N°d'ordre: .....

République Algerienne Démocratique et Populaire
Ministére de l'enseignement supérieur et de la Recherche scientifique

Université Djillali Liabès de Sidi Bel Abbès
Faculté des Sciences Exactes
Département d'informatique

Thèse de Doctorat en Sciences

Filière : Informatique
Spécialité : Intéropérabilité et intégration des systèmes d'information dans le Web

Par

Mm ARDJANI Fatima

Evolution des Données Liées : Maintenance des liens

Soutenue le 14/12/2017 devant le jury :

Dr.	Sofiane BOUKLI HACENE	UDL SBA	Président du jury
Dr.	Reda ADJOUDJ	UDL SBA	Examinateur
Dr.	Moussa ALI CHERIF	UDL SBA	Examinateur
Pr.	Sidi Mohammed BENSLIMANE	ESI-SBA	Examinateur
Dr.	Djelloul BOUCHIHA	CU-Naâma	Directeur de thèse
Pr.	Mimoun MALKI	ESI-SBA	Co-Directeur de thèse

Année Universitaire : 2017/2018

A mon mari Rafik
Cher fils Louei et ma petite fille Loudjeine…

Remerciements

≺ Ce n'est pas de vivre selon la science qui procure le bonheur ; ni même de réunir toutes les sciences à la fois, mais de posséder la seule science du bien et du mal.≻ [**Platon**]

Je tiens tout d'abord, à remercier Dieu le Tout Puissant qui m'a donné le privilège d'étudier et de suivre le chemin de la science.

Je remercie **Dr Sofiane BOUKLI HACENE**, d'avoir accepté de présider le jury de ma soutenance.

Je remercie également **Pr Sidi Mohammed BENSLIMANE** et **Dr Reda ADJOUDJ**, **Dr Moussa ALI CHERIF** d'avoir accepté d'être mes rapporteurs.

Merci à mon directeur de thèse **Dr Djelloul BOUCHIHA** pour ses remarques éclairées et son encadrement tout au long de cette thèse.

Merci à mon co-directeur de thèse **Pr Mimoun MALKI** d'avoir accepté de diriger mon travail.

Sincèrement merci à tous mes collègues du centre universitaire de Naâma pour leur accueil et leur soutien tout au long de ces 4 ans, avec une mention spéciale sans ordre particulier à Bouagada Benamer, Abdelghani Bouziane, Yasser Yahiaoui, Arbaoui nadjia et à ceux dont le nom m'échappe au moment ou j'écris ces lignes.

Merci évidemment à ma grande et petite famille, pour tout.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Liste des listings

Introduction 1

Sommaire

1.1 Contexte

Le Web de données a été conçu pour étendre le Web par des données structurées partagées. Son idée de base, exprimée par Tim Berners-Lee en 2001 dans [T.Berners-Lee *et al.* 2001], est inspirée de la structure des pages Web - liées entre elles par des liens hypertextes - pour proposer une nouvelle représentation uniformisée de données, exploitable aussi bien par l'humain que par la machine.

Le Web de données se fonde sur le framework RDF (Resource Description Framwork) qui représente les données sous forme de triplets. Un triplet est composé de trois éléments : sujet, prédicat et objet. Ces triplets posent en relation des ressources RDF qui désignent des ressources du Web, du monde réel ou des concepts généraux. Chaque ressource RDF a un identifiant unique URI (Uniform Resource Identifier, identifiant de ressource uniforme) d'une page Web en relation avec la ressource. Les langages RDFS (RDF Shema) et OWL (Ontology Web Language) sont utilisés pour organiser les ressources RDF en classes hiérarchisées et pour définir les relations qui peuvent lier les ressources.

Ces langages permettent également de faire des inférences à partir de données RDF en se basant sur les logiques de description. L'utilisation des langages RDFS et OWL pour structurer les données et leur organisation en triplets pour former des graphes de données rend les bases RDF plus simples que les bases de données relationnelles. Les bases RDF sont interrogeables par les requêtes du langage SPARQL.

La forme la plus tangible du Web de données est le Linked Data , apparu en 2008, en même temps que SPARQL. Il s'agit d'une collection de bases qui concernent des domaines variés et qui suivent des règles communes de structuration. Ces bases sont reliées entre elles par des relations d'équivalence entre ressources RDF représentants les mêmes éléments dans des bases différentes.

Les ressources RDF des bases du Linked Data sont toutes associées à des pages Web lisibles par l'utilisateur et fournissent toutes un moyen d'accès à leur contenu.

L'initiative "Données Liées" (Linked Data) vise à publier des données structurées et liées sur le Web en utilisant les technologies du Web sémantique. Ces technologies offrent différents langages pour exprimer les données sous forme de graphes RDF et les interroger en SPARQL .

Le Web de données permet la création des bases RDF et des services basés sur les données RDF. RDF "Resource Description Framework" est un modèle de données simple pour la représentation de connaissances sur le Web. RDF sert à décrire des ressources RDF. Chaque ressource RDF est unique dans tout le linked data. Un triplet RDF exprime une relation entre un sujet et un objet, c'est-à-dire un triplet décrit une propriété du sujet ayant pour valeur l'objet du triplet. Le sujet est une ressource, identifiée explicitement par un URI. Un URI - Uniform Resource Identifier- est un identifiant unique sur le web. La relation est toujours une ressource identifiée (URI) ; L'objet est soit une ressource (identifiée ou non), soit une donnée brute, aussi appelée littéral. La base doit contenir des liens vers d'autres bases du Linked Data.

La liaison décrit la relation entre deux ressources et consistent en trois

références d'URI. Les URI dans le sujet et l'objet du lien identifient les ressources liées. L'URI du prédicat définit le type de relation entre les ressources. On a deux types de lien RDF, les liens RDF internes, c'est à dire les liens décrits dans la même base "les liens **intra-Base**" et les liens RDF externes, c'est à dire les liens décrits entre un ensemble de bases "les liens **inter-Base**". Les liens internes connectent des ressources dans une seule source de données liées. Ainsi, les URI des sujets et des objets sont dans le même espace de noms. Les liens externes connectent des ressources dans des sources de données liées différentes. Les URI des sujets et objets des liens externes sont dans des espaces de noms différents. Un lien externe est une collection de liens RDF externes entre deux ensembles de données. Il s'agit d'un ensemble de triples RDF où tous les sujets se trouvent dans un ensemble de données et tous les objets se trouvent dans un autre ensemble de données.

1.2 Problématique

Deux constats sur le Web des données, en particulier sur le Linked Data, nous semblent importants, et ont motivé les directions de recherche de cette thèse. Notre premier constat est que la qualité des liens à l'intérieur d'une base RDF "les liens **intra-Base**" n'est pas bien assurée.

Le nombre de bases RDF dans le Web des données et leurs tailles sont en croissance constante au fur et à mesure que de nouvelles données sont ajoutées et quelles données actuelles du Web sont converties en RDF. Cette accumulation de données se fait sans mécanisme standardisé ou reconnu d'évaluation de la qualité des liens entre ressources, à l'intérieur d'une base RDF.

Dans le linked data, les ontologies fournissent l'encadrement de la structure des données d'une base RDF. L'ontologie définit l'ensemble des classes et relations qui sont utilisées dans la base RDF. Leurs définitions sont faites de façon à encadrer leurs utilisations par des contraintes pour garder la cohérence des données de la base. À chaque changement, il est nécessaire de vérifier s'il ne cause pas l'apparition des nouvelles incohérences, par exemple, des liens morts ou erronées. A partir de ce constat un défi important dans le contexte de la découverte et la maintenance des liens est de développer des algorithmes efficaces pour détecter les liens morts ou erronées dans la même base (liens intra-base) et de donner une méthode de maintenance pour éviter cette hétérogénéité.

Notre deuxième constat est que la qualité des liens entre un ensemble de bases RDF "les liens **inter-Base**" n'est pas bien assurée.

Les Données Liées permettent la mise en œuvre des applications qui réutilisent des données distribuées sur le Web. Pour faciliter l'interopérabilité entre ces applications, les données issues de différents fournisseurs doivent être liées . Cela signifie que la même entité dans différents ensembles de données doit être identifiée. L'un des principaux défis des Données Liées est de faire face à cette hétérogénéité par la détection des liens entre les ensembles de données.

Dans un tel environnement dynamique, le Web de données évolue : de nouvelles données sont ajoutées, des données obsolètes sont supprimées ou modifiées. Ainsi, les liens entre les données doivent aussi évoluer.

Puisque les liens ne devraient pas être rétablis chaque fois qu'un changement de données se produit, le Web sémantique a besoin des méthodes qui tiennent compte de l'évolution.

Avec le temps, des **liens morts** peuvent apparaître. Les liens morts sont ceux pointant vers des URI qui ne sont plus entretenues, et sont aussi les liens potentiels non définis même quand de nouvelles données sont publiées. Trop de liens morts mènent à un grand nombre de requêtes HTTP inutiles envoyées par les applications clientes. Un problème de recherche actuel abordé par la communauté des Données Liées est celui de la maintenance des liens. Figure.1.1 : représente une vue du problème de l'hétérogénéité des données.

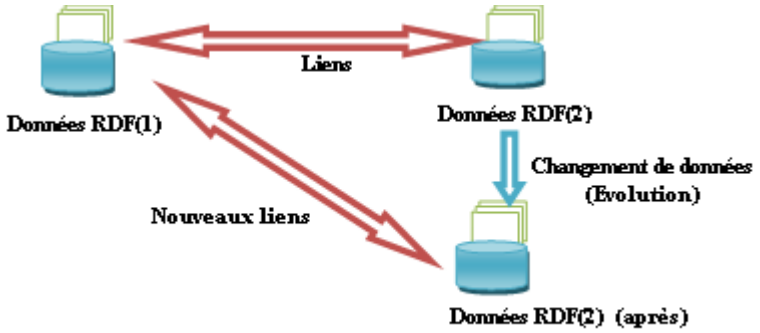

FIGURE 1.1 – *Problème d'hétérogénéité des données*

A partir de ce constat un défi important dans le contexte de la maintenance des liens est de développer des algorithmes efficaces pour la découverte et la maintenance des liens entre des jeux de données. Cet aspect est essentiel pour s'assurer qu'un jeu de données est intégré au Web et que tous les liens décrits sont totalement corrects une fois le jeu localisé.

De nombreuses techniques sont directement applicables dans un contexte de découverte et de maintenance des liens " inter-base ". En principe, il y a trois techniques principales. Une simple, fondée sur des clés, qu'exploite les schémas de nommage communs aux deux sources. Parmi les approches fondées sur des clés on peut citer : KD2R, C-SAKey. Dans le cas où il n'existerait pas d'identifiants communs entre les jeux de données, il est nécessaire d'utiliser des heuristiques de couplage plus complexes, fondées sur des similarités. Donc elle compare les items et les lie si leur similarité dépasse un certain seuil. Parmi ces outils on peut citer SILK, LIMES. Outre ces outils, qui obligent leurs utilisateurs à définir explicitement l'heuristique de correspondance, il en existe aussi qui apprennent l'heuristique de correspondance directement des données. Parmi eux, citons Knofuss, RiMOM.

L'outil LiQuate utilise une approche d'évaluation fondée sur les réseaux bayésiens pour identifier les ambiguïtés et les mises en relations incomplètes dans les liens entre les ressources de bases différentes. Le résultat final de cette évaluation suggère à un expert les ressources potentiellement mal liées.

Donc les chercheurs focalisent leurs travaux de découverte et de maintenance des liens entre un ensemble de bases RDF. Par contre notre ap-

proche donne des solutions de maintenance des liens à l'interieur d'une base RDF "intra-Base" et entre un ensemble de base RDF "inter-Base".

1.3 Contributions de la thèse

1.3.1 Contributions

De nombreux jeux de données sont publiés sur le web à l'aide des technologies du web sémantique. Ces jeux de données contiennent des données qui représentent des liens vers des ressources similaires. Si ces jeux de données sont liés entre eux par des liens construits correctement, les utilisateurs peuvent facilement interroger les données à travers une interface uniforme, comme s'ils interrogeaient un jeu de données unique.

Pour garder la stabilité et l'efficacité d'un système qui se base sur les données liées, malgré les changements qui peuvent toucher ses ressources, nous avons :

1. Tout d'abord, proposée deux méthodes de découverte de liens :
   — La première est une nouvelle approche pour découvrir automatiquement les liens morts et erronées entre les données RDF dans la même base (intra-base) en se basant sur les modèles de liens qui apparaissent autour des ressources et les contraintes d'intégrités.
   — La deuxième est une nouvelle approche pour découvrir automatiquement les liens morts et erronées entre deux bases (inter-base) en se basant sur l'alignement d'ontologie.
2. Nous avons développé un système d'alignement d'ontologie "**ABCMap**" fondé sur une méthode d'optimisation qui se base sur les colonies d'Abeilles Artificielles (ABC).
3. Nous avons aussi proposé une approche pour la maintenance des liens.
4. Dans le cadre de l'évaluation de notre système d'alignement d'ontologie, nous avons considéré la campagne d'évaluation de systèmes d'alignements OAEI 2012. L'analyse des résultats expérimentaux montre que notre approche est meilleure que celle de tous les participants à l'OAEI 2012 en termes de rappel et précision.

1.3.2 Principales publications

Le travail sur notre thèse de doctorat a conduit aux publications suivantes :

1 Fatima Ardjani, Djelloul Bouchiha, Mimoun Malki,"An Approach for Discovering and Maintaining Links in RDF Linked Data", International Journal of Modern Education and Computer Science (IJMECS), Vol.9, No.3, pp.56 – 63, 2017.DOI : 10.5815/ijmecs.2017.03.07

2 Fatima Ardjani, Djelloul Bouchiha, Mimoun Malki,"Ontology-Alignment Techniques : Survey and Analysis", IJMECS, vol.7, no.11, pp.67 – 78, 2015.DOI : 10.5815/ijmecs.2015.11.08

4. Organisation du document Le reste de la thèse est organisé en deux grandes parties :

La première partie "Background et Etat de l'art" comporte :

Le chapitre 2 (Background) : décrit tous les concepts relatifs au problème abordé, notamment le Web sémantique, les Données Liées, l'alignement d'ontologies.

Les chapitres 3 et 4 (État de l'art) : décrits tous les travaux relatifs, notamment dans le domaine "découverte et maintenance des liens" et "l'évolution de l'alignement d'ontologie".

La deuxième partie "Découverte et maintenance des liens" implique :

Le chapitre 5 (Approche proposée) : Nous détaillons dans ce chapitre les deux processus, découverte et maintenance des liens entre les Données Liées RDF.

Le chapitre 6 (Implémentation et évaluation) : Nous évaluons nos approches proposées.

Enfin, on clôture la thèse par "Conclusion et perspectives" là où on présente une synthèse et un bilan du travail réalisé. On présente ainsi les perspectives liées à la poursuite de ce travail, ainsi qu'aux nouveaux thèmes de recherche qui nous paraissent les plus pertinents.

Partie I : Background et Etat de l'art

Background 2

Sommaire

Le Web a connu une évolution pendant 28 ans et cela a conduit à passer du Web de documents au Web de données, ouvrant la voie à de nombreuses applications novatrices dans des domaines trop variés. Alors que les changements apportés par le Web 2.0 sont essentiellement sociaux, le passage du Web que nous connaissons au Web de données (ou Web sémantique, si l'on y inclut également des possibilités plus avancées de raisonnement logique) requiert une évolution plus profonde en termes de représentation des données [T.Heath & C.Bizer 2010].

Ce chapitre présente toutes les facettes du Web de données nécessaires pour la compréhension de cette thèse. Nous présentons les technologies du Web de données, ainsi que le Linked Data et nous en faisons un état des lieux. Enfin, nous définissons les concepts avancés de manipulation des documents RDF nécessaires à nos contributions.

2.1 Evolution du Web : du Web de documents vers un Web de données

Le Web sémantique est un ensemble de technologies visant à rendre les contenus web intelligibles pour des programmes afin d'améliorer l'indexation et la navigation dans un ensemble de données toujours croissant.

Le Web est apparu à la fin des années 1980. Pour dater l'invention du Web, on fait souvent référence à l'article de Tim Berners-Lee [T.Berners-Lee 1989]. Les requêtes du protocole HTTP permettent d'accéder à des ressources mais également de les modifier, de les créer et de les supprimer. Pourtant, au début du Web, la plupart des navigateurs ne proposaient quasiment que de la consultation des ressources crées par le biais d'autres protocoles comme FTP. Ces ressources sont dites statiques car, d'une consultation à une autre, il y a peu de chances qu'elles aient évolué. Il n'est question pratiquement que de lire des documents et de naviguer de document en document.

En 1995, Ward Cunningham invente le premier wiki [1]. On découvre un Web où tout le monde peut créer des données, du contenu et des documents à partir de son navigateur. On commencera aussi à parler du Web dynamique car les informations peuvent changer d'une visite à l'autre, elles sont produites en fonction du contexte (la page d'un wiki va être différente après chaque modification).

Dès qu'il y a eu des liens entre documents HTML, la question de leur raison d'être s'est posée. La navigation hypertexte repose sur le principe de la navigation par hyperliens. En parcourant un document HTML, vous pouvez être amenés à cliquer sur un lien qui vous envoie vers un autre document HTML. En lisant le texte de la page HTML, l'humain peut comprendre la raison de ce lien, ce dont un programme est incapable, car il ne peut analyser le texte.

Les premiers articles de Tim Berners-Lee au sujet du Web ont présenté la notion des liens. Ce principe est fondamental dans le Web sémantique. L'objectif des technologies du Web sémantique est, d'après le W3C, "de fournir une couche technologique servant à construire un Web de données permettant aux ordinateurs de rendre des services plus utiles et de développer des systèmes d'interactions fiables". L'objectif du W3C est de développer et de supporter des standards ouverts encourageant l'expansion du Web à long terme [T.Heath & C.Bizer 2010].

Le Web sémantique n'est pas présenter comme un nouveau paradigme du Web. Il s'agit plutôt d'un ensemble d'outils, de technologies, dont RDF est l'élément primordial, venant soutenir l'évolution des pratiques. Le Web sémantique, c'est un ensemble d'outils proposés en réponse à l'évolution du Web pour le faire tendre toujours plus vers l'échange de connaissances entre individus. Il ne vient donc pas s'opposer au Web social ou Web 2.0. Au contraire, il vient proposer des solutions techniques pour en faciliter la pratique.

Les données liées se fondent sur les technologies du Web sémantique et proposent un ensemble de bonnes pratiques pour exposer, partager et

1. Le Portland Pattern Repository est le site dans lequel Ward Cunningham introduit les fonctionnalités, aujourd'hui à la base de toute implémentation du principe de wiki (http ://c2.com/ppr/)..

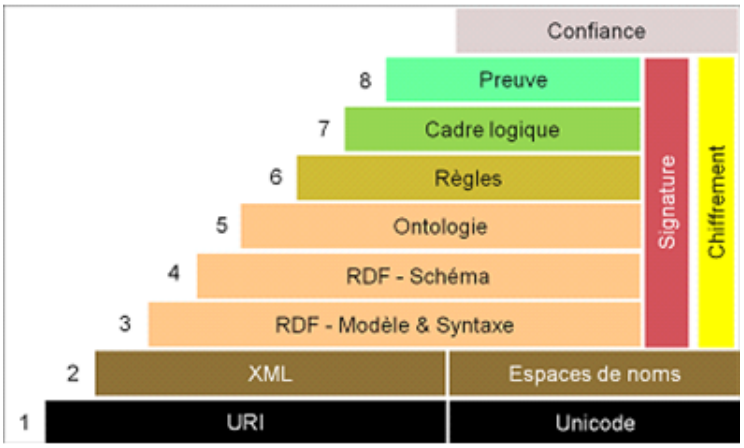

FIGURE 2.1 – *Architecture du Web sémantique.*

connecter des données, de l'information et des connaissances. Certains disent que les données liées sont le meilleur exemple d'application du Web sémantique [T.Heath & C.Bizer 2010].

2.1.1 Le Web de documents

Le Web de documents s'appuie sur le principe de liens existants entre des documents pouvant résider sur des serveurs différents. Les hyperliens permettent aux internautes de naviguer entre les serveurs, et aux moteurs de recherche d'explorer le Web afin de fournir des fonctionnalités sophistiquées à partir du contenu exploré. C'est pourquoi ces hyperliens sont centraux dans la connexion du contenu de différents serveurs, afin de créer un espace d'information unique et global. En combinant simplicité, décentralisation et ouverture, le Web semble atteindre l'architecture idéale, ce que sa croissance de ces vingt dernières années ne cesse de souligner.

Le Web est construit sur un ensemble de standards simples :

— Des URI (Uniform Resource Identifiers, identifiants de ressource uniformes) comme mécanisme d'identification unique et global [T.Berners-Lee *et al.* 1998] ;

— HTTP (HyperText Transfer Protocol, protocole de transfert hypertexte), le mécanisme d'accès universel [R.Fielding 1999] ;

— HTML (HyperText Markup Language, langage de balisage hypertexte), le format de contenu largement utilisé [D.Raggett & I.Jacobs 1999].

la structuration des données joue un role principale en leur réutilisation et même facilite la création d'outils pour réutiliser les données de manière fiable. Le langage dans lequel la plupart des sites web sont écrits, HTML, est dirigé vers la structuration de documents textuels plutôt que de données. Les données s'entremêlant au texte, il est difficile pour des applications logicielles d'extraire des bribes structurées à partir de pages HTML. Pour résoudre ce probleme, plusieurs microformats ont été inventés [T.Heath & C.Bizer 2010].

Microformats

L'utilisation des microformats nous aide à publier des données structurées décrivant des types spécifiques d'entités. Ils servent à intégrer de manière très précise des données dans des pages HTML, ce qui permet aux applications de les extraire sans ambiguïté. Mais l'inconvénient des microformats, c'est la limite de leur représentation de données à peu de types différents d'entités. Ils ne donnent qu'un petit ensemble d'attributs pour les décrire. Or, souvent, on ne peut pas exprimer les relations entre entités. Par exemple, distinguer le fait qu'une personne soit le conférencier d'un événement plutôt qu'un participant ou un organisateur. Par conséquent, les microformats ne sont pas convenues pour le partage de données sur le Web [T.Heath & C.Bizer 2010].

Les API web

Les API web sont une méthode plus générique pour devenir les données structurées disponibles sur le Web. Elles donnent, par le biais de requêtes, un accès simple à des données structurées via le protocole HTTP.

L'arrivée de cette méthode a fait une explosion de petites applications composites qui combinent les données de plusieurs sources, chacune d'elles étant accessible via une API propre au fournisseur de données. Les API web offrent généralement des résultats dans des formats de données comme XML et JSON, supportés par plusieurs langages de programmation. Toutefois, dans le Web, elles ont des limites, qui s'expliquent mieux par une comparaison avec le langage HTML.

Le langage HTML définit l'élément d'ancrage, **a**, dont l'un des attributs valides est **href**. quand on utilise la balise d'ancrage et l'attribut **href** ensemble, on obtient un lien sortant du document courant. Les navigateurs et les robots des moteurs de recherche, sont programmés pour le reconnaitre, soit en affichant un lien cliquable, soit en parcourant le lien directement pour récupérer et traiter le document référencé. C'est cette connectivité entre les documents, controlés par une syntaxe standard pour indiquer des liens, qui a validé le Web de documents. Par contre, les données renvoyées par la majorité des API web ne possèdent pas l'équivalent de la balise d'ancrage HTML et de l'attribut href, pour indiquer les liens à suivre afin de trouver des données connexes.

Plusieurs API web pointent sur des éléments en utilisant des identifiants qui n'ont pas qu'une portée locale. Donc, on a aucun mécanisme standard disponible qui permettrait de se référer aux éléments décrits par une API dans les données renvoyées par une autre. Par conséquent, les données retournées par une API web figurent généralement sous forme de fragments isolés, sans liens fiables signalant le chemin vers les données connexes[T.Heath & C.Bizer 2010].

Les mêmes principes peuvent être appliqués aux données du Web, pour faciliter la découverte des liaisons.

Lier des données distribuées sur le Web exige un mécanisme standard afin de spécifier l'existence et la signification de liaisons entre les éléments décrits dans ces données.

2.1.2 Le Web de données

Plusieurs organisations ont utilisé les données liées pour publier leurs données, pas uniquement pour les mettre en ligne mais aussi de les ancrer dans le Web [N.Mendelsohn 2009]. Donc on obtient un espace global de données qui s'appelle le Web de données [T.Heath *et al.* 2009]. Il forme un grand graphe global [T.Berners-Lee 2007] composé de milliards de triplets RDF provenant de plusieurs sources. Le Web de données peut être vu comme une couche additionnelle fortement entrelacée avec les documents classiques du Web et ayant plusieurs propriétés communes :

- Comprendre n'importe quel type de données.
- N'importe qui peut y publier leur données.
- Les entités sont connectées par des liens RDF, créant un graphe global de données par des ponts entre des sources différentes, ce qui permet de découvrir de nouvelles sources.
- Les éditeurs de données ont les choix de vocabulaire pour représenter leurs données.
- Déréférencement : utilisation des URI pour identifier les termes du vocabulaire afin de trouver leurs définitions.
- L'utilisation de HTTP comme mécanisme standardisé d'accès aux données et de RDF comme modèle standard de données facilite l'accès aux données, comparé aux API web qui se base sur des modèles de données et des interfaces hétérogènes.

Les origines de ce Web se fondent sur un effort de la communauté de chercheurs du Web sémantique, et particulièrement, sur les activités du projet Linking Open Data (LOD) du W3C, démarré en janvier 2007. L'objectif de ce projet, était de commencer le Web de données par une identification de jeux de données disponibles et accessibles sous des licences ouvertes [T.Heath *et al.* 2009], de les convertir en RDF convenablement aux principes des données liées et de les publier sur le Web.

2.2 Données liées " Linked Data "

vise à publier des données structurées et liées sur le Web en utilisant les technologies du Web sémantique. Ces technologies offrent différents langages pour exprimer les données sous forme de graphes RDF et les interroger en SPARQL [T.Heath *et al.* 2009]. Les principes de linked data sont ;

1 Utilisation des références URI pour identifier des objets réels et des concepts abstraits.

2 Exporation du contenu pointé par un point d'accès HTTP. Il faut que toute ressource soit identifiée par un URI HTTP.

3 Utilisation d'un modèle unique pour publier les données structurées, c'est le RDF. Ce modèle de données est basé sur une structure de graphe concue pour le contexte du Web.

4 Utilisation des hyperliens pour connecter toutes sortes d'éléments et pas seulement des documents web. Dans le linked data les hyperliens sont appelés des liens RDF [T.Heath *et al.* 2009].

2.3 RDF " Resource Description Framework "

Le modèle de données RDF [G.J.lyne & J.Carroll 2004] représente l'information comme un graphe orienté avec des nœuds et des arcs nommés. Ce modèle, prévu pour une représentation intégrée de l'information de diverses origines, est structuré de manière hétérogène et représenté à l'aide de plusieurs schémas. Il est décrit en détail dans le W3C RDF Primer [F.Manola & E.Miller 2004]. Un apercu de ce modèle est présenté dans la Figure 2.2.

