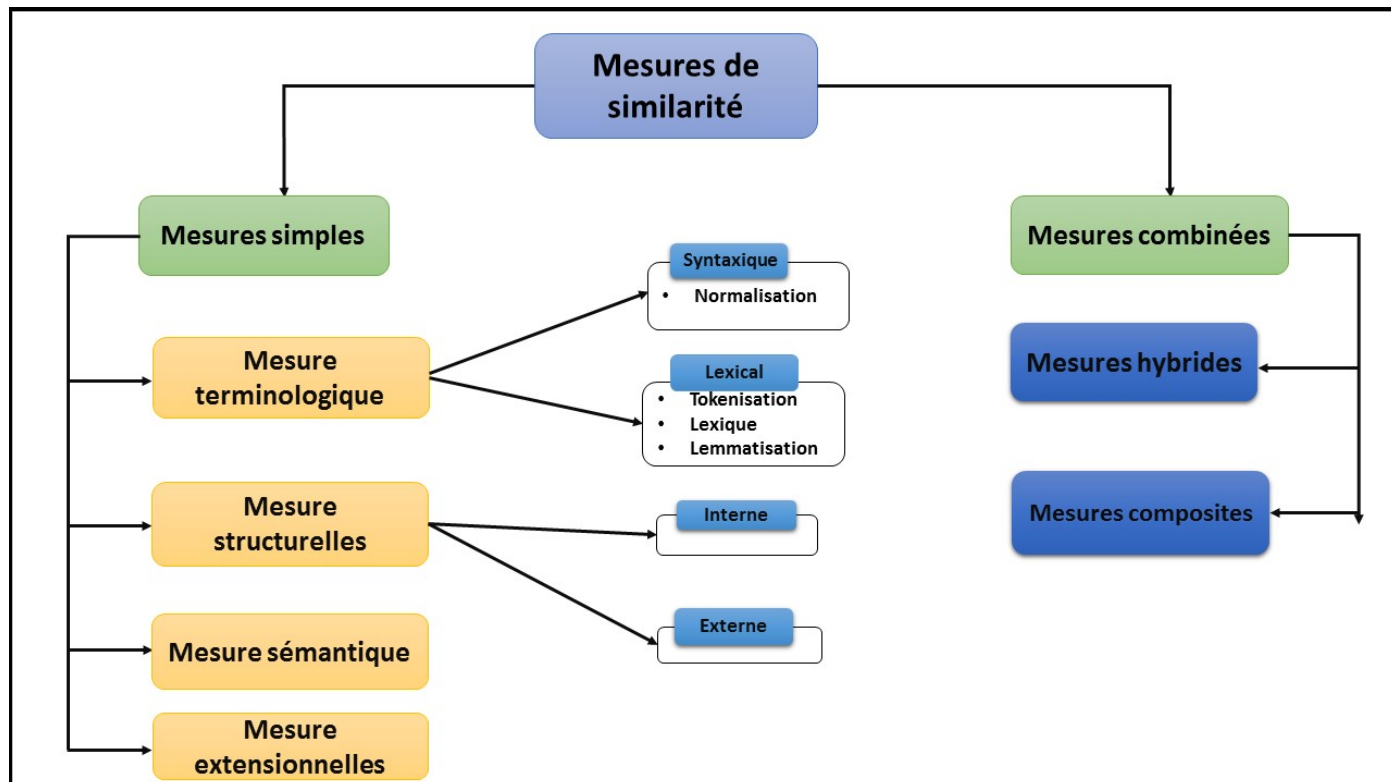


FIGURE 2.1 – Mesures de calcul de similarités [11].

### 2.4.1 Mesures simples

Elles sont basées sur l'analyse du contenu à l'intérieur de donnée. Il se compose d'un ensemble de classes et de leurs propriétés.

Les mesures simples sont identifiées par quatre types de mesures.

#### 1. Mesure terminologique

Ces méthodes sont utilisées pour calculer la valeur de similarité des entités textuelles (comparent les chaînes de caractères afin d'en déduire la similarité, telles que des noms, des métadonnées sur les noms, des étiquettes, des commentaires, des descriptions, etc.

Nous trouvons dans cette méthode deux approches essentielles l'approche syntaxique et l'approche lexicale, appelée aussi linguistique [30].

##### — Mesure syntaxique

Cette approche analyse la structure des chaîne à comparer, plus l'ordre des caractères dans la chaîne, le nombre d'apparition d'une lettre dans une chaîne pour concevoir des mesures de la similarité, plus elles partageront de caractères en commun. Par contre, elles n'exploitent pas la signification des termes.

Généralement, ces méthodes exigent un pré-traitement qui consiste à normaliser les chaînes à comparer avant de les fournir aux fonctions de calcul de la similarité.

Nous citons la distance de Jaro, Jaro Winkler, Levenshtein, Éditer la distance, distance de Jaccard et la distance TF/IDF [45].

#### a) La distance de Jaro

Elle mesure la distance entre deux chaînes de caractère. Il est principalement utilisé pour détecter les doublons. Plus la distance est grande la valeur de Jaro entre les deux chaînes est plus élevée et plus les deux chaînes sont similaires.

Cette mesure est particulièrement adaptée au traitement de chaînes courtes. Les résultats sont normalisés de telle sorte que a une valeur comprise entre 0 et 1, (0 signifie que les deux chaînes sont complètement différentes, 1 signifie qu'elles sont complètement similaires).

La distance de Jaro entre deux chaînes de caractères S1 et S2 est définie par [21] :

$$d_{jaro}(s1, s2) = \left(\frac{1}{3}\right) * \left(\left(\frac{m}{|s1|}\right) + \left(\frac{m}{|s2|}\right) + \left(\frac{m-t}{|s1|}\right)\right) \quad (2.6)$$

Où :

$|s1|$  (resp.  $|s2|$ ) est la longueur de la chaîne de caractères s1 ( resp. s2).

$m$  : est le nombre de caractères correspondants.

$t = N/2$  : est le nombre de transpositions.

$N$  : est le nombre de couples de caractères correspondants qui ne sont pas dans le même ordre dans leurs chaînes respectives.

Deux caractères identiques de s1 et s2 sont considérés comme correspondantes si leur éloignement (la différence entre leurs positions dans leurs chaînes respectives) ne dépasse pas la valeur val :

$$val = \max\left(\frac{|s1|, |s2|}{2}\right) - 1 \quad (2.7)$$

#### b) La distance de Jaro-Winkler

Elle est dérivée de la distance de Jaro. Elle utilise le coefficient de préfixe p et prend en charge les chaînes commençant par un préfixe de longueur l ( $l \leq 4$ ).

La distance de Jaro-Winkler entre deux chaînes de caractères s1 et s2 sont définis comme [42] :

$$d_{jaro-winkler} = d_{jaro} + (L * p(1 - d_{jaro})) \quad (2.8)$$

Où :

$l$  : est la longueur du préfixe commun (jusqu'à 4 caractères).

$P$  : Coefficient Préférez les chaînes avec un préfixe commun.

Winkler offre de la valeur  $p = 0,1$ .

### c) Distance de Levenshtein

La distance de Levenshtein entre deux chaînes  $x$  et  $y$  est définie comme le nombre minimum d'opérations d'édition pour convertir  $x$  en  $y$  en remplaçant les lettres de  $x$  par des lettres de  $y$  et/ou supprimer une lettre de  $x$  et insérer une lettre de  $y$ .

En d'autres termes, il est égal au nombre minimum de caractères qui doivent être supprimés, insérés ou remplacés d'une chaîne à l'autre [20].

### d) Éditer la distance

Transformer une chaîne de caractère  $x$  à un autre  $y$ , nous pouvons utiliser trois opérations de base, supprimer, insérer et remplacer.

La distance d'édition entre deux chaînes est définie comme le nombre minimum d'opérations d'édition nécessaires pour convertir une chaîne en l'autre. Là encore, le coût de chacune de ces opérations a été déterminé. Utilisez ensuite la distance d'édition pour trouver une Chaîne à l'autre avec un coût minime [22].

En prenant le coût de suppression et d'insertion égal à 1 et le coût de remplacement égal à 2, la distance d'édition est définie par :

$$DE(ax, bx) = DE(a, b), \text{ si } x = y \quad (2.9)$$

$$DE(ax, bx) = \min(DE(a, b) + 2, DE(ax, b) + 1, DE(a, by) + 1), \text{ si } x \neq y \quad (2.10)$$

$$DE(a, f) = |a| \text{ et } DE(f, b) = |b| \quad (2.11)$$

Lorsque tous les coûts sont égaux à 1, on tombe sur la distance de Levenshtein. La distance d'édition peut être considérée comme une généralisation de la distance Hamming.

### e) L'indice et la distance de Jaccard

C'est Deux métriques utilisées dans les statistiques pour comparer la similarité et la diversité entre les échantillons. Le coefficient exponentiel ou Jaccard est le rapport entre la cardinalité (amplitude) de l'intersection des ensembles de comparaison et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles [19].

Étant donné deux ensembles  $A$  et  $B$ , l'indice de Jaccard est défini comme :

$$\text{Indicejaccard} = \left( \frac{|A \cap B|}{|A \cup B|} \right) \quad (2.12)$$

L'indice de Jaccard mesure la différence entre les ensembles. Il consiste simplement à soustraire l'exposant de Jaccard à 1 :

$$Djaccard = 1 - Indicedjaccard = \left( \frac{|A \cup B| - |A \cap B|}{A \cup B} \right) \quad (2.13)$$

$$TF = \left( \frac{n(t)}{N} \right) \quad (2.14)$$

$$IDF = \left( \log \frac{|D|}{d(t)} \right) \quad (2.15)$$

Tel que :

- **D** : Un corpus de documents.
- **|D|** : Le nombre de documents dans le corpus D.
- **n(t)** : Le nombre d'occurrences du terme t dans le document.
- **N** : Le nombre total de Termes dans le document.
- **d(t)** : Le nombre de documents qui contiennent au moins une fois le terme t.

#### — **Mesure lexicale**

Les méthodes lexicales utilisant des ressources externes (dictionnaires, taxonomies,...etc.), ces méthodes permettent de déterminer la similarité entre deux entités. Ces entités sont représentées par des termes (ou mots).

La similarité est calculée à partir des liens sémantiques déjà existants dans les ressources externes comme WordNet [45].

WordNet est une grande base de données lexicale pour la langue anglaise où les noms, les verbes, les adjectifs et les adverbes sont regroupés dans des ensembles de synonymes cognitifs (appelés synsets). Les synsets sont liés par moyens de relations conceptuelles sémantiques et lexicales.

La formule de similarité lexicale entre S1 et S2 se calcule ainsi :

$$lexsim(s1, s2) = \left( \frac{\beta}{\min(\text{syn}(S1), \text{syn}(S2))} \right) \quad (2.16)$$

Où :

Min (syn(S1), (syn(S2)) le minimum des cardinalités de deux ensembles syn(C1) et syn(C2) et  $\beta = \text{syn}(S1)\text{syn}(S2)$ .

Cette mesure renvoi 1 si au moins S1 et S2 ont 1 synset commun. 0 est retournée dans le cas où S1 et S2 ne sont pas synonymes et n'ont pas de relation lexicale (antonymes, hyponymes...).

## 2. Mesures structurelles

Ce sont les méthodes qui déduisent la similarité entre deux entités en exploitant leurs positions dans une hiérarchie et en fonction des informations structurelles.

En effet, les entités sont reliées entre elles par des liens sémantiques ou syntaxiques. On peut distinguer entre deux méthodes structurelles.



## 2. MÉTHODES EXISTANTES POUR LA DÉCOUVERTE DE LIENS DANS LES DONNÉES LIÉES

---

L'une qui n'exploite que des informations concernant des attributs d'entités (les méthodes structurelles interne) et l'autre qui considère des relations entre des entités (les méthodes structurelles externes) [45].

### — Mesures interne

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...etc.).

Les premiers systèmes qui se basent sur ce principe sont les systèmes d'intégration et d'alignement des schémas de bases de données.

### — Mesures externes

Contrairement aux méthodes structurelles internes, qui exploitent des informations des attributs d'entité, les méthodes structurelles externes traitent la structure externe de l'entité et exploitent des relations entre elles-mêmes.

## 3. Mesures extensionnelles

Elles déduisent la similarité entre deux entités qui sont notamment des concepts ou des classes en analysant leurs extensions (leurs ensembles d'instances), tel que chaque instance peut être représentée par un vecteur de noms et/ou de valeurs des calculs de similarités entre vecteurs permettent de comparer les instances [44].

On distingue Deux approches pour comparer les ressourcés à partir des instances associées aux triplets :

- Soit les deux datasets à comparer référencent les mêmes instances et dans ce cas on génère une similarité entre les concepts qui partagent les mêmes instances.
- Soit les deux datasets à comparer ne référencent pas les mêmes instances et dans ce cas on fait des recherches par mots-clés dans les instances (souvent des documents ou autres fichiers). La similarité est ensuite calculée entre les instances à l'aide de ces mots-clés.

Dans le cas où les ensembles d'instances partagent une partie commune, on peut avoir des mesures extensionnelles qui emploient des opérations de l'ensemble, telles que la distance de Hamming.

### **Distance de Hamming**

La distance de Hamming fournit une Manière simple mais pas toujours pertinente de comparer deux chaînes.

Il mesure la distance entre deux chaînes de caractère x et y de même longueur [22] :

$$D_{hamming}(x, y) = \text{card}\left(\frac{i}{x[i]}\right) x \neq y \text{ et } 0 \leq i \leq n-1 \quad (2.17)$$

#### 4. Mesure sémantique

La similarité sémantique est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens.

La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun (plus ils ont de caractéristiques communes, plus les concepts sont similaires) et à leurs différences (plus deux concepts sont différents, moins ils sont similaires). La similarité maximale est obtenue lorsque deux concepts sont identiques [27].

### 2.4.2 Mesure combinée

Les différentes techniques décrites ci-dessus peuvent alors être utilisées un ensemble de paramétrage d'un algorithme "hybride" (deux ou plusieurs techniques au sein d'un même algorithme) ou d'un algorithme exécuté "composite" en parallèle [24].

#### 1. Mesure Hybride

Le moyen le plus simple de combiner des métriques et de les utiliser séquentiellement en choisissant l'ordre d'exécution [10].

#### 2. Mesure composite

Une autre manière de combiner les résultats des différentes mesures (c.-à-d. les valeurs de similarité) consiste tout d'abord à lancer parallèlement plusieurs mesures, puis par la suite à combiner leurs résultats [10].

## 2.5 Les méthodes de découverte des liens dans le contexte du web de données

### 2.5.1 RDF-AI

Est une architecture pour aligner et fusionner des ensembles de données. Par conséquent, cet outil génère un alignement qui peut être utilisé pour fusionner deux ensembles de données ou générer un ensemble lié contenant des triplets owl : sameAs.

RDF-AI prend en entrée un fichier XML spécifiant les paramètres pré-opérationnels : réorganisation du nom, traduction, spécifiant la structure du jeu de données, les techniques d'alignement pour chaque ressource à comparer.

L'outil fonctionne avec une copie locale de l'ensemble de données implémentée en Java [35].

### 2.5.2 SILK

Est paramétré par un langage de spécification de liens (le Silk Link Spécification Language).

L'utilisateur spécifie les entités à associer et indique la mesure de similarité à utiliser.

## 2. MÉTHODES EXISTANTES POUR LA DÉCOUVERTE DE LIENS DANS LES DONNÉES LIÉES

---

SILK utilise diverses méthodes pour aligner les chaînes, mesurer la similarité numérique, la similarité des dates, la distance entre les concepts dans les taxonomies et les méthodes de similarité collective.

Les ensembles de données préparés pour la transformation peuvent être spécifiés avant le processus d'alignement pour améliorer ses résultats. Silk prend en entrée deux ensembles de données accessibles via les points de terminaison SPARQL.

IL fournit une sortie à l'aide du triplets owl :sameAs ou d'un autre prédicat spécifié par l'utilisateur. Silk a été testé sur plusieurs jeux de données [41].

### 2.5.3 KNOFUSS

L'architecture de Knofuss est conçue pour fusionner des ensembles de données. L'une des caractéristiques de Knofuss est sa capacité à aligner des jeux de données décrits en termes d'ontologies hétérogènes.

Le processus de découvert des liens des données est piloté par des ontologies spécialisées qui spécifient les ressources à comparer, et la technique d'alignement appropriée à utiliser.

Les ressources sont sélectionnées en spécifiant une requête SPARQL sur l'ensemble de données. Cet outil fournit un ensemble d'algorithmes d'alignement de chaînes.

Lorsque les deux jeux de données à aligner sont décrits par des ontologies différentes, l'alignement peut être spécifié dans le format d'alignement, permettant d'utiliser l'un des nombreux systèmes d'alignement d'ontologies disponibles [33].

### 2.5.4 LIMES

Le Framework LIMES (links Discovery Framework) utilise des caractéristiques mathématiques échelles de l'espace de mesure pour réduire le nombre de calcul à effectuer par le système.

En particulier, il utilise des inégalités triangulaires dans l'espace intermédiaire pour calculer une estimation pessimiste de la similitude entre les instances.

Sur la base de ces approximations, LIMES peut filtrer un grand nombre de paires instances qui ne peuvent pas répondre aux conditions de correspondance définies par l'utilisateur Soutien.

Ensuite, il calcule la similarité entre les paires d'instances restantes et retourne le résultat de Matching à l'utilisateur [32].

### 2.5.5 Travail de [Yuliu et al, 2015]

Ce travail propose une nouvelle approche pour la découverte de liens des propriétés entre les ensembles de données liées, toutes ces propriétés sont liées par des fonctions.

Le processus de la découverte de liens de ces propriétés des données liées basé sur la similarité entre les fonctions peut récupérer certaines propriétés correspondantes que d'autres

méthodes ne peuvent pas les trouver, et peut identifier les propriétés non appariées qui sont retourné par d'autres méthodes [25].

### 2.5.6 STS [John et al, 2018]

STS (Similarité Textuelle Sémantique) est une méthode dans le domaine du traitement du langage naturel évalue la relation entre des textes ou des documents à l'aide d'une métrique définie.

Dans le cas d'un ensemble de données qui liées des concepts sont beaucoup plus courtes que les phrases utilisées dans la similarité textuelle sémantique et beaucoup moins complexes.

Dans cette méthode on trouve des travaux supplémentaires pour développer des caractéristiques spécialisées pour l'équivalence des termes seraient d'un grand avantage, et aussi qu'un obstacle important de découverte de liens est la diversité des jeux de données, en termes de nature des textes, l'existence d'informations supplémentaires utiles pour la mise en relation (telles que les propriétés les descriptions, etc.) [26].

### 2.5.7 Travail de [Armando et al, 2022]

Cette méthode a présenté une approche semi-automatique pour aligner des ensembles de données et des scénarios du monde réel.

Cette approche proposée est nécessaire en raison du besoin de solutions capables d'aligner de manière fiable les données avec le moins de connaissances possibles du domaine.

De plus, la solution permet d'exécuter l'alignement directement dans le triple stockage de sorte qu'il n'est pas nécessaire de générer des fichiers à aligner.

De plus, il a été possible de noter des problèmes liés aux informations fournies par les auteurs. Cette recherche a également mené une expérience pour évaluer l'approche proposée et la comparée à l'efficacité avec d'autres outils en utilisant les métriques de précision, de rappel et F-mesure.

Dans l'expérience, ces métriques ont été évaluées dans deux scénarios d'alignement, dans lesquels l'approche proposée a obtenu de bons résultats compatibles avec les outils d'appariement des instances supérieures. Bien qu'elle n'ait pas les meilleures valeurs dans les deux évaluations [4].

## 2.6 La comparaison des méthodes

Nous présentons un tableau récapitulatif (tableau 2.1) de notre analyse des méthodes de découvert des liens des données en prenant comme critères les entrées du système, automatisation, mesure de similarité, sortie, accès aux données, domaine, technique de mesure de similarité et niveau de connaissances.

### 2.6.1 Critères de comparaison

#### 1. Automatisation

C'est un critère qui nous montre si le système fonctionne automatiquement, manuellement par l'intervention de l'utilisateur, ou un fonctionnement semi-automatique, intervention de l'utilisateur dans quelques tâches.

#### 2. Techniques de mesure de similarité

Ce critère nous indique les techniques utilisées par chaque méthode, il existe un très grand nombre de technique dans la littérature (mesure terminologie, sémantique ...).

#### 3. Sortie

Le type des liens de la sortie finale.

#### 4. Accès aux données

Comment accède aux jeux de données (à travers un point d'accès SPRQL, à partir d'une URL, à partir d'une copie locale des jeux de données).

#### 5. Domaine

Le domaine qu'est applicable par ses méthodes.

#### 6. Mesure de similarité

Représente les types des mesures de similarité utilisées pour faire la comparaison entre les ressources.

#### 7. Niveau de connaissance

L'utilisation des connaissances externes, comme par exemple les connaissances définies par le schéma ontologique qui représente des ensembles de données reposent sur des relations entre individus.



	Format d'entre	Automatisation	Mesure de similarité	Technique de mesure de similarité	Type de lien de sortie	Accès aux données	Domaines	Niveaux de connaissance
RDF-AI [Yandin et al, 2009]	RDF, SPRQL, Xml	Semi-Auto	Syntaxique, Lexicale	Wordnet, Taxonomie	OWL :sameAs	Copie local	Plusieurs	Taxonomie distance
SILK [Juluis et al, 2009]	RDF, SPRQL, CSV	Semi-Auto	Syntaxique	Taxonomie	Plusieurs liens	SPARQL	Science de la vie	-
Knofuss [Andrly et al, 2012]	RDF, SPRQL	Semi-Auto	Syntaxique	Alignement des mesures de chaîne de caractère	OWL :sameAs	Copie local	Plusieurs	Schéma externe
LIMS [Kevin et al, 2014]	RDF, SPRQL, CSV	Automatique	Syntaxique, Sémantique	Alignement des mesures de chaîne de caractère	OWL :sameAS	SPARQL	Plusieurs	-
Travail de [Yuliu et al, 2015]	RDF	Semi-Auto	Syntaxique	Normalisation, Toxinization, jaro Winkler, Levinstein	OWL :sameAs	Copie local	Plusieurs	-
STS [John et al, 2018]	Plusieurs formats	Semi-Auto	Syntaxique, Structurelle	Thésaurus, Wu Palmer	OWL :sameAS	Copie local	Traitement du langage naturel	-
Travail de [Armando et al, 2022]	RDF, SPARQL	Semi-Auto	Syntaxique	Alignement des mesures de chaîne de caractère	OWL :sameAS	SPARQL	Plusieurs	-

TABLE 2.1 – Comparaison des techniques de découverte des liens

## 2.6.2 Analyse

Après une étude comparative des différents systèmes de découverte des liens présentés dans La table 2.1 nous constatons ce qui suit :

- Ces approches acceptent en entrée des fichiers RDF, elles utilisent des requêtes SPARQL pour récupérer des données à partir des SPARQL.  
SILK et LIMES peuvent découvrir des liens entre des fichiers CSV.
- Parmi les différentes méthodes citées auparavant toutes les méthodes sont semi-automatiques sauf LIMES qui complètement automatique.
- Ces framework génèrent en sortie des liens owl : sameAs. Le framework SILK Peuvent générer d'autres types de liens qui doivent être spécifiés par l'utilisateur.
- Pour l'utilisation de la similarité entre les données liées, la technique de calcul de similarité n'est pas la même, la plupart des méthodes utilisent la mesure de similarité syntaxique, quelques méthodes utilisent deux méthodes de similarité, syntaxique et lexicale, ou structurelle.
- Toutes les méthodes prennent en charge les mesures de similarité, mais la définition de la sémantique ça diffère d'une méthode à une autre, par exemple la méthode RDF-AI défini la sémantique en combinant la mesure syntaxique et lexicale.
- Peu de méthodes traitent le niveau de connaissances. Parmi les méthodes mentionnées dans le tableau, il y a que les méthodes RDF-AI et KNOFUSS qui traitent le niveau de connaissance.

## 2.7 Conclusion

Dans ce chapitre, nous avons présentons un état de l'art sur les méthodes de découverte de liens de données existantes, en les comparant selon des critères bien précis tels que les mesures de similarité, la manipulation de concepts composés, les aspects sémantiques, etc.

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 3

---

# CONCEPTION DU SYSTÈME

---

### 3.1 Introduction

Après avoir donné un aperçu général sur le domaine de découverte des liens. Dans ce chapitre nous proposons notre solution pour résoudre le problème de découverte des liens. Nous commencerons d'abord par la présentation des caractéristiques de notre système par rapport aux systèmes existants présentés dans le chapitre précédent, et présenter le schéma global de notre système et son fonctionnement.

Ensuite, nous allons donner une description détaillée de notre solution, en présentant les différentes étapes, et calculer la similarité en combinant plusieurs méthodes pour avoir les meilleurs résultats.

### 3.2 Caractéristiques de notre système

La majorité des travaux existants que nous avons déjà analysé dans le chapitre précédent utilisent la combinaison de quelques mesures de similarités ce qui diminue la précision de la similarité sémantique. Nous avons distingué aussi que la majorité des travaux sont semi-automatique, ce qui nécessite une intervention d'un expert du domaine.

L'objectif principal est de proposer un système pour la découverte des liens, qui combine plusieurs mesures de similarité. Pour avoir une mesure de similarité global de haute qualité, et ainsi avoir un système purement automatique. Notre système se caractérise par :

- L'utilisation du standard RDF comme un langage de représentation des ressources.
- L'automatisation du processus de découverte des liens.
- La réalisation d'une catégorisation par domaine entre les ressources.
- L'utilisation de plusieurs mesures de similarités qui sont combinées pour avoir la similarité sémantique (terminologie (syntaxique, lexicale), structurelle (interne, externe), et extensionnelle).

- L'utilisation des ressources externe pour avoir un alignement sémantique avec l'utilisation de WordNet et le site officiel de DBpedia.
- La réalisation des tests du système sur des jeux volumineux des données.

## 3.3 Schéma global du système

Notre système consiste à effectuer une découverte sémantique entre les données liées. Dans ce qui suit nous présentons le schéma global de notre solution, ensuite nous détaillons les différentes étapes et algorithmes utilisés.

Le schéma global proposé est illustré dans la figure suivante :

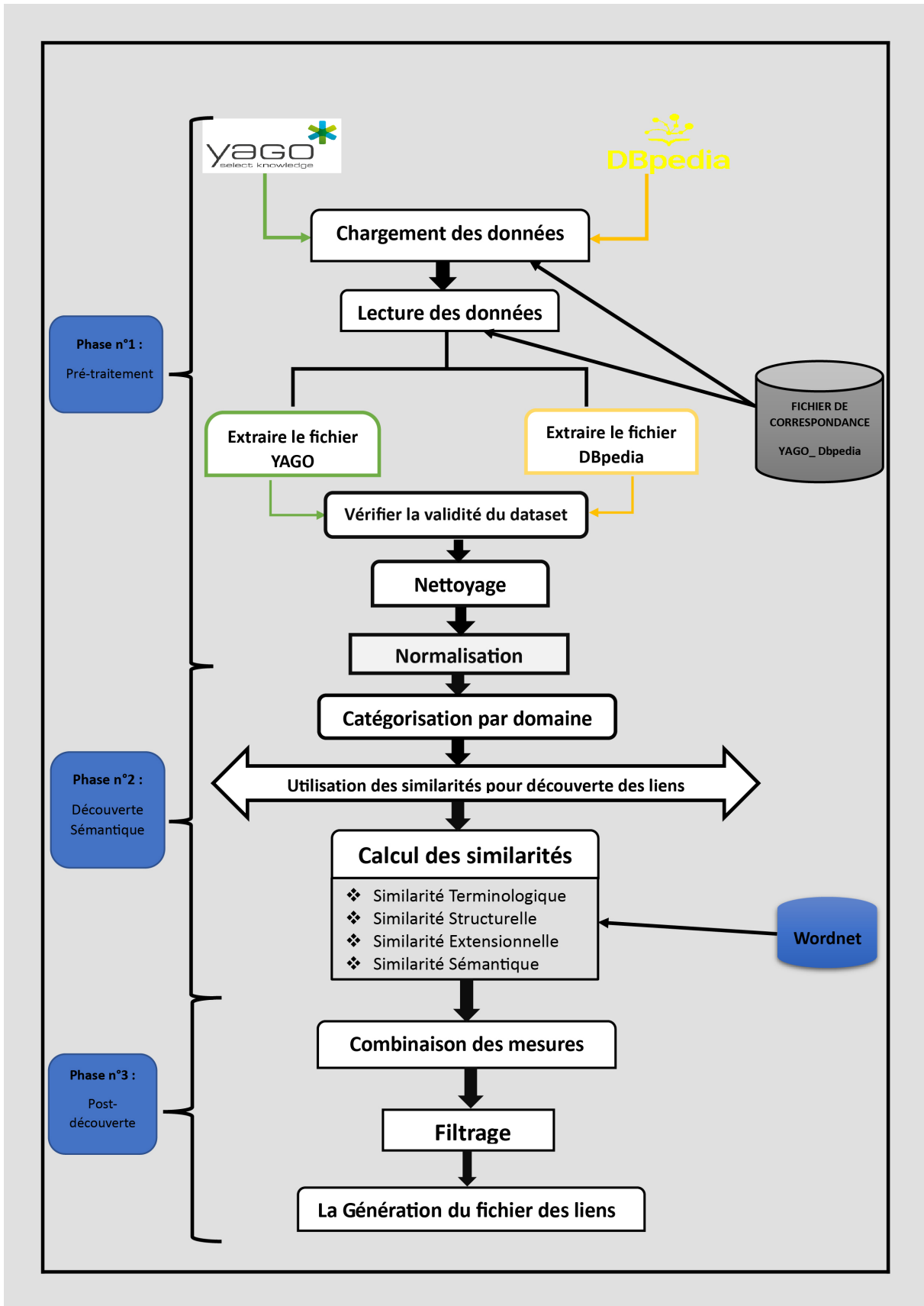


FIGURE 3.1 – Schéma Global.