En RDF, la description d'une ressource est représentée comme un certain nombre de triplets. Les trois parties de chaque triplet sont appelées sujet, prédicat et objet, pour correspondre à la structure d'une phrase simple, comme celle-ci :

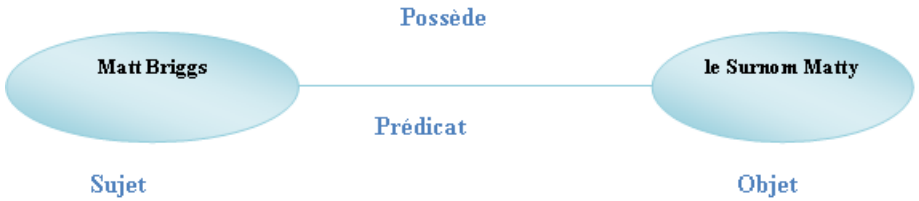

Figure 2.2 – *Triplet RDF.*

Le sujet d'un triplet est l'URI identifiant la ressource décrite. L'objet peut être soit une valeur littérale (une chaine de caractères, un nombre ou une date), soit une URI d'une autre ressource qui est liée, d'une manière ou d'une autre, à l'objet. Le prédicat, au milieu, indique la sorte de relation qu'existe entre le sujet et l'objet (par exemple, le nom ou la date de naissance, pour une valeur littérale ; l'employeur ou une connaissance, pour une autre ressource). Il est, lui aussi, identifié par une URI. Ces URI de prédicats proviennent des "vocabulaires", collections d'URI qui peuvent être utilisées pour représenter l'information dans un certain domaine [G.J.lyne & J.Carroll 2004].

2.3.1 Les types de triplets RDF

On distingue deux types principaux de triplets RDF : les triplets littéraux et les liens RDF.

Les triplets littéraux

Ils ont un objet RDF littéral (une chaine de caractères, un nombre ou une date) et servent à décrire les propriétés des ressources, par exemple, le nom ou la ville de naissance d'une personne. Ces valeurs littérales peuvent être brutes ou typées. Une valeur littérale brute est une chaine de caractères combinée avec un tag de langue, qu'identifie une langue naturelle, comme l'anglais, l'allemand ou le francais. Une valeur littérale typée est une chaine de caractères combinée avec une URI de type de données, qu'identifie le type de données de la valeur littérale. Dans le cas de types courants, comme les entiers, les nombres à virgule flottante et les dates,

ces URI sont définies dans la spécification des types de données de XML Schema [P.Biron & A.Malhotra 2004].

Les liens RDF

Ils décrivent les relations entre deux ressources, et consistent en trois références d'URI. Les URI dans le sujet et l'objet du lien identifient les ressources liées. L'URI du prédicat définit le type de relation entre les ressources. Une distinction utile peut être effectuée entre les liens RDF internes et externes.

— Les liens internes connectent des ressources dans une seule source de données liées. Ainsi, les URI des sujets et des objets sont dans le même espace de noms [T.Heath & C.Bizer 2010].

— Les liens externes connectent des ressources dans des sources de données liées différentes. Les URI des sujets et objets des liens externes sont dans des espaces de noms différents. Ces liens sont cruciaux dans le Web de données, puisqu'ils forment le "pont" qui relie les îlots de données dans un espace de données global et interconnecté [T.Heath & C.Bizer 2010].

On peut représenter un ensemble de triplets RDF par un graphe RDF. Les URI sujet et objet sont les nœuds du graphe et chaque triplet est un arc orienté du sujet vers l'objet. Puisque les URI dans un contexte de données liées sont globalement uniques et peuvent être déréférencées en ensembles de triplets RDF, il est possible d'imaginer toutes les données liées comme un seul grand graphe global, comme l'a proposé Tim Berners-Lee [T.Berners-Lee 2007]. Les applications de données liées opèrent sur ce grand graphe et en récupèrent les parties requises en déréférencant des URI.

2.3.2 Les liens RDF externes

Le principe de données liées est de définir des liens RDF pointant vers d'autres sources de données sur le Web. Ces liens RDF externes sont fondamentaux, puisqu'ils sont le "pont" qu'unit les îlots de données dans un espace de données global interconnecté, et qui permet aux applications de découvrir intuitivement de nouvelles sources [T.Heath & C.Bizer 2010].

D'un point de vue technique, un lien RDF externe est un triplet RDF dans lequel le sujet est une référence URI dans un espace de noms d'un jeu de données, alors que le prédicat et/ou l'objet sont des références URI pointant vers des espaces de noms d'autres jeux de données. Déréférencer ces URI donne une description des ressources liées fournies par le serveur distant. Cette description contient habituellement d'autres liens RDF qui pointent vers d'autres URI qui, à leur tour, peuvent être déréférencés et ainsi de suite. C'est de cette manière que les descriptions individuelles des ressources peuvent être liées. De même, on peut surfer sur le Web de données à l'aide d'un navigateur de données liées ou bien de robots d'un moteur de recherche. Il y a trois types importants de liens RDF :

Liens de relation

Le Web de données contient des informations sur une multitude d'éléments : gens, entreprises, lieux mais aussi films, livres, musique, gènes, etc. Grâce aux liens RDF, les références peuvent être définies d'un jeu de données vers un autre, qui peut, à son tour, contenir des descriptions qui font référence à des entités dans un troisième jeu de données, un cycle que l'on peut répéter à l'infini. Le fait de définir des liens RDF connecte des sources de données, bien sûr, mais aussi cela permet de créer un réseau de données potentiellement infini qui peut être utilisé par des applications clientes [T.Heath & C.Bizer 2010].

Listing 2.1 – *Exemple d'un Lien de relation*

```
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-
   ns#> .
@prefix foaf:<http://xmlns.com/foaf/0.1/> .
<http://biglynx.co.uk/people/dave-smith>
rdf:type foaf:Person ;
foaf:name "Dave Smith" ;
foaf:based_near <http://sws.geonames.org/3333125/> ;
foaf:based_near <http://dbpedia.org/resource/
   Birmingham> ;
foaf:topic_interest <http://dbpedia.org/resource/
   Wildlife_photography> ;
foaf:knows <http://dbpedia.org/resource/
   David_Attenborough> .
```

Cet exemple [T.Heath & C.Bizer 2010] montre la manière dont **Big Lynx** utilise les liens RDF qui pointent vers des entités liées pour enrichir les données publiées sur **Dave Smith**. Pour fournir des informations sur l'endroit où réside ce dernier, il existe des liens RDF indiquant que **Dave** vit près de **basednear**, quelque chose identifié par l'URI *"http ://sws.geonames.org/3333 125/"*.

Les applications de données liées qui vont regarder le contenu associé récupéreront une description complète de **Birmingham** de **Geonames**, une source qui fournit les noms de lieux (en différentes langues), les coordonnées géographiques et des informations sur les structures administratives. Les données **Geonames** sur **Birmingham** contiendront un lien supplémentaire sur "http ://dbpedia.org/resource/Birmingham". En suivant ce lien, des applications pourront trouver le chiffre de la population totale, les codes postaux, les descriptions dans quatre-vingt-dix langues et les personnes célèbres et groupes liés à **Birmingham**. Cette description fournie par **DBpedia**, à son tour, contiendra d'autres liens RDF vers d'autres sources disposant d'informations sur **Birmingham**. Ainsi, en définissant un seul lien RDF, **Big Lynx** a permis aux applications de récupérer des données d'un réseau de sources de données liées.

Liens d'identité

Les URI HTTP ne sont pas seulement des identifiants, elles permettent aussi d'accéder à une information. Plusieurs URI peuvent être employées

pour se référer à un seul et même objet réel. Ainsi, quelqu'un qui veut publier des données sur le Web pour se décrire (que l'on nommera **Jeff**) doit d'abord définir une URI pour s'identifier, dans un espace de noms qu'il possède ou dans lequel le possesseur du nom de domaine l'a autorisé à le faire. Ensuite, il installe un serveur pour retourner les données le décrivant, en réponse à quelqu'un regardant le contenu de son URI par le protocole HTTP. Un consommateur d'informations connaitra alors deux choses : d'abord, toutes les données sur Jeff ; ensuite, l'origine de ces données, puisqu'il les a récupérées d'une URI sous le contrôle de **Jeff**. Mais, que se passe-t-il si **Jeff** veut publier des données décrivant un endroit ou une personne célèbre sur le Web ? Il applique la même procédure : il définit des URI identifiant l'emplacement et la personne célèbre dans son espace de noms, et il fournit les données à qui les demande par leur URI. Les consommateurs de données connaissent donc maintenant les données de **Jeff** et savent qui les a publié. Dans un environnement ouvert comme le Web, **Jeff** ne sera probablement pas le seul à parler de cet endroit ou de cette personne, il y aura certainement plusieurs fournisseurs d'informations sur ces mêmes entités. Comme chacun utilisera sa propre URI pour une personne ou un endroit, on aura au final plusieurs URI identifiant la même entité. Ces URI sont appelées des alias d'URI. Pour que l'on puisse toujours voir tous les fournisseurs d'informations qui parlent de la même entité, les données liées se basent sur des liens RDF entre les alias d'URI. D'un commun accord, les éditeurs utilisent le type de lien "**http ://www.w3.org/2002/07/owl#sameAs**" pour déclarer que deux alias d'URI désignent la même ressource. Par exemple, si Dave Smith veut aussi maintenant une page d'accueil pour ses données privées au-delà de ce que **Big Lynx** publie à son propos, il peut ajouter un lien "**http ://www.w3.org/2002/07/owl#sameAs**" dans sa page privée, qui déclarera que l'URI employée pour ce document et celle donnée par le lien se réfèrent à la même entité [T.Heath & C.Bizer 2010]. "**http ://www.dave-smith.eg.ukme**", "**http ://www.w3.org/2002/07/owl#sameAs**", "**http ://biglynx.co.uk/people/dave-smith**". Il peut sembler fastidieux d'utiliser plusieurs URI pour une même entité et d'employer des liens **owl :sameAs** pour les connecter. Cependant, c'est essentiel pour que le Web de données fonctionne comme un système social. En voici les raisons :

- **Plusieurs opinions.** Des alias d'URI ont une fonction sociale importante sur le Web de données, puisqu'ils sont déréférencés en descriptions des mêmes ressources fournies par différents éditeurs ; ainsi, plusieurs opinions peuvent être exprimées.
- **Tracabilité.** Utiliser différentes URI permet d'associer l'éditeur à ce qu'il veut dire d'une entité spécifique, et cela en déréférencant l'URI utilisée pour identifier l'entité.
- **Pas de point d'échec centralisé.** Si tout élément dans le monde ne pouvait avoir qu'une et une seule URI, cela imposerait la création d'une autorité centralisée de nommage pour assigner les URI. La complexité de coordination, les surcharges administratives et bureaucratiques alors introduites seraient une barrière majeure au développement du Web de données.

Ce dernier point s'éclaire particulièrement lorsque l'on considère la taille de nombreux ensembles de données. Par exemple, **Geonames** fournit des informations sur huit millions d'endroits. S'il avait dû chercher les URI communément acceptées pour tous ces lieux, il n'aurait pas pu publier des données liées. Définir des URI pour des endroits dans leur propre espace de noms réduit la barrière d'entrée, puisqu'il n'est pas nécessaire de connaitre les URI utilisées par d'autres pour ces mêmes endroits [T.Heath & C.Bizer 2010].

Plus tard, **Geonames** (ou d'autres) pourraient chercher et publier des liens **owl :sameAs** pointant vers des données sur ces lieux dans d'autres jeux de données, ce qui permet d'adopter progressivement les principes des données liées. Ainsi, au lieu de se baser sur un accord sur les URI, le Web de données résout le probleme des identités d'une manière évolutionniste et distribuée : de plus en plus de liens **owl :sameAs** peuvent être ajoutés et différents éditeurs peuvent les publier.

L'effort de création des liens peut être partagé entre plusieurs parties. Il y a eu une certaine incertitude majeure ces dernières années pour savoir si owl :sameAs ou d'autres prédicats devaient être utilisés pour exprimer les liens d'identité [H.Halpin *et al.* 2010]. Une source majeure de cette incertitude provenait du fait que la sémantique d'OWL [P.Patel-Schneider & I.Horrocks 2004] traitait les déclarations RDF comme des faits plutôt que comme des revendications de la part des fournisseurs d'informations.

Aujourd'hui, **owl :sameAs** est largement utilisé dans un contexte de données liées et des centaines de millions de ces liens sont publiés sur le Web. On recommande d'employer **owl :sameAs** pour exprimer des liens d'identité, mais aussi pour garder en mémoire l'idée que le Web est un système social et que tout son contenu doit être considéré comme des affirmations par différentes parties plutôt que comme des faits.

Liens de vocabulaire

La promesse du Web de données n'est pas seulement de faire en sorte que les applications clientes découvrent de nouvelles données en suivant des liens RDF à l'exécution, mais aussi de les aider à intégrer des données de ces sources. Cette intégration requiert un lien entre les schémas utilisés par les différentes sources pour publier leurs données. Le terme schéma est compris, dans un contexte de données liées, comme le mélange de termes distincts de vocabulaires RDF différents, employés par des sources pour publier des données sur le Web. Ce mélange peut inclure des termes de vocabulaires largement répandus tout comme des termes propriétaires [T.Heath & C.Bizer 2010].

Le Web de données utilise une approche à deux faces pour gérer le cas des représentations hétérogènes [T.Berners-Lee & L.Kagal 2008]. D'un côté, il tente d'éviter cette situation en promouvant la réutilisation des termes de vocabulaires largement déployés. Quand ces vocabulaires contiennent les termes requis pour représenter un jeu de données spécifique, ils devraient être utilisés. Cela aide à résorber l'hétérogénété à partir d'un accord ontologique.

D'un autre côté, le Web de données tente de gérer cette hétérogénéité

en rendant les données aussi auto-descriptives [N.Mendelsohn 2009] que possible. Cela signifie qu'une application de données liées qui découvre des informations sur le Web représentées avec un vocabulaire inconnu devrait être capable de trouver toutes les méta-informations requises pour transformer les données en une représentation compréhensible.

De manière plus technique, on suit une double approche. D'abord, on rend déréférencables les URI identifiant des termes de vocabulaire, pour que les clients puissent chercher les définitions RDFS et OWL de ces termes (chaque terme de vocabulaire pointe vers sa propre définition [D.Berrueta & J.Phipps 2008]). Ensuite, on publie des correspondances entre les termes de différents vocabulaires sous la forme de liens RDF [N.Mendelsohn 2009].

Ensemble, ces techniques permettent aux applications des données liées de découvrir les méta-informations requises pour intégrer intuitivement les données à l'aide de liens RDF.

Un éditeur de données liées devrait donc adopter la démarche suivante : d'abord, rechercher les termes largement utilisés qui pourraient être réutilisés pour représenter les données ; si les vocabulaires largement déployés ne les fournissent pas, ces termes devraient être définis dans un vocabulaire propriétaire, et employés en plus de ceux des autres vocabulaires répandus. Alors, dans la mesure du possible, cet éditeur devrait chercher à convaincre ses confrères de recourir à ce nouveau vocabulaire pour décrire les données en question.

Si, plus tard, l'éditeur de données découvre qu'un autre vocabulaire contient le même terme que le sien, il devrait définir un lien RDF entre les URI identifiant les deux termes, qui préciserait que les deux URI se réfèrent au même concept. OWL (Ontology Web Language, langage d'ontologies pour le Web [D.L.McGuinness & Harmelen 2004]), RDFS (RDF Schema, schéma RDF [D.Brickley & R.V.Guha 2004]) et SKOS (Simple Knowledge Organization System, système simple d'organisation des données [A.Miles & S.Bechhofer 2009]) définissent des types de liens RDF qui peuvent être choisis à cette fin. **owl :equivalentClass** et **owl :equivalentProperty** peuvent être utilisés pour déclarer que les termes de différents vocabulaires sont équivalents. Si une correspondance moins forte est recherchée, alors on peut utiliser **rdfs :subClassOf**, **rdfs :subPropertyOf**, **skos :broadMatch** et **skos :narrowMatch**.

Listing 2.2 – *Exemple de Lien de vocabulaire*

```
http://biglynx.co.uk/vocab/sme#SmallMediumEnterprise
   est lie a un terme lie de
DBpedia, Freebase, UMBEL et OpenCyc.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
   ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix co: <http://biglynx.co.uk/vocab/sme#>.
<http://biglynx.co.uk/vocab/sme#SmallMediumEnterprise>
    rdf:type rdfs:Class ;
rdfs:label "Small or Medium-sized Enterprise";
rdfs:subClassOf <http://dbpedia.org/ontology/Company>.
```

```
rdfs:subClassOf <http://umbel.org/umbel/sc/Business>;
rdfs:subClassOf <http://sw.opencyc.org/concept/
   Mx4rvVjQNpwpEbGdrcN5Y29ycA>;
rdfs:subClassOf <http://rdf.freebase.com/ns/m/0qb7t >.
```

De la même manière que les liens **owl :sameAs** interconnectent de facon incrémentale des **alias** d'URI, des liens entre des termes de vocabulaires peuvent être définis par diverses parties. Plus il y a de liens entre les vocabulaires, plus les applications clientes pourront intégrer les données représentées avec ces derniers. Le Web de données se fonde sur une approche distribuée de l'intégration de données, ce qui permet de répartir l'effort d'intégration dans le temps et entre plusieurs acteurs [M.J.Franklin *et al.* 2005] [J.Madhavan *et al.* 2007] [A.Schultz 2010].

2.4 Métadonnées

Pour s'assurer que les ressources d'un jeu de données sont abondamment décrites, on peut appliquer les mêmes principes au jeu lui-même : il devrait inclure des informations sur ses auteurs, sa crédibilité (soit la date de la dernière mise à jour) et ses termes de licence. Ces métadonnées donnent au consommateur la clarté nécessaire sur la provenance et la fiabilité, ainsi que les termes sous lesquels les données peuvent être réutilisées ; chaque élément est important pour encourager la réutilisation des données.

En outre, les descriptions du jeu peuvent inclure des pointeurs vers des exemples de ressources, ce qui peut aider les robots d'indexation à découvrir et à indexer les données. Si la charge créée par les robots d'indexation est trop grande, les descriptions peuvent aussi inclure des liens vers des fichiers RDF, téléchargeables et indexables séparément [T.Heath & C.Bizer 2010].

Deux mécanismes de base sont disponibles pour publier des descriptions d'un jeu de données : les sitemaps sémantiques [R.Cyganiak *et al.* 2008] et les descriptions voiD [K.lexander *et al.* 2009].

2.4.1 Sitemaps sémantiques

Les sitemaps sémantiques sont une extension du protocole Sitemaps bien établi. Ils fournissent aux moteurs de recherche des indices sur les pages web disponibles pour l'exploration [D.Wood *et al.* 2014].

Un sitemap consiste en un document XML, généralement nommé sitemap.xml et stocké à la racine d'un site web. Le schéma Sitemaps définit des éléments, comme url, loc, lastmod et changefreq, grâce auxquels le propriétaire du site peut fournir des informations de base sur les pages qui constituent son site et la fréquence à laquelle elles changent en moyenne, ce qui permet aux moteurs de recherche d'optimiser leur comportement en exploration [T.Heath & C.Bizer 2010].

L'extension sémantique des Sitemaps définit des éléments supplémentaires, appropriés pour accroître la quantité des données d'un fichier avec des informations utiles dans un contexte de données liées. Ces informations peuvent être le label et l'URI pour le jeu de données, les URI

d'exemples caractéristiques de ce jeu, ainsi que l'emplacement des points d'accès SPARQL (SPARQL endpoint) et des fichiers de triplets.

En utilisant l'élément datasetURI dans un Sitemap, les éditeurs de données peuvent informer les moteurs de recherche et les autres applications clientes de l'URI du jeu de données, d'où ils peuvent récupérer de plus informations sur ce jeu en RDF [D.Wood *et al.* 2014].

2.4.2 VoiD

VoiD (Vocabulary of Interlinked Datasets, vocabulaire de jeux de données interliées) [2] est un standard de facto pour les vocabulaires de description de jeux de données liées. Il reprend quelques fonctionnalités des Sitemaps sémantiques (par exemple, les termes dataDump et sparqlEndpoint), mais en RDF. Il permet aussi de décrire les vocabulaires utilisés dans un jeu de données, les liens vers d'autres jeux ainsi que les partitions logiques (ou sous-ensembles) d'un jeu spécifique [D.Wood *et al.* 2014].

La possibilité de définir et de lier des sous-ensembles de données est particulièrement utile. On peut publier des données RDF riches sur des descriptions RDF, qui peuvent à leur tour être définies comme des sous-ensembles d'un ensemble plus large.

Dans ce cas, le sujet des triplets RDF devrait être l'URI de la description RDF elle-même (c'est-à-dire une ressource d'informations), pas l'URI de la ressource décrite [D.Wood *et al.* 2014].

2.4.3 Métadonnées de provenance

La possibilité de tracer l'origine des données est une composante-clé dans la construction d'applications sérieuses et fiables. L'utilisation d'URI déréférencables lie en dur cette capacité dans les données liées, car n'importe qui peut déréférencer une URI particulière pour déterminer ce que le propriétaire de cet espace de noms dit sur une ressource particulière. Cependant, comme des fournisseurs d'informations différents pourraient publier des données dans le même espace de noms, il est important de tracer l'origine de fragments de données particuliers. Les sources de données devraient donc fournir, avec les données, des métadonnées de provenance, lesquelles devraient être représentées comme des triplets RDF décrivant le document où sont contenues les données originalles [D.Wood *et al.* 2014].

Un vocabulaire largement déployé pour représenter ces données est **Dublin Core**, particulièrement les prédicats **dc :creator**, **dc :publisher** et **dc :date**. Lors de l'utilisation de **dc :creator** et de **dc :publisher** dans un contexte de données liées, on devrait employer les URI et non les noms littéraux identifiant les créateurs et éditeurs.

Cela permet à des tiers d'y faire référence de manière non ambiguë et, par exemple, de connecter ces URI à des informations disponibles sur le Web qui pourraient être utilisées pour vérifier la qualité et la fiabilité des données publiées. Le profil personnel de **Dave Smith**, introduit précédemment, montre l'utilisation de ces prédicats pour fournir des informations simples sur la provenance.

2. http ://www.w3.org/2001/sw/interest/void/.fnt :1.

Le modèle de provenance ouverte (Open Provenance Model) fournit une autre piste, un vocabulaire plus expressif qui décrit la provenance en termes d'Agents, Artifacts et Processes. Une comparaison des différents vocabulaires de provenance ou d'autres ressources sur la publication des informations de provenance est disponible sur le site du W3C Provenance Incubator Group.

Pour que les consommateurs puissent vérifier l'attribution des métadonnées, les éditeurs pourraient décider de signer leurs données de manière digitale. Une bibliothèque open-source peut être utilisée pour produire ces signatures, comme NG4J(Named Graphs API for Jena) [D.Wood *et al.* 2014].

2.5 Vocabulaires pour décrire des données

RDF fournit un modèle de données générique et abstrait qu'utilise des triplets sujet, prédicat, objet pour décrire des ressources. Cependant, il ne fournit pas des termes propres à un domaine pour décrire des classes d'éléments dans le monde, et la manière dont elles sont liées. Cette fonction est remplie par les taxonomies, vocabulaires et ontologies exprimées en SKOS (Simple Knowledge Organization System, système d'organisation simple des connaissances) [A.Miles & S.Bechhofer 2009], RDFS (schéma RDF) [D.Wood *et al.* 2014] et OWL (Ontology Web Language).

SKOS est un vocabulaire servant à exprimer les hiérarchies conceptuelles, souvent appelées taxonomies, alors que RDFS et OWL fournissent des vocabulaires pour décrire les modèles conceptuels en termes de classes et leurs propriétés. Par exemple, quelqu'un peut définir un vocabulaire RDFS pour les animaux de compagnie qu'inclut une classe Dog, contenant tous les chiens. Il peut aussi définir une propriété comme hasColour, permettant ainsi aux propriétaires de chiens de publier des descriptions RDF à l'aide de ces termes.

Ensemble, SKOS, RDFS et OWL fournissent un continum d'expressivité. SKOS est largement employé pour représenter des thésaurus, des taxonomies, des systèmes de titres de sujets et des hiérarchies de sujets (par exemple, la mécanique appartient au sujet plus large qu'est la physique). RDFS et OWL sont utilisés dans des cas où les prémisses des relations entre les termes devraient être représentées (par exemple, tous les athlètes sont aussi des personnes)[D.Wood *et al.* 2014].

Couplés à un moteur de raisonnement convenable, les modèles RDFS et OWL permettent des relations implicites, inférées par rapport aux données. Dans un contexte de données liées, il est souvent suffisant d'exprimer des vocabulaires en RDFS.

Cependant, certaines primitives de OWL, comme sameAs, sont utilisées régulièrement pour exprimer le fait que deux URI identifient la même ressource. La combinaison de RDFS et quelques primitives d'OWL est souvent appelée, de manière informelle, RDFS++ [D.Wood *et al.* 2014].

2.5.1 Taxonomie

C'est un vocabulaire dans lequel les termes sont organisés de manière hiérarchique. Chaque terme peut partager une relation parent-enfant

avec un ou plusieurs autres éléments de la taxonomie. L'une des relations parent-enfant les plus couramment utilisées est la spécialisation-généralisation, dans laquelle un terme est une forme plus ou moins spécifique d'un autre terme. Les relations parent-enfant peuvent être aussi des relations de plusieurs à plusieurs. Cependant, beaucoup de taxonomies adoptent la restriction précisant que chaque élément ne peut avoir qu'un seul parent. Dans ce cas, la taxonomie est un arbre ou une collection d'arbres (forêt) [D.Wood *et al.* 2014].

2.5.2 Vocabulaire

C'est une collection de termes non ambigus utilisés pour la communication. Ils ne doivent pas être redondants sans identification explicite de la redondance. En outre, ils sont censés avoir un sens cohérent dans tous les contextes [D.Wood *et al.* 2014].

2.5.3 Ontologie

L'ontologie est la base de ce que nous appelons la représentation des connaissances. Ces connaissances sont exprimées sous forme de symboles auxquels nous donnons une "sémantique" (un sens). Supposons que nous voulions interroger une base de données contenant diverses ressources (textes, images, vidéos) à l'aide d'une requête (question ou mot[s]- clé[s]) : comment trouver les données stockées dans cette base qui correspondent à cette requête ? [T.Heath & C.Bizer 2010]

Pour résoudre ce probleme, nous construisons ce que l'on appelle des "bases de connaissances" qui sont constituées des éléments suivants :

1. Une ontologie. Une collection de classes et de relations (que nous appelons "propriétés") entre ces classes..
2. Des règles. Un ensemble de contraintes sur les propriétés et les classes de l'ontologie.
3. Des faits. Des instances de l'ontologie.

Les ontologies fournissent un vocabulaire précis pour représenter la connaissance et permette de spécifier les entités qui seront représentées, comment elles peuvent être regroupées et quelles relations les connectent. Le vocabulaire peut être considéré comme un contrat social entre un fournisseur de données et un consommateur. Plus l'ontologie est précise et facilement compréhensible, plus son potentiel d'utilisation sera grand. Si elle est trop complexe, qu'elle inclut des concepts et des relations dont l'utilisateur n'a pas besoin, alors elle peut devenir une source de confusion, compliquée et difficile à employer, à maintenir et à étendre.

Le Web sémantique utilise une combinaison de langage de schéma et de langage d'ontologie pour fournir les capacités des vocabulaires, des taxonomies et des ontologies. RDF Schema (RDFS) fournit un vocabulaire spécifique pour RDF qui peut être utilisé pour définir des taxonomies de classes et de propriétés et de simples spécifications de domaines et de portée pour des propriétés. Le langage d'ontologie du Web (OWL) permet de définir des ontologies qui capturent la sémantique des connaissances du domaine [T.Heath & C.Bizer 2010].

2.5.4 RDFS

RDF permet de déclarer des ressources qui sont identifiées par des URI. Ces déclarations prennent la forme de triplets qui associent une ressource (un sujet) à une valeur (un objet) en utilisant une propriété (un prédicat). RDF fournit un modèle extrêmement puissant et expressif pour capturer l'information, mais il ne donne aucun moyen de saisir le sens de l'information - ce n'est que de la modélisation [D.Wood *et al.* 2014].