## 3.4 Description de la solution proposée

La solution proposée est composée de trois phases. Chaque phase présente une ou plusieurs étapes que nous avons suivies pour faire la découverte des liens de données.

### Phase 1 : Pré-traitement

Le pré-traitement 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 vérifié leur validité. Nous avons terminée la phase du pré-traitement par une étape de normalisation des données.

Cette phase comporte quatre étapes principales :

- **Étape 1** : Téléchargement de données.
- **Étape 2** : Vérification de la validité des datasets.
- **Étape 3** : Nettoyage.
- **Étape 4** : Normalisation.

### Phase 2 : Découverte sémantique

Cette phase répondre au besoin de la découverte de liens de données.

Elle comporte trois étapes essentielles :

- **Étapes 1** : Catégorisation par domaines.
- **Étapes 2** : Détermination de l'équivalence entre les domaines.
- **Étapes 3** : Découverte de lien entre les triplets.

### 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 et pour ce avoir une mesure globale (sémantique). Ainsi une génération du fichier des liens.

Elle comporte trois étapes principales :

- **Étape 1** : Combinaison des mesures.
- **Étape 2** : Filtrage.
- **Étape 3** : Génération du fichier des liens.

### Les datasets utilisés

La solution que nous avons proposée consiste à concevoir un système de découverte des liens entre deux datasets. Pour expliquer la présente solution, le choix s'est porté sur deux dataset : **YAGO** et **DBpedia**. Les deux datasets ont été téléchargé depuis le site officiel de DBpedia.

#### 1. Yago<sup>1</sup>

YAGO (Yet Another Great Ontology) est une base de connaissance créée par l'institut Max-Planck d'informatique à Sarrebruck. Elle est constituée à partir d'informations extraites de Wikipédia et d'autres sources.

---

1. <https://www.yago.org/>

### 2. DBpedia<sup>2</sup>

Est un projet universitaire d'exploration et extraction automatiques de données dérivées de Wikipédia. Son principe est de proposer une version structurée et normalisée au format du web sémantique des contenus de Wikipedia.

DBpedia vise aussi à interconnecter Wikipédia avec d'autres ensembles de données ouvertes provenant du Web des données. Ce projet est conduit par l'Université de Leiozig, l'université libre de Berlin et l'entreprise OpenLink Software [5].

Dans la section ci-dessus, nous allons expliquer en détail la solution proposée.

### 3.4.1 Phase 1 : Pré-traitement

Le pré-traitement est la première phase de notre solution. Il sera question de téléchargement des deux datasets, et de la vérification leur validité, ainsi que du nettoyage et la normalisation.

#### 3.4.1.1 Étape 1 : Téléchargement des données

Cette étape consiste à extraire tous les triplets de chaque dataset d'une manière automatique. Nous avons tout d'abord chargé les deux datasets **Yago** et **Dbpedia**, ainsi que le fichier de correspondance, ce dernier représente les correspondances de l'expert du domaine. Puis nous avons parcouru les datasets afin d'extraire tous les triplets. Les triplets extraits constituent essentiellement les ressources destinées à être alignées qui seront stockées dans deux dictionnaires différents. Les triplets extraits du dataset **Yago** seront stockés dans un dictionnaire "Dict-Yago", les triplets du dataset **DBpedia** seront stockés dans un dictionnaire "Dict-DBpedia", les triplets de fichier de **correspondance** seront stockés dans deux listes différentes. une liste pour la partie DBpedia "SameAs-Subject-DBpedia", et une autre liste pour la partie yago "SameAs-Subject-yago".

#### Exemples de triplets de YAGO :

Le figure ci-dessous montre quelques triplets du dataset yago. La première colonne représente les sujets, alors que la deuxième colonne représente les propriétés, et la troisième représente les objets.

---

2. <https://www.dbpedia.org/>

Sujet	propriété	objet
<http://dbpedia.org/resource/1903_World_Series>	<http://dbpedia.org/resource/Bluetongue_disease>	<http://dbpedia.org/resource/Bruce_Perens>
<http://dbpedia.org/resource/1941>	<http://dbpedia.org/resource/13th_century>	<http://dbpedia.org/resource/4th_century>
<http://dbpedia.org/resource/100BaseVG>	<http://dbpedia.org/resource/10BASE2>	<http://dbpedia.org/resource/10BASE5>
<http://dbpedia.org/resource/15th_century_BC>	<http://dbpedia.org/resource/L%C3%A9on_(film)>	<http://dbpedia.org/resource/American_Goldfinch>
<http://dbpedia.org/resource/16_mm_film>	<http://dbpedia.org/resource/8_mm_film>	<http://dbpedia.org/resource/Anthony_Zinni>
<http://dbpedia.org/resource/1938_FIFA_World_Cup>	<http://dbpedia.org/resource/Ideal_gas_law>	<http://dbpedia.org/resource/Blast_beat>
<http://dbpedia.org/resource/1950_FIFA_World_Cup>	<http://dbpedia.org/resource/Willie_Davenport>	<http://dbpedia.org/resource/Mamo_Wolde>
<http://dbpedia.org/resource/2000_Summer_Olympics>	<http://dbpedia.org/resource/Universal_Turing_machine>	<http://dbpedia.org/resource/Vaf%C3%BEr%C3%BA%C3%9F>
<http://dbpedia.org/resource/2002_Bali_bombings>	<http://dbpedia.org/resource/Jemaah_Islamiyah>	<http://dbpedia.org/resource/Outer_product>
<http://dbpedia.org/resource/1990s_in_film>	<http://dbpedia.org/resource/Fulling>	<http://dbpedia.org/resource/Cai_Lun>

FIGURE 3.2 – Exemple des triplets YAGO.

### Exemples de triplets DBpedia :

Le figure ci-dessous montre quelques triplets du dataset DBpedia. La première colonne représente les sujets, la deuxième représente les propriétés, alors que la troisième représente les objets.

Sujet	propriété	objet
<http://mpii.de/yago/resource/Abu_Dhabi>	<http://mpii.de/yago/resource/Alabama>	<http://mpii.de/yago/resource/Achilles>
<http://mpii.de/yago/resource/Abraham_Lincoln>	<http://mpii.de/yago/resource/Aristotle>	<http://mpii.de/yago/resource/An_American_in_Paris>
<http://mpii.de/yago/resource/Academy_Award>	<http://mpii.de/yago/resource/Actrius>	<http://mpii.de/yago/resource/Animalia_(book)>
<http://mpii.de/yago/resource/International_Atomic_Time>	<http://mpii.de/yago/resource/Ang_Lee>	<http://mpii.de/yago/resource/Ayn_Rand>
<http://mpii.de/yago/resource/Alain_Connes>	<http://mpii.de/yago/resource/Allan_Dwan>	<http://mpii.de/yago/resource/Algeria>
<http://mpii.de/yago/resource/Characters_in_Atlas_Shrug>	<http://mpii.de/yago/resource/Atlas_Shrugged>	<http://mpii.de/yago/resource/Austria>
<http://mpii.de/yago/resource/American_Samoa>	<http://mpii.de/yago/resource/Amoeboid>	<http://mpii.de/yago/resource/ASCII>
<http://mpii.de/yago/resource/Apollo>	<http://mpii.de/yago/resource/Andre_Agassi>	<http://mpii.de/yago/resource/Austro-Asiatic_languages>
<http://mpii.de/yago/resource/Afro-Asiatic_languages>	<http://mpii.de/yago/resource/Andorra>	<http://mpii.de/yago/resource/Animal_Farm>
<http://mpii.de/yago/resource/Alaska>	<http://mpii.de/yago/resource/Aldous_Huxley>	<http://mpii.de/yago/resource/Analysis_of_variance>

FIGURE 3.3 – Exemple de triplets DBpedia.

### Exemples des triplets du fichier de correspondance :

Yago	OWL :sameAs	Dbpedia
http://mpii.de/yago/resource/wordnet_pocket_veto_100209680	http://www.w3.org/2002/07/owl#sameAs	http://dbpedia.org/class/yago/wordnet_pocket_veto_100209680
http://mpii.de/yago/resource/wordnet_sincer_104218773	http://www.w3.org/2002/07/owl#sameAs	http://dbpedia.org/class/yago/wordnet_silencer_104218773
http://mpii.de/yago/resource/wordnet_pocket__108665101	http://www.w3.org/2002/07/owl#sameAs	http://dbpedia.org/class/yago/wordnet_pool__108665101

TABLE 3.1 – Exemple des triplets du fichier de correspondance.

Le tableau ci-dessus illustre quelques ressources du fichier de correspondance. La première colonne représente les ressources de dataset yago. La troisième colonne représente les ressources de dataset DBpedia. La deuxième colonne représente le lien d'identité qui relie les deux datasets.

#### 3.4.1.2 Étape 2 : Vérification de la validité du dataset

Pour réaliser les étapes suivantes, nous devons tout d'abord vérifier la validité des datasets. Pour faire, nous avons introduit deux algorithmes. Le premier va faire l'extraction de propriétés et d'objets pour chaque ressource. Le deuxième consiste à compter le nombre d'occurrences pour les éléments qui existent au niveau du fichier d'équivalence.

##### 1. L'algorithme d'extraction de propriété et d'objet pour chaque ressource :

L'objectif de cet algorithme consiste à extraire toutes les propriétés et les objets pour chaque ressource donnée.

##### **Algorithme Extraction-propriétés-objets**

**Entres :** Liste-ressources

**Sortes :** Liste-final

**Début**

1. /\*Parcourir tous les ressources des datasets \*/

**Foreach** (ressources  $\in$  Liste – ressource) **do**

2. /\*Pour chaque ressources qui existe chercher tous ses propriétés et objets\*/

**Foreach** (ressource) **do**

Liste-final.ressource := ressource

L-propriété := propriétés

L-objets := objets

3./\* Affecter pour chaque ressources ses propriétés et objets \*/

**Foreach** (L-propriété) **do**

Liste-final := propriétés

**Foreach** (L-objets) **do**

Liste-final := objets

4. /\*Afficher liste finale des propriétés et des objets pour chaque ressource \*/

**Afficher** (Liste-final) **End for**

**End for**

**End for**

**End for**

**End**

**2. L'algorithme qui calcule le nombre d'occurrences qui existent pour chaque élément :**

L'objectif de cet algorithme est de compter le nombre d'occurrences pour les éléments qui existent au niveau du fichier d'équivalence.

**Algorithme qui calcule le nombre d'occurrences existantes**

**Inputs :** dbpedia-dict, dbpedia-yago, sameAs-subject-yago, sameAs-subject-yago .

**Outputs :** shared-subj

**Début**

1. /\*Importation des datasets\*/

lire(datasets)

2. /\*Parcourir la liste des ressources des datasets\*/

**For**  $i \in (\text{sameAs} - \text{subject} - \text{dbpedia})$

**For**  $i \in (\text{sameAs} - \text{subject} - \text{yago})$

3. /\*Rechercher les éléments qui existent dans yago et DBpedia par rapport au fichier de correspondance\*/

**if** (dbpedia-dict[sameAs-sujets-dbpedi*a*][*i*])>0 And

(yago-dict[sameAs-sujets-yago][*i*])>0 **than**

4. /\*Ajouter les résultats dans la liste \*/

**Ajouter** (shared-subj)

**End if**

**End for**

**End**



### 3. CONCEPTION DU SYSTÈME

---

Voici quelques exemples qui nous montrent l'existence des mêmes éléments pour les datasets YAGO et DBpedia par rapport au fichier de correspondance :

**Exemple de nombre d'éléments qui existe au niveau de fichier d'équivalence et dans YAGO :**

```
tuple numéro 177424 → prop_obj ajoutés = 25212
tuple numéro 177425 → prop_obj ajoutés = 25212
tuple numéro 177426 → prop_obj ajoutés = 25212
tuple numéro 177427 → prop_obj ajoutés = 25212
tuple numéro 177428 → prop_obj ajoutés = 25212
tuple numéro 177429 → prop_obj ajoutés = 25212
tuple numéro 177430 → prop_obj ajoutés = 25212
tuple numéro 177431 → prop_obj ajoutés = 25212
tuple numéro 177432 → prop_obj ajoutés = 25212
```

FIGURE 3.4 – Exemple de nombre d'éléments qui existe au niveau du fichier d'équivalence et dans YAGO .

Cependant, il existe des datasets pour lesquels nous n'avons pas trouvé d'éléments qui existent au niveau des fichiers d'équivalence, tel que le cas dans l'exemple des deux datasets DBpedia et DBLP.

```
tuple numéro 440897 → prop_obj ajoutés = 0
tuple numéro 440898 → prop_obj ajoutés = 0
tuple numéro 440899 → prop_obj ajoutés = 0
tuple numéro 440900 → prop_obj ajoutés = 0
tuple numéro 440901 → prop_obj ajoutés = 0
tuple numéro 440902 → prop_obj ajoutés = 0
tuple numéro 440903 → prop_obj ajoutés = 0
tuple numéro 440904 → prop_obj ajoutés = 0
tuple numéro 440905 → prop_obj ajoutés = 0
```

FIGURE 3.5 – Exemple de nombre d'éléments qui n'existent pas au niveau de fichier d'équivalence et dans DBLP

### 3.4.1.3 Étape 3 : Nettoyage

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

Ensuite, nous avons sauvegardé les résultats obtenus de tous les triplets dans deux dictionnaires différents.

### 3.4.1.4 Étape 4 : 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 casse :** C'est un processus qui rend les séquences des mots plus uniformes (exemple figure 3.7).

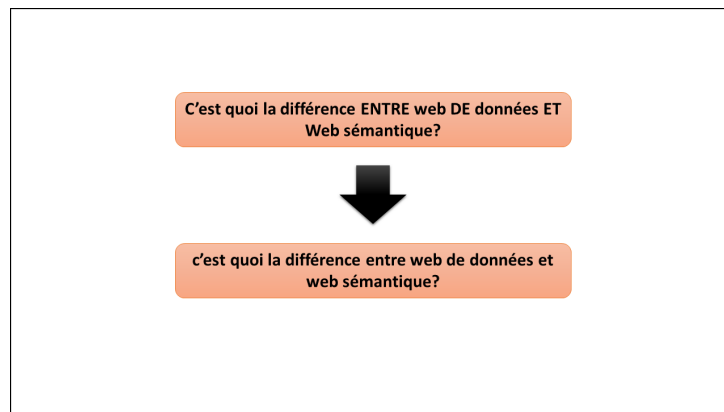


FIGURE 3.6 – Normalisation de casse.

- 2. Suppressions des mots vides :** Dans cette phase nous allons supprimer tous les mots vides (pronoms personnels, prépositions, ...etc.). (Exemple figure 3.8).

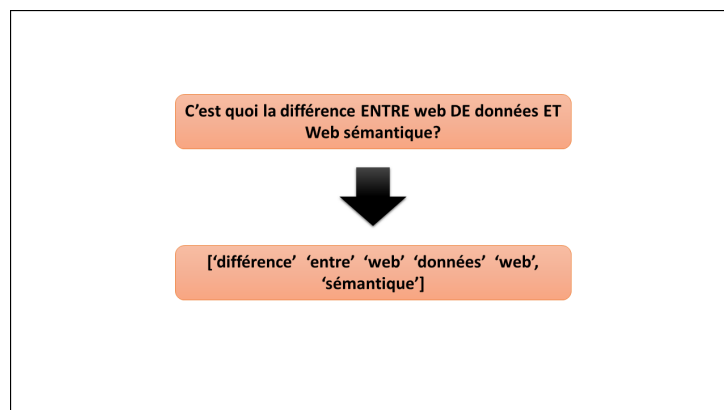


FIGURE 3.7 – Suppressions des mots vides.

**3. Normalisation des blancs et des liaisons :** Convertir toute tabulation, répétition d'espace, apostrophe, tiret, ...etc en un espace blanc.

Par exemple : "Bluetongue-disease " devient " Bluetongue disease ".

**4. Suppression des ponctuations :** Par exemple : " D !A,T.A ? " devient " DATA ".

#### 3.4.2 Phase 2 : Découverte sémantique

La phase de découverte sémantique se compose de trois étapes. Dans la première étape nous avons réalisé la catégorisation par domaine. L'objectif de cette étape étant de réduire l'espace de recherche, afin d'avoir des catégories par domaines. Dans la deuxième étape nous avons réalisé la détermination de l'équivalence entre les domaines des deux datasets. L'objectif, cette fois-ci est la détermination des domaines équivalents entre les deux datasets. Par la suite, nous avons procédé à la découverte sémantique sur les triplets qui existent dans des domaines équivalents.

##### 3.4.2.1 Étape 1 : Catégorisation par domaine

L'étape de catégorisation par domaine proposée consiste à catégoriser les deux datasets **Yago** et **DBpedia** par domaine. Pour cela, nous avons utilisé le langage des requêtes **SPARQL**.

Le processus de construction des catégories par domaine se base sur la récupération des domaines qui existent au niveau de DBpedia et yago. Il s'agit des ressources externes que nous avons utilisé dans la solution proposé.

Voici le processus de construction des catégories par domaines :

**1.** Tout d'abord nous avons introduit les deux liens DBpedia et yago.

**2.** Ensuite, nous avons développé la requête SPARQL, pour récupérer tous les domaines de chaque sujet d'un triplet donné. Le processus de construction des catégories se fait triplet par triplet.

**3.** A la fin de processus, nous n'avons gardé que les domaines qui existent dans les deux datasets.

**La requête SPARQL :**

```
PREFIX rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns>
```

```
PREFIX rdfs : <http://www.w3.org/2000/01/rdf-scemas>
```

```
PREFIX dbr : <http://dbpedia.org/resource>
```

```
PREFIX dbo : <http://dbpedia.org/ontology>
```

```
SELECT DISTINCT ? obj WHERE
```

```
"""+lien"""+ rdf :type ?obj
```

```
FILTER strstarts(str(?obj), str(dbo :))
```

Les exemples ci-dessous montrent un exemple des catégories par domaine pour chaque ressource des deux datasets.

### Exemple des domaines des triplets YAGO :

Yago	Domaine
<http://yago-knowledge.org/resource/Abu-Dhabi>	City/ Place/ Thing/ Capital-city
<http://yago-knowledge.org/resource/Alfons-Maria-Jakob>	Person/ Thing/ Human
<http://yago-knowledge.org/resource/Book-of-Nehemiah>	Book/ Creative-Work /Product /Religious-text
<http://yago-knowledge.org/resource/Common-Desktop-Environment>	Creative-Work /Product/ Desktop-environment

TABLE 3.2 – La catégorisation par domaines du dataset yago.

### Exemple des domaines des triplets de DBpedia :

DBEPDIA	Domaine
<http://dbpedia.org/resource/1903-World-Series>	Television-Show
<http://dbpedia.org/resource/2005-Canadian-federal-budget>	Film
<http://dbpedia.org/resource/100th-Infantry-Battalion-(United-States)>	Agent/ MilitaryUnit/ Organisation
<http://dbpedia.org/resource/1999-FIFA-U-17-World-Championship>	CSocietalEvent /Event/ /SportsEvent

TABLE 3.3 – La catégorisation par domaines du dataset DBpedia.

#### 3.4.2.2 Étape 2 : Détermination de l'équivalence entre les domaines

Pour assurer la détermination des équivalences entre les domaines en entrée. Nous allons utiliser les domaines des deux dataset. En sortie l'algorithme établit les résultats des équivalences entre les domaines. Nous avons utilisé la mesure terminologie syntaxique sur les domaines la formule (3.1).

$$Indicejaccard = \left( \frac{|A \cap B|}{|A \cup B|} \right) \quad (3.1)$$

La figure ci-dessous illustre un schéma pour déterminer des équivalences entre domaines.

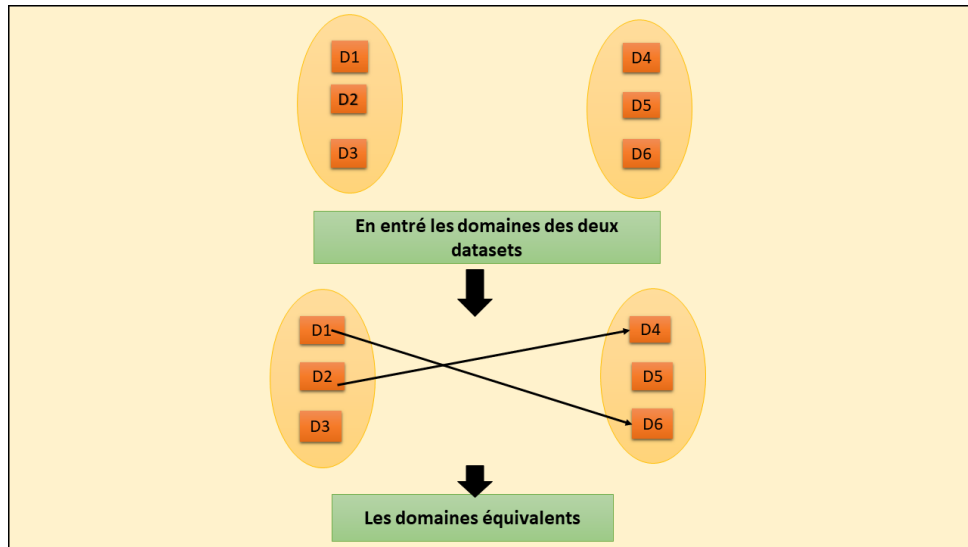


FIGURE 3.8 – Domaines équivalents.

#### 3.4.2.3 Étape 3 : Découverte de lien entre les triplets

Après avoir trouvé les correspondances entre les domaines de chaque dataset. Nous déclenchons le processus de découverte des liens entre les triplets, pour le calcul des similarités entre chaque triplet qui se trouve dans des domaines équivalents, c'est à dire faire un calcul de similarité sémantique entre les triplets de domaine **A** avec les triplets du domaine **B**. Et pour chaque triplet du domaine **A**, nous calculons la valeur de similarité avec les triplets du domaine **B**, pour déterminer les triplets équivalents.

La figure (3.10) représente le processus de découverte de liens entre les triplets comme suit :

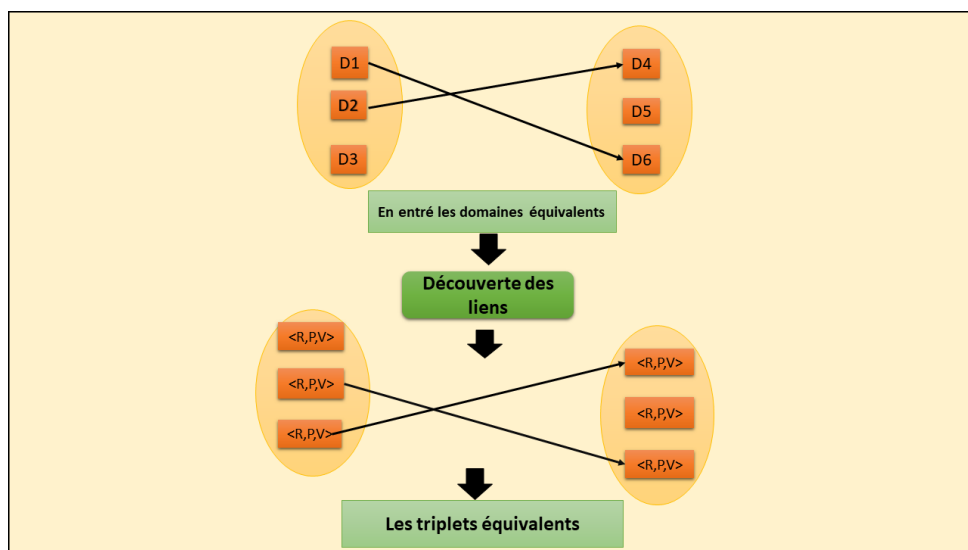


FIGURE 3.9 – Triplets équivalents.

Pour expliquer comment nous avons calculé les mesures de similarité. Nous avons profité d'utiliser la mesure structurelle interne et externe sur les propriétés, ainsi que la mesure extensionnelle sur les objets. Les résultats obtenus seront combinés avec une stratégie de combinaison des mesures de similarités, pour atteindre une seule valeur de similarité globale.

En examinant cette valeur de similarité globale par rapport à un seuil prédéfini, deux triplets sont considérés comme similaires (équivalents) ou différents. La figure ci-dessous résume toutes les mesures de similarité utilisées.

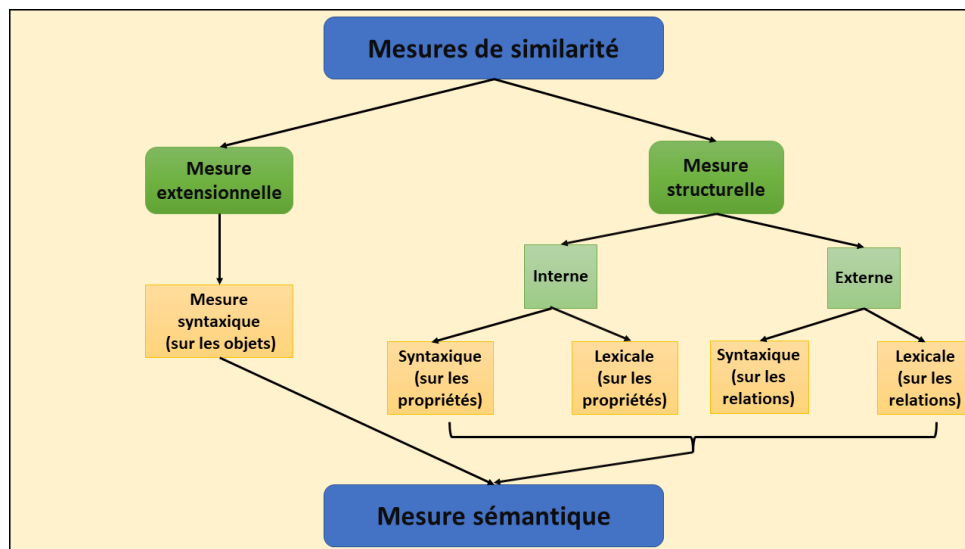


FIGURE 3.10 – Mesures de similarités.

## 1. Similarité structurelle

Dans cette étape, nous avons utilisé la similarité structurelle, afin de chercher les correspondances entre les triplets des deux datasets. Elle est décomposée en deux types de similarité interne et externe.

### a) Interne

Afin de calculer la similarité structurelle interne entre les propriétés. Nous allons utiliser les deux mesures suivantes :

#### i. Syntaxique

Pour évaluer la correspondance entre les triplets des deux datasets, nous avons appliqué la distance 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 la cardinalité de l'union des ensembles de comparaison et les cardinalités de l'union des ensembles.

$$Indice_{jaccard} = \left( \frac{|A \cap B|}{|A \cup B|} \right) \quad (3.2)$$

On considère A comme étant une propriété de dataset **yago** et B une propriété de dataset **DBpedia**.

La similarité entre A et B est calculée en traitant chaque caractère qui compose les deux triplets.