La première étape consiste à développer un vocabulaire commun ou une collection de ressources, qui a une signification bien comprise et qui est utilisé de manière cohérente pour décrire d'autres ressources. RDF Schema ne tente pas de définir ces vocabulaires partagés, mais, plutôt, de fournir un langage avec lequel nous pouvons développer nos propres vocabulaires partagés [D.Wood *et al.* 2014].

Les vocabulaires RDFS décrivent les classes de ressources et les propriétés employées dans un modèle RDF. En utilisant RDFS, nous pouvons organiser les classes et les propriétés en spécialisation-généralisation d'hiérarchies, définir les attentes de domaine et de portée des propriétés, affirmer l'appartenance à une classe et préciser et interpréter les types de données.

Toutes les ressources en RDFS sont considérées comme membres de classe de toutes les ressources RDF : en tant que telles, elles sont donc toutes des instances. Vous pouvez décrire plus en détail ces cas en faisant des déclarations à leur sujet par les propriétés ou en les rendant explicitement membres des autres classes définies dans un vocabulaire RDFS [T.Heath & C.Bizer 2010].

2.5.5 OWL

Le langage d'ontologies du Web (OWL) est un langage RDF développé par le W3C pour la définition de classes et de propriétés, et aussi pour permettre un raisonnement et une inférence plus puissants sur les relations. OWL a été construit comme une extension à RDFS, un vocabulaire schéma plus simple et qui repose sur un grand nombre de travaux antérieurs sur le développement de langages d'ontologies. C'est le standard du W3C pour la définition des schémas du Web sémantique. Les outils et l'API qui le supportent sont en pleine expansion. OWL est un langage très vaste avec beaucoup de parties compliquées. Il est divisé en trois sous-langages de complexité et d'expressivité croissantes appelés : OWL Lite ; OWL DL ; OWL Full [D.Wood *et al.* 2014].

OWL Lite, complexité faible

OWL Lite est destiné à des utilisateurs qui ont surtout besoin d'une hiérarchie de concepts, de classifications et d'une expressivité limitée, parce que c'est le moins expressif des trois. Il est particulièrement adapté à ceux qui souhaitent bénéficier d'une plus grande expressivité de RDF/RDFS, tout en conservant une certaine facilité d'utilisation. Pourquoi OWL Lite est-il intéressant ? Parce que, avec ce langage, nous pouvons ajouter des contraintes sur les concepts et les relations, ce que nous ne pouvons pas faire avec RDF/RDFS [D.Wood *et al.* 2014].

Un cas d'utilisation est la migration rapide de thésaurus ou d'anciennes taxonomies vers les ontologies. Cependant, pour ce travail, SKOS pourrait être plus approprié parce qu'il a été concu pour faciliter la publication de vocabulaires structurés et leur utilisation dans le Web sémantique [D.Wood *et al.* 2014].

OWL DL, complexité moyenne

OWL DL a été créé pour les utilisateurs qui veulent une expressivité maximale, sans perdre la complétude du calcul (toutes les inférences sont fournies) et la décidabilité des systèmes de raisonnement (tous les calculs sont terminés dans un intervalle de temps fini). OWL DL est fondé sur les logiques de description, un champ de recherche qui a étudié un fragment particulier décidable de la logique du premier ordre.

Un cas d'utilisation est un système de raisonnement automatisé [D.Wood *et al.* 2014].

OWL DL inclut tous les éléments du langage OWL avec des restrictions comme la séparation des types (une classe ne peut pas être un individu ou une propriété et une propriété ne peut pas être un individu ou une classe).

OWL Full, complexité forte

OWL Full est parfait pour les utilisateurs qui veulent une expressivité maximale et la liberté syntaxique de RDF sans garantie de calcul (complétude comme décidabilité)[D.Wood *et al.* 2014].

Dans OWL Full, une classe peut se traiter simultanément comme une collection d'individus et comme un individu à part entière.

Ce langage utilise aussi des URI dans les espaces de noms de RDF, RDFS et OWL. Il emploie aussi les définitions du schéma XML littéral (XML Schema Literal).

Comme le soulignent toutes ces indications, chacun de ces sous-langages représente une extension plus simple par rapport à son prédécesseur (pour ce qu'il est possible d'exprimer et pour ce qu'il est possible de conclure de manière valide). Les affirmations suivantes sont vraies, mais leur symétrique ne l'est pas [T.Heath & C.Bizer 2010] :

Toute ontologie OWL Lite conforme est une ontologie OWL DL conforme.

Toute ontologie OWL DL conforme est une ontologie OWL Full conforme.

Toute inférence OWL Lite valide est une inférence OWL DL valide.

Toute inférence OWL DL valide est une inférence OWL Full valide.

2.5.6 Contraintes d'intégrité

Les contraintes d'intégrité permettent de définir des règles sur les propriétés et les classes de l'ontologie. Elles ont pour rôle d'assurer la cohérence d'une base RDF. Les contraintes d'intégrité permettent ainsi de vérifier que toutes les descriptions des ressources sont correctes. par exemple, elles sont utlisées lors de classification (organiser la hiérarchie des types dans l'ontologie) et d'instanciation (attribuer le type - le plus spécifique -

à un individu ou un littéral). On peut définir trois classes de contraintes d'intégrité :

1. Les contraintes de typage, nécessitent que les ressources liées par une relation donnée aient un type précis, engendrées par les domaines et co-domaines des propriétés ainsi que par les disjonctions.
2. Les contraintes d'unicité, nécessitent qu'une ressource ne puisse pas être présente de la même facon dans plus d'un triplet contenant la même relation, engendrées par les propriétés fonctionnelles.
3. Les contraintes de définition, nécessitent qu'une ressource soit liée à une autre par un triplet contenant une relation ou des ressources nœuds précises, engendrées par les définitions de classes par restriction en OWL.

2.6 SPARQL

Le langage de requêtes SPARQL est largement utilisé pour l'interrogation des données RDF, et il est mis en œuvre par tous les magasins RDF majeurs. Outre les simples graphes RDF, SPARQL permet également d'interroger des ensembles de graphes nommés.

L'application composite veut afficher des données sur **Birmingham** à côté du profil de **Dave** sur le site web de **Big Lynx**. Afin de récupérer toutes les informations que LDspider a trouvée dans toutes les sources de données sur **Birmingham**, l'application composite va exécuter la requête SPARQL suivante dans le magasin RDF. L'échange de requêtes et de résultats s'appuiera sur le protocole SPARQL [D.Wood *et al.* 2014].

Listing 2.3 – *Exemple du protocole SPARQL.*

```
SELECT DISTINCT ?p ?o ?g WHERE
{
{ GRAPH ?g
{ <http://dbpedia.org/resource/Birmingham> ?p
   ?o . }
}
UNION
{ GRAPH ?g1
{ <http://dbpedia.org/resource/Birmingham>
<http://www.w3.org/2002/07/owl#sameAs> ?y }
GRAPH ?g
{ ?y ?p ?o }
}
}
```

Les symboles commencant par un point d'interrogation dans la requête sont des variables liées à des valeurs provenant de graphes différents durant l'exécution [D.Wood *et al.* 2014].

La première ligne de la requête précise que l'on veut récupérer les prédicats de tous les triplets (**?p**) ainsi que les objets (**?o**) de tous les triplets qui décrivent **Birmingham**. De plus, on veut récupérer les noms des graphes (**?g**) dont chaque triplet est issu. Ensuite, on utilise le nom du

graphe pour grouper les triplets sur la page web et afficher, à côté de chacun d'eux, l'URI de l'endroit où il a été récupéré. Le modèle de graphe des lignes 3 à 5 correspond à l'ensemble de données sur **Birmingham** à partir de DBpedia. Les modèles de graphe des lignes 7-10 correspondent à tous les triplets dans les autres graphes qui sont reliés par des liens **owl :sameAs** avec l'URI DBpedia pour Birmingham. **Jena** envoie les résultats de la requête vers l'application comme un résultat SPARQL sous forme de documents XML, et l'application les met en forme pour s'adapter à la présentation de la page web.

L'application de données liées minimaliste précédente néglige de nombreux aspects importants qu'implique la consommation de données liées [T.Heath & C.Bizer 2010].

Conclusion

Dans ce chapitre nous avons présenté tous les concepts relatifs au probleme abordé notamment le Web de données, les principes de base des données liées, et a décrit la manière dont ils interagissent pour étendre le Web à un espace de données global. Tout comme le web classique de documents, le Web de données est construit sur un petit ensemble de standards et sur l'idée d'utiliser des liens pour connecter le contenu de différentes sources.

On a aussi décrit comment SKOS, RDFS et OWL offrent un continuum d'expressivité à travers lequel les relations entre les classes d'objets peuvent être définies.

Un état de l'art de tous les travaux de découverte et de maintenance des liens et évolution d'alignement d'ontologies sera détaillé dans les chapitres suivants.

Alignement des ontologies : Etat de l'art

3

Sommaire

L'évolution des technologies Internet produit des avantages dans la recherche sur le partage et l'intégration de sources dispersées dans un environnement distribué. Dans une architecture décentralisée, le Web sémantique permet aux agents logiciels de comprendre des sources liées sémantiquement. Les ontologies ont été considérées comme une composante fondamentale afin de partagé des connaissances. Dans la réalité, les ontologies de différents domaines sont construites de façon indépendante les unes des autres par plusieurs différentes communautés. Donc il faut établir des correspondances sémantiques entre les ontologies qui décrivent des données et des connaissances partagées.

l'alignement d'ontologies est trés nécessaire dans les systèmes d'intégration, car il rend les ressources décrites par des ontologies différentes d'une manière conjointe.

3.1 L'alignement d'ontologie

L'alignement d'ontologie permet de générer des correspondances entre les entités. Ces entités sont des concepts ou des propriétés ou encore des instances. Cependant la production automatique des correspondances entre deux ontologies est d'une extrême difficulté qui est due aux divergences (conceptuelle, habitudes, etc.) entre différentes communautés de développement des ontologies [J.Euzenat & P.Shvaiko 2013].

Techniques d'alignement

Dans cette partie, nous explorons les techniques et les méthodes utilisées dans la littérature qui s'attaquent au probleme de recherche de la similarité, de la dissimilarité ou de la correspondance entre deux entités en général, qu'elles figurent dans des schémas, ou dans des ontologies représentées soit en RDF(S), soit en OWL [J.Euzenat & P.Shvaiko 2013].

- **Les méthodes terminologiques :** Ces méthodes font la comparaison des termes ou des chaînes de caractères ou bien des textes. Elles sont utilisées pour calculer la valeur de similarité des entités textuelles, telles que les noms, les étiquettes, les commentaires, les descriptions, etc. Ces méthodes sont divisées en deux sous-catégories : des méthodes qui comparent des termes, et des méthodes reposant sur certaines connaissances linguistiques [J.Euzenat & P.Shvaiko 2013].
- **Les méthodes structurelles :** Ce sont des méthodes qui calculent la similarité de deux entités en employant des informations structurelles lorsque les entités en question sont reliées aux autres par des liens sémantiques ou syntaxiques, formant ainsi une hiérarchie ou un graphe d'entités [J.Euzenat & P.Shvaiko 2013]. Nous appelons :
  1. **méthodes structurelles internes :** les méthodes qui n'emploient que des informations concernant les attributs d'entité,
  2. **méthodes structurelles externes :** les méthodes qui considèrent les relations entre des entités.
- **Les méthodes extensionnelles :** Ces méthodes calculent la similarité entre deux entités qui sont notamment des concepts ou des classes en analysant leurs ensembles d'instances [J.Euzenat & P.Shvaiko 2013].
- **Les méthodes sémantiques :** Il existe :
  1. **Les techniques fondées sur les ontologies externes :** Pour aligner deux ontologies il faut que les comparaisons se fassent selon un capital de connaissances communes. Ces méthodes reposent sur l'utilisation d'une ontologie formelle intermédiaire pour répondre à ce besoin [J.Euzenat & P.Shvaiko 2013].
  2. **Les techniques déductives :** Les méthodes sémantiques se fondent sur des modèles de logique (tels que la satisfiabilité propositionnelle (SAT), la SAT modale ou les logiques de description) et sur des méthodes de déduction pour calculer la similarité entre deux entités. Les techniques des logiques de description (telles que le test de subsomption) peuvent être utilisées

pour vérifier les relations sémantiques entre les entités comme l'équivalence (la similarité est égale à 1), la subsomption (la similarité est de 0 à 1) ou l'exclusion (la similarité est égale à 0)[J.Euzenat & P.Shvaiko 2013].

3.2 Approches d'alignement

Les approches d'alignement sont issues de différentes communautés, comme la recherche d'information, le traitement du langage naturel, l'ingénierie des connaissances,...etc.

Le tableau 3.1 résume les approches d'alignement d'ontologie. La plupart des approches reposent sur la même méthode, à savoir la méthode terminologique qui est composée de deux techniques : la première est basée sur la comparaisons des termes ; la deuxième utilise une ressource externe comme "WordNet". Mais les méthode sémantiques ne sont pas employées que dans quelques approches, par exemple, CtxMatch, S-Match et LogMap.

On note encore que :

- COMA et COMA++, S-Match, gèrent plusieurs types d'ontologies.
- DCM, HSM, IceQ, leurs entrées ont de multiples ontologies.
- COMA et COMA ++ et GeRoMeSuite, leurs présentations internes sont sous forme de Graphes acycliques orientés.
- S-Match, DSSim, et TaxoMap mesurent la similarité entre différentes entités d'ontologie, en utilisant aussi la disjonction et la subsomption. Par contre, les autres approches calculent seulement les relations d'équivalence.
- Plusieurs approches ont introduit de nouvelles façons de coder le processus d'alignement, par exemple, à travers des réseaux de Markov, comme iMatch et CODI, ou en proposant les imbrications intéressantes entre les interconnexion de données et le schéma alignement, comme Paris, ou en analysant dans le processus d'alignement également les données d'image, tels que VSBM et GBM.
- Plusiurs approches, comme Scarlet, OMviaUO et BLOOMS et BLOOMS++, allaient au-delà en utilisant WordNet comme source de connaissances de base.
- Plusieurs approches récentes ont mis en place la vérification d'alignement dans le processus d'appariement, comme Lily, YAM ++ et LogMap.
- Falcon, Anchor-Flood, Lily, AgreementMaker, LogMap et FSM gèrent efficacement des ontologies à grande échelle.
- COMA ++, S-Match , AgreementMaker, DSSim, SAMBO et YAM++, sont équipés d'une interface utilisateur graphique.

Dans le tableau suivant :

- La colonne **Entrée** représente l'entrée prise par les systèmes.
- La colonne **Besoins** représente les ressources qui doivent être disponibles pour que le système fonctionne. Cela couvre l'aspect **manuel**, qui est ici désigné par "**utilisateur**" lorsque le retour de l'utilisateur est requis, "**semi**" lorsque le système peut profiter de la rétroaction des utilisateurs, mais peut être "**automatique**" lorsque le système fonctionne sans intervention de l'utilisateur. De même,

la valeur "**instances**" indique que le système nécessite des instances de données.

— Les colonnes **Mesures terminologiques**, **Mesures structurelles**, **Mesures extensionnelles** et **Mesures sémantiques** spécifient les techniques d'alignement adoptées par l'approche en question.

Tableau 3.1 – *Synthèse des approches d'alignement*

Approche	**Entrée**	**Besoins**	**Mesures terminologiques**	**Mesures structurelles**	**Mesures extensionnelles**	**Mesures sémantiques**	**Observation**
YAM/YAM++ [F.Duchateau *et al.* 2009]	XML, OWL	Auto, Semi	WordNet	Profils de structure, similarité d'inondation			
LSD [A.H.Doan *et al.* 2001]	Schémas : bases de données relationnelles, Taxonomie XML	Auto, Instances	La technique d'apprentissage Bayes naïf	Structure hiérarchique	basée sur des contraintes de domaine		
COMA/ COMA++ [H.Do & E.Rahm 2002]	Schémas : bases de données relationnelles, XML, OWL	Utilisateur	Basée sur des caractères, Basée sur la langue, Types de données, Thésaurus auxiliaire	DAG (arbre) correspondant à un biais vers différentes structures, par exemple feuilles, référentiel des structures			

Approche	Entrée	Besoins	Mesures terminologiques	Mesures structurelles	Mesures extensionnelles	Mesures sémantiques	Observation
S-Match [W.Winkler 1999]	Classification, schéma XML, OWL	Auto	Basée sur des caractères, Basée sur la langue, WordNet			Propositionnelle SAT modale	
IceQ [J.Euzenat & P.Shvaiko 2013]	Formulaire Web	Auto, Semi	Basée sur des caractères	Clustering	Basée sur des contraintes		
OLA [J.Euzenat & P.Shvaiko 2013].	RDF, OWL	Auto, Instances	Basée sur des caractères, Basée sur la langue, Similarité des types des données, WordNet	Calcul itératif de point fixe, Matching de voisins, structure taxonomique			
iMAP [R.Dhamankar *et al.* 2004]	Schémas : bases de données relationnelles	Auto, Instances	La technique d'apprentissage Bayes naïf	Structure hiérarchique	Basée sur des Contraintes de domaine		
DCM [K.Chang *et al.* 2005]	Formulaire Web	Auto		Corrélation, Statistiques			Intégration des données

Approche	Entrée	Besoins	Mesures terminologiques	Mesures structurelles	Mesures extensionnelles	Mesures sémantiques	Observation
SAMBO [P.Lambrix & H.Tan 2006]	OWL	Auto, Documents	Basée sur des caractères, La technique d'apprentissage Bayes naïf, WordNet	Similarité de structure itérative sur la base de is-a, (partie-tout)			Fusion des ontologies
HSM [W.Su *et al.* 2006]	Ontologies	Auto		Modèles de co-occurrence, Statistiques			
CtxMatch/ CtxMatch2 [P.Bouquet *et al.* 2006]	Classification, OWL	Utilisateur	Basée sur des caractères, Basée sur la langue, WordNet			basée sur les logiques de description	
GeRoMeSuite [D.Kensche *et al.* 2007]	SQL DDL, XML, OWL	Auto, Semi	Basée sur des caractères	oui			Fusionner, composer

Approche	Entrée	Besoins	Mesures terminologiques	Mesures structurelles	Mesures extensionnelles	Mesures sémantiques	Observation
Scarlet [M.Sabou *et al.* 2008]	OWL	Auto	Basée sur des caractères			Basée sur des règles d'inférence	
PORSCHE [K.Saleem *et al.* 2008]	XSD	Auto	Basée sur des caractères, Basée sur la langue, Thésaurus de domaine	Clustering , Exploitation de l'arbre			Schéma de médiation
Falcon-AO [W.Hu *et al.* 2008]	RDF, OWL	Auto, Instances	Basée sur des caractères, WordNet	Affinité structurelle			
FSM [L.Ivantysynova & T.Scheffer 2008]	Schémas : bases de données	Auto, Instances	Basée sur des caractères				
Anchor-Flood [M.S.Hanif & M.Aono 2009]	RDFS, OWL	Auto	Basée sur des caractères, Basée sur la langue, WordNet	Interne, externe			

Approche	Entrée	Besoins	Mesures terminologiques	Mesures structurelles	Mesures extensionnelles	Mesures sémantiques	Observation
AOAS [S.Zhang & O.Bodenreider 2007]	OWL	Auto	Basée sur des caractères, Basée sur la langue	exploitent des relations entre des entités hyponyme-hyperonyme (is-a)		Basée sur des règles d'inférence	
GLUE [A.H.Doan *et al.* 2004]	Schémas : bases de données relationnelles, Taxonomie XML	Auto, Instances,	La technique d'apprentissage Bayes naïf	Structure hiérarchique	Mesure basée sur des instances, basée sur des contraintes de domaine		
OMviaUO [V.Mascardi *et al.* 2010]	RDFS, OWL	Auto	Basée sur des caractères, Basée sur la langue	Structure taxonomique		Basée sur des règles d'inférence	
VSBM et GBM [K.Todorov & C.Hudelot 2010]	Ontologies	Auto, Instances	Statistiques, SVM	Les corrélations dans un graphe multimédia			

Approche	**Entrée**	**Besoins**	**Mesures terminologiques**	**Mesures structurelles**	**Mesures extensionnelles**	**Mesures sémantiques**	**Observation**
BLOOMS/ BLOOMS+ [P.Jain *et al.* 2010]	RDFS, OWL	Auto	Basée sur la langue, API d'alignement	Structure taxonomique		Basée sur des règles d'inférence	
DSSim [M.Nagy & M.Vargas-Vera. 2010]	OWL, SKOS	Auto	Basée sur des caractères, Basée sur la langue, WordNet		Mesures basées sur des instances		Question-réponse
TaxoMap [F.Hamdi *et al.* 2010]	OWL	Auto, Semi	Basée sur des caractères, Basée sur la langue	Comparaison de la structure par l'intermédiaire des hiérarchies is-a			
LogMap [E.Jimenez-Ruiz & B.C.Grau 2011]	OWL	Auto, Semi	Basée sur des caractères, Basée sur WordNet	Comparaison de structure		Satisfiabilité, propositionnelle	

Approche	Entrée	Besoins	Mesures terminologiques	Mesures structurelles	Mesures extensionnelles	Mesures sémantiques	Observation
XMAP ++ [W.Djeddi & M.T.khadir 2014]	Ontologies en OWL-DL	Semi	Basée sur des caractères, Basée sur la langue,WordNet, Similarités agrégées	Basée sur des informations sur la présence des propriétés et sur leurs contraintes de cardinalités			RNA permet de calculer la meilleure correspondance entre des couples d'entités, permet de maximiser la découverte du nombre de couples similaires et réduire le nombre de ceux qui sont dissimilaires.
RiMOM-IM [C.Shao *et al.* 2014]	Ontologies	Semi	Mesures basées sur des tokens, Similarités agrégées		Basée sur des instances		

3.3 Statistiques

Figure(3.1) montre que les recherches dans le domaine d'alignement des ontologies ont commencé début des années quatre vingt dix. Début des années deux milles, le nombre de travaux dans ce domaine devient de plus en plus important avec l'apparition de la notion du Web sémantique. Ce nombre atteint son maximum l'année 2010. Les recherches se poursuivent jusqu'à ce jour [F.Ardjani *et al.* 2015].

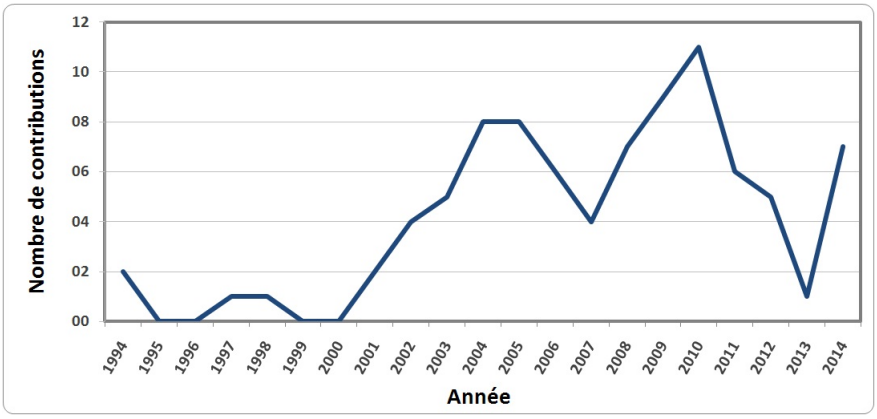

Figure 3.1 – *Évolution du nombre des travaux dans le domaine d'alignement des ontologies [F.Ardjani et al. 2015]*

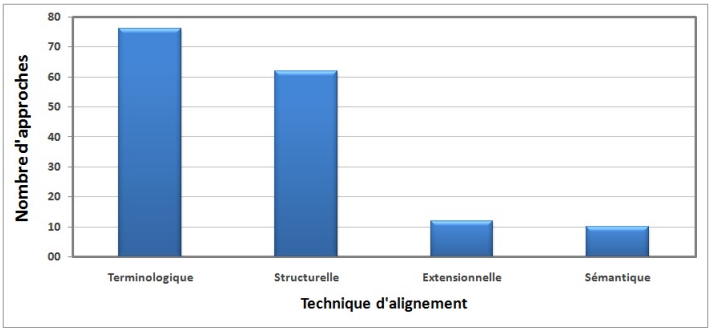

Figure 3.2 – *Taux d'utilisation des techniques d'alignement (terminologique, structurelle, extensionnelle et sémantique)[F.Ardjani et al. 2015]*

A partir de la Figure(3.2) il est clair que la méthode terminologique intervient dans un nombre important d'approches. La méthode structurelle marque aussi son importance parmi les autres techniques. Ceci peut être expliqué par le fait que les méthodes basées sur les termes ou la structure sont souvent maîtrisables et faciles à mettre en œuvre. A la différence des méthodes sémantiques qui nécessitent la disponibilité de sources sémantiques difficiles à construire. Elles nécessitent aussi des moteurs de raisonnement complexes pour déduire des relations sémantiques[F.Ardjani *et al.* 2015].

Conclusion

Le but de ce chapitre était de recenser les travaux dans le domaine d'alignement des ontologies. C'est un travail qui peut éclaircir le chemin des chercheurs dans ce domaine. Ils peuvent ainsi choisir l'approche adéquate à leur probleme. Ils peuvent aussi voir les insuffisances et les corrigés, ou encore proposer de nouvelles approches d'alignement.

Notre travail dans cette thèse est motivé par l'hépothèse suivante : si un alignement spécifie de manière explicite les correspondances entre entités similaires de deux ontologies, alors Cet alignement spécifie aussi quelles ressources sont le plus probable d'*ê*tre liées.

Nous présentons dans le chapitre **5** une nouvelle approche pour découvrir automatiquement les liens morts et erronées entre deux bases (inter-base) en se basant sur l'alignement d'ontologie.

Ainsi, nous présentons un nouveau système d'alignement d'ontologie "**ABCMap**" fondé sur une méthode d'optimisation qui se base sur les colonies d'Abeilles Artificielles (ABC).

Découverte et Maintenance des liens : Etat de l'art

4

Sommaire

Ce Chapitre étudie le processus de Découverte et de Maintenance des liens entre des jeux de données. cet aspect est essentiel pour s'assurer qu'un jeu de données est intégré au Web et que tous les liens décrits sont totalement correctes une fois le jeu localisé.

Les liens RDF peuvent être définis manuellement ou automatiquement. Le choix de la méthode dépendra du jeu de données et du contexte de publication. Des liens manuels sont généralement utilisés pour des petits jeux statiques, alors que des méthodes automatiques ou semi-automatiques seront choisies pour des jeux de grande taille.

La méthode manuelle ne s'adapte pas bien à de grands jeux de données, comme les 413 000 lieux de DBpedia et les entrées correspondantes dans **Geonames**. La stratégie habituelle consiste alors à utiliser des heuristiques de liens automatiques, ou semi-automatiques, pour générer des liens RDF entre les sources de données.

Le couplage d'enregistrements, aussi appelé résolution d'identité ou détection de duplications, est un probleme bien connu dans les bases de données, et dans la communauté d'alignement d'ontologies [T.Heath & C.Bizer 2010].

De nombreuses techniques de ces champs sont directement applicables dans un contexte de données liées. En principe, il y a deux techniques principales de couplage. Une simple, fondée sur des **clés**, qu'exploite les schémas de nommage communs aux deux sources. La seconde, plus compliquée, se fonde sur des **similarités** : elle compare les items et les lie pour voir si leur similarité dépasse un certain seuil.

4.1 Approches fondées sur des clés

Dans un grand nombre de domaines, il y a des schémas de nommage communément acceptés. Par exemple, les GTIN (Global Trade Item Numbers, numéros d'item en commerce mondial) sont généralement utilisés pour identifier des produits. Pour la publication, il existe les numéros ISBN ; dans la finance, les identifiants ISIN. Si ces identifants figurent dans un jeu de données, ils devraient être exposés dans l'URL ou dans les valeurs des propriétés. Ces propriétés sont appelées des propriétés inversement fonctionnelles. Leurs valeurs identifiant de manière unique l'objet d'un triplet. Elles devraient être défnies comme telles dans le vocabulaire correspondant, et déclarées de type **owl :InverseFunctionalProperty**. En incluant les identifiants communément acceptés dans les URI, ou comme des propriétés inversement fonctionnelles dans des données publiées, on construit les fondations pour utiliser des algorithmes basés sur des motifs pour générer des liens RDF entre les données. ProductDB est un exemple de source de données utilisant des codes GTIN pour les produits dans ses URI. Il associe l'URI **http ://productdb.org/gtin/0978185326780 2** à une version particulière de l'origine des espèces de **Charles Darwin**. Les alias d'URI sont aussi créés à partir des identifiants ISBN et EAN du livre, ce qu'aide par ailleurs les liens basés sur des clés [T.Heath & C.Bizer 2010]. Une clé est un ensemble de propriétés particulièrement discriminant car les valeurs de ces propriétés permettent d'identifier un objet du monde réel sans ambiguïté. Ces clés peuvent être utilisées par un raisonneur pour inférer logiquement des liens d'identité ou pour construire des fonctions de similarité plus complexes prenant en compte des mesures de similarités élémentaires entre littéraux. Parmi les approches fondées sur des clés en peut cité : "*KD2R* [D.Symeonidou *et al.* 2011], *C-SAKey*[D.Symeonidou *et al.* 2015]".

4.2 Approches se basant sur des similarités

Dans le cas où il n'existerait pas d'identifiants communs entre les jeux de données, il est nécessaire d'utiliser des heuristiques de couplage plus complexes, fondées sur des similarités. Ces heuristiques peuvent comparer les propriétés des entités à lier autant que celles des entités liées. Elles agrègent les différents scores de similarité et lient les entités si la valeur agrégée de similarité dépasse un certain seuil. Par exemple, **Geonames** et **DBpedia** fournissent tous deux des informations sur des lieux géographiques. Pour identifier des endroits qu'apparaissent dans les deux jeux de données, on pourrait utiliser une heuristique qui compare les noms de lieux, grâce à une fonction de similarité de chaînes de caractères, les valeurs de longitude et de latitude avec un outil de correspondance géographique, le nom du pays où les endroits sont localisés, ainsi que la population totale. Si toutes (ou la plupart) des comparaisons donnent de bons résultats de similarité, on considère que les deux lieux sont identiques [T.Heath & C.Bizer 2010].

Comme on ne peut pas supposer que les sources de données du Web fournissent des descriptions complètes des ressources, les heuristiques de similarité pourraient être choisies pour tolérer certaines valeurs man-

quantes. DBpedia, par exemple, ne contient que les chiffres totaux de population pour une partie des endroits décrits. Une heuristique de couplage appropriée pourrait dès lors donner un plus grand poids au pays dans le cas où ce total est manquant. Il y a plusieurs outils pour définir des heuristiques de couplage d'une manière déclarative. Ils automatisent le processus de génération de liens RDF en se basant sur ces déclarations.

4.2.1 Silk-Link Discovery Framework

Silk fournit un langage déclaratif flexible pour spécifier des heuristiques de correspondance, qui peuvent combiner différents comparateurs de chaînes, numériques autant que géographiques. Il peut transformer les valeurs des données avant l'utilisation dans le processus de correspondance. Il accepte aussi l'agrégation des scores de similarité en utilisant diverses fonctions d'agrégation. Silk peut faire correspondre des jeux de données tant locaux que distants, auxquels on accède par le biais du protocole SPARQL [T.Heath & C.Bizer 2010].

Les tâches de correspondance qui requièrent un grand nombre de comparaisons peuvent être gérées par le biais de fonctionnalités de blocage ou par le lancement de Silk dans un cluster Hadoop.

Silk fournit une boîte à outils pour découvrir et maintenir les liens entre les sources de données du Web.

Silk se compose de trois composants :

1. Un moteur de découverte de liens, qui calcule les liens entre les sources de données ;

2. Un outil pour évaluer les liens de données générés afin d'affiner la spécification de liaison ;

3. Un protocole pour maintenir les liens entre les sources de données.

4.2.2 LIMES-Link Discovery Framework for Metric Spaces

LIMES [T.Heath & C.Bizer 2010] implémente une méthode rapide, mais sans perte, pour la découverte des liens à grande échelle, en se basant sur les caractéristiques des espaces métriques. Il fournit cependant un langage moins expressif pour spécifier les heuristiques de correspondance.

4.2.3 Outils fondés sur l'apprentissage

Outre ces outils, qu'obligent leurs utilisateurs à défnir explicitement l'heuristique de correspondance, il en existe aussi qu'apprennent l'heuristique de correspondance directement des données. Parmi eux, citons *Knofuss*, *RiROM*, *idMesh* et *ObjectCoref* [T.Heath & C.Bizer 2010]. L'avantage des heuristiques d'apprentissage est que les systèmes ne doivent pas être manuellement configurés pour chaque type de liens créés entre les jeux. L'inconvénient est que les approches fondées sur un apprentissage automatique ont généralement une précision inférieure à celles qui s'appuient sur la connaissance du sujet apportée par des humains sous la forme d'une description de correspondances.

Tableau 4.1 – *Comparaison des outils de découverte et de maintenance des liens*

	Silk [J.Volz *et al.* 2009]	**LIMES** [A.Cyrille *et al.* 2011]	**Knofuss** [A.Nikolov *et al.* 2007]	**RIMOM** [J.Tang *et al.* 2004]
Les données en entré	RDF, SPARQL, CSV	RDF, SPARQL, CSV	RDF, SPARQL	RDF, OWL
Les types de liens	owl :sameAs "utilsateur spécifie d'autres types"	owl :sameAs "utilsateur spécifie d'autres types"	owl :sameAs	owl :sameAs
Configuration	manuel	manuel	manuel	adaptatif, moyenne pondérée
Techniques d'alignement	chaînes de car	chaînes de car	chaînes de car apprentissage adaptatif	chaînes de car
Alignement ontologie	non	non	oui, en entrée	oui

La tâche d'appariement d'instances de l'initiative pour l'évaluation d'alignements entre ontologies en 2010 a comparé la qualité des liens qui ont été produits par différents outils reposant sur de l'apprentissage. L'évaluation a révélé des valeurs de précision entre 0,7 et 0,97 et a montré que la qualité des liens dépend largement de la tâche de couplage spécifique.

Une tâche liée à la génération de liens est leur maintenance à long terme, puisque les sources de données changent. Dans [P.Niko & H.Bernhard 2010], les auteurs proposent *DSNotify*, un framework qui surveille les sources de données liées et informent les applications consommatrices des changements.

L'outil *LiQuate* [Ruckhaus *et al.* 2013] utilise une approche d'évaluation fondée sur les réseaux bayésiens pour identifier les ambiguïtés et les mises en relations incomplètes dans les liens entre les ressources de bases différentes. Le résultat final de cette évaluation suggère à un expert les ressources potentiellement mal liées.

Conclusion

Le but de ce chapitre est de recenser les travaux dans le domaine de la découverte et de la maintenance des liens.

Donc les chercheurs focalisent leurs travaux de découverte et de maintenance des liens entre un ensemble de bases RDF. Par contre nos approches, présenté dans le chapitre suivant donnent des solutions de maintenance des liens à l'intérieur d'une base RDF "intra-Base" et entre un ensemble de base RDF "inter-Base".

Nous proposons dans le chapitre suivant une approche pour identifier automatiquement les liens "corrects, erronés et morts" entre les données RDF en se basant sur les modèles de liens. Nos approches inclut aussi un processus pour maintenir les liens quand un changement de données se produit.

Partie II : Détection et maintenance de liens

La découverte et la maintenance des liens entre les données liées RDF

5

Sommaire

De nombreux jeux de données sont publiés sur le web à l'aide des technologies du web sémantique. Ces jeux de données contiennent des données qui représentent des liens vers des ressources similaires. Si ces jeux de données sont liés entre eux par des liens construits correctement, les utilisateurs peuvent facilement interroger les données à travers une interface uniforme, comme s'ils interrogeaient un jeu de données unique. Dans ce chapitre, nous proposons une approche pour découvrir automatiquement les liens entre les données RDF en se basant sur les modèles de liens qui apparaissent autour des ressources. Notre approche inclut aussi un processus automatique pour maintenir les liens quand un changement de données se produit.

Le travail présenté dans ce chapitre à fait l'objet de deux publications internationales

1 Une approche pour la découverte et la maintenance des liens entre les données liées RDF a été proposée dans [F.Ardjani *et al.* 2017]. Elle vise à détecter les liens corrects et les liens erronés dans la même base (les liens intra-base), et dans un ensemble de base (les liens inter-base). On donne aussi une méthode de maintenance pour éviter hétérogénéité.

2 Une étude sur les approches d'alignement d'ontologie a été présentée dans un article intitulé "Alignement des ontologies : État de l'art" [F.Ardjani *et al.* 2015].

5.1 Concepts de base

5.1.1 Modèles de liens intra-Base

Soit S un ensemble de liens RDF.

Nous définissons deux ensembles pour la base RDF : l'ensemble des ressources relations R_s, et l'ensemble des ressources nœuds N_s, avec :

— L'ensemble R_s contient les relations des liens de S.

— N_s contient les couples (sujet, objet) participant aux liens de S.

Le contenu d'un document RDF peut être vu comme l'ensemble des descriptions des ressources relations qu'il contient. Dans la terminologie RDF, une description d'une ressource relation désigne en particulier l'ensemble des ressources nœuds autour d'une ressource relation donnée, c'est-à-dire son voisinage.

On définit la fonction voisinage comme suit :

Voisinage : $R_s \rightarrow N_s$

$r \rightarrow voisinage(r) = \{(n_i, n_j)/(n_i, r, n_j) \in S\}$

Nous définissons le modèle de liens intra-base, le regroupement d'une ressource relation et de son voisinage. Le modèle de liens est composé de la ressource relation et de sa description dans la base RDF, c'est-à-dire le voisinage de la ressource relation. Un modèle de liens est donc composé de liens qui contiennent une ressource relation en communs et un ensemble de couples de ressources nœuds $(n_1; n_2)$.

La taille d'un modèle de liens est le nombre de liens qu'il contient.

5.1.2 Modèles de liens inter-Base

Soit deux bases RDF B_1 et B_2. Soit aussi S, l'ensemble de liens externes entre les bases B_1 et B_2.

Nous définissons N_{ob} l'ensemble de nœuds objets de B_1, N_{sj} l'ensemble de nœuds sujets de B_1.

Nous définissons aussi l'ensemble des ressources relations R_s, et l'ensemble des ressources nœuds N_s, avec :

— L'ensemble R_s contient les relations des liens externes de S.

— N_s contient les couples (sujet, objet) des liens externes de S.

Une description d'une ressource relation désigne en particulier l'ensemble des ressources nœud autour d'une ressource relation donnée, c'est-à-dire son voisinage.

On définit la fonction voisinage comme suit :

Voisinage : $R_s \rightarrow N_s$

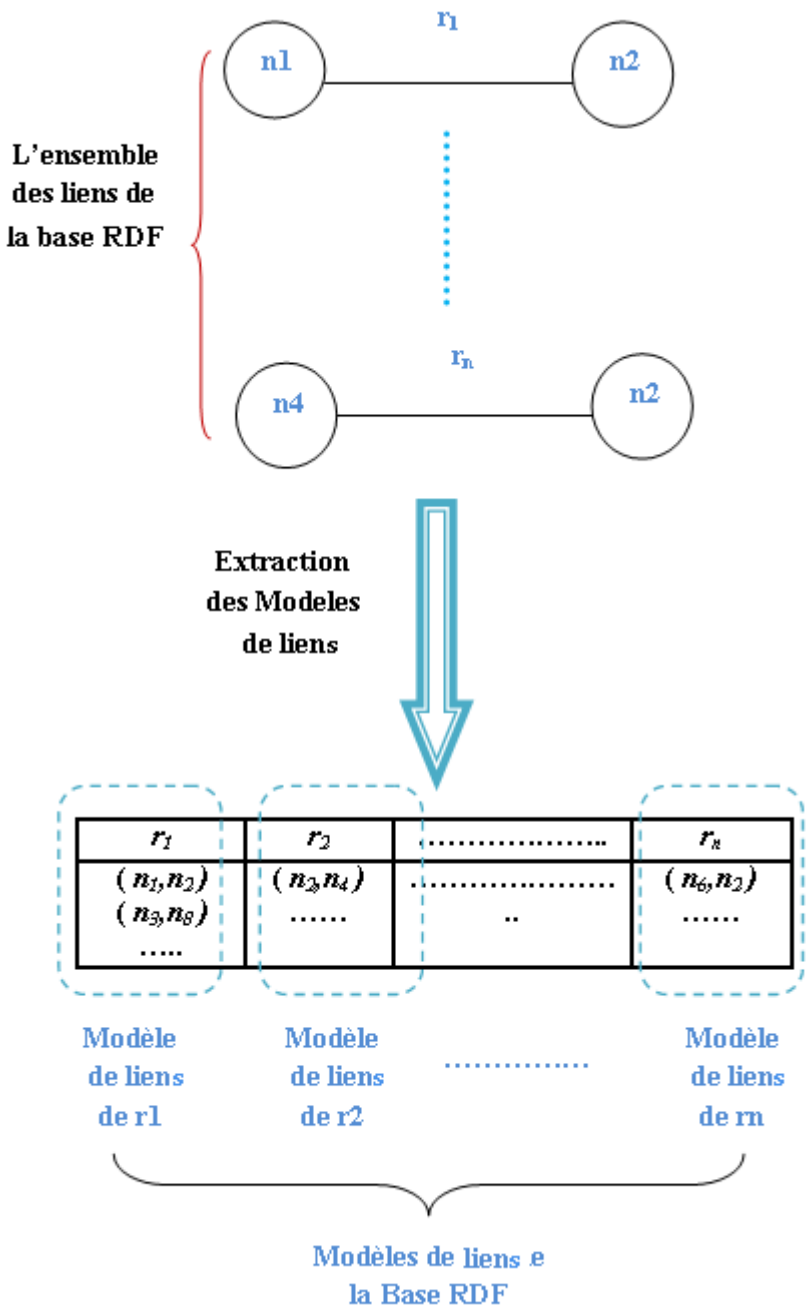

FIGURE 5.1 – *Extraction des Modèles de liens internes*

$r \rightarrow voisinage(r) = \{(n_i, n_j)/(n_i, r, n_j) \in S\}$

Nous définissons le modèle de liens inter-base, le regroupement d'une ressource relation et de son voisinage. Le modèle de liens est composé de la ressource relation et de sa description, c'est-à-dire le voisinage de la ressource relation. Un modèle de liens est donc composé de liens externes qui contiennent une ressource relation en communs et un ensemble de couples de ressources nœuds $(n_1; n_2)$.

La taille d'un modèle de liens est le nombre de liens externes qu'il contient.

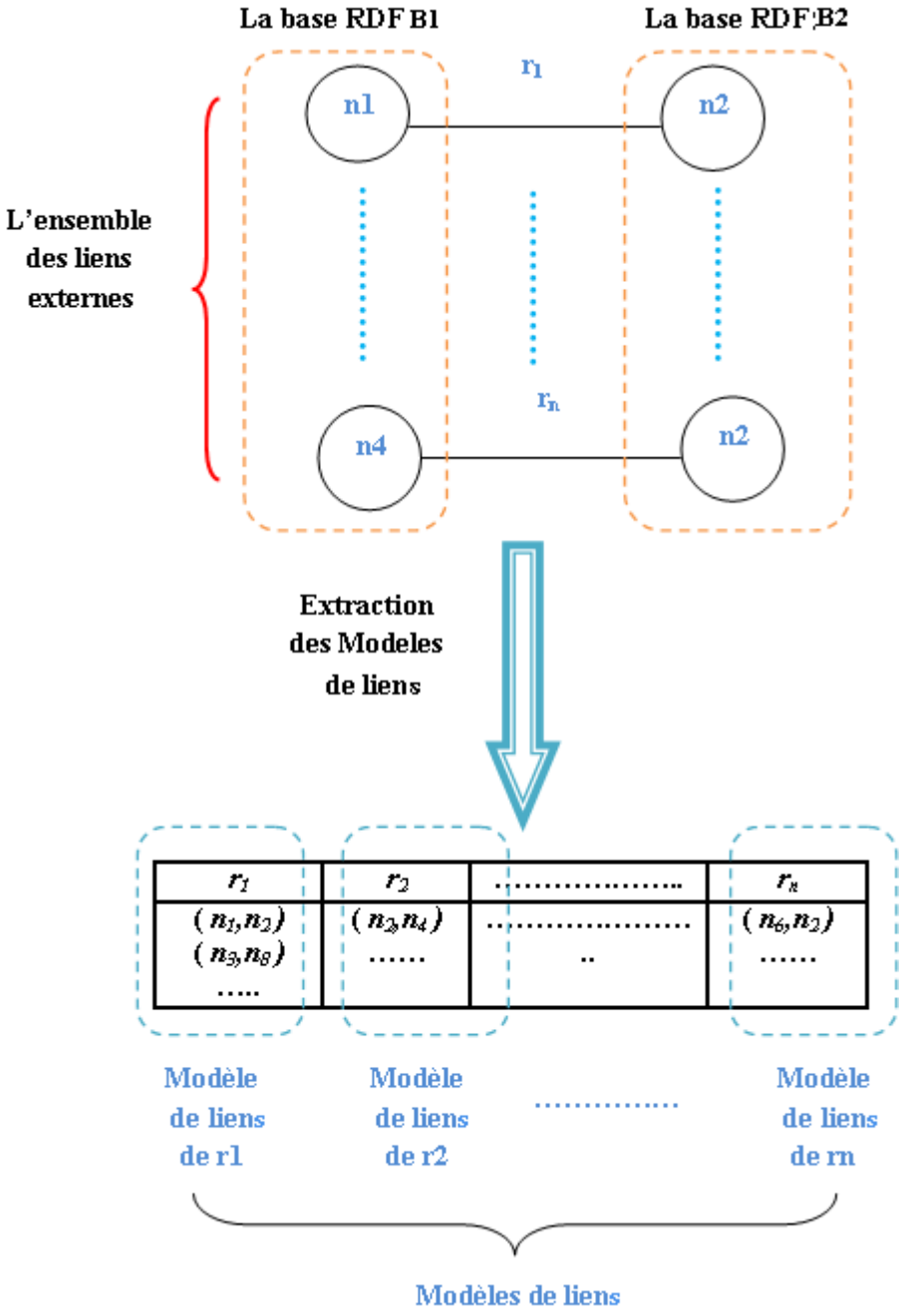

FIGURE 5.2 – *Extraction des Modèles de liens externes*

5.2 APPROCHE PROPOSÉE

Dans le linked data, les ontologies fournissent l'encadrement de la structure de données d'une base RDF. L'ontologie définit l'ensemble des classes et relations qui sont utilisées dans la base. Leurs définitions sont faites de façon à encadrer leurs utilisations par des contraintes pour garder la cohérence des données de la base.

Avec le temps, des liens morts ou erronées peuvent apparaître. Des liens morts sont ceux pointant vers des URI qui ne sont plus entretenues, et sont aussi des liens potentiels non définis même quand de nouvelles

données sont publiées. Trop de liens morts mènent à un grand nombre de requêtes HTTP inutiles envoyées par les applications clientes. Un probleme de recherche actuel abordé par la communauté des Données Liées est celui de la maintenance des liens.

À chaque changement, il est nécessaire de vérifier s'il ne cause pas l'apparition des nouvelles incohérences. Il est nécessaire de vérifier si les équivalences entre ressources de différentes bases se maintiennent à chaque évolution, et de garder la trace des changements effectués.

Le but de notre approche est de détecter les liens corrects et les liens erronés dans la même base (liens intra-base) et entre un ensemble de bases (liens inter-base) et de donner une méthode de maintenance pour éviter cette hétérogénéité.

5.2.1 Processus de découverte des liens "intra-base"

La méthode se fonde sur les données de la base RDF qui sont incohérentes par rapport à l'ontologie. Pour détecter les liens qui respectent ou non l'ontologie, nous utilisons les contraintes d'intégrité.

Les contraintes d'intégrité permettent d'extraire les ressources de la base qui respectent ou qui ne respectent pas l'ontologie. Pour déterminer si certaines de ces ressources sont semblables entre elles, nous définissons les modèles de liens. Les modèles de liens donnent un résumé des points communs d'un ensemble de ressources sous forme d'un ensemble de triplets. Les modèles positifs sont ceux qui respectent une contrainte et les modèles négatifs sont ceux qui ne la respectent pas.

Les modèles positifs sont ceux qui respectent une contrainte c'est à dire un ensemble de triplets qui respectent la contrainte d'intégrité d'une ontologie. On peut donc extraire le nombre de liens corrects dans cet ensemble.

Les modèles négatifs sont ceux qui ne respectent une contrainte, c'est à dire un ensemble de triplets qui ne respectent la contrainte d'intégrité d'une ontologie. On peut donc extraire le nombre de liens erronées dans cet ensemble. La méthode est schématisée dans figure.5.3

Notre approche utilise deux contraintes d'intégrité : les contraintes de typage de domaine et co-domaine pour vérifier les liens qui respectent ou non l'ontologie.

Pour bien comprendre les deux contraintes, on va présenter un exemple illustratif d'un lien.

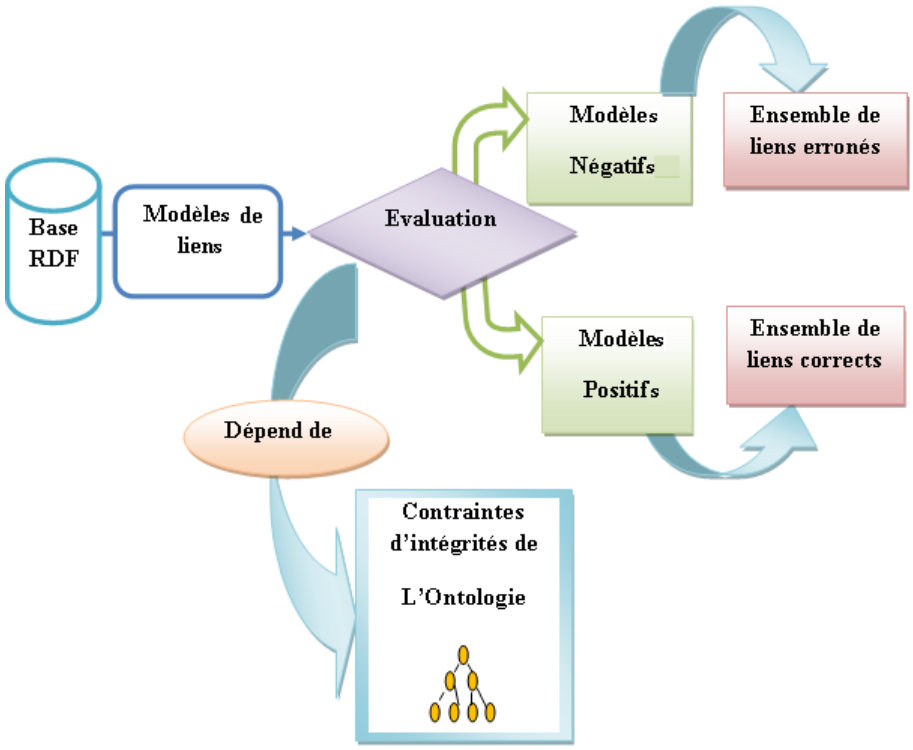

FIGURE 5.3 – *L'approche proposée pour la détection des liens intra-base*

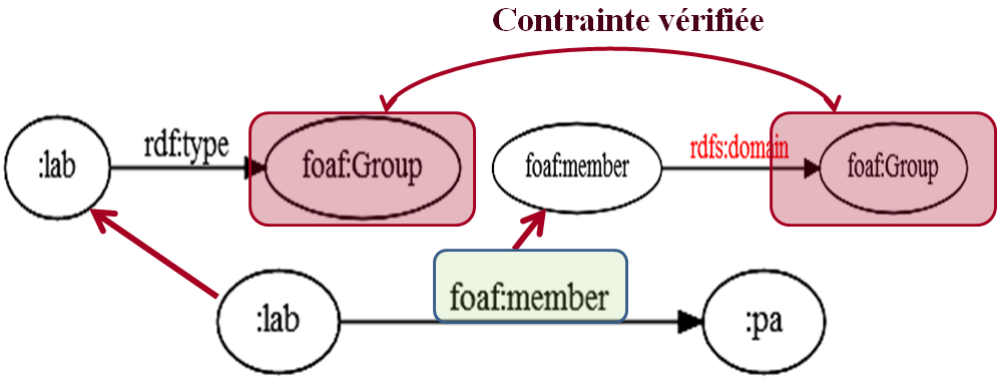

FIGURE 5.4 – *Contrainte de typage de Domaine*

La figure représente un lien avec une ressource sujet "**lab**" et une ressource objet "**pa**" et une ressource prédicat "**member**".

D'aprés ce lien on constate bien que la propriété "**RDF :type**" précise que le sujet "**lab**" est une instance de la classe "**groupe**", et que la propriété "**rdfs :domain**" spécife la propriété "**member**" pour le domaine de la classe "**groupe**". En pratique, cela signifie que lorsque la propriété spécifiée est utilisée dans un triplet, le sujet de ce dernier sera toujours une instance de la classe spécifiée par la propriété **rdfs :domain** dans l'ontologie. Donc on peut dire que la contrainte de typage de Domaine est vérifiée.

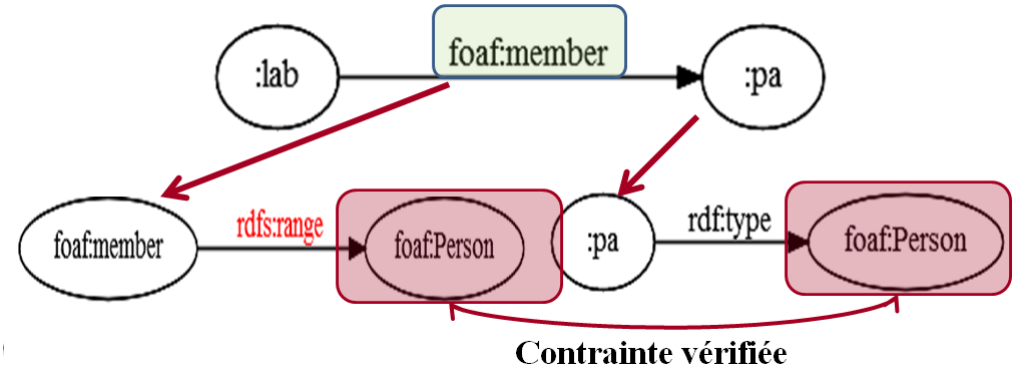

FIGURE 5.5 – *Contrainte de typage de Co-Domaine*

D'aprés ce lien, on constate bien que la propriété "**RDF :type**" précise que l'objet "**pa**" est une instance de la classe "**groupe**", et que la propriété **"rdfs :range"** spécifie la propriété "**member**" a la portée de la classe "**groupe**".

En pratique, cela signifie que lorsque la propriété spécifiée est utilisée dans un triplet, l'objet de ce dernier sera toujours une instance de la classe spécifiée par la propriété **rdfs :range** dans l'ontologie. Cela peut être employé pour spécifier aussi bien les propriétés de portée littérale. Donc on peut dire que la contrainte de typage de Co-Domaine "Range" est vérifiée.