Le résultat de cette mesure est compris entre 0 et 1, plus la valeur est proche de 1 plus les paires de chaînes sont similaires. Si le résultat est égal à 1, cela signifie que les paires sont identiques.

#### **Exemple**

L'exemple d'une partie des deux dataset qui contiennent les différentes propriétés des données utilisés.

- propriété-yago = Alabama
- propriété-DBpedia = Bluetongue-disease .

Étant donné les deux propriétés composées ("Alabama" et "Bluetongue-disease"), la similarité syntaxique est calculée selon le principe ci-dessus :

Les caractères des deux propriétés composées respectifs sont :

Alabama : A,l,a,b,a,m,a = 7 caractère

Bluetongue-disease : B,l,u,e,t,o,n,g,u,e,d,i,s,e,a,s,e = 17 caractère

$$SimSyn = \left( \frac{|Alabama \cap Bluetongue|}{|Alabama \cup Bluetongue|} \right) \quad (3.3)$$

Donc :

**SimSyn = 0.2**

#### **ii. Lexicale**

La mesure lexicale consiste à utiliser des ressources externes (dictionnaires, taxonomies, etc.). Elles permettent de déterminer la similarité entre deux triplets. Ces triplets sont représentés par des termes. Plusieurs types de ressources peuvent être utilisés, notre choix s'est porté sur WordNet.

WordNet est une ressource lexicale de langue anglaise, disponible sur internet, qui regroupe des termes (noms, verbes, adjectifs et adverbes) en ensembles de synonymes appelés synsets.

WordNet contient des liens entre les synsets qui représentent plusieurs relations : est-un(e), fait-partie-de, synonymie, antonymie, hyponymie, homonymie etc.

L'objectif de cette mesure est de renvoyer un score indiquant à quel point les synsets de deux propriétés sont similaires.

Dans notre cas le calcul de la similarité lexicale se fait entre les propriétés des triplets de deux datasets.

La fonction  $Syn(P)$  calcule l'ensemble des Synsets de WordNet des triplets retenu soit :

$$S = syn(P1) \cap syn(P2) \quad (3.4)$$

L'ensemble des sens communs entre P1 et P2 à comparer, la cardinalité de S est :

$$\lambda(S) = |syn(P1) \cap syn(P2)|; \quad (3.5)$$

Soit :

$\min(|Syn(P1)|, |Syn(P2)|)$  le minimum entre les cardinalité des deux ensembles  $Syn(P1)$  et  $Syn(P2)$  alors la mesure de similarité lexicale entre deux Propriétés P1 et P2 est définit comme suit :

$$simlex(P1, P2) = \frac{\lambda(S)}{\min(|Syn(P1)|, |Syn(P2)|)} \quad (3.6)$$

### Exemple

Soit les deux propriétés des deux datasets respectivement DBpedia, YAGO :

- propriété-DBpedia = Gushikami
- propriété-yago = Anglesey

$SimLiX(Synset(Gushikami), Synset(Anglesey)) = 0.83$

Par la suite, nous avons combiné la mesure syntaxique et lexicale pour en avoir un résultat global, nous avons aussi ajouté un poids pour la mesure lexicale grâce à sa grande importance. Ce poids a été désigné grâce à des tests afin d'obtenir des meilleurs résultats d'alignement, comme le montre la formule suivante :

$$SimSem(P1, P2) = \frac{SimTer(P1, P2) + (2 * Simlexi(P1, P2))}{3} \quad (3.7)$$

## b) Externe

Afin de calculer la similarité structurelle externe entre les relations, nous allons utiliser les deux mesures suivantes :

### i. Syntaxique

Pour le calcul de la mesure syntaxique sur les relations. La valeur de cette relation elle doit être un URI c'est à dire un lien vers une autre ressource.

Nous avons appliqué l'indice de Jaccard sur les relations

### ii. Lexicale

Pour le calcul de la mesure lexicale, nous avons utilisé wordnet sur les relations.



## 2. Similarité extensionnelle

### a) Syntaxique

Pour calculer la mesure de similarité extensionnelle sur les objets, la valeur de l'objet doit être une valeur littérale. Nous avons utilisé l'indice de Jaccard.

L'indice de Jaccard est défini comme :

$$Indicejaccard = \left( \frac{|O1 \cap O2|}{|O1 \cup O2|} \right) \quad (3.8)$$

**Exemple** soit les deux objets des datasets respectivement YAGO, DBpedia :

- objet-yago = Ankara
- objet-DBpedia = Krna .

Étant donné les deux objets composés ("Ankara" et " Krna"). La similarité extensionnelle est calculée selon le principe ci-dessous :

Les caractères des deux objets composés respectifs sont :

Ankara : a,n,k,a,r,a = 6 caractère

Krna : k,r,n,a = 4 caractère

$$SimExte = \left( \frac{|Ankara \cap Krna|}{|Ankara \cup Krna|} \right) \quad (3.9)$$

Donc SimExte = 0.66

### 3.4.3 Phase 3 : Post-découverte

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

#### 3.4.3.1 Étape 1 : Combinaison des mesures

La similarité sémantique est une évaluation du lien sémantique afin d'estimer à quel degré deux triplets sont proches dans leurs sens, elle est calculée par la combinaison des similarités extensionnelle et structurelle, donc elle est définie selon la formule suivante :

$$SimSem(T1, T2) = \frac{SimExten(O1, O2) + SimStru(P1, P2)}{2} \quad (3.10)$$

#### 3.4.3.2 Étape 2 : Filtrage

Le filtrage consiste à déterminer une valeur de seuil bien précise de sorte qu'on garde que les triplets qui ont une valeur de similarité supérieure à la valeur du seuil et on élimine les triplets qu'ils ont une valeur inférieure.

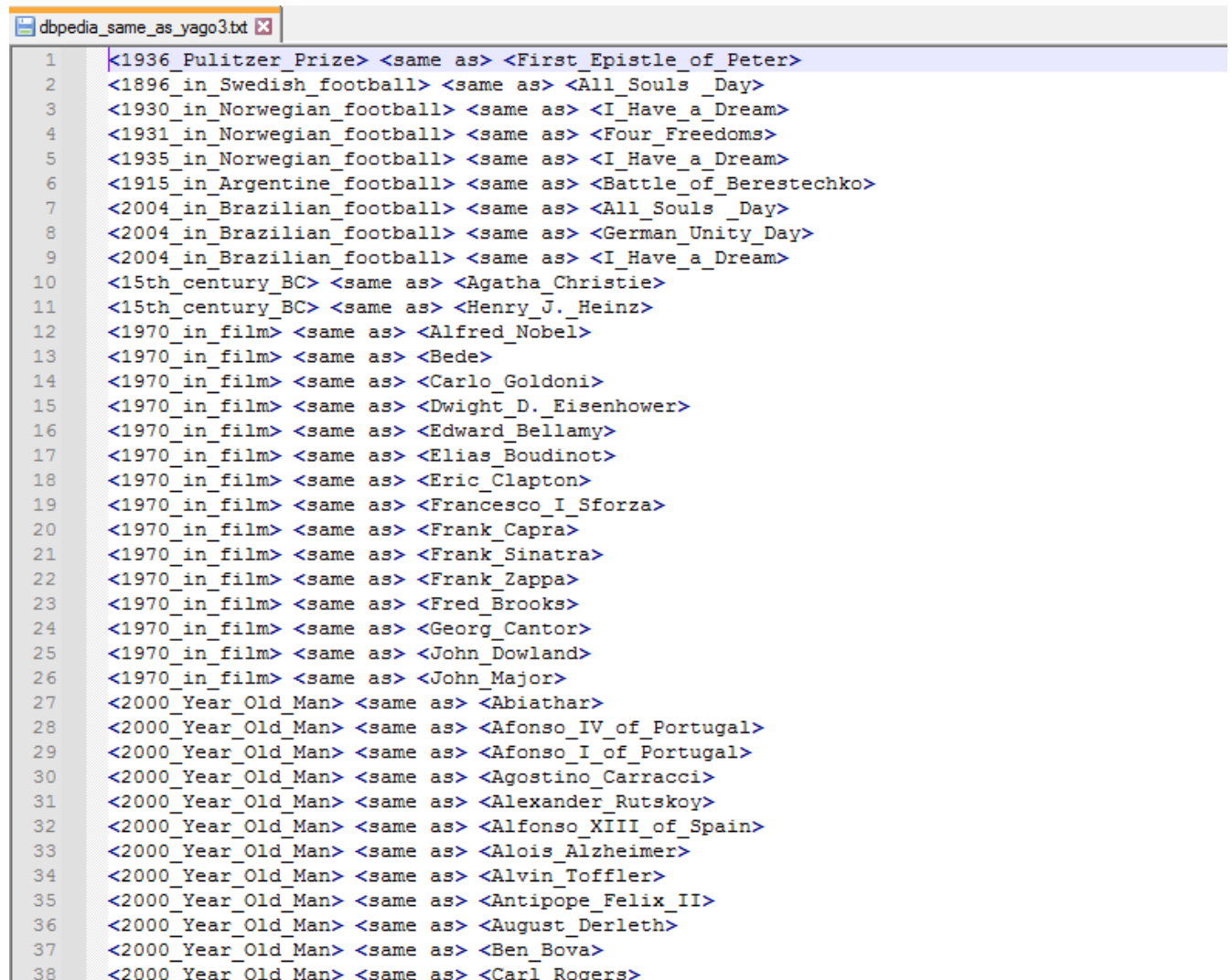
Nous obtenons donc un fichier des ressources les plus similaires. Le résultat sera par la suite décrit sous format RDF qui présente l'alignement des datasets.

dbpedia_same_as_yago1.txt	
1	<1936_Pulitzer_Prize> <same as> <First_Epistle_of_Peter> <0.6901881720430108>
2	<1896_in_Swedish_football> <same as> <All_Souls_Day> <0.7372685185185185>
3	<1930_in_Norwegian_football> <same as> <I_Have_a_Dream> <0.6543528934833283>
4	<1931_in_Norwegian_football> <same as> <Four_Freedoms> <0.742222222222221>
5	<1935_in_Norwegian_football> <same as> <I_Have_a_Dream> <0.653623188405797>
6	<1915_in_Argentine_football> <same as> <Battle_of_Berestechko> <0.6224350205198359>
7	<2004_in_Brazilian_football> <same as> <All_Souls_Day> <0.656084656084656>
8	<2004_in_Brazilian_football> <same as> <German_Unity_Day> <0.6639329805996472>
9	<2004_in_Brazilian_football> <same as> <I_Have_a_Dream> <0.7032967032967034>
10	<15th_century_BC> <same as> <Agatha_Christie> <0.6434782608695652>
11	<15th_century_BC> <same as> <Henry_J._Heinz> <0.6071428571428572>
12	<1970_in_film> <same as> <Alfred_Nobel> <0.6524195885250652>
13	<1970_in_film> <same as> <Bede> <0.608974358974359>
14	<1970_in_film> <same as> <Carlo_Goldoni> <0.700297287253809>
15	<1970_in_film> <same as> <Dwight_D._Eisenhower> <0.6086601307189543>
16	<1970_in_film> <same as> <Edward_Bellamy> <0.6135265700483092>
17	<1970_in_film> <same as> <Elias_Boudinot> <0.6400966183574879>
18	<1970_in_film> <same as> <Eric_Clapton> <0.6111111111111112>
19	<1970_in_film> <same as> <Francesco_I_Sforza> <0.6151053013798111>
20	<1970_in_film> <same as> <Frank_Capra> <0.6590909090909092>
21	<1970_in_film> <same as> <Frank_Sinatra> <0.6270009139574356>
22	<1970_in_film> <same as> <Frank_Zappa> <0.6570806100217865>
23	<1970_in_film> <same as> <Fred_Brooks> <0.672463768115942>
24	<1970_in_film> <same as> <Georg_Cantor> <0.6217948717948718>
25	<1970_in_film> <same as> <John_Dowland> <0.6333333333333333>
26	<1970_in_film> <same as> <John_Major> <0.6848484848484848>
27	<2000_Year_Old_Man> <same as> <Abiathar> <0.6091415830546265>
28	<2000_Year_Old_Man> <same as> <Afonso_IV_of_Portugal> <0.6155913978494624>
29	<2000_Year_Old_Man> <same as> <Afonso_I_of_Portugal> <0.617271505376344>
30	<2000_Year_Old_Man> <same as> <Agostino_Carracci> <0.8161616161616163>
31	<2000_Year_Old_Man> <same as> <Alexander_Rutskoy> <0.7120958751393534>
32	<2000_Year_Old_Man> <same as> <Alfonso_XIII_of_Spain> <0.6117424242424243>
33	<2000_Year_Old_Man> <same as> <Alois_Alzheimer> <0.6086074609753946>
34	<2000_Year_Old_Man> <same as> <Alvin_Toffler> <0.6283068783068784>
35	<2000_Year_Old_Man> <same as> <Antipope_Felix_II> <0.6247276688453158>
36	<2000_Year_Old_Man> <same as> <August_Derleth> <0.6066666666666667>
37	<2000_Year_Old_Man> <same as> <Ben_Bova> <0.9112903225806451>
38	<2000_Year_Old_Man> <same as> <Carl_Rogers> <0.6791526374859709>

FIGURE 3.11 – Le fichier des résultats RDF.