On a déjà vue que notre approche se base sur les modèles de liens, donc on va commencer par une base RDF contenant un ensemble de lien, donc après l'étape d'extraction des modèles de liens, on obtient un ensemble de modèles de liens. Chaque modèle de liens est composé de liens qui contiennent une ressource relation (*prédicat*) en commun et un ensemble de couples de ressources nœuds (n_x, n_y), c'est-à-dire, sujet et objet.

La structure définie pour l'extraction des modèles de liens est représentée sous forme d'arbre avec une racine Modèles de liens (Figure 5.6). Sous la racine, on trouve plusieurs modèles, chacun correspond à un prédicat. Chaque modèle regroupe comme fils un ensemble de liens, ayant bien sur un seul prédicat en commun. Chaque lien possède deux fils : Sujet et Objet. Les fils Sujets et Objets représentent les feuilles de l'arbre.

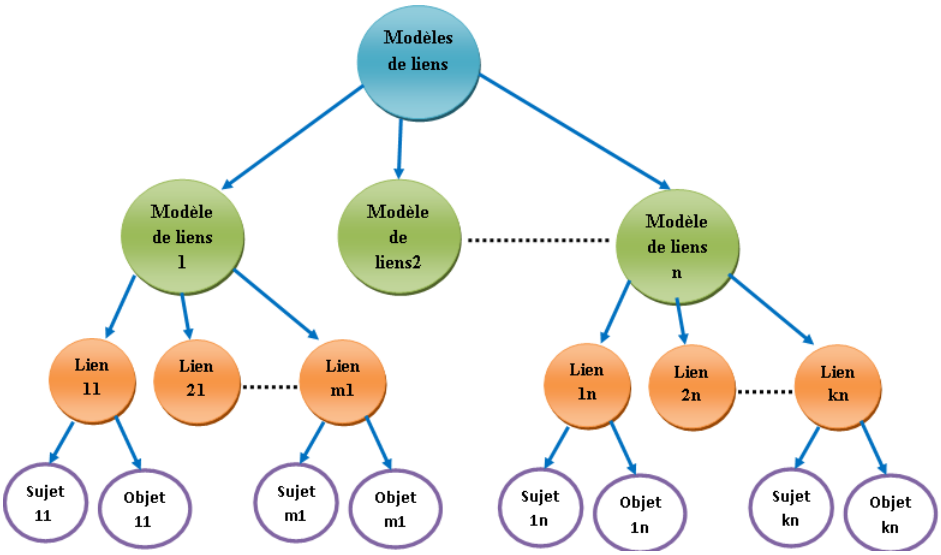

FIGURE 5.6 – *Structure utilisée pour l'extraction des Modèles de liens*

L'algorithme proposé décrit le processus d'extraction des modèles de liens.

Algorithme 1 Algorithme d'extraction des modéles de liens

Entrèes : Un ensemble de liens **L**.
Sorties : Un ensemble de modèles de liens **ML**.
Début
  Pour chaque lien **I** de l'ensemble **L**
    Si Prédicat n'appartient pas *à* l'ensemble de prédicats de **ML**
      Ajouter un nouveau modèle de liens **MLI** dans l'ensemble **ML** avec le nouveau prédicat dans l'ensemble **ML**
      Ajouter sujet et objet du lien en entrée au nouveau modèle de liens
    Sinon
      Ajouter le sujet et l'objet du lien en entrée au modèle de liens avec le prédicat qui est équivalent au prédicat du lien en entrée
    Fin Si
  Fin Pour
Fin

Complexité : la complexité de cet algorithme est de l'ordre **O(n)**, avec **n** le nombre de liens, donc c'est une compléxité linéaire.

On peut dire que la structure du modèle de liens est plus fiable en application, au lieu d'extraire le domaine et le co-domain de chaque prédicat de chaque lien pour les vérifier avec les types des ressources , on extrait seulement le domaine et le co-domaine du prédicat du modèle de liens.

Après l'extraction des modèles de liens, on va les vérifier pour découvrir les liens corrects et les liens erronés en utilisant les contraintes de typage de domaine "Domain "et co-domaine "Rang".

L'algorithme proposé décrit le processus de vérification des modèles de liens.

Algorithme 2 Algorithme de vérification des modéles de liens "Intra-Base"

Entrèes : Un ensemble de modèles de liens *ML*, la Base *B*, l'ontologie de la base *O*.
Sorties : Un ensemble de liens corrects *C*, Un ensemble de liens erronés *E*.
Début
  Pour chaque modèle de liens *M* de l'ensemble *ML*
    Extraire le domaine "**rdfs :domain**" et le co-domaine "**rdfs :rang**" pour le prédicat *P* du modèle de liens *M* à partir de l'ontologie *O*
    Extraire le type "**rdf : type**" pour le sujet *S* et l'objet *Ob* à partir de la base *B*
    Si Type $S = DomaineP$ et Type $Ob = Co - DomaineP$
      Ajouter le lien *à* l'ensemble des liens corrects *C*
    Sinon
      Ajouter le lien *à* l'ensemble des liens erronés E
    Fin Si
  Fin Pour
Fin

Complexité : la complexité de cet algorithme est de l'ordre O(n*m), avec **n** le nombre de modèles de liens et **m** le nombre de sujets et d'objets, donc la complexité est quadratique.

5.2.2 Processus de Maintenance "intra-Base"

1. Maintenance locale

Après l'étape de vérification des liens, on obtient un ensemble de liens corrects et un ensemble de liens erronés. Notre Système DML crée une base RDF appelée base RDF correcte, et il met l'ensemble des liens corrects dans cette base. Ensuite, il crée une autre base RDF appelée base RDF erronée, et il met l'ensemble des liens erronés dans cette base, afin de la présenter à l'expert s'il existe, pour corriger les erreurs d'édition. Donc le résultat de la maintenance de la base RDF en entrée c'est la base RDF correcte en sortie.

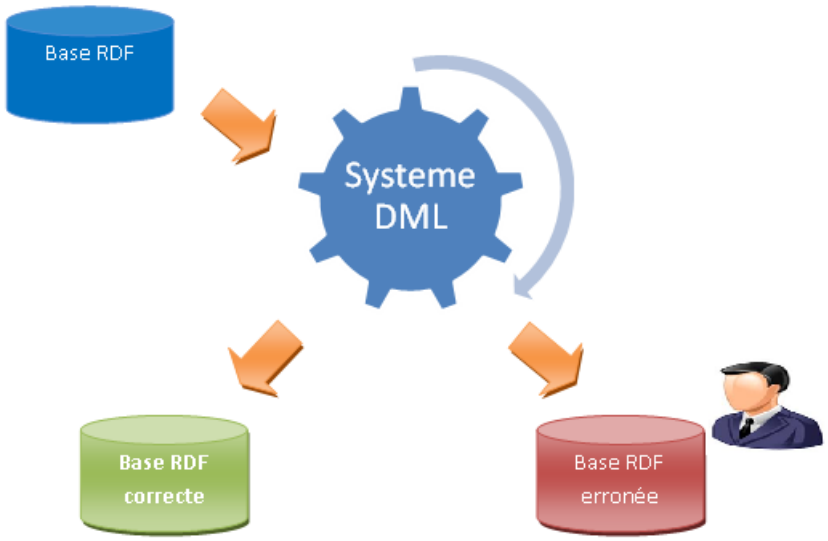

FIGURE 5.7 – *Maintenance locale "intra-Base"*

2. Maintenance globale

Pour maintenir la qualité des données liées, il faut évaluer la qualité des modifications des données, et les filtrer selon le résultat de leur évaluation. Lorsqu'une mise à jour ne respecte pas les contraintes de domaine et co-domaine de l'ontologie, on dit qu'elle est incohérente avec l'ontologie. Donc la modification est refusée par notre système. Lorsqu'une mise à jour respecte les contraintes de domaine et co-domaine de l'ontologie, on dit qu'elle est conforme avec l'ontologie. Donc la modification est acceptée.

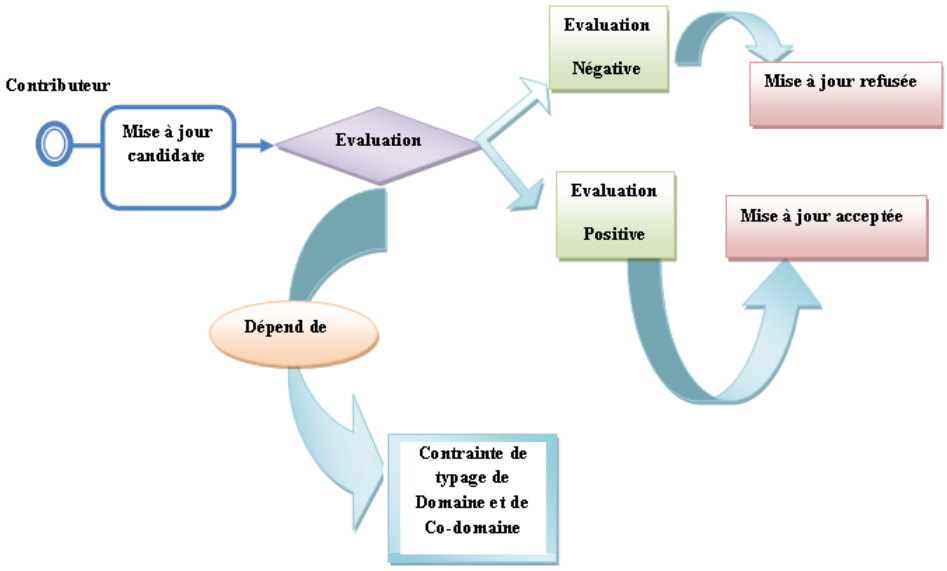

FIGURE 5.8 – *L'approche proposée pour la maintenance "intra-Base"*

5.2.3 Processus de découverte des liens "inter-base"

Notre méthode se base sur les données de deux bases conforment à deux ontologies différentes. Pour identifier les liens, nous utilisons les modèles de liens et l'alignement des deux ontologies.

— Les modèles positifs-base1-base2 sont ceux qui respectent les correspondances de l'alignement des deux ontologies, c'est à dire un ensemble de liens qui respectent les correspondances de l'alignement des deux ontologies. On peut donc extraire le nombre de liens corrects dans cet ensemble.

— les modèles négatifs-base1-base2 sont ceux qui ne respectent pas les correspondances de l'alignement, c'est à dire un ensemble de liens qui ne respectent pas les correspondances de l'alignement des deux ontologies. On peut donc extraire le nombre de liens erronés dans cet ensemble.

Notre méthode de détection des liens inter-base est schématisée dans la Figure 5.9

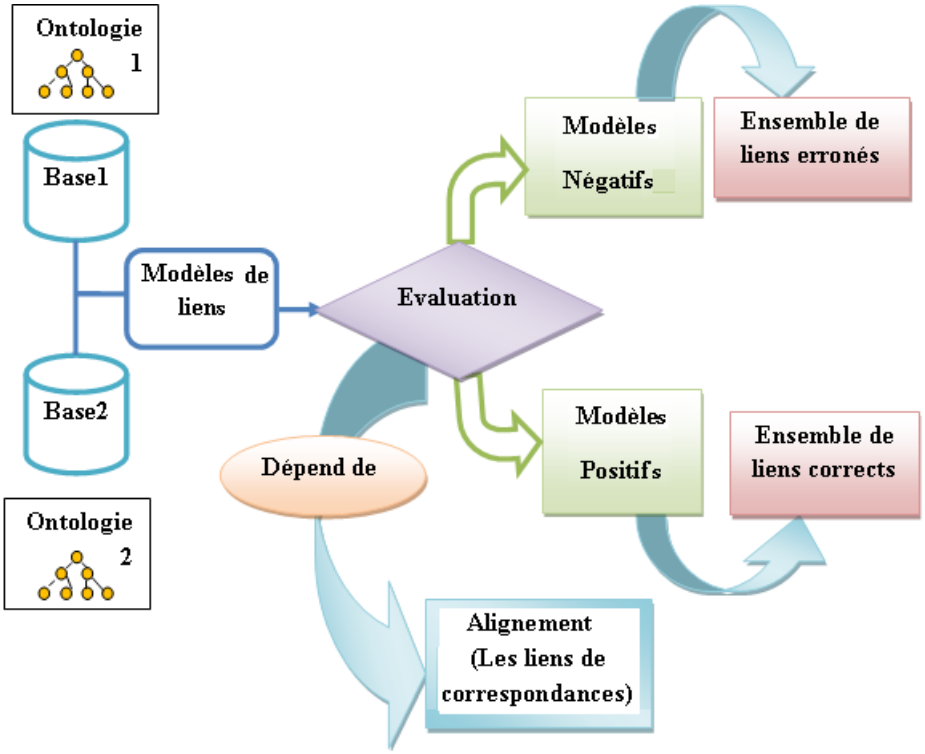

FIGURE 5.9 – *L'approche proposée pour la détection des liens inter-base*

Méthode d'alignement d'ontologie :

L'alignement de deux ontologies consiste à identifier les correspondances entre les entités de ces deux ontologies. On appelle cette phase de découverte des mappings : le processus de matching. Ce processus fait la combinaison de plusieurs méthodes de comparaison (Matchers) pour calculer la similarité entre deux entités.

Ces méthodes calculent la meilleure correspondance entre des couples

d'entités. Elles peuvent augmenter le nombre de couples similaires et réduire le nombre de couples dissimilaires [J.Euzenat & P.Shvaiko 2013].

Définition 1 (Similarité).

La similarité $S : \mathbb{O}X\mathbb{O} \rightarrow R$ est une fonction d'une paire d'entités à un nombre réel exprimant la similarité entre ces deux entités, telle que [J.Euzenat & P.Shvaiko 2013] :

$\forall a, b \in \mathbb{O}, S(a,b) \geq 0$ (positivité)

$\forall a, b, c \in \mathbb{O}, S(a,a) \geq S(b,c)$ et $S(a,a) = S(a,b) \Longleftrightarrow a = b$ (autosimilarité ou maximalité)

$\forall a, b \in \mathbb{O}, S(a,b) = S(b,a)$ (symétrie)

$\forall a, b, c \in \mathbb{O}, S(a,b) = S(b,c) \Longrightarrow S(a,b) = S(a,c)$ (transitivité)

$\forall a, b \in \mathbb{O}, S(a,b) \leq \infty$ (finitude)

La dissimilarité est parfois utilisée au lieu de la similarité. Elle est définie de manière analogue à la similarité, sauf qu'elle n'est pas transitive :

Définition 2 (Dissimilarité).

La dissimilarité $S : \mathbb{O}X\mathbb{O} \rightarrow R$ est une fonction d'une paire d'entités à un nombre réel exprimant la dissimilarité entre ces deux entités, telle que [J.Euzenat & P.Shvaiko 2013] :

$\forall a, b \in \mathbb{O}, DS(a,b) \geq 0$ (positivité)

$\forall a, b, c \in \mathbb{O}, DS(a,a) \leq DS(b,c)$ et $DS(a,a) = 0$ (minimalité)

$\forall a, b \in \mathbb{O}, DS(a,b) = DS(b,a)$ (symétrie)

$\forall a, b \in \mathbb{O}, DS(a,b) \leq \infty$ (finitude)

La distance mesure la dissimilarité de deux entités : si la valeur de la fonction de similarité de deux entités est élevée, la distance entre elle est petite et vice-versa. Elle est définit dans [J.Euzenat & P.Shvaiko 2013] comme suit :

Définition 3 (distance).

La distance $D : \mathbb{O}X\mathbb{O} \rightarrow R$ est une fonction de dissimilarité satisfaisant la définitivité et l'intérate triaunulaire[J.Euzenat & P.Shvaiko 2013] :

$\forall a, b \in \mathbb{O}, D(a,b) = 0 \Longleftrightarrow a = b$ (définitivité)

$\forall a, b, c \in \mathbb{O}, D(a,b) + D(b,c) \geq D(a,c)$ (intératé triaunulaire)

Les valeurs de similarité sont souvent normalisées pour pouvoir être combinées dans des formules plus complexes. Si la valeur de similarité et la valeur de dissimilarité entre deux entités sont normalisées, notées $\overline{S}$ et $\overline{DS}$, alors on a $\overline{S} + \overline{DS} = 1$ [J.Euzenat & P.Shvaiko 2013].

Architecture de notre système d'alignement ABCMap

Dans cette section, nous présentons l'algorithme développé, nommé ABCMap, pour mettre en correspondance deux ontologies représentées en OWL. L'algorithme calcule la similarité entre deux entités de deux ontologies en se basant sur des mesures syntaxiques et linguistiques.

Les valeurs de similarité calculées par les mesures linguistiques et syntaxiques sont appelées les valeurs de similarité partielles entre deux entités, et elles sont stockées dans une base de similarités (Vecteurs). Les valeurs de similarité partielles sont calculées par des mesures normalisées, elles sont donc comprises dans l'intervalle de 0 à 1.

Ces valeurs de similarité partielles sont ensuite agrégées pour atteindre une seule valeur de similarité finale entre deux entités, qui est aussi comprise entre 0 et 1. En examinant cette valeur de similarité finale par rapport à un seuil prédéfini, deux entités sont considérées comme similaires (équivalentes) ou différentes.

Nous commençons par une présentation générale de l'algorithme ABCMap. Cette phase contient les différentes étapes de l'algorithme afin de produire les mappings.

Principe général de l'algorithme ABCMap

Afin de produire des mappings entre deux ontologies, ABCMap exécute un processus qui contient deux phases(voir Figure 5.10).

Phase 1 : Calcul de similarité :

ABCMap calcule trois différentes mesures de similarité pour chaque paire d'entités. Les matchers à base de comparaisons syntaxiques et linguistiques effectuent un calcul de similarité. dans lequel chaque entité de l'ontologie source est comparée avec toutes les entités de l'ontologie cible. Ceci génère une matrice de similarité qui contient un vecteur pour chaque paire d'entités. Ce vecteur se compose de trois valeurs de similarité. Nous employons dans notre système plusieurs méthodes syntaxiques qui calculent par exemple la distance de Jaro-Winkler entre deux entités, et une méthode linguistique basée sur le dictionnaire WordNet. Dans ce qui suit nous donnons plus de détails concernant les matchers linguistiques et syntaxiques utilisés dans notre système.

Matchers linguistiques et syntaxiques ou matchers terminologiques

Nous définissons ci-après les matchers utilisés dans ABCMap :

1. Les matchers syntaxiques : calculent la similarité (ou la dissimilarité) entre deux suites de caractères avec des fonctions et des méthodes de comparaison basées sur les chaînes de caractères.
2. Les matchers lexicaux ou linguistiques : utilisent des ressources auxiliaires pour comparer les mots. ABCMap emploie une source d'information auxiliaire à savoir le dictionnaire WordNet.

1. Matcher syntaxiques

On utilise la distance de Jaro-Winkler [Rijsbergen 1975] et la distance N-gramme pour caculer la similarité entre les entités.

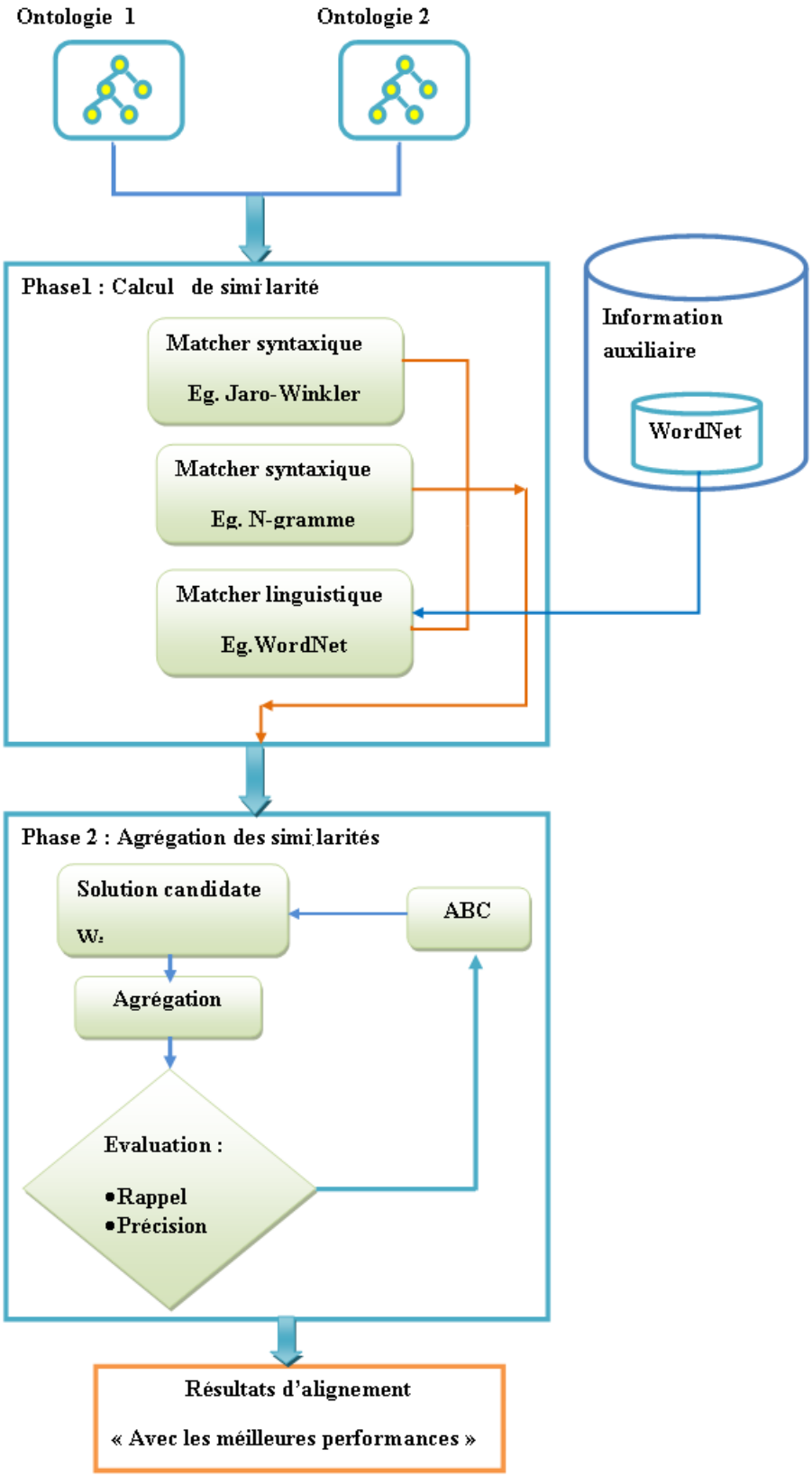

FIGURE 5.10 – *Architecture de notre système d'alignement ABCMap*

Distance de Jaro-Winkler :

La métrique Jaro[W.Winkler 1999] calcule la similarité entre deux chaînes de caractères prenant en compte le nombre et l'ordre des caractères communs entre elles.

Soit s et t deux chaînes de caractères. Soit N_C le nombre des caractères communs figurant dans les deux chaînes dans une distance de moitié de la longueur de la chaîne la plus courte. Soit N_t le nombre des caractères transposés, qui sont des caractères communs figurant dans des positions différentes. La distance de Jaro est une fonction de dissimilarité DS_{jaro} : $SXS \rightarrow [0,1]$ telle que :

$$DS_{jaro}(s,t) = 1 - \frac{1}{3}(\frac{N_C}{|S|} + \frac{N_C}{|t|} + \frac{N_C - N_{t/2}}{N_C}) \tag{5.1}$$

Il existe aussi des distances qui sont des variantes de la distance de Jaro, telle que la distance Jaro-Winkler[W.Winkler 1999] :

(***Distance de Jaro-Winkler***). Soit s et t deux chaînes de caractères. Soit P la longueur du préfixe commun le plus long de s et t. Soit n un nombre positif. La distance de Jaro-Winkler est une fonction de dissimilarité $DS_{Jaro-Winkler} : SXS \rightarrow [0,1]$, telle que :

$$\overline{DS_{Jaro-Winkler}(s,t)} = \overline{DS_{jaro}(s,t)} - \frac{max(P,n)}{10}\overline{DS_{jaro}(s,t)} \tag{5.2}$$

Distance N-gramme :

La distance n-gram [J.Euzenat & P.Shvaiko 2013] calcule le rapport entre le nombre de *n-grams* communs au-dessus du nombre total de *n-grams* entre deux suites de caractères. Typiquement, soit *ngram(s,n)* l'ensemble de toutes les sous-chaînes de s de longueur n, la distance *ngram* entre deux chaînes s et t est définie par la fonction de dis-similarité suivante :

$$DS_{n-grams}(s,t) = |ngram(s,n) \cap ngram(t,n)|/n * Min(|s|;|t|) \tag{5.3}$$

2. Matcher linguistiques

Dans notre approche, le matcher linguistique utilise le dictionnaire WordNet [C.Fellbaum 1998]. WordNet est actuellement la ressource sémantique la plus populaire. Sa connaissance est facile à mettre en pratique dans les applications linguistiques, parce que WordNet prend la forme d'un réseau sémantique simple. En outre, WordNet est disponible gratuitement sur internet à la communauté et facile à obtenir. WordNet est une base de données lexicale de la langue anglaise, qui regroupe les termes (noms, verbes, adverbes et adjectifs) en ensemble de synonymes appelés synsets. Un synset rassemble tous les termes dénotant un concept donné. Un synset contient un ensemble de mots synonymes et leur brève description appelée **Gloss**.

Phase 2 : Combinaison et génération des mappings candidats

Après la première phase, un opérateur d'agrégation pondérée sera appliqué pour combiner les trois similarités calculées par différents matchers en une seule similarité agrégée.

Il s'agit d'associer à chaque valeur de similarité un coefficient de pondération "poids" et à faire la somme des similarités pondérées pour atteindre une nouvelle et unique similarité.

Généralement, les coefficients de pondération " poids " sont attribués manuellement ou par l'utilisation d'une méthode, mais la plupart des méthodes qui sont disponibles, souffrent d'un manque d'optimalité.

Nous développons un Algorithme d'optimisation basé sur les colonies d'Abeilles Artificielles, pour générer une agrégation de similarité basée sur les poids optimaux, afin d'obtenir un alignement bien optimisé.

Dans notre système, deux objectifs dont la précision et le rappel sont optimisés.

Dans le domaine de l'alignement d'ontologies, les métriques de Précision, Rappel, F-mesure [J.Euzenat & P.Shvaiko 2013] sont utilisées pour évaluer la qualité des alignements obtenus. L'OAEI (Ontology Alignement Evaluation Initiative) [J.Euzenat & P.Shvaiko 2013] prend ces métriques pour l'évaluation de la qualité de l'alignement.

La première étape dans le processus d'évaluation de la qualité de l'alignement est de résoudre le probleme manuellement. Le résultat obtenu est considéré comme le mapping de référence.

La comparaison du résultat de l'alignement de référence avec celui obtenu par notre méthode d'alignement produit trois ensembles : $N_{trouvé}, N_{attendu}, N_{correct}$.

L'ensemble $N_{trouvé}$ représente les paires produites par le système.

L'ensemble $N_{attendu}$ désigne l'ensemble des couples appariés dans l'alignement de référence.

L'ensemble$N_{correct}$ est l'intersection des deux ensembles $N_{trouvé}$ et $N_{attendu}$ Il représente l'ensemble des paires appartenants à la fois à l'alignement obtenu et à l'alignement de référence. La précision est le rapport du nombre des paires pertinentes trouvées $N_{correct}$ sur le nombre total des paires produites $N_{trouvé}$:

$$Précision = |N_{correct}|/|N_{trouvé}| \tag{5.4}$$

Le rappel est le rapport du nombre de paires pertinentes trouvées $N_{correct}$ sur le nombre total de paires pertinentes $N_{attendu}$. Il spécifie ainsi la portion des vraies correspondances trouvées. Le rappel est définie par la fonction :

$$Rappel = |N_{correct}|/|N_{attendu}| \tag{5.5}$$

Optimisation de l'alignement par les Colonies d'Abeilles Artificielles

L'algorithme ABC (Artificiel Bee Colony) est présenté par Karaboga et Basturk en 2005, en inspectant les comportements des abeilles réelles pour trouver les emplacements de la source de nourriture "*nectar*", et donner les informations des sources de nourriture aux autres abeilles qui se trouvent dans la ruche.

Dans cet algorithme, les abeilles artificielles sont organisées en trois groupes : abeilles actives (abeilles qui recherche la nourriture), inactives (abeilles d'observation) et scouts ou éclaireuses (abeilles chargées de trouver de nouvelles nourritures, i.e.,le nectar de nouvelles sources) [Mezura-Montes *et al.* 2010].

Pour chaque source de nourriture, il y a seulement une abeille active. C'est-à-dire, le nombre d'abeilles actives est égal au nombre de sources de nourriture [Karaboga & Basturk 2007].

Si l'abeille active ne réussit pas de trouver la source de nourriture, elle va être forcément devenir une abeille éclaireuse pour rechercher aléatoirement de nouvelles sources de nourriture. Les abeilles actives partagent les informations avec les abeilles inactives dans la ruche pour que les abeilles inactives puissent choisir une source de nourriture pour l'explorer. Le processus de l'algorithme ABC est présenté comme suit :

Dans notre approche, l'optimisation par colonie d'abeille a été conçue pour coder différents poids comme des emplacements de la source de nourriture.

Le processus de ABCMap est initialisé avec une population de SN solutions distribuées de façon aléatoire, et l'algorithme recherche alors des solutions optimales " poids ", afin d'obtenir un alignement bien optimisé.

Au cours d'évaluation, la fonction d'agrégation a été calculée en multipliant les poids de sortie par les valeurs de similarités comme indiqué dans l'équation 5.6.

$$func_{agg}(ontologie1_i, ontologie2_j) = \sum_{k=1}^{3} w_k * F_k(smap_{ij}), avec \sum_{k=1}^{3} w_k = 1 \tag{5.6}$$

Par conséquent, on prend les paires d'entités qui ont une similarité supérieure au seuil et les similarités inférieures au seuil sont rejetées.

Dans notre approche, la population s'appelle une colonie, et se compose d'un nombre de solutions ou d'emplacements de source de nourriture candidats $x_i (i = 1, 2, ..., SN)$.

Chaque emplacement x a $n * m$ positions qui représentent $n * m$ vecteurs de poids correspondent à $n * m$ différentes mesures de similarité considérées par l'algorithme, sachant que n est le nombre d'entités dans l'ontologie source et m est le nombre d'entités dans l'ontologie cible.

Un vecteur de poids contient trois cellules ou positions représentent trois poids (valeur de cellule normalisée) pour les trois mesures de similarité, la figure 5.11 présente le codage des sources de nourriture.

La $i^{ème}$ source de nourriture avec trois cellules est convertie en trois poids en utilisant la formules $W_{ij} = X_{ij} / \sum_{i=1}^{3} X_i$; avec $0 \leq Wij \leq 1$ et $\sum_{i=1}^{3}(W_{ij}) = 1$

Le processus ABC est présenté comme suit :

Etape 1- Initialisation :

On commence par sélectionner la population de SN abeilles de façon aléatoire dans l'espace de recherche, sachant que chaque abeille porte une matrice x de n*m positions, chaque position représente un vecteur de "poids". Dans notre cas on a n=3 :

Dans le vecteur de "poids", y a trois positions qui contiennent des valeurs comprises entre 0 et 1. Chaque position représente un poids par rapport à une mesure de similarité.

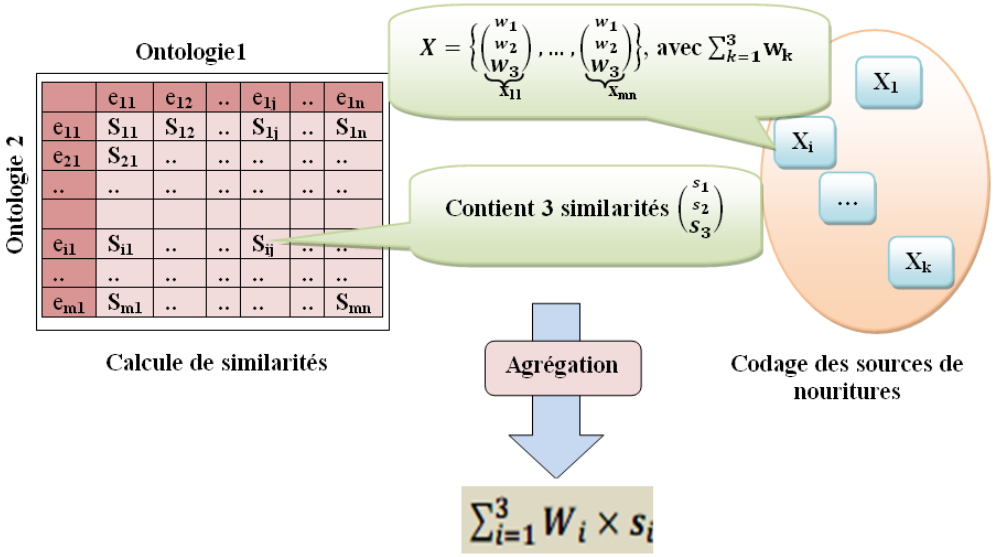

FIGURE 5.11 – *Codage des sources de nourriture*

$$X = \left\{ \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \cdots \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \right\} \tag{5.7}$$

$$\begin{cases} abeille_1 = X_1 \\ abeille_2 = X_2 \\ \vdots \\ abeille_i = X_i \end{cases}$$

Ensuite, on calcule la fitnesse selon cette formule 5.8 :

$$Fitness = \frac{1}{F_{objective}} \tag{5.8}$$

Une fois que ces populations sont placées dans l'espace de recherche, elles prennent le nom : abeilles actives.

Etape 2- Déplacement des abeilles actives : Calculer la probabilité de choisir une source de nourriture par l'équation 5.9,

$$P_i = \frac{o,9 * Fitnesse_i}{\max(Fitnesse)} + 0.1 \tag{5.9}$$

Puis sélectionner une source de nourriture, et ensuite déterminer ses quantités de nectar. L'équation de mouvement des abeilles inactives est donnée comme suit :

$$m_{ij}(t+1) = x_{ij} + yx_{ij}(t) - x_{kj}(t) \tag{5.10}$$

tel que m_{ij} est la $i^{ième}$ position de l'abeille inactive, t est le nombre d'itération, x_{ij} est l'abeille utilisée choisie aléatoirement, **'j'** représente la dimension du vecteur de solution qui produit une série de variables aléatoires dans la gamme [0,1] ; $k \in \{1,2,3,..,N\}$ et $j \in \{1,2,3,..,D\}$ sont choisis aléatoirement ; D est le nombre de paramètre à optimiser ; Kest aussi choisi aléatoirement mais doit être différent de l'indice i.

Etape 3- Déplacer les abeilles éclaireuses : Si la valeur de Fitnesse des abeilles actives ne sont pas améliorées par un nombre d'itérations prédéterminé, appelé "*max-cycle*", ces sources de nourriture sont abandonnées, et l'abeille trouvée dans cet emplacement passera aléatoirement pour explorer d'autres nouveaux emplacements. (Abeilles actives deviennent des abeilles éclaireuses). Cette explication est traduite mathématiquement par l'équation 5.11 :

$$v_{ij} = v_{ij}^{min} + \phi_{ij}(v_{ij}^{max} - v_{ij}^{min})\phi_{ij} \in [0,1] \tag{5.11}$$

Étape 4- Mettre à jour la meilleure solution trouvée jusqu'ici :

Chaque abeille prend la meilleure position que celles trouvées dans les itérations. Cette meilleure position est également appelée meilleure ou meilleure local, cette meilleure position élimine les autres positions acquises par cette abeille.

Ensuite, on prend la meilleure valeur de Fitnesse et de position, qui sont trouvées par les abeilles, et les mémorise.

Étape 5- Critère d'arrêt : Vérifier le processus de calcul jusqu'à ce que le nombre d'itérations atteint la valeur maximale prédéfinie, ou qu'une solution de la fonction objective acceptable soit trouvée.

Toutes ces étapes sont résumées dans l'algorithme.3.

Algorithme 3 Algorithme d'Optimisation par les colonies d'abeilles

Entrèes : S : nombre d'abeilles éclaireuses, W : nombre d'abeilles actives, O : nombre d'abeilles inactives.
Sorties : Meilleure solution.
Initialiser la population avec $S + W$ solutions aléatoires.
Début
    Evaluer la fitnesse de la population
    Tant que le critère d'arrêt n'est pas satisfait "le nombre d'itération"
        Recruter O abeilles inactives et attribuer
        chacune *à* un membre de la population
        Pour chaque abeille inactive affectée *à* un membre **n** de la population
            Effectuer une itération de l'algorithme de
            recherche de nouvelle source
        Fin Pour
        Evaluer la fitnesse de la population
        Si un membre de la population ne s'est pas amélioré au cours des itérations
            Sauver la solution et remplacer la par une solution aléatoire
        Fin Si
        Trouver S solutions aléatoires et remplacer les S membres de la population qui ont la mauvaise fitnesse
    Fin Tant que
    Retourner la meilleure solution
Fin

Processus de vérification des liens "inter-base"

Notre méthode se base sur les données de deux bases conforment à deux ontologies différentes. Pour identifier les liens, nous utilisons les modèles de liens et l'alignement des deux ontologies.

— Les modèles positifs-base1-base2 sont ceux qui respectent les correspondances de l'alignement des deux ontologies, c'est-à-dire, un ensemble de liens qui respectent les correspondances de l'alignement des deux ontologies. On peut donc extraire le nombre de liens corrects dans cet ensemble.
— les modèles négatifs-base1-base2 sont ceux qui ne respectent pas les correspondances de l'alignement, c'est-à-dire, un ensemble de liens qui ne respectent pas les correspondances de l'alignement des deux ontologies. On peut donc extraire le nombre de liens erronés dans cet ensemble.

Pour dire qu'un lien est correct :

— premièrement, il faut vérifier si le lien respecte les correspondances de l'alignement de deux ontologies,
— deuxièment, il faut évaluer l'étiquette (label) du sujet avec l'étiquette de l'objet à l'aide d'une mesure de similarité (jaro Similarity), si le résultat est supérieur à un certain **Seuil** alors on peut dire que le lien est correct sinon le lien est eronné.

L'algorithme ci-dessous décrit le processus de vérification des liens.

Algorithme 4 Algorithme de vérification des modéles de liens "Inter-Base"

Entrèes : Un ensemble de modèles de liens *ML*, Les corespondances de l'alignement des deux ontologies *CR*.
Sorties : Un ensemble de liens corrects *C*, Un ensemble de liens erronés *E*.
Début
  Pour chaque modèle de liens *M* de l'ensemble *ML*
    Pour chaque Lien $L \in M$
      Pour chaque sujet $S \in L$ et objet $Ob \in L$
        Si (S et Ob) rdf :type cr $\in$ *CR* et $Sim_{jaro}(label_{sujet}; label_{objet}) > seuil$
          Ajouter le lien *à* l'ensemble des liens corrects *C*
        Sinon
          Ajouter le lien *à* l'ensemble des liens erronés *E*
        Fin Si
      Fin Pour
    Fin Pour
  Fin Pour
Fin

Complexité : la complexité de cet algorithme est de l'ordre **O(n*m*k)**, avec **n** le nombre de modèles de liens et **m** le nombre de lien et **k** le nombre de sujets et d'objets .

5.2.4 Processus de maintenance "Inter-Base"

2. Maintenance locale

Après l'étape de vérification des liens, on obtient un ensemble de liens externes corrects et un ensemble de liens externes erronés. Notre Système crée une base RDF appelée base RDF externe correcte, et il met l'ensemble des liens externes corrects dans cette base. Ensuite, il crée une autre base RDF appelée base RDF externe erronée, et il met l'ensemble des liens erronés externes dans cette base, afin de la présenter à l'expert s'il existe, pour corriger les erreurs d'édition. Donc le résultat de la maintenance de la base RDF en entrée est la base RDF correcte en sortie.

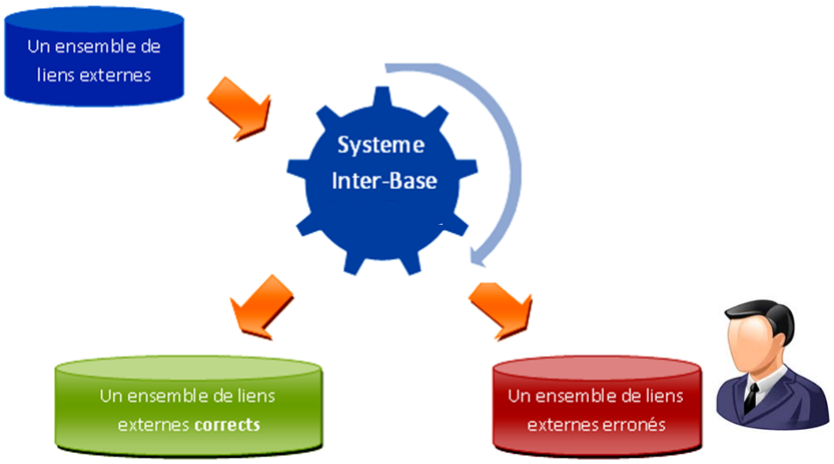

FIGURE 5.12 – *Maintenance "Inter-Base"*

2. Maintenance globale

Pour maintenir la qualité des données liées, il faut évaluer la qualité des modifications des données et les filtrer selon le résultat de leur évaluation. Lorsqu'une mise à jour ne respecte pas les correspondances de l'alignement d'ontologies, on dit qu'elle est incohérente avec les correspondances de l'alignement d'ontologies, donc la modification est refusée.

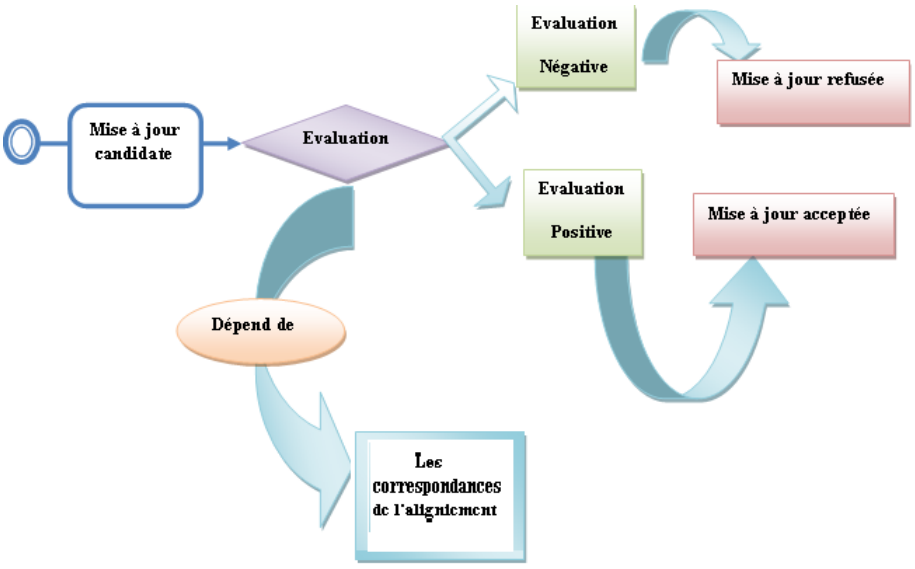

FIGURE 5.13 – *L'approche proposée pour la maintenance "Inter-Base"*

CONCLUSION

Le but de ce chapitre était de proposer :

— Une approche de découverte des liens corrects et erronés, inter et intra base, en utilisant les modèles de liens, a été proposée. Les modèles de liens sont utilisés comme synthèse ou résumé de la base RDF.

  - Pour la découverte des liens intra-base, on se base sur les contraintes d'intégrité qui existent dans l'ontologie de domaine.
  - Pour la découverte des liens inter-base, on utilise l'alignement entre les ontologies.
- Une approche d'alignement entre ontologies est aussi proposée. Pour obtenir les mapping, chaque entité de l'ontologie source est comparée avec toutes les entités de l'ontologie cible, produisant une matrice de similarité, qui contient un vecteur pour chaque paire d'entités. Notre vecteur se compose de trois valeurs de similarité. On doit ensuite associer à chaque valeur de similarité un coefficient de pondération "poids" et à faire la somme des similarités pondérées pour atteindre une nouvelle et unique similarité agrégée. Pour choisir les poids optimaux générant un alignement optimal, nous avons choisi l'optimisation par les Colonies d'Abeilles Artificielles.
- Approche de maintenance des liens RDF.

On a déjà vu dans le chapitre précédant que les chercheurs focalisent leurs travaux "SILK, LIMES, KD2R, C-SAKey, RIMOM, LogMap,...." de découverte et de maintenance des liens entre un ensemble de bases RDF. Par contre, notre approche, présentée dans ce chapitre, donne des solutions de découverte et de maintenance des liens à l'interieur d'une base RDF " intra-Base " et entre un ensemble de base RDF " inter-Base ".

Les outils "KD2R, C-SAKey", ce sont des approches fondées sur des clés. Donc elles exploitent les schémas de nommage communs aux deux sources. Les outils "SILK, LIMES", ce sont des approches fondées sur des similarités. Donc elles comparent les items et les lient si leur similarité dépasse un certain seuil, c'est-à-dire, ils utilisent un *alignement implicite* entre les ontologies. Les outils "Knofuss, RiMOM, LogMap" obligent leurs utilisateurs à définir explicitement l'heuristique de correspondance, c'est-à-dire, elle utilise un *alginement explicite* entre les ontologies.

L'outil LiQuate [Ruckhaus *et al.* 2013] utilise une approche d'évaluation fondée sur les réseaux bayésiens pour identifier les ambiguïtés et les mises en relations incomplètes dans les liens entre les ressources de bases différentes. Le résultat final de cette évaluation suggère à un expert les ressources potentiellement mal liées.

Notre approche de découverte et de maintenance des liens " inter-Base " fait partie de la même catégorie que les autres outils "Knofuss, RiMOM, LogMap", c'est à dire utilise un *alignement initial*.

Ce premier travail va être expérimenté sur des données réelles. Les résultats vont être comparés avec d'autres résultats des autres approches "RiMOM, LogMap,...". Par la suite, dans le chapitre suivant, nous allons présenter encore une application réelle, là o*ù* on va résoudre le probleme de détection et de maintenance des liens.

Expérimentation 6

Sommaire

Dans le chapitre précédent, nous avons présenté notre approche de découverte et de maintenance des liens. Tout d'abord, un premier travail consiste à détecter les liens corrects et les liens erronés dans la même base (liens intra-base), et dans l'ensemble de base (liens inter-base).
Le premier processus (liens intra-base) permet de détecter les liens dans la même base en se basant sur les " modèles de liens " et les " contraintes d'intégrité " de l'ontologie. Le deuxième processus (liens inter-base) permet de détecter les liens entre deux bases en se basant sur une méthode d'alignement d'ontologie qui a été optimisée par les Colonies d'Abeilles Artificielles "ABC". Ensuite, nous avons présenté la deuxième proposition qui inclut aussi un processus automatique pour maintenir les liens quand un changement de données se produit.
Le reste de ce chapitre est organisé comme suit : premièrement, nous présentons dans la section 6.2.1 les outils utilisés pour l'implémentation de notre système. Ensuite, nous présentons dans la section 6.2.2 les résultats expérimentaux de la première solution de découverte et de maintenance des liens " intra-base ". Ensuite, nous présentons les résultats expérimentaux de la deuxième solution de découverte et de maintenance des liens inter-base dans la section 6.2.3. Une brève conclusion conclut ce chapitre.

6.1 Expérimentation et Evaluation

Dans cette section, nous présentons les implémentations des méthodes présentées dans le chapitre précédent. Ces implémentations sont regroupées dans le système DML, Système de Découverte et de Maintenance des Liens, que nous avons conçu.
Tout d'abord, nous présentons les implémentations suivantes :

— La première méthode " Découverte des liens intra-Base ".
— La deuxième méthode " Maintenance des liens intra-Base ".

Ensuite, nous présentons les implémentations suivantes :

— La première méthode " Découverte des liens inter-Base ".
— La deuxième méthode " Maintenance des liens inter-Base ".

6.1.1 Implémentation

Cette partie sera consacrée à la présentation des outils et des environnements utilisés pour l'implémentation de notre Système DML.

6.2.1.1 Les outils utilisés

1. Eclipse oxygen

Nous avons utilisé *Eclipse SDK version 4.7*. Eclipse est un IDE (EDI en français pour Environnement de Développement Intégré). Ce logiciel est disponible gratuitement. Il a été développé par IBM, et il est compatible pour la plupart des systèmes d'exploitation. Il fournit des outils modulaires capables non seulement de faire du développement en Java, mais aussi en d'autres langages et d'autres activités.

2. API Jena

Les données Web, qui ont été mises en cache localement, sont généralement accessibles via soit des requêtes SPARQL, soit une API RDF. Une liste d'API RDF pour différents langages de programmation est disponible dans le cadre des outils de développement du Web sémantique du W3C2 [1]Sweet Tools [2]qui offre une autre série d'outils de développement du Web sémantique.
On trouve, notamment, Jena [3]et Sesame [4]qui sont des API RDF renommées. Dans notre système, l'extraction des ressources à partir de la base RDF et l'ontologie est réalisée par l'intermédiaire de l'API Jena [5]
Cette API permet l'extraction de toutes les données sous forme de triplets RDF (Sujet, Prédicat, Objet). Toutes les informations liées à la base RDF et l'ontologie sont présentées sous forme de triplets RDF. Ces triplets sont exploités pour la découverte et la maintenance des liens.

1. http ://www.w3.org/2001/sw/wiki/Tools .
2. http ://www.mkbergman.com/sweet-tools/. .
3. http ://incubator.apache.org/jena/index.html..
4. http ://www.openrdf.org/doc/sesame/users/ch07.html..
5. http ://jena.sourceforge.net.

6.1.2 Processus de découverte des liens intra-Base

Nous avons choisi d'évaluer DML sur la base RDF Dbpedia Core 2016. [6]
Le contenu de DBPedia [N.Popitsch & B.Haslhofer 2010] est issu du récolte des pages et des mises à jours de Wikipedia. La base DBPedia est accessible par son end point officiel [7], et la mise à jour est annuelle. La mise à jour de DBPedia est faite à partir de la base DBPediaLive [8], miroir de DBPedia, qui est mise à jour directement à partir de Wikipedia. Les ajouts et suppressions des données des pages de Wikipedia sont extraites et appliquées à DBPedia Live quotidiennement, et annuellement une mise àjour générale est extraite de DBPedia Live pour être appliquée à DBPedia.
Notre système DML exécute un processus qui contient plusieurs étapes. Chacune est basée sur un ou plusieurs mécanismes.

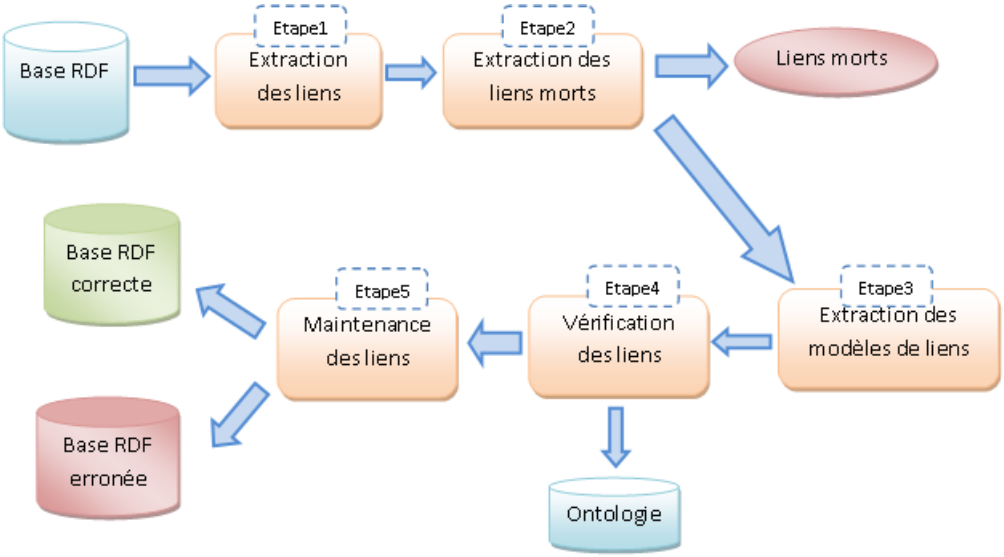

FIGURE 6.1 – *Le système DML " Intra-base "*

6.2.2.1 Etape 1 : Extraction des liens internes " Intra-Base "

Les liens RDF décrivent les relations entre deux ressources, et consistent en trois références d'URI. Les URI, dans le sujet et l'objet du lien, identifient les ressources liées. L'URI du prédicat définit le type de la relation entre les ressources.
Les liens internes " intra-Base " connectent des ressources dans une seule source de données liées. Ainsi, les URI des sujets et des objets sont dans le même espace de noms.
Nous avons choisi "Dbpedia-fr-2016 et Dbpedia-ja-2016" [9]pour évaluer notre système. Les résultats de l'extraction des liens depuis Dbpedia-fr-2016 et Dbpedia-ja-2016 sont présentés dans le Tableau 6.1.

6. http ://downloads.dbpedia.org/2016-04/core/.
7. http ://dbpedia.org/sparql.
8. http ://live.dbpedia.org/sparql.
9. http ://wiki.dbpedia.org/downloads-2016-04.

Base RDF	Nbr triplets	Nbr liens
Dbpedia fr 2016	16840	16036
Dbpedia ja 2016	15832	7896

Tableau 6.1 – *Extraction des liens "Intra-Base"*

6.2.2.2 Etape 2 : Extraction des liens Morts " Filtrage "

Les liens morts sont ceux pointant vers des URI qui ne sont plus entretenues, c'est-à-dire, l'objet n'appartient pas à l'ensemble des espaces des noms définis dans la base.
Nous définissons *NS(URI)*, l'espace de noms d'un URI et *LNS(B)* l'ensemble des espaces des noms des ressources défini dans la base RDF. Par exemple, dans la base Dbpedia-fr-2016, les espaces des noms définis sont présentés dans le Listing 6.1.

Listing 6.1 – *Les espaces des noms définis dans la base Dbpedia fr 2016*

```
http://xmlns.com/foaf/0.1/
http://creativecommons.org/ns#
http://purl.org/vocab/vann/
http://purl.org/dc/terms/
http://purl.org/dc/elements/1.1/
http://www.w3.org/2003/01/geo/wgs84_pos#
http://purl.org/NET/cidoc-crm/core#
http://www.wikidata.org/entity/
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#
http://www.ontologydesignpatterns.org/ont/d0.owl#
http://www.w3.org/2002/07/owl#
http://www.w3.org/2001/XMLSchema#
http://www.w3.org/1999/02/22-rdf-syntax-ns#
http://www.w3.org/2000/01/rdf-schema#
http://www.w3.org/ns/prov#
http://dbpedia.org/ontology/
http://fr.dbpedia.org/resource/
<http://dbpedia.org/property/
```

Un lien mort $< S; P; O >$, entre un sujet et un objet, existe dans la base B si $NS(O)$ n'appartien pas $LNS(B)$ et le triplet $\in B$. C'est-à-dire, un lien mort signifie que l'objet respectif n'est pas trouvé dans l'ensemble des espaces de noms de la base B. Un lien mort extrait à partir de la base Dbpedia-fr-2016 est présentés dans le Listing 6.2.