#### 3.4.3.3 Étape 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 3.11).



```
dbpedia_same_as_yago3.txt
1 <1936_Pulitzer_Prize> <same as> <First_Epistle_of_Peter>
2 <1896_in_Swedish_football> <same as> <All_Souls_Day>
3 <1930_in_Norwegian_football> <same as> <I_Have_a_Dream>
4 <1931_in_Norwegian_football> <same as> <Four_Freedoms>
5 <1935_in_Norwegian_football> <same as> <I_Have_a_Dream>
6 <1915_in_Argentine_football> <same as> <Battle_of_Berestechko>
7 <2004_in_Brazilian_football> <same as> <All_Souls_Day>
8 <2004_in_Brazilian_football> <same as> <German_Unity_Day>
9 <2004_in_Brazilian_football> <same as> <I_Have_a_Dream>
10 <15th_century_BC> <same as> <Agatha_Christie>
11 <15th_century_BC> <same as> <Henry_J_Heinz>
12 <1970_in_film> <same as> <Alfred_Nobel>
13 <1970_in_film> <same as> <Bede>
14 <1970_in_film> <same as> <Carlo_Goldoni>
15 <1970_in_film> <same as> <Dwight_D_Eisenhower>
16 <1970_in_film> <same as> <Edward_Bellamy>
17 <1970_in_film> <same as> <Elias_Boudinot>
18 <1970_in_film> <same as> <Eric_Clapton>
19 <1970_in_film> <same as> <Francesco_I_Sforza>
20 <1970_in_film> <same as> <Frank_Capra>
21 <1970_in_film> <same as> <Frank_Sinatra>
22 <1970_in_film> <same as> <Frank_Zappa>
23 <1970_in_film> <same as> <Fred_Brooks>
24 <1970_in_film> <same as> <Georg_Cantor>
25 <1970_in_film> <same as> <John_Dowland>
26 <1970_in_film> <same as> <John_Major>
27 <2000_Year_Old_Man> <same as> <Abiathar>
28 <2000_Year_Old_Man> <same as> <Afonso_IV_of_Portugal>
29 <2000_Year_Old_Man> <same as> <Afonso_I_of_Portugal>
30 <2000_Year_Old_Man> <same as> <Agostino_Carracci>
31 <2000_Year_Old_Man> <same as> <Alexander_Rutskoy>
32 <2000_Year_Old_Man> <same as> <Alfonso_XIII_of_Spain>
33 <2000_Year_Old_Man> <same as> <Alois_Alzheimer>
34 <2000_Year_Old_Man> <same as> <Alvin_Toffler>
35 <2000_Year_Old_Man> <same as> <Antipope_Felix_II>
36 <2000_Year_Old_Man> <same as> <August_Derleth>
37 <2000_Year_Old_Man> <same as> <Ben_Bova>
38 <2000_Year_Old_Man> <same as> <Carl_Rogers>
```

FIGURE 3.12 – Le fichier RDF.

## 3.5 Conclusion

Dans ce chapitre, nous avons présenté les étapes détaillées de notre solution pour la découverte des liens. En détaillant les trois phases : pré-traitement, et découverte sémantique, ainsi que les mesures des similarités utilisées, au final poste découverte.

Dans le chapitre suivant, nous allons implémenter et mettre en oeuvre ce que nous avons déjà proposé dans ce chapitre, autrement dit l'implémentation et la validation des résultats de notre système.



## Chapitre 4

---

# IMPLÉMENTATION ET TEST DU SYSTÈME

---

## 4.1 Introduction

Après avoir réalisé la phase de conception de notre système. Ce chapitre couvre tous les détails relatifs à l'aspect d'implémentation de notre approche qui s'appuie sur la découverte des liens définis dans le chapitre précédent.

Pour ce faire, nous avons tout d'abord présenté les différents environnements de développement et les différents outils utilisés. Puis nous décrivons de façon visuelle notre implémentation via des captures d'écran des différentes interfaces de notre système. Nous passons par la suite à la validation et les tests de notre système. À la fin nous montrons l'évaluation du mapping avec quelques résultats et discussions.

## 4.2 Environnement de développement

Avant de commencer l'implémentation de notre système de découverte des liens, nous allons spécifier les différents outils utilisés dans notre travail :

### 4.2.1 Matériel utilisé

Tout le travail et les tests présents dans ce mémoire ont été réalisés sur une machine puissante avec les spécifications suivantes :

**CPU** : AMD Ryzen™ 5 2400G avec processeur graphique Radeon™ RX Vega 11.

- Nombre de cœurs : 4
- Nombre de threads : 8
- Fréquence de base : 3.6Ghz
- Fréquence Boost maximale : 3.9Ghz

**RAM** : 16 Gb /DDR4

**SYSTÈME D'EXPLOITATION** : Windows 10 - Édition 64 bits

## 4.2.2 Langage utilisé

Nous avons utilisé le langage de programmation Python.

### 4.2.2.1 Python

Python est un langage de programmation de haut niveau interprété pour la programmation à usage général. Créé par Guido van Rossum, repose sur une philosophie de conception qui permette la lisibilité du code. Il fournit des constructions permettant une programmation claire.

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.

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 [39].

Le choix de ce langage présente les avantages suivants :

- Python est entièrement gratuit.
- Python est un langage complet et puissant dans de nombreux domaines.
- Python est orienté objet mais n'impose pas ce type de programmation.
- La syntaxe du python reste très simple et le code peut être très lisible.

### 4.2.2.2 Paquets utilisés dans Python

#### 1. 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.

#### 2. NLTK

NLTK (Natural Language Toolkit) 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 tutoriels, ainsi que la documentation de l'interface de programmation (API).

#### 3. WordNet

WordNet est une base de données lexicale pour la langue anglaise, créée par Princeton et faisant partie du corpus NLTK.

## 4. IMPLÉMENTATION ET TEST DU SYSTÈME

WordNet peut être utilisé avec le module NLTK pour trouver la signification des mots, des synonymes, des antonymes, etc.

### 4. Visual Studio Code

Est un éditeur de code open-source développé par Microsoft pour Windows et Linux qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, python, et C++ grâce à des extensions, et utilisé pour développer des applications Web.

### 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.

## 4.3 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 de découverte des liens, et ses différentes fonctionnalités.

La (figure 4.1) représente l'interface d'accueil de notre système.

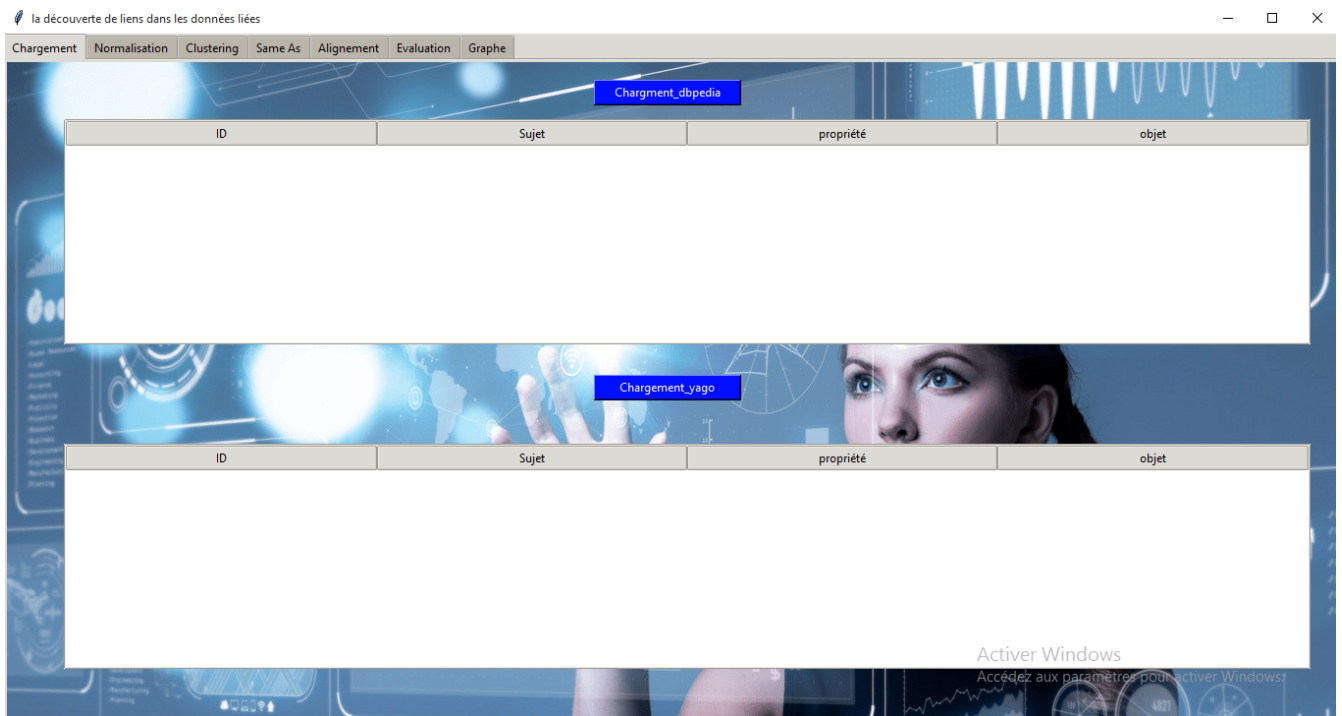


FIGURE 4.1 – L'interface d'accueil.

En cliquant sur les deux boutons (**chargement-yago**) et (**chargement-dbepedia**), le processus de chargement se lance et son résultat s'affiche illustré par la figure (4.2).

ID	Sujet	propriété	objet
339	<http://dbpedia.org/resource/1903_World_Series>	<http://dbpedia.org/resource/Bluetongue_disease>	<http://dbpedia.org/resource/Bruce_Perens>
3308	<http://dbpedia.org/resource/1941>	<http://dbpedia.org/resource/13th_century>	<http://dbpedia.org/resource/4th_century>
3319	<http://dbpedia.org/resource/100BaseVG>	<http://dbpedia.org/resource/10BASE2>	<http://dbpedia.org/resource/10BASE5>
3761	<http://dbpedia.org/resource/15th_century_BC>	<http://dbpedia.org/resource/L_C3%A9on_(film)>	<http://dbpedia.org/resource/American_Goldfinch>
4346	<http://dbpedia.org/resource/16_mm_film>	<http://dbpedia.org/resource/8_mm_film>	<http://dbpedia.org/resource/Anthony_Zinni>
5626	<http://dbpedia.org/resource/1938_FIFA_World_Cup>	<http://dbpedia.org/resource/Ideal_gas_law>	<http://dbpedia.org/resource/Blast_beat>
5628	<http://dbpedia.org/resource/1950_FIFA_World_Cup>	<http://dbpedia.org/resource/Willie_Davenport>	<http://dbpedia.org/resource/Mamo_Wolde>
6706	<http://dbpedia.org/resource/2000_Summer_Olympics>	<http://dbpedia.org/resource/Universal_Turing_machine>	<http://dbpedia.org/resource/Vaf%C3%BEr%C3%BA%C3%99>
10361	<http://dbpedia.org/resource/2002_Bali_bombings>	<http://dbpedia.org/resource/Jemaah_Islamiyah>	<http://dbpedia.org/resource/Outer_product>
21367	<http://dbpedia.org/resource/1990s_in_film>	<http://dbpedia.org/resource/Fulling>	<http://dbpedia.org/resource/Cai_Lun>

ID	Sujet	propriété	objet
0	<http://mpii.de/yago/resource/Abu_Dhabi>	<http://mpii.de/yago/resource/Alabama>	<http://mpii.de/yago/resource/Achilles>
1	<http://mpii.de/yago/resource/Abraham_Lincoln>	<http://mpii.de/yago/resource/Aristotle>	<http://mpii.de/yago/resource/An_American_in_Paris>
2	<http://mpii.de/yago/resource/Academy_Award>	<http://mpii.de/yago/resource/Actrius>	<http://mpii.de/yago/resource/Animalia_(book)>
3	<http://mpii.de/yago/resource/International_Atomic_Tim>	<http://mpii.de/yago/resource/Ang_Lee>	<http://mpii.de/yago/resource/Ayn_Rand>
4	<http://mpii.de/yago/resource/Alain_Connes>	<http://mpii.de/yago/resource/Allan_Dwan>	<http://mpii.de/yago/resource/Algeria>
5	<http://mpii.de/yago/resource/Characters_in_Atlas_Shrug>	<http://mpii.de/yago/resource/Atlas_Shrugged>	<http://mpii.de/yago/resource/Austria>
6	<http://mpii.de/yago/resource/American_Samoa>	<http://mpii.de/yago/resource/Amoeboid>	<http://mpii.de/yago/resource/ASCII>
7	<http://mpii.de/yago/resource/Apollo>	<http://mpii.de/yago/resource/Andre_Agassi>	<http://mpii.de/yago/resource/Austro-Asiatic_languages>
8	<http://mpii.de/yago/resource/Afro-Asiatic_languages>	<http://mpii.de/yago/resource/Andorra>	<http://mpii.de/yago/resource/Animal_Farm>
9	<http://mpii.de/yago/resource/Alaska>	<http://mpii.de/yago/resource/Aldous_Huxley>	<http://mpii.de/yago/resource/Analysis_of_variance>

FIGURE 4.2 – Chargement des datasets.

La figure ci-dessous représente l'interface de normalisation. Nous devons tout d'abord appuyer sur les deux boutons (**normalisation-dbepedia**) et (**normalisation-yago**), afin d'obtenir les résultats de la normalisation.

ID	Sujet	propriété	objet
0	1903 World Series	Bluetongue disease	Bruce Perens
1	1941	13th century	4th century
2	100BaseVG	10BASE2	10BASE5
3	15th century BC	L C3 A9on (film)	American Goldfinch
4	16 mm film	8 mm film	Anthony Zinni
5	1938 FIFA World Cup	Ideal gas law	Blast beat
6	1950 FIFA World Cup	Willie Davenport	Mamo Wolde
7	2000 Summer Olympics	Universal Turing machine	Vaf C3 BEr C3 BA C3 B0nr
8	2002 Bali bombings	Jemaah Islamiyah	Outer product
9	1990s in film	Fulling	Cai Lun

ID	Sujet	propriété	objet
0	Abu Dhabi	Alabama	Achilles
1	Abraham Lincoln	Aristotle	An American in Paris
2	Academy Award	Actrius	Animalia (book)
3	International Atomic Time	Ang Lee	Ayn Rand
4	Alain Connes	Allan Dwan	Algeria
5	Characters in Atlas Shrugged	Atlas Shrugged	Austria
6	American Samoa	Amoeboid	ASCII
7	Apollo	Andre Agassi	Austro Asiatic languages
8	Afro Asiatic languages	Andorra	Animal Farm
9	Alaska	Aldous Huxley	Analysis of variance

FIGURE 4.3 – Normalisation.



## 4. IMPLÉMENTATION ET TEST DU SYSTÈME

Après avoir normalisé les deux datasets, nous avons procédé aux différentes catégories par domaine pour chaque ressource voir (figure 4.4).