Listing 6.2 – *Extrait de la base Dbpedia fr 2016*

```
Lien
sujet       http://fr.dbpedia.org/resource/
   Communaute_de_Robinson
predicat    http://dbpedia.org/ontology/affiliation
objet       http://www.eglise-protestante-unie.fr/
```

On remarque dans l'exemple précédent que *NS(objet)* n'appartient pas aux espaces des noms définis dans la base Dbpedia-fr-2016, mais l'espace de nom du sujet et du prédicat appartient aux espaces de noms définis dans la base Dbpedia-fr-2016. Les résultats obtenus après l'analyse de tous les liens de la base Dbpedia-fr-2016 sont présentés dans la Figure 6.2.

FIGURE 6.2 – *Extraction des liens morts*

D'après ces résultats on a le nombre de liens conformes au DNS est égal à *16026*, et le nombre de liens morts est égale à *5*. Donc le pourcentage de liens morts par rapport au nombre total des liens " *16031*" est égal à *0.03*.
Après l'étape de filtrage des liens morts, nous présentons l'étape de l'extraction des modèles de liens.

6.2.2.3 Étape 3 : Extraction des modèles de liens

Pour chaque modèle de liens, nous avons mentionné le nombre de liens identifiés automatiquement selon l'algorithme.1. Le Tableau 6.2 récapitule les résultats obtenus après l'extraction des modèles de liens pour les deux bases *Dbpedia-fr-2016* et *Dbpedia-ja-2016* :

Les Modèles de liens	**Dbpedia fr 2016** *"Nbr liens"*	**Dbpedia ja 2016** *"Nbr liens"*
Modèle de liens 0	328	4998
Modèle de liens 1	4999	653
Modèle de liens 2	1421	288
Modèle de liens 3	3952	20
Modèle de liens 4	609	811
Modèle de liens 5	3240	539
Modèle de liens 6	10	13
Modèle de liens 7	1060	76
Modèle de liens 8	389	3
Modèle de liens 9	18	39
Modèle de liens 10	/	8
...	...	...
Modèle de liens 16	/	2

Tableau 6.2 – *Modèles de liens pour les deux bases Dbpedia fr 2016 et Dbpedia ja 2016*

On remarque que la base *Dbpedia-fr-2016* ne contient que *10* modèles de liens par rapport à la base *Dbpedia-ja-2016* qui contient *17* modèles de liens.
Les résultats de la base *Dbpedia-fr-2016* obtenus par notre système DML sont présentés dans la Figure 6.3.

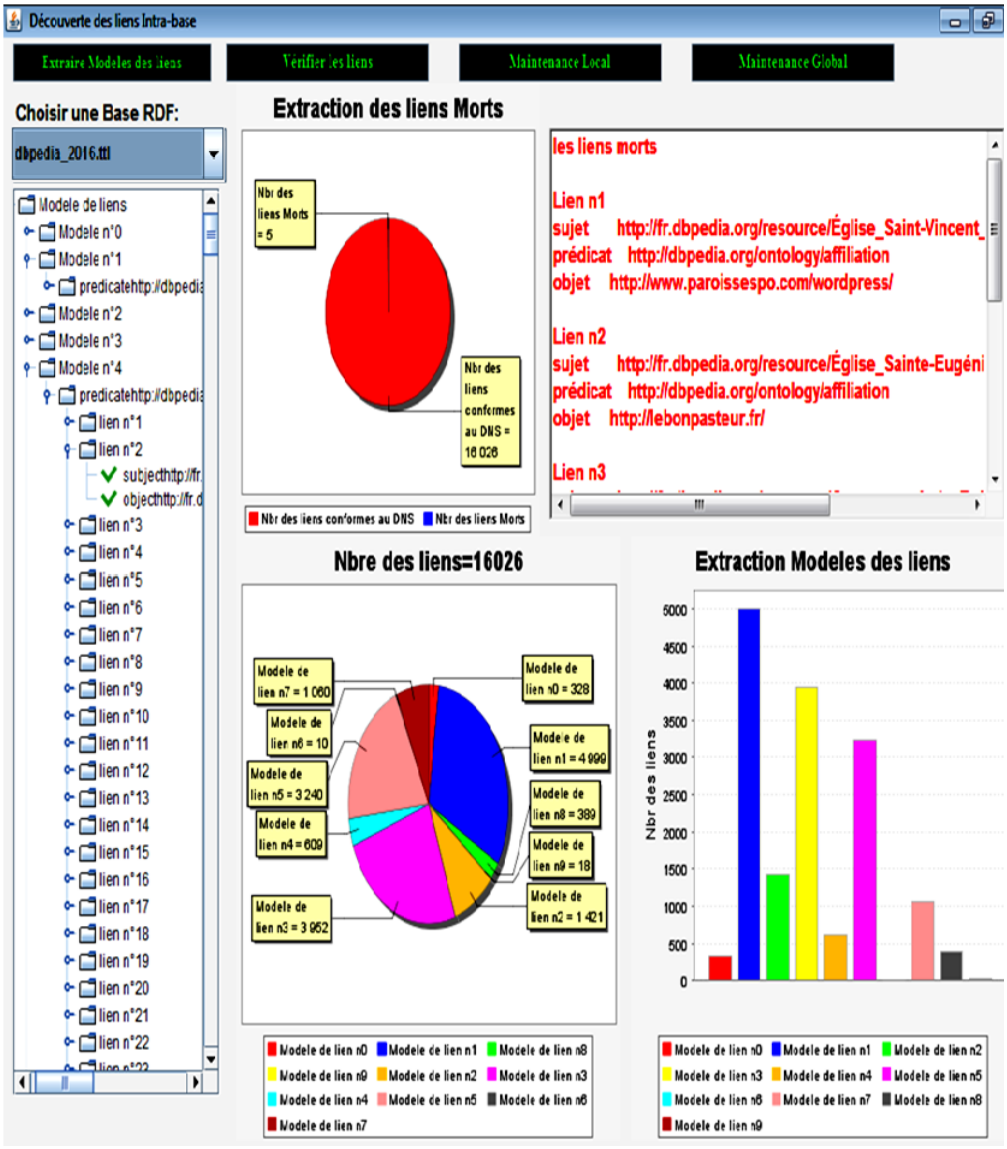

FIGURE 6.3 – *Résultats d'extraction des modèles de liens pour " Dbpedia-fr-2016 "*

Cette Figure représente l'interface graphique de notre logiciel montrant l'arborécence des modèles de liens. La base Dbpedia-fr-2016 contient *10* modèles de liens, et chaque modèle contient un ensemble de liens. Par exemple, le modèle de lien numéro *5* contient *3240* liens qui contiennent une ressource relation (*prédicat*) en commun. D'après la figure d'extraction des modèles de liens pour " Dbpedia-fr-2016 ", on constate que le modèle de liens numéro *1* possède le plus grand nombre de liens par rapport aux autres modèles.

6.2.2.4 Etape4 : Découverte des liens corrects et des liens erronés

Après l'extraction des modèles de liens, on va les vérifier pour découvrir les liens corrects et les liens erronés en utilisant les contraintes de typage de domaine (*Domain*) et co-domaine (*Rang*).

Nous avons utilisé le langage SPARQL pour extraire le *domaine* et le *co-domaine* du prédicat pour chaque modèle de liens.

Le Listing 6.3 décrit la requête d'extraction du *domaine*.

Listing 6.3 – *Requête d'extraction du domaine "Domaine"*

```
#filename: dbpedia 2016.owl
PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema
   #>
PREFIX dcat:    <http://www.w3.org/ns/dcat#>
PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-
   syntax-ns#>
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX owl:     <http://www.w3.org/2002/07/owl#>
PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbponto: <http://dbpedia.org/ontology/>
PREFIX dbpprop: http://dbpedia.org/property/

SELECT ?p ?domain FROM <dbpedia_2016.owl>
WHERE
{?p rdfs:domain ?domain.}
```

Le Listing 6.4 décrit la requête d'extraction du *co-domaine*.

Listing 6.4 – *Requête d'extraction du co-domaine "* Rang *"*

```
#filename: dbpedia_2016.owl
PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema
   #>
PREFIX dcat:    <http://www.w3.org/ns/dcat#>
PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-
   syntax-ns#>
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX owl:     <http://www.w3.org/2002/07/owl#>
PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbponto: <http://dbpedia.org/ontology/>
PREFIX dbpprop: http://dbpedia.org/property/

SELECT?p ?rang FROM <dbpedia_2016.owl>
WHERE
{?p rdfs:rang ?rang.}
```

Le Listing 6.5 décrit la requête d'extraction de Type du sujet et de Type de l'objet à partir de la base dbpedia-fr-2016.

Listing 6.5 – *Requête d'extraction du Type*

```
#filename: dbpedia_fr_2016.ttl
PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema
   #>
PREFIX dcat:    <http://www.w3.org/ns/dcat#>
PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-
   syntax-ns#>
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX owl:     <http://www.w3.org/2002/07/owl#>
PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbponto: <http://dbpedia.org/ontology/>
PREFIX dbpprop: http://dbpedia.org/property/

SELECT?ressource ?Type FROM <dbpedia_fr_2016.ttl >
WHERE
{?ressourcerdf:type ?Type.}
```

Résultats de vérification des modèles de liens

Pour chaque modèle de liens, nous avons mentionné le nombre de liens corrects identifiés automatiquement selon l'algorithme de vérification (voir Algorithme 2). Le Tableau 6.3 récapitule les résultats obtenus après vérification des modèles de liens pour les deux bases Dbpedia-fr-2016 et Dbpedia-ja-2016 :

Les Modèles de liens	**Dbpedia fr 2016 "Nbr liens corrects"**	**Dbpedia ja 2016 "Nbr liens corrects"**
Modèle de liens 0	133	4999
Modèle de liens 1	1847	0
Modèle de liens 2	611	0
Modèle de liens 3	1697	0
Modèle de liens 4	263	0
Modèle de liens 5	681	0
Modèle de liens 6	348	13
Modèle de liens 7	148	0
Modèle de liens 8	7	1
Modèle de liens 9	0	39
Modèle de liens 10	/	8
...	...	...
Modèle de liens 16	/	2

Tableau 6.3 – *Liens corrects pour chaque modèle de liens*

Les résultats obtenus par notre système DML sont présentés dans la Figure 6.4.

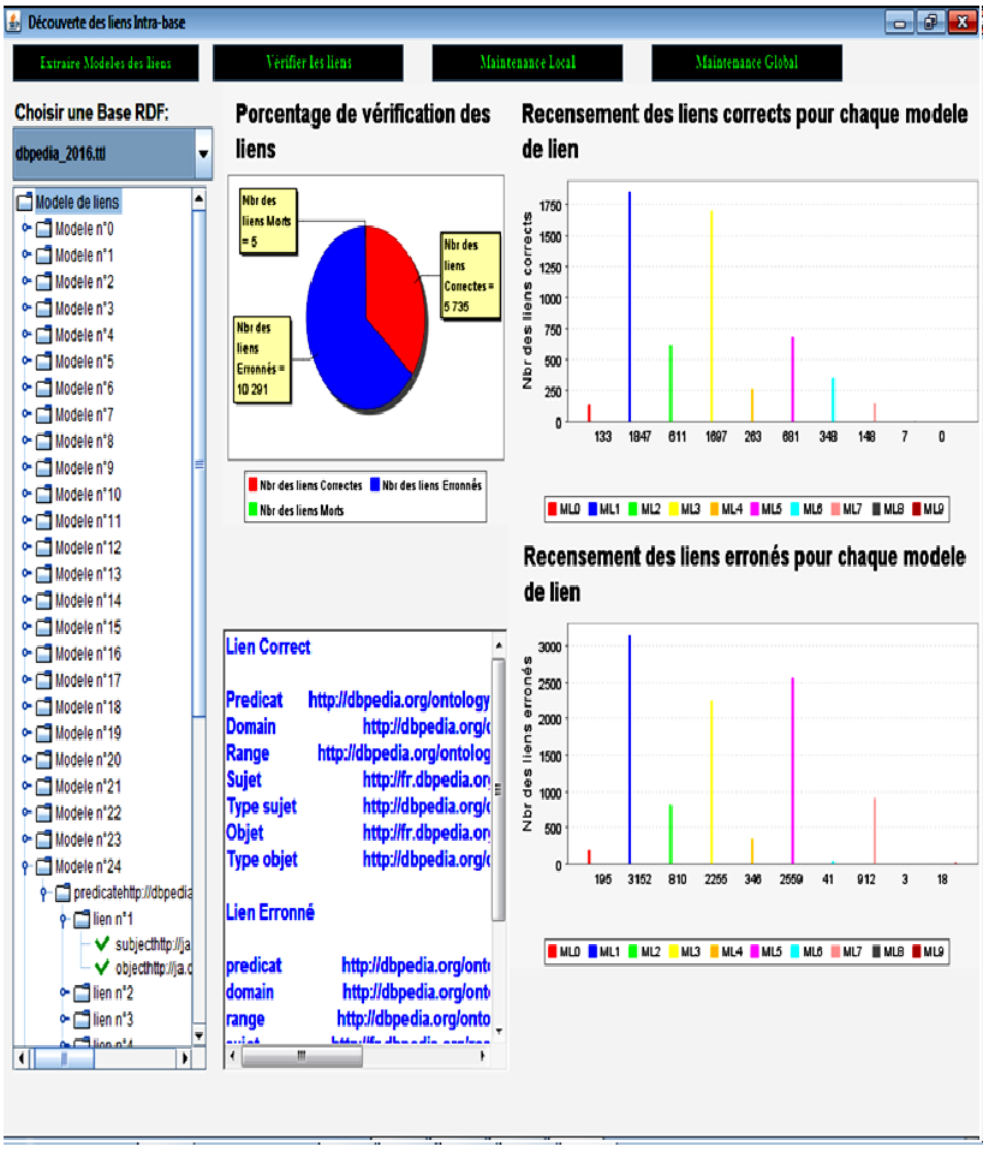

FIGURE 6.4 – *Résultats de vérification des modèles de liens pour " Dbpedia-fr-2016 "*

Les histogrammes montrent le nombre de liens corrects et de liens eronnés pour chaque modèle de liens par exemple, le modèle numéro "*1*" contient "*1847*" liens corrects et "*3152*" liens erronés, par contre le modèle numéro "*6*" contient "*348*" liens corrects et *41* liens erronés. Le cercle illustre le nombre de liens corrects "*5735*", le nombre des liens erronés "*10291*" et le nombre des liens morts "*5*". On voit aussi au dessous, un exemple de lien correct et un autre erroné.
On peut dire que la structure du modèle de liens est plus fiable en application, car elle est plus rapide. Au lieu d'extraire le domaine et le co-domaine de chaque prédicat de chaque lien pour les vérifiés avec les types des ressources, on extrait seulement le domaine et le co-domaine du prédicat du modèle de liens. Donc, on peut dire que notre méthode est plus performante en terms de temps et d'espace mémoire.
D'après "*le lien correct*" de listing 6.6, on constate bien que le domaine du prédicat est égal au type du sujet. Donc la contraite de typage du domaine est vérifiée. On constate aussi que le co-domaine "*Rang*" du prédicat

est égal au type de l'objet. On peut dire alors que la contraite de typage du co-domaine est vérifiée. Après la vérification des deux contraintes, on résulte que le lien est correct.

Listing 6.6 – *Un exemple de lien correct et un autre erroné*

```
Lien correct
Predicat     http://dbpedia.org/ontology/locationCity
Domain           http://dbpedia.org/ontology/
   Organisation
Rang             http://dbpedia.org/ontology/City
Sujet            http://fr.dbpedia.org/resource/
   Theatre_de_la_Taganka
Type sujet       http://dbpedia.org/ontology/
   Organisation
Objet            http://fr.dbpedia.org/resource/Moscou
Type objet       http://dbpedia.org/ontology/City
Lien errone
predicat     http://dbpedia.org/ontology/nationalTeam
Domain       http://dbpedia.org/ontology/Athlete
Rang         http://dbpedia.org/ontology/SportsTeam
sujet        http://fr.dbpedia.org/resource/
   Pauline_Biscarat__3
Type sujet   http://dbpedia.org/ontology/SportsEvent
objet        http://fr.dbpedia.org/resource/
   Rugby_a_sept
Type objet   http://dbpedia.org/ontology/Sport
```

D'après "*le lien erroné*" de listing 6.6, on constate que le domaine du prédicat n'est pas égal au type du sujet. Donc la contraite de typage du domaine n'est pas vérifiée. On constate aussi que le co-domaine " Rang " du prédicat n'est pas égal au type de l'objet. Donc la contraite de typage du co-domaine n'est pas vérifiée. Après la vérification des deux contraintes, on résulte que le lien est erroné.

6.2.2.5 Maintenance

1. Maintenance locale

Après l'étape de vérification des liens, on obtient un ensemble de liens corrects et un ensemble de liens erronés. Notre Système DML crée une base RDF appelée base RDF correcte, et il met l'ensemble des liens corrects dans cette base. Ensuite, il crée une autre base RDF appelée base RDF erronée, et il met l'ensemble des liens erronés dans cette base. Donc le résultat de la maintenance de la base RDF en entrée c'est la base RDF correcte en sortie.

2. Maintenance globale

Pour maintenir la qualité des données liées, il faut évaluer la qualité des modifications des données, et les filtrer selon le résultat de leur évaluation. Lorsqu'une mise à jour ne respecte pas les contraintes de domaine

et co-domaine de l'ontologie, on dit qu'elle est incohérente avec l'ontologie. Donc la modification est refusée par notre système DML. Lorsqu'une mise à jour respecte les contraintes de domaine et co-domaine de l'ontologie, on dit qu'elle est conforme avec l'ontologie. Donc la modification est acceptée. La Figure 6.5 montre un exemple d'évaluation.

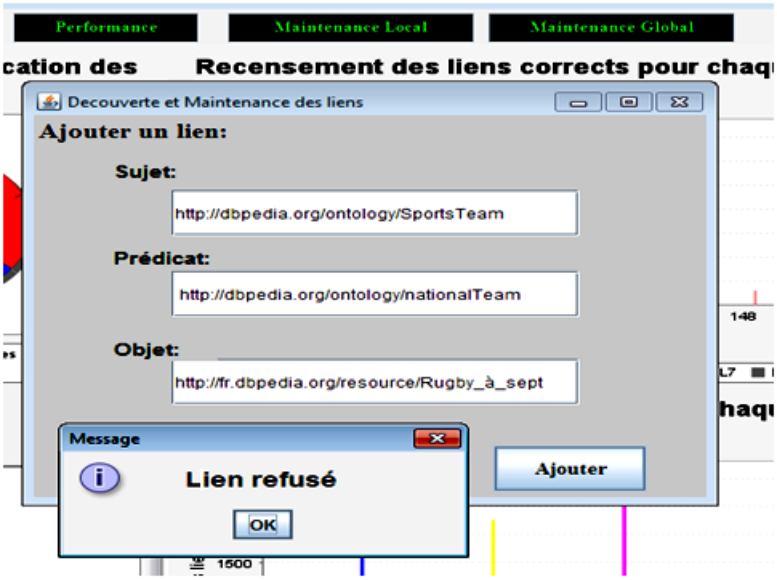

FIGURE 6.5 – *Maintenance globale "Exemple d'évaluation"*

Après l'évaluation, si la modification est acceptée, notre système DML utilise le langage SPARQL pour ajouter le lien accepté à la base. Dans Listing 6.7, nous présentons la requête d'ajout d'un lien dans la base dbpedia-fr-2016.

Listing 6.7 – *Requête d'ajout d'un lien dans la base dbpedia fr 2016*

```
#filename: dbpedia_fr_2016.ttl
PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat:     <http://www.w3.org/ns/dcat#>
PREFIX rdf:      <http://www.w3.org/1999/02/22-rdf-
   syntax-ns#>
PREFIX foaf:     <http://xmlns.com/foaf/0.1/>
PREFIX owl:     <http://www.w3.org/2002/07/owl#>
PREFIX xsd:      <http://www.w3.org/2001/XMLSchema#>
PREFIX dbpedia: <http://fr.dbpedia.org/resource/>
PREFIX dbponto: <http://dbpedia.org/ontology/>
PREFIX dbpprop: http://dbpedia.org/property/

INSERT DATA
{
dbpedia:Th\'{e}\^{a}tre_de_la_Taganka  dbponto:
   locationCity   dbpedia:Moscou.
}
```

6.1.3 Processus de découverte des liens inter-Base

Le système DML " Inter-base " exécute un processus qui contient plusieurs étapes. Chacune est basée sur un ou plusieurs mécanismes.

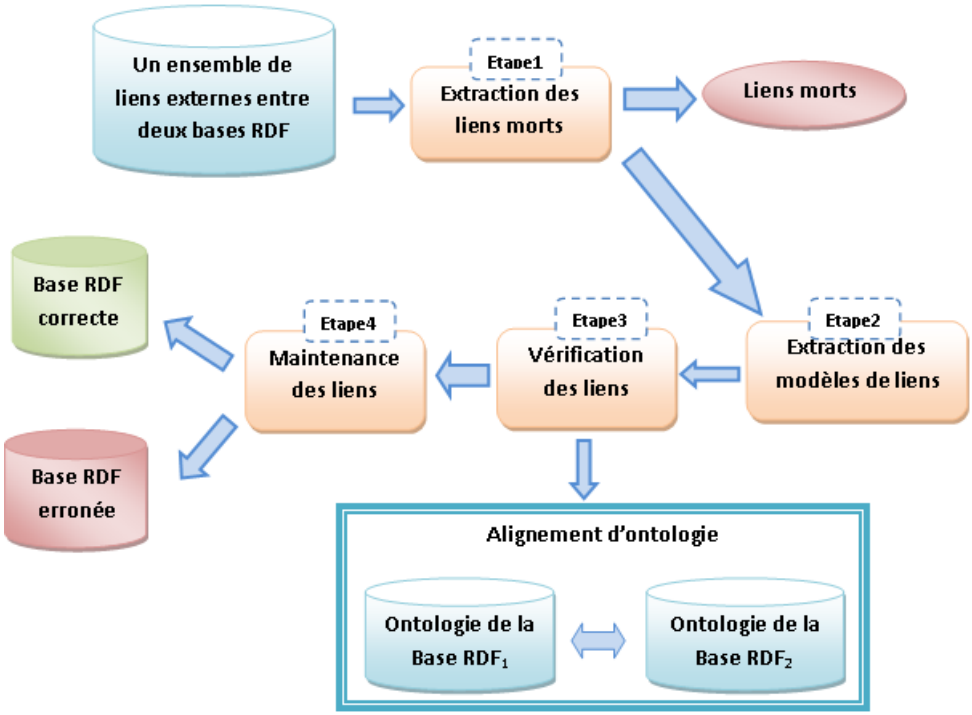

FIGURE 6.6 – *Le système DML " Inter-base "*

Les liens externes connectent des ressources dans des sources de données liées différentes. Les URI des sujets et objets des liens externes sont dans des espaces de noms différents.
Nous avons choisi **Geonames-links-2016** pour évaluer notre système. Dbpedia fournit la base **Geonames-links-2016**, et contient un ensemble de liens "*owl :sameAs*" pointant vers des données sur la même entité au sein de la source de données **Geonames**.
Owl-sameas est une primitives de OWL, et est utilisée pour exprimer le fait que deux URI identifient la même ressource. OWL fournit la propriété *owl :sameAs* pour affirmer que deux individus avec différentes URI sont une seule entité du monde réel. Une fois cette relation établie, les deux individus sont traités comme s'ils étaient les mêmes.

6.2.3.1 Etape 1 : Extraction des liens Morts " Filtrage "

Un lien RDF externe est un triplet RDF dans lequel le sujet est une référence URI dans un espace de noms d'un jeu de données, alors que le prédicat et/ou l'objet sont des références URI pointant vers des espaces de noms d'autres jeux de données.

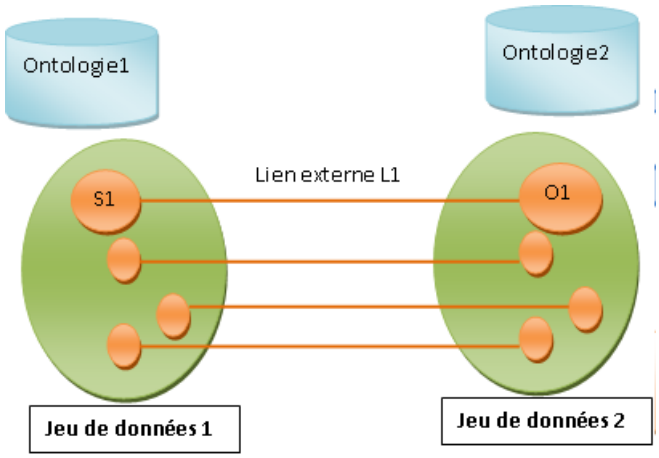

FIGURE 6.7 – *Liens Externe*

Les liens morts sont ceux pointant vers des *URI* qui ne sont plus entretenues, c'est-à-dire, la référence d'URI de l'objet n'appartient pas à l'ensemble des espaces de noms définis par leur jeu de données.
Nous définissons *NS(URI)*, l'espace de noms d'un URI, et *LNS(B)* l'ensemble des espaces des noms d'un jeu de données.
Le sujet S_1 est une référence d'URI. $NS_1(URI)$ est définit dans un espace de noms d'un jeu de données $LNS(B_1)$.
L'objet O_1 est une référence d'URI. $NS_2(URI)$ est définit dans un espace de noms d'un autre jeu de données $LNS(B_1)$.

Les espaces de noms définis par le jeu de données Geonames 2016 sont présentés dans le Listing.6.8.

Un lien mort $< S; P; O >$, entre un sujet et un objet, existe entre deux jeux de données si $NS_2(O)$ n'appartient pas $LNS(B_2)$, et le triplet $\in B_1$. C'est-à-dire, un lien mort signifie que l'objet respectif n'est pas trouvé dans l'ensemble des espaces de noms définis par le jeu de données B_2.

Listing 6.8 – *Les espaces de noms définis par le jeu de données Geonames 2016*

```
http://xmlns.com/foaf/0.1/
http://creativecommons.org/ns#
http://purl.org/vocab/vann/
http://purl.org/dc/terms/
http://purl.org/dc/elements/1.1/
http://www.w3.org/2003/01/geo/wgs84_pos#
http://purl.org/NET/cidoc-crm/core#
http://www.wikidata.org/entity/
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#
http://www.ontologydesignpatterns.org/ont/d0.owl#
http://www.w3.org/2002/07/owl#
http://www.w3.org/2001/XMLSchema#
http://www.w3.org/1999/02/22-rdf-syntax-ns#
http://www.w3.org/2000/01/rdf-schema#
http://www.w3.org/ns/prov#
http://geonames.org/ontology/
http://sws.geonames.org/
```

Après l'analyse de tous les liens de Geonames-links-2016, les résultats obtenus montrent que tous les liens sont conformes au DNS. Donc on a aucun lien mort.

6.2.3.2 Etape 2 : Extraction des modèles de liens

Pour chaque modèle de liens, nous avons mentionné le nombre de liens identifiés automatiquement. Le Tableau 6.4 récapitule les résultats obtenus après l'extraction des modèles de liens pour la base **Geoames-links-2016** :

Les Modèles de liens	**Geoames links 2016 "Nbr liens"**
Modèle de lien 0	535377

Tableau 6.4 – *Modèles de liens pour la base Geoames links 2016.*

On remarque que la base geoames-links-2016 ne contient qu'un seul modèle de liens car on a un seul prédicat " *Owl-sameas* ".