**Catého\_dbpedia**

ID	Sujet	Propriété	Objet	Domaine
678	1948 Pulitzer Prize	Polski Fiat 125p	Gallows Hill (novel)	Book
682	1983 Pulitzer Prize	Zain Mahmood	Chris Burnett Quartet	Book
743	1 New York Place	Ruggiero (music)	Mary Alden	Building
785	1743 English cricket season	Somebody Loves You (1932 song)	Concierto serenata	Building
973	1886 Mine Falls Gatehouse	Scrivener (software)	Buzzard (DC Comics)	Building
1095	190 Coltrin Road	Valkyrie (band)	Zahra Ahmadi	Building
469	18 Crown 6	Ascension of the Watchers	Nobuteru Taniguchi	ChemicalSubstance
504	1 Butyl 3 methylimidazolium hexafluorophosph	Her Sanity	Paweraa	ChemicalSubstance
522	2 Methylheptane	Lydia Rabinowitsch Kempner	Karmarkar s algorithm	ChemicalSubstance
565	1,2,4 Trimethylbenzene	Tour D C3 Afense 2000	1980 24 Hours of Le Mans	ChemicalSubstance

**Catého\_yago**

ID	Sujet	Propriété	Objet	Domaine
937	Epistle to Philemon	Elliptic curve cryptography	Eightfold Path (policy analysis)	Book
1075	First Epistle of Peter	First Epistle of John	File format	Book
1176	Gospel of Luke	Gospel of Matthew	Gospel of John	Book
293	Athlon	Amnon	Amu Darya	Brand
750	Datsun	Douglas Coupland	David Fincher	Brand
810	DKW	Doctor Syn	Dhystone	Brand
1287	Holden	Hank Greenberg	Heinrich Schliemann	Brand
1299	Honda	Team handball	Hilbert s basis theorem	Brand
410	BBC	BBC Radio 1	BBCi	BroadcastService
594	Channel 4	Carolina Parakeet	Church (building)	BroadcastService

FIGURE 4.4 – Catégorisation par domaine.

En cliquant sur le bouton (**DBpedia same as yago**), l'interface ci-dessous n'affiche que la représentation que des domaines équivalents dans les deux datasets (figure 4.5).

**DBpedia same as yago**

ID	Sujet	Propriété	Objet	Domaine
172	1893 in film	Chile at the 2004 Summer Olympics	DD Smash	Book
297	1890 in film	Abad C3 Adn	Anderson Gray McKendrick	Book
376	1001 Ways to Beat the Draft	WBAP	Kuickshow	Book
408	1924 Pulitzer Prize	Montebello, Antioquia	Mutat C3 A1	Book
412	1927 Pulitzer Prize	Gurston Down Motorsport Hillclimb	...Or Stay Tuned	Book
578	1933 Pulitzer Prize	Steve Slaton	Kerala Kaumudi	Book
646	1986 Pulitzer Prize	Daron McFarland	Channel 3 (Israel)	Book
677	1936 Pulitzer Prize	Biological pigment	Summy Airport	Book
678	1948 Pulitzer Prize	Polski Fiat 125p	Gallows Hill (novel)	Book
682	1983 Pulitzer Prize	Zain Mahmood	Chris Burnett Quartet	Book

ID	Sujet	Propriété	Objet	Domaine
45	And Then There Were None	Hercule Poirot	Miss Marple	Book
176	Acts of the Apostles	Assyria	Abijah	Book
408	Book of Joshua	Book of Ezra	Book of Daniel	Book
412	Book of Nehemiah	Book of Jeremiah	Book of Isaiah	Book
416	Book of Ruth	Book of Esther	British Rail	Book
417	Book of Job	Book of Proverbs	Book of Lamentations	Book
418	Book of Ezekiel	Big Brother (TV series)	Bristol City F.C.	Book
425	Book of Hosea	Book of Obadiah	Book of Jonah	Book
426	Book of Micah	Book of Nahum	Book of Haggai	Book
427	Book of Malachi	Book of Zechariah	Book of Zephaniah	Book

FIGURE 4.5 – Les mêmes domaines.

La figure (4.6) représentant les résultats obtenus après le calcul des différentes mesures de similarités entre ressources, ainsi que la similarité globale.

propriété_dbpeida	propriété_yago	similarité_syntaxique	similarité_lexical	similarité_structurale
1969 in film	Jabal Ram	0.16666666666666666	0.5263157894736842	0.4064327485380117
1969 in film	Javelin throw	0.19047619047619047	0.5714285714285714	0.4444444444444444
1969 in film	Jazz dance	0.1	0.75	0.5333333333333333
Gushikami, Okinawa	Anglesey	0.18181818181818182	0.8333333333333334	0.6161616161616162
Gushikami, Okinawa	Battle of Okinawa	0.25	1.0	0.75
Gushikami, Okinawa	Flag of Greenland	0.16666666666666666	0.8333333333333334	0.6111111111111112
Gushikami, Okinawa	Foreign relations of Japan	0.1891891891891892	0.6666666666666666	0.5075075075075075
1913 Indianapolis 500	Age Khan III	0.13793103448275862	0.75	0.5459770114942529
1913 Indianapolis 500	Agessilaus II	0.17857142857142858	0.75	0.5595238095238095
1913 Indianapolis 500	Ahmed III	0.15384615384615385	0.75	0.5512820512820512

objet_dbpeida	objet_yago	similarité
American Goldfinch	Joseph Conrad	0.4090909090909091
Tamagusuku, Okinawa	Running amok	0.4090909090909091
Falsterbo Canal	Albertus Magnus	0.42857142857142855
Falsterbo Canal	Alexander I of Scotland	0.4074074074074074
Falsterbo Canal	Alfonso the Battler	0.4782608695652174
Falsterbo Canal	Amalric of Bena	0.5
Falsterbo Canal	Albert Camus	0.5
Falsterbo Canal	Baralong Incident	0.45454545454545453
Falsterbo Canal	Barcelonnette	0.47368421052631576
Falsterbo Canal	Battle of Ramillies	0.4166666666666667

objet_dbpeida	objet_yago	similarité_globale
1936_Pulitzer_Prize	First_Epistle_of_Peter	0.6901881720430108
1896_in_Swedish_football	All_Souls_Day	0.7372685185185185
1930_in_Norwegian_football	I_Have_a_Dream	0.6543528934833283
1931_in_Norwegian_football	Four_Freedoms	0.7422222222222221
1935_in_Norwegian_football	I_Have_a_Dream	0.653623188405797
1915_in_Argentine_football	Battle_of_Berestechko	0.6224350205198359
2004_in_Brazilian_football	All_Souls_Day	0.656084656084656
2004_in_Brazilian_football	German_Unity_Day	0.6639329805996472
2004_in_Brazilian_football	I_Have_a_Dream	0.7032967032967034
15th_century_BC	Agatha_Christie	0.6434782608695652

FIGURE 4.6 – Résultats des mesures de similarités.

La figure (4.7) représente les résultats obtenus par les mesures de performance.

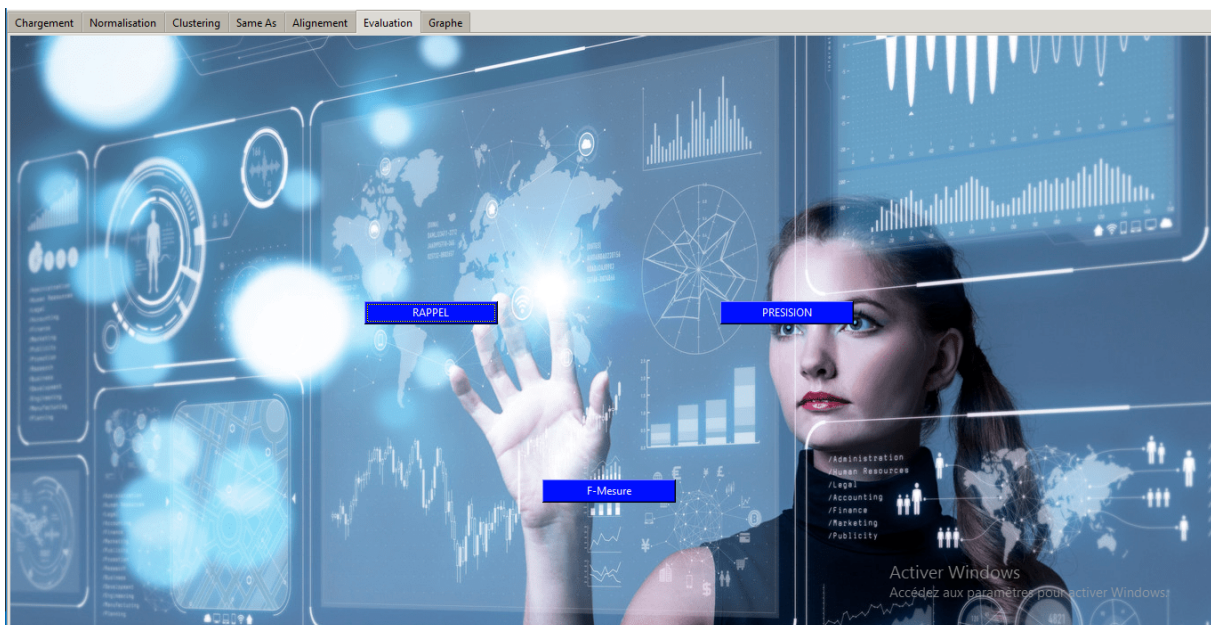


FIGURE 4.7 – Évaluation.

## 4. IMPLÉMENTATION ET TEST DU SYSTÈME

Le code ci-dessous, représente comment nous avons calculé la similarité globale.

```
def jacc_sim_sementique():
    # coefficient
    coefjac = 1
    coefwn = 2
    # Connecting to sqlite,
    conn = sqlite3.connect('bdd/semantic.db')

    # Creating a cursor object using the cursor() method
    cpt = 0

    sameasfile = open("dbpedia_same_as_yago.txt", "w")

    cur2 = conn.cursor()

    cur2.execute(
        "SELECT distinct * FROM dbpedia_yago where domaine not like '' order by domaine")

    rowsdbpedia = cur2.fetchall()
    # dbsujet, dbprop, dbobj, ygsujet, ygprop, ygobj, domaine

    for rowd in rowsdbpedia:
        mot1 = normalizer(rowd[1])
        mot2 = normalizer(rowd[4])
        jac = jaccard(mot1, mot2)
        wnet = wn_sim(mot1, mot2)
        mot3 = normalizer(rowd[2])
        mot4 = normalizer(rowd[5])
        jac2 = jaccard(mot3, mot4)
        glob = (((coefjac * float(jac)+coefwn * float(wnet)) /
                (coefjac+coefwn))+float(jac2))/2
```

FIGURE 4.8 – Code de calcul de la similarité globale.

### 4.4 Test du système

Afin de tester la fiabilité et la validité de notre système. Nous avons choisi deux datasets **yago** et **DBpedia**. Nous avons comparé les correspondances obtenus avec le fichier d'équivalence qui représente les correspondances de l'expert du domaine.

#### 4.4.1 Résultats expérimentaux et discussion

Dans cette section nous présentons les résultats obtenus par les différentes mesures de similarité. Nous avons testées et utilisées pour la découverte de liens entre les différentes ressources.

Nous avons appliqué la mesure syntaxique sur les propriétés, de même pour la mesure l'exicale. Nous avons appliqué la mesure extensionnelle sur les objets. À la fin nous avons utilisé la combinaison des deux mesures de similarités structurelle et extensionnelle pour en avoir une mesure globale.

Les résultats obtenus sont mentionnés dans les tableaux ci-dessous.

### 1. La similarité entre les propriétés des triplets des deux datasets "DBpedia" et "yago"

Propriété-DBpedia	Propriété-yago	Syntaxique	Lexicale	Structurelle
Andrew Matthews	Bunge Born	0.20	0.57	0.44
Andrew Matthews	Cy Young	0.12	0.66	0.48
County School	Arkanas	0.09	0.75	0.53
County School	Centaurus	0.25	0.77	0.59
Descanso Gardens	Abdera	0.25	0.75	0.58
Descanso Gardens	Arkansas	0.20	0.52	0.42

TABLE 4.1 – Les résultats des similarités obtenus entre les propriétés des deux datasets.

- Concernant la mesure lexicale, nous avons utilisé "WorNet", d'après les résultats mentionnés dans le tableau ci-dessus nous pouvons remarquer que la mesure "lexicale" montre des résultats plus élevés par rapport à la mesure "syntaxique", nous prenons l'exemple de "County scool" et "Centaurus", la mesure syntaxique donne 0.25 et la mesure lexicale donne 0.77 cela veut dire que les résultats de la mesure lexicale sont pertinent par rapport la mesure syntaxique, vu que ces dernières utilisent wordnet.
- Nous avons obtenus les résultats de la mesure "structurelle" par la combinaison de la mesure syntaxique et lexicale.

### 2. La similarité entre les objets des triplets des deux datasets "DBpedia" et "yago"

Concernant la mesure de similarité extensionnelle, nous avons appliqué la mesure de similarité syntaxique sur les objets.

Objet-DBpedia	Objet-yago	Extensionnelle
American Goldfinch	File archiver	0.40
American Goldfinch	Andronicus of Rhodes	0.4
American Goldfinch	Amalric of Bena	0.5
American Goldfinch	Adlof Eicmanm	0.6
American Goldfinch	Gregory Chaitin	0.43
American Goldfinch	Film genre	0.47

TABLE 4.2 – Les résultats des similarités obtenus entre les objets des deux datasets.

### 3. La similarité globale en combinant la mesure structurelle et extensionnelle

Pour le calcul de la similarité globale (sémantique) entre les différentes ressources. Nous avons combiné les résultats obtenus par les deux mesures structurelles et extensionnelle.

Les résultats de la similarité globale sont mentionnés dans le tableau ci-dessous :

Sujet-dbpedia	Sujet-yago	Similarité globale
Pulitzer-Prize	First-Epistle-of-Peter	0.69
Brazilian-football	All-Souls -Day	0.65
film	Carlo-Goldoni	0.7
Year-Old-Man	Ben-Bova	0.9
Year-Old-Man	Agostino-Carracci	0.8

TABLE 4.3 – Similarité globale.

#### 4.4.2 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 :

$$TP = S \cap H \quad (4.1)$$

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

$$FP = S - S \cap H \quad (4.2)$$

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

$$FN = H - S \cap H \quad (4.3)$$

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

$$\text{Precision} = \frac{|TP|}{|TP| + |FP|} \quad (4.4)$$

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

$$\text{Rappel} = \frac{|TP|}{|TP| + |FN|} \quad (4.5)$$

Les valeurs obtenues par le calcul des correspondances entre les datasets ne sont comparables par le biais de la précision et du rappel. Le rappel peut prendre des valeurs importantes aux dépens de la précision, en retournant toutes les correspondances possibles.

En même temps, la précision peut prendre des valeurs importantes aux dépens du rappel, en retournant que les correspondances correctes cependant peu nombreuses.

C'est pour ces raisons qu'il est préférable de prendre en considération les deux mesures simultanément via une mesure qui combine le rappel et la précision telles que : la F-mesure qui se calcule de la manière suivante :

$$\text{F - mesure} = \frac{2 * (\text{Precision} * \text{Rappel})}{(\text{Precision} + \text{Rappel})} \quad (4.6)$$

• **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 parcourt les résultats obtenus par la similarité globale qui ont un degré de similarité élevé à un seuil  $> 0.65$ .

Nous avons comparé nos résultats avec les résultats du fichier de référence pour calculer le rappel et la précision et F-mesure.

### **Rappel**

$$\text{Rappel} = \frac{107}{107 + 0} = 1 \quad (4.7)$$

### **Précision**

$$\text{Precision} = \frac{107}{107 + 7} = 0.938 \quad (4.8)$$

### **F-mesure**

$$\text{F - mesure} = \frac{2 * (1 * 0.938)}{1 + 0.938} = 0.97 \quad (4.9)$$

## 4. IMPLÉMENTATION ET TEST DU SYSTÈME

Pour une meilleure lisibilité et visualisation des résultats, nous avons illustré grâce à l'histogramme suivant :

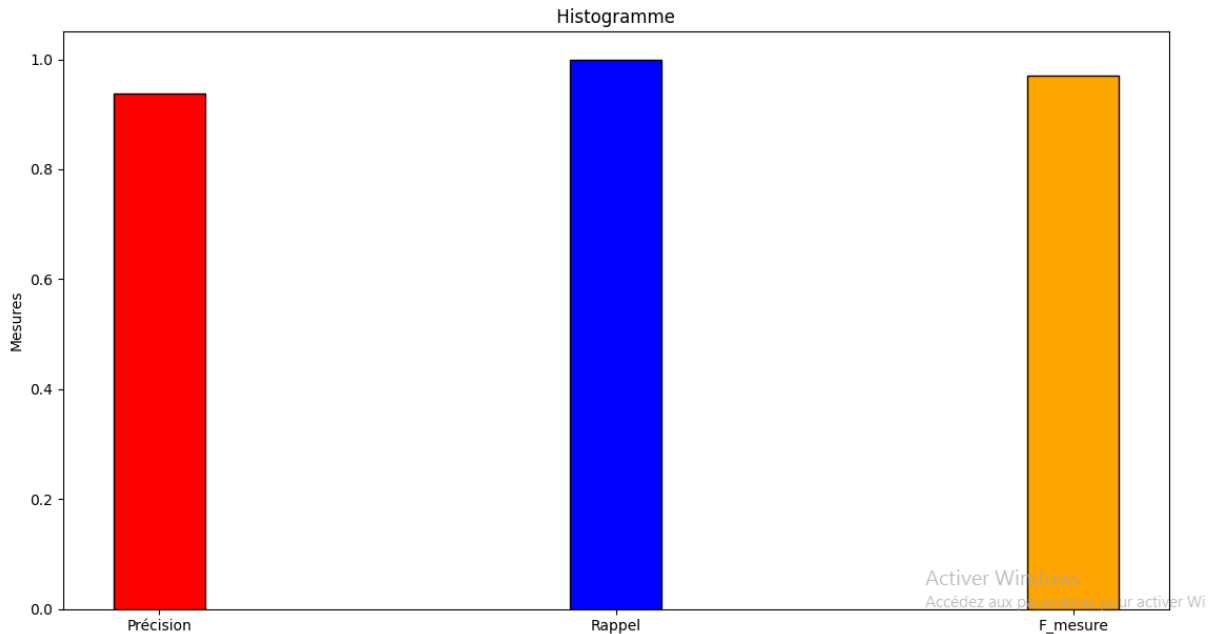


FIGURE 4.9 – Mesure de performance

D'après la figure (4.9) nous pouvons voir que notre système donne de très bons résultats pour les dataset utilisées pour ce test.

### 4.4.3 Interprétation des résultats

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

- La valeur de rappel obtenu par notre système restitue un meilleur résultat avec un score de 1, ce qui signifie que les résultats sont à 100% complets.
- En termes de précision, notre système confirme son niveau de performance obtenant une valeur de 0.93, ce qui veut dire qu'il a trouvé 93% des réponses pertinentes possibles.
- Concernant la F-mesure elle obtient un résultat de 0.97 signifie que 97% des réponses pertinents, le taux de F-mesure est élevé grâce aux taux élevés de Précision et Rappel.

## 4.5 Conclusion

Dans ce chapitre nous avons décrit les outils et l'environnement utilisés pour le développement de notre système. Nous avons aussi présenté l'interface graphique.

Nous avons également présenté les différentes mesures d'évaluation du notre système (Rappel, Précision, F-mesure), et nous avons clôturé le chapitre par une interprétation des résultats obtenus.





---

# CONCLUSION GÉNÉRALE ET PERSPECTIVES

---

Le web de données permet de publier des données structurées et non structurées sur le web, non pas sous la forme de silos de données isolés les uns des autres, mais en les reliant pour constituer un réseau d'informations global. Les données liées visent à partager et à interconnecter des données structurées sur le web selon les principes des données liées, sous forme d'une représentation lisible par la machine pour former un seul espace de données global.

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans le domaine de l'ingénierie des connaissances et plus particulièrement web de données liées. Notre objectif a été de chercher et proposer une solution découverte des liens des données liées.

Dans le premier chapitre, nous sommes intéressés aux web de données et les données liées. Pour cela, nous sommes partis des définitions du web de données. Par la suite, nous avons mentionnés les composants d'une ontologie, les langages d'ontologies et domaine d'application des données liées. Nous avons aussi parlé à propos de processus découverte de correspondance est l'objectif principal de ce travail.

Le deuxième chapitre a été consacré aux mesures de similarités ainsi qu'aux différentes méthodes de découverte des liens dans le contexte des données liées qui existent et qui ont traité la découverte des liens dans leur système d'alignement. Cette partie a été finie par une étude comparative en étudiant tous les points qui peuvent servir à mener notre travail.

Dans le troisième chapitre, afin de répondre à notre objectif, nous avons proposé une solution de découverte des liens qui permet de traiter les différentes données. Cette solution se base sur une combinaison des mesures de similarité syntaxique, lexicale et structurelle, extensionnelle et sémantique pour obtenir des bons résultats de découverte des liens.

Sur la base de cette solution, nous avons implémenté notre système de découverte des liens afin de qualifier le travail effectué et de montrer l'efficacité de notre solution avec des dataset de différents et volumineux, afin de qualifier le travail effectué et de démontrer l'efficacité de notre la solution dans la production de bonne qualité, des tests sur les différentes dataset générées ont été effectués en utilisant les mesures d'évaluation "Rappel" et "Précision" et "F-mesure".

Enfin, on a eu quelques perspectives tel que l'utilisation d'autres mesures de similarité terminologie et structurelles, afin d'améliorer la qualité du mapping, et aussi de prendre en considération le temps d'exécution, vu que le temps d'exécution est un point important dans le système de mapping et surtout avec un traitement en ligne.

---

# Bibliographie

---

- [1] Aghaei, S., Nematbakhsh, M. A., and Farsani, H. K. (2012). Evolution of the world wide web : From web 1.0 to web 4.0. *International Journal of Web & Semantic Technology*, 3(1) :1–10.
- [2] Amiar, S. (2017). Techniques et outils pour l ’ alignement d ’ ontologies Remerciements.
- [3] Antelme, D., Delestre, N., and Malandain, N. (2019). Un schéma owl pour la description d ’ éléments de formation. In *Environnements Informatiques pour l ’ Apprentissage Humain*.
- [4] Barbosa, A., Bittencourt, I. I., Siqueira, S. W., Dermeval, D., and Cruz, N. J. (2022). A context-independent ontological linked data alignment approach to instance matching. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 18(1) :1–29.
- [5] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3) :154–165.
- [6] Bleiholder, J. and Naumann, F. (2009). Data fusion. *ACM computing surveys (CSUR)*, 41(1) :1–41.
- [7] BOUNEFFA co directeur, M. (2017). Ontologies et web sémantique pour une construction évolutive d’applications dédiées à la logistique.
- [8] Collignon, A. and Cuxac, P. (2017). Istex : des enrichissements au web de données. *I2D-Information, donnees documents*, 54(4) :8–15.
- [9] Da Sylva, L. (2017). Les données et leurs impacts théoriques et pratiques sur les professionnels de l’information. *Documentation et bibliothèques*, 63(4) :5–34.
- [10] Elbyed, A. (2009). *ROMIE, une approche d’alignement d’ontologies à base d’instances*. PhD thesis, Evry, Institut national des télécommunications.

- 
- [11] Euzenat, J., Loup, D., Touzani, M., and Valtchev, P. (2004). Ontology alignment with ola. In *Proc. 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON)*, pages 59–68. No commercial editor.
- [12] Euzenat, J. and Shvaiko, P. (2007). Basic techniques. *Ontology Matching*, pages 73–116.
- [13] Grau, B. C., Dragisic, Z., Eckert, K., Euzenat, J., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A. O., Lambrix, P., et al. (2013). Results of the ontology alignment evaluation initiative 2013. In *OM : Ontology Matching*, pages 61–100. No commercial editor.
- [14] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220.
- [15] Guha, R. V., Brickley, D., and Macbeth, S. (2016a). Schema. org : evolution of structured data on the web. *Communications of the ACM*, 59(2) :44–51.
- [16] Guha, R. V., Brickley, D., and Macbeth, S. (2016b). Schema. org : evolution of structured data on the web. *Communications of the ACM*, 59(2) :44–51.
- [17] Heath, T. and Bizer, C. (2011a). Linked data : Evolving the web into a global data space. *Synthesis lectures on the semantic web : theory and technology*, 1(1) :1–136.
- [18] Heath, T. and Bizer, C. (2011b). Linked data : Evolving the web into a global data space. *Synthesis lectures on the semantic web : theory and technology*, 1(1) :1–136.
- [19] Hunter, J., Drennan, J., and Little, S. (2004). Realizing the hydrogen economy through semantic web technologies. *IEEE Intelligent Systems*, 19(1) :40–47.
- [20] JAN+, S., KHAN, I., SHAH, I., AHMAD, G., KHATTAK, A., and AL-SULTANY, G. (2013). An overview of concept comparison practices used in ontology alignment. *Sindh University Research Journal-SURJ (Science Series)*, 45(1).
- [21] Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406) :414–420.
- [22] Klein, M. C. and Fensel, D. (2001). Ontology versioning on the semantic web. In *SWWS*, pages 75–91.
- [23] Klyne, G., Carroll, J. J., and McBride, B. (2004). Resource description framework (rdf) : Concepts and abstract syntax. w3c recommendation, feb. 2004.
- [24] Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet : An electronic lexical database*, 49(2) :265–283.

- [25] Liu, Y., Chen, S.-H., and Gu, J.-G. (2015). Property alignment of linked data based on similarity between functions. *International Journal of Database Theory and Application*, 8(4) :191–206.
- [26] McCrae, J. P. and Buitelaar, P. (2018). Linking datasets using semantic textual similarity. *Cybernetics and information technologies*, 18(1) :109–123.
- [27] Mellal, N. (2007). *Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information*. PhD thesis, Chambéry.
- [28] Mestiri, M. A. (2007). Vers une approche web sémantique dans les applications de gestion de conférences.
- [29] Miller, E. et al. (2001). The w3c semantic web activity. *Semantic Web Activity Statement*. *Disponível*.
- [30] Monge, A. E., Elkan, C., et al. (1996). The field matching problem : algorithms and applications. In *Kdd*, volume 2, pages 267–270.
- [31] Ngomo, A.-C. N. and Auer, S. (2011a). Limes—a time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- [32] Ngomo, A.-C. N. and Auer, S. (2011b). Limes—a time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- [33] Nikolov, A., d'Aquin, M., and Motta, E. (2012). Unsupervised learning of link discovery configuration. In *Extended Semantic Web Conference*, pages 119–133. Springer.
- [34] Rissland, E. L. (2006). Ai and similarity. *IEEE Intelligent Systems*, 21(03) :39–49.
- [35] Scharffe, F., Liu, Y., and Zhou, C. (2009). Rdf-ai : an architecture for rdf datasets matching, fusion and interlink. In *Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US)*, page 23.
- [36] Seaborne, A., Manjunath, G., Bizer, C., Breslin, J., Das, S., Davis, I., Harris, S., Idehen, K., Corby, O., Kjernsmo, K., et al. (2008). Sparql update. *A language for updating RDF graphs. Member submission, W3C*.
- [37] Slimane, A. (2017). *Techniques et outils pour l'alignement d'ontologies*. PhD thesis, Université Mouloud Mammeri.
- [38] Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering : principles and methods. *Data & knowledge engineering*, 25(1-2) :161–197.

- [39] Van Rossum, G. and Drake Jr, F. L. (1995). *Python tutorial*, volume 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.
- [40] Villata, S., Delaforge, N., Gandon, F., and Gyrard, A. (2011). An access control model for linked data. In *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pages 454–463. Springer.
- [41] Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. (2009). Discovering and maintaining links on the web of data. In *International Semantic Web Conference*, pages 650–665. Springer.
- [42] Winkler, A. (2006). Fatal in theory and strict in fact : An empirical analysis of strict scrutiny in the federal courts. *Vand. L. Rev.*, 59 :793.
- [43] Wu, Z., Eadon, G., Das, S., Chong, E. I., Kolovski, V., Annamalai, M., and Srinivasan, J. (2008). Implementing an inference engine for rdfs/owl constructs and user-defined rules in oracle. In *2008 IEEE 24th International Conference on Data Engineering*, pages 1239–1248. IEEE.
- [44] Ziani, M., Boulanger, D., and Talens, G. (2010). Système d’aide à l’alignement d’ontologies métier-application au domaine géotechnique. In *Congrès INFORSID 2010*, pages p345–360.
- [45] Ziani, M., Boulanger, D., and Talens, G. (2011). Système d’aide à l’alignement d’ontologies métier. *Ingénierie des systèmes d’information*, 16(1) :89–112.