6.2.3.3 Etape 3 : Découverte des liens corrects et des liens erronés

Après l'extraction des modèles de liens, on va les vérifier pour découvrir les liens corrects et les liens erronés en utilisant l'alignement des deux ontologies.

Un alignement entre les ontologies hétérogènes est utilisé pour indiquer au système les correspondances entre les entités des ontologies. Le système fonctionne en suite de façon similaire à un système a une seule ontologie.

pour dire qu'un lien est correct :

— premièrement, il faut vérifier si le lien respecte les correspondances de l'alignement de deux ontologies,

— deuxièment, il faut évaluer l'étiquette (label) du sujet avec l'étiquette de l'objet à l'aide d'une mesure de similarité (jaro Similarity), si le résultat est supérieur à **Seuil = 0.9** alors on dit que le lien est correct sinon le lien est eronné.

Notre méthode se base sur les données de la base *geoames-links-2016* conforme à deux ontologies différentes, Pour vérifier les liens, nous utilisons l'alignement des deux ontologies " *Dbpedia-2016.owl* " et " *geoames-2016.owl* ".

Nous avons développé un système d'alignement d'ontologie "**ABCMap**" fondé sur une méthode d'optimisation qui se base sur les colonies d'Abeilles Artificielles (ABC).

Pour évaluer notre système "**ABCMap**" , nous avons opté pour la campagne d'évaluation de systèmes d'alignements "*OAEI 2012*".

Evaluation de notre système ABCMap

Nous avons considéré la campagne d'évaluation des systèmes d'alignements OAEI. Cette campagne propose, depuis 2005, une méthode pour évaluer les systèmes d'alignement d'ontologies. Plusieurs tâches permettent de tester plusieurs aspects d'un système d'alignement.

Cette campagne propose un certain nombre de catégories, afin d'évaluer les systèmes d'alignement suivant plusieurs critères.

Nous présentons les résultats obtenus de notre système d'alignement d'ontologie **ABCMap** en testant différentes séries (OAEI 2012) avec "un nombre d'itérations égal à *10*, nombre d'abeilles actives égal à *3*, nombre d'abeilles inactive égale à *1* et un seuil égal à *0.7*". On prend les paires d'entités qui ont une similarité supérieure au seuil, et les similarités inférieures au seuil sont rejetées. On a utilisé l'API Wordnet pour le calcul de la similarité linguistique [Meilicke & Stuckenschmidt 2007]. Le tableau (6.5) présente une brève description des tests de référence.

ID	**Brève description**
101-104	Les ontologies alignées sont identiques, ou la première est la restriction OWL-Lite de la seconde
201-210	Les ontologies alignées ont la même structure, mais avec différentes fonctionnalités lexicales et linguistiques
301-304	Les ontologies alignées sont des cas réels

Tableau 6.5 – *Brève description des tests de référence*

Le tableau 6.6 montre les meilleurs résultats obtenus par notre système. la Précision et le Rappel sont des métriques utilisées pour évaluer la qualité de notre système d'alignement d'ontologies.

ID	**Rappel**	**Précision**
101	0,93	0,98
103	0,81	0,90
104	0,86	0,93
201	0,91	0,97
206	0,70	0,92
302	0,61	0,89

Tableau 6.6 – *Les meilleurs résultats obtenus par notre système d'alignement ABCMap.*

Notre approche de découverte et de maintenance des liens " inter-Base " fait partie de la même catégorie des autres outils "Knofuss, RiMOM, LogMap", c'est-à-dire utilise un *alignement initial*.

Nous avons comparé nos résultats avec les deux outils "RiMOM, LogMap" de la campagne OAEI 2012. Nous avons choisi de comparer notre système ABCMap avec "RiMOM, LogMap" pour les raisons suivantes :

— Nôtre approche fait partie de la même catégorie que les autres approches "RiMOM, LogMap".
— "RiMOM, LogMap" sont parmis les systèmes les plus connus de sa catégorie.
— Les outils "RiMOM, LogMap" utilisent aussi un alignement initial.

Le tableau 6.7 montre une comparaison de notre approche avec les

participants à l'OAEI 2012, où les chiffres à l'intérieur sont la moyenne de tous les cas de test.

L'analyse des résultats expérimentaux montre que nos résultats sont meilleurs par rapport à ceux de tous les participants à l'OAEI 2012 en termes de rappel et précision sauf pour l'outil RIMOM, o*ù* les résultats sont légèrement meilleurs que notre approche, mais cet outil ne permet que de générer des liens entre deux ensembles de données conforment à deux ontologies différentes, alors notre approche permet de détecter les liens incohérente avec l'ontologie, c'est-à-dire les liens morts et les liens erronés.

Système	**Rappel**	**précision**
Optima	0,49	0,89
LogMap	0,45	0,73
RIMOM	0,96	1,00
MaasMatch	0,57	0,54
ASE	0,54	0,49
Notre approche	0.80	0,93

Tableau 6.7 – *Comparaison de notre approche avec les participants à l'OAEI 2012*

Résultats de vérification des modèles de liens

Pour chaque modèle de liens, nous avons mentionné le nombre de liens corrects identifiés automatiquement selon l'algorithme de vérification (voir Algorithme 4). Le Tableau 6.8 récapitule les résultats obtenus après vérification des modèles de liens pour la bases *geonames-links-2016* :

Modèles de liens	**Nbr liens corrects**	**Nbr liens erronés**
Modèle de liens N_0	160643	374734

Tableau 6.8 – *Les liens corrects de la base Geonames links 2016*

Les résultats obtenus par notre système DML sont présentés dans la Figure 6.8

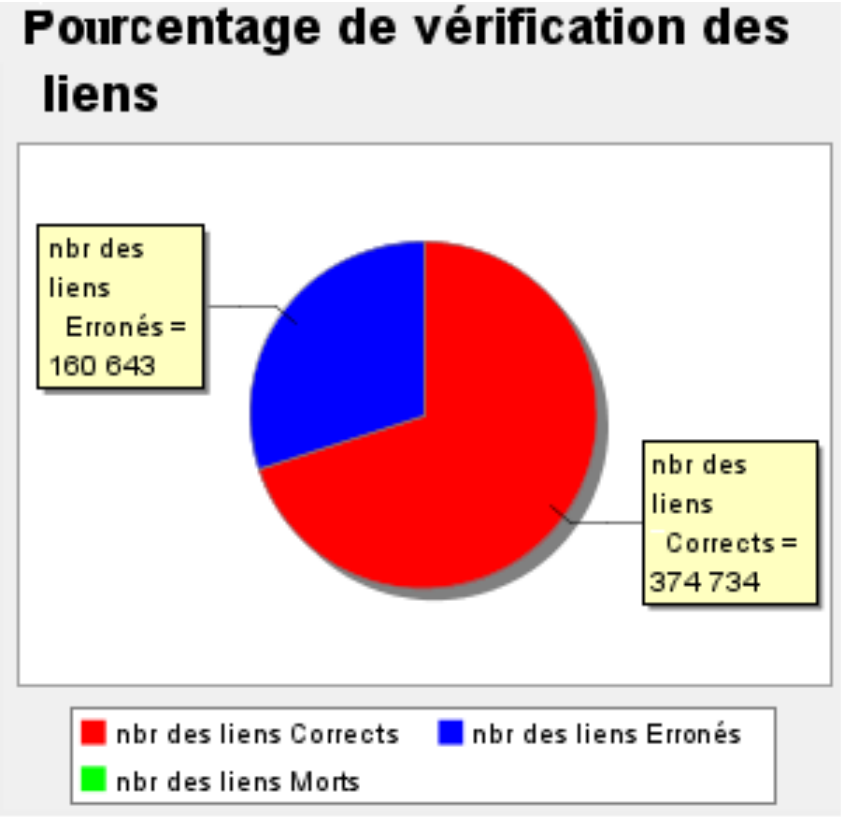

FIGURE 6.8 – *Résultats de vérification des liens pour "**Geonames-links-2016**"*

Cette figure illustre les résultats de vérification des liens (le nombre total de liens égal à *535377*). On constate que le nombre de liens corrects est égal à "*374734*" par contre, nous avons "*160643*" liens erronés. On conclut que la plupart des liens sont incohérents avec les correspondances de l'alignement d'ontologies.

6.2.2.5 Maintenance

Après l'étape de vérification des liens, on obtient un ensemble de liens externes corrects et un ensemble de liens externes erronés. Notre Système DML crée une base RDF appelée base RDF externe correcte, et il met l'ensemble des liens externes corrects dans cette base. Ensuite, il crée une autre base RDF appelée base RDF externe erronée, et il met l'ensemble des liens erronés externes dans cette base. Donc le résultat de la maintenance de la base RDF en entrée est la base RDF correcte en sortie.

CONCLUSION

Dans ce chapitre, nous avons présenté les résultats obtenus par notre système DML. Nous avons testé et évalué notre système selon la campagne d'évaluation OAEI 2012. Nous avons comparé notre approche "intra-base" par rapport à d'autres systèmes. Nous avons choisi de comparer notre système ABCMap avec "RiMOM, LogMap" pour les raisons suivantes : Notre approche de découverte et de maintenance des liens " inter-Base "

fait partie de la m*ê*me catégorie que les autres outils "RiMOM, LogMap", c'est-à-dire, elle utilise un *alignement initial*. L'analyse des résultats expérimantaux montre que nos résultats sont meilleurs par rapport à ceux de tous les participants à l'OAEI 2012 en termes de rappel et précision, sauf pour l'outil RIMOM o*ù* les résultats sont légèrement meilleurs que notre approche, mais cet outil ne permet que de générer des liens entre deux ensembles de données conforment à deux ontologies différentes, par contre notre approche permet de détecter les liens qui sont incohérente avec l'ontologie, c'est-à-dire les liens morts et les liens erronés.

Conclusion générale 7

Sommaire

Le chapitre présent conclut cette thèse. Tout d'abord, nous présentons un résumé de notre approche proposée pour la découverte et la maintenance des liens. Ensuite, nous présentons quelques perspectives pour les futurs travaux.

7.1 Synthèse des contributions (Apport de la thèse)

Le travail présenté dans cette thèse vise donc la maintenance des liens dans les données liées RDF. Pour ce faire nous avons tout d'abord présenté :

- Un Etat de l'art sur :
  - L'alignement des ontologies
  - Maintenance des liens RDF

Nous avons ensuite proposé :

- Une approche de découverte des liens corrects et erronés, inter et intra base, en utilisant les modèles de liens, a été proposée. Les modèles de liens sont utilisés comme synthèse ou résumé de la base RDF.
  - Pour la découverte des liens intra-base, on se base sur les contraintes d'intégrité qui existent dans l'ontologie de domaine.
  - Pour la découverte des liens inter-base, on utilise l'alignement entre les ontologies.
- Une approche d'alignement entre ontologies est aussi proposée. Pour obtenir les mapping, chaque entité de l'ontologie source est comparée avec toutes les entités de l'ontologie cible, produisant une matrice de similarité, qui contient un vecteur pour chaque paire d'entités, notre vecteur se compose de trois valeurs de similarité. On doit ensuite associer à chaque valeur de similarité un coefficient de pondération "poids" et à faire la somme des similarités pondérées pour atteindre une nouvelle et unique similarité agrégée. Pour choisir les poids optimaux générant un alignement optimal, nous avons choisi l'optimisation par les Colonies d'Abeilles Artificielles.
- Approche de maintenance des liens RDF.
- Une étude expérimentale a été faite au niveau des différentes étapes du processus de maintenance des liens. L'étude a montré l'efficacité du système implémenté par rapport aux autres systèmes existants. Nous avons testé et évalué notre Système selon la campagne d'évaluation OAEI 2012. Nous avons comparé notre approche "intra-base" par raport à d'autres systèmes. Nous avons choisi de comparer notre système ABCMap avec "RiMOM, LogMap" pour les raisons suivantes : Notre approche de découverte et de maintenance des liens " inter-Base " fait partie de la même catégorie des autres outils "RiMOM, LogMap", c'est-à-dire, elle utilise un *alginement initial*. L'analyse des résultats expérimantaux montre que nos résultats sont meilleurs par rapport à ceux de tous les participants à l'OAEI 2012 en termes de rappel et précision sauf pour l'outil RIMOM o*ù* les résultats sont légèrement meilleurs que notre approche, mais cet outil ne permet que de générer des liens entre deux ensembles de données conforment à deux ontologies différentes, par contre notre approche permet de détecter les liens qui sont incohérente avec l'ontologie, c'est-à-dire les liens morts et les liens erronés.

Comme déjà vu, les chercheurs focalisent leurs travaux "SILK, LIMES, KD2R, C-SAKey, RIMOM, LogMap,...." de découverte et de maintenance des liens entre un ensemble de bases RDF. Par contre, notre approche donne des solutions de découverte et de maintenance des liens à l'interieur

d'une base RDF "intra-Base" et entre un ensemble de base RDF "inter-Base".

7.2 Perspectives

Le travail présenté dans cette thèse pour la découverte et la maintenance des liens peut être amélioré sur différents niveaux :

1. Considérer d'autres contraintes d'integrité, telles que les contraintes d'unicité et les contraintes de définition afin de :
   (a) Découvrir de nouveaux liens.
   (b) Améliorer la maintenance des liens.
2. Considérer d'autres Matchers "structurelle et sémantique" dans le processus d'alignement d'ontologies.
3. Utiliser d'autres méthodes d'optimisation pour optimiser l'alignement d'ontologi, afin de découvrir de nouvelles correspondances.

Bibliographie

[A.Cyrille *et al.* 2011] A.Cyrille, N.Ngomoand et S.Auer. *LIMES-A Time-Efficient Approach for Large Scale Link Discovery on the Web of Data*. In the Twenty Second international joint conference on Artificial Intelligence, volume 3, pages 2312–2317, 2011.

[A.H.Doan *et al.* 2001] A.H.Doan, P.Domingos et A.Halevy. *Reconciling schemas of disparate data sources : a machinelearning approach*. In 20th International Conference on Management of Data (SIGMOD), pages 509–520, 2001.

[A.H.Doan *et al.* 2004] A.H.Doan, J.Madhavan, P.Domingos et A.Halevy. Ontology matching : a machine learning approach. Springer, 2004.

[A.Miles & S.Bechhofer 2009] A.Miles et S.Bechhofer. *Skos simple knowledge organization system reference*. W3C, pages 24,56, 2009.

[A.Nikolov *et al.* 2007] A.Nikolov, V.Uren et E.Motta. *KnoFuss :A comprehensive architecture for knowledge fusion*. In the 4 th international conference on Knowledge capture, pages 185–186, 2007.

[A.Schultz 2010] C.Bizer A.Schultz. *The R2R Framework : Publishing and discovering mappings on the Web*. In the 1st International Workshop on Consuming Linked Data, volume 25, 2010.

[C.Fellbaum 1998] C.Fellbaum. *WordNet : An electronic lexical database*. MIT Press, 1998.

[C.Shao *et al.* 2014] C.Shao, L.Hu et J.Li. *RiMOM-IM results for OAEI 2014*. In the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), 2014.

[D.Berrueta & J.Phipps 2008] D.Berrueta et J.Phipps. *Best practice recipes for publishing RDF vocabularies*. W3C note, pages 24, 58, 63, 83, 2008.

[D.Brickley & R.V.Guha 2004] D.Brickley et R.V.Guha. *RDF Vocabulary Description Language 1.0 : RDF Schema*. W3C Recommendation, pages 17, 24, 56, 2004.

[D.Kensche *et al.* 2007] D.Kensche, C.Quix, M.A.Chatti et M.Jarke. *GeRoMe : a Generic Role-based Metamodel for model management*. J. Data Semant, vol. 3, pages 82–117, 2007.

[D.L.McGuinness & Harmelen 2004] D.L.McGuinness et F.van Harmelen. *OWL Web Ontology Language Overview*. W3C Recommendation, pages 17, 24, 56, 2004.

[D.Raggett & I.Jacobs 1999] D.Raggett et A.Le Hors I.Jacobs. *HTML 4.01 specification*. W3C Recommendation, page 7, 1999.

[D.Symeonidou *et al.* 2011] D.Symeonidou, N.Pernelle et F.Saïs. *KD2R : A Key Discovery Method for Semantic Reference Reconciliation*. In OTM Workshops, pages 392–401, 2011.

[D.Symeonidou *et al.* 2015] D.Symeonidou, N.Pernelle et F.Saïs. *C-SAKey : une approche de découverte de clés conditionnelles dans des données RDF*. IC, 2015.

[D.Wood *et al.* 2014] D.Wood, M.Zaidman, L.Ruth, M.Hausenblas et T.Berners-Lee. Linked data structured data on the web. Manning Publications Co, 2014.

[E.Jimenez-Ruiz & B.C.Grau 2011] E.Jimenez-Ruiz et B.C.Grau. *LogMap : logic-based and scalable ontology matching*. In 10th International Semantic Web Conference (ISWC), volume 7031, pages 273–288, 2011.

[F.Ardjani *et al.* 2015] F.Ardjani, D.Bouchiha et M.Malk. *Ontology-Alignment Techniques : Survey and Analysis*. IJMECS, vol. 7, no. 11, pages 67–78, 2015.

[F.Ardjani *et al.* 2017] F.Ardjani, D.Bouchiha et M.Malk. *An Approach for Discovering and Maintaining Links in RDF Linked Data*. IJMECS, vol. 9, no. 3, pages 56–63, 2017.

[F.Duchateau *et al.* 2009] F.Duchateau, R.Coletta, Z.Bellahsene et R.Miller. *(not) Yet Another Matcher*. In 18th ACM Conference on Information and Knowledge Management (CIKM), pages 1537–1540, 2009.

[F.Hamdi *et al.* 2010] F.Hamdi, C.Reynaud et B.Safar. *Pattern-based mapping refinement*. In 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW), volume 6317, pages 1–15, 2010.

[F.Manola & E.Miller 2004] F.Manola et E.Miller. *RDF Primer*. W3C, page 15, 2004.

[G.J.lyne & J.Carroll 2004] G.J.lyne et J.Carroll. Resource description framework (rdf) : Concepts and abstract syntax w3c recommendation. 2004.

[H.Do & E.Rahm 2002] H.Do et E.Rahm. *COMA-a system for flexible combination of schema matching approaches*. In 28th International Conference on Very Large Data Bases (VLDB), pages 610–621, 2002.

[H.Halpin *et al.* 2010] H.Halpin, P.Hayes, J.McCusker, D.Mcguinness et H.Thompson. *When owl :sameas isn't the same : An analysis of identity in linked data*. In 9th International Semantic Web Conference., 2010.

[J.Euzenat & P.Shvaiko 2013] J.Euzenat et P.Shvaiko. Ontology matching. Springer, 2013.

[J.Madhavan *et al.* 2007] J.Madhavan, J.R.Shawn, S.Cohen, X.Dong, D.Ko, C.Yu et A.Halevy. *Web-scale data integration : You can only afford to pay as you go*. In Proceedings of the Conference on Innovative Data Systems Research, volume 25 of *107*, 2007.

[J.Tang *et al.* 2004] J.Tang, B.Liang, J.Li et K.Wang. *Risk Minimization Based Ontology Mapping*. In Content Computing, editeur, Lecture Notes in Computer Science Springer, volume 3309, pages 469–480, 2004.

[J.Volz *et al.* 2009] J.Volz, C.Bizer, M.Gaedke et G.Kobilarov G. *Discovering and Maintaining Links on the Web of Data*. In Heidelberg. Springer-Verlag, editeur, the 8 th International Semantic Web Conference,ISWC 09, 2009.

[Karaboga & Basturk 2007] Dervis Karaboga et Bahriye Basturk. *Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems*. Foundations of fuzzy logic and soft computing, pages 789–798, 2007.

[K.Chang *et al.* 2005] K.Chang, B.He et Z.Zhang. *Toward large scale integration : building a metaquerier over databases on the web*. In 2nd Biennial Conference on Innovative Data Systems Research (CIDR), pages 44–55, 2005.

[K.lexander *et al.* 2009] K.lexander, R.Cyganiak, M.Hausenblas et J.Zhao. *Describing linked datasets*. In Workshop on Linked Data on the Web, 48, 2009.

[K.Saleem *et al.* 2008] K.Saleem, Z.Bellahsene et E.Hunt. *PORSCHE : Performance ORiented SCHEma mediation*. Inf. Sci, vol. 33, no. 7-8, pages 637–657, 2008.

[K.Todorov & C.Hudelot 2010] N.James K.Todorov et C.Hudelot. *Combining visual and textual modalities for multimedia ontology matching*. In 5th International Conference on Semantic and Digital Media Technologies (SAMT), volume 6725, pages 95–110, 2010.

[L.Ivantysynova & T.Scheffer 2008] S.Jaroszewicz L.Ivantysynova et T.Scheffer. *Schema matching on streams with accuracy guarantees*. Intell. Data Anal, vol. 12, no. 3, pages 253–270, 2008.

[Meilicke & Stuckenschmidt 2007] Christian Meilicke et Heiner Stuckenschmidt. *Analyzing mapping extraction approaches*. In Proceedings of the 2nd International Conference on Ontology Matching-Volume 304, pages 25–36. CEUR-WS. org, 2007.

[Mezura-Montes *et al.* 2010] Efrén Mezura-Montes, Mauricio Damián-Araoz et Omar Cetina-Domingez. *Smart flight and dynamic tolerances in the artificial bee colony for constrained optimization*. In Evolutionary Computation (CEC), 2010 IEEE Congress on, pages 1–8. IEEE, 2010.

[M.J.Franklin *et al.* 2005] M.J.Franklin, A.Halevy et D.Maier. *From databases to dataspaces : A new abstraction for information management*. SIGMOD Record, vol. 34, no. 4, pages 27–23, 2005.

[M.Nagy & M.Vargas-Vera. 2010] M.Nagy et M.Vargas-Vera. *Towards an automatic semantic data integration : multi-agent framework approach*. Semantic Web, pages 107–134, 2010.

[M.Sabou *et al.* 2008] M.Sabou, M.d'Aquin et E.Motta. *Exploring the semantic web as background knowledge for ontology matching*. J. Data Semant. XI, pages 156–190, 2008.

[M.S.Hanif & M.Aono 2009] M.S.Hanif et M.Aono. *An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size*. J. Web Semant, vol. 7, no. 4, pages 344–356, 2009.

[N.Mendelsohn 2009] N.Mendelsohn. *The self-describing web tag finding*. vol. 106, no. 29, page 24, 2009.

[N.Popitsch & B.Haslhofer 2010] N.Popitsch et B.Haslhofer. *DSNotify : handling broken links in the web of data*. In Proceedings of the 19th international conference on World wide web, pages 761–770. ACM, 2010.

[P.Biron & A.Malhotra 2004] P.Biron et A.Malhotra. *XML schema part 2 : Datatypes*. W3C Recommendation, page 16, 2004.

[P.Bouquet *et al.* 2006] P.Bouquet, L.Serafini, S.Zanobini et S.Sceffer. *Bootstrapping semantics on the web : meaning elicitation from schemas*. In 15th International World Wide Web Conference (WWW), Edinburgh, pages 505–512, 2006.

[P.Jain *et al.* 2010] P.Jain, P.Hitzler, A.Sheth, K.Verma et P.Yeh. *Ontology alignment for linked open data*. In 9th International Semantic Web Conference (ISWC), volume 6496, pages 401–416, 2010.

[P.Lambrix & H.Tan 2006] P.Lambrix et H.Tan. *SAMBO-a system for aligning and merging biomedical ontologies*. J. Web Semant, vol. 4, no. 1, pages 196–206, 2006.

[P.Niko & H.Bernhard 2010] P.Niko et H.Bernhard. *Dsnotify : Handling broken links in the web of data*. In the 19th International World Wide Web Conference, ACM, 2010.

[P.Patel-Schneider & I.Horrocks 2004] P.Patel-Schneider et I.Horrocks. *OWL Web Ontology Language Semantics and Abstract Syntax*. W3C Recommendation, page 23, 2004.

[R.Cyganiak *et al.* 2008] R.Cyganiak, R.Delbru, H.Stenzhorn, G.Tummarello et S.Decker. *Semantic sitemaps : Efficient and flexible access to datasets on the semantic web*. In the 5th European Semantic Web Conference, 2008.

[R.Dhamankar *et al.* 2004] R.Dhamankar, Y.Lee, A.Doan et A.Halevy. *iMAP : discovering complex semantic matches between database schemas*. In 23rd International Conference on Management of Data (SIGMOD), pages 383–394, 2004.

[R.Fielding 1999] R.Fielding. *Hypertext transfer protocol*. request for comments, 1999.

[Rijsbergen 1975] C.J.Van Rijsbergen. *Information Retrieval*. Springer, 1975.

[Ruckhaus *et al.* 2013] Edna Ruckhaus, Oriana Baldizán et María-Esther Vidal. *Analyzing linked data quality with liquate*. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pages 629–638. Springer, 2013.

[S.Zhang & O.Bodenreider 2007] S.Zhang et O.Bodenreider. *Experience in aligning anatomical ontologies*. Int. J. Semantic Web Inf, vol. 3, no. 2, pages 1–26., 2007.

[T.Berners-Lee & L.Kagal 2008] T.Berners-Lee et L.Kagal. *The fractal nature of the semantic web*. AI Magazine, vol. 29, no. 3, pages 62,101,107, 2008.

[T.Berners-Lee *et al.* 1998] T.Berners-Lee, R.Fielding et L.Masinter. *Uniform Resource Identifiers (URI) : Generic Syntax*. RFC 2396, page 7, 1998.

[T.Berners-Lee *et al.* 2001] T.Berners-Lee, J.Hendler et O.Lassilia. *The semantic web*. Scientifc American, vol. 284, no. 5, pages 34–44, 2001.

[T.Berners-Lee 1989] T.Berners-Lee. *Information management : A proposal*. Rapport technique, 1989.

[T.Berners-Lee 2007] T.Berners-Lee. Giant global graph. 2007.

[T.Heath & C.Bizer 2010] T.Heath et C.Bizer. Linked data : Evolving the web into a global data space. Synthesis Lectures on the Semantic Web : Theory and Technology, morgan et claypool édition, 2010.

[T.Heath *et al.* 2009] T.Heath, C.Bizer et T.Berners-Lee. *Linked data the story so far*. International Journal on Semantic Web and Information Systems, vol. 5, no. 3, pages 1–22, 2009.

[V.Mascardi *et al.* 2010] V.Mascardi, A.Locoro et P.Rosso. *Automatic ontology matching via upper ontologies : a systematic evaluation*. IEEE Trans. Knowl. Data Eng, vol. 22, no. 5, pages 609–623, 2010.

[W.Djeddi & M.T.khadir 2014] W.Djeddi et M. T M.T.khadir. *XMap++ : Results for OAEI 2014*. In the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), 2014.

[W.Hu *et al.* 2008] W.Hu, Y.Qu et G.Cheng. *Matching large ontologies : a divide-and-conquer approach. Data Knowl*. Eng, vol. 67, no. 1, pages 140–160, 2008.

[W.Su *et al.* 2006] W.Su, J.Wang et F.Lochovsky. *Holistic schema matching for web query interfaces*. In 10th Conference on Extending Database Technology (EDBT), volume 3896, pages 77–94, 2006.

[W.Winkler 1999] W.Winkler. *The state record linkage and current research problems*. Internal Revenue Service Publication, 1999.

Titre Evolution des Données Liées :
Maintenance des liens

Résumé Le résumé en français (≈ 1000 caractères)

Mots-clés Les mots-clés en français

Title Le titre en anglais

Abstract Le résumé en anglais (≈ 1000 caractères)

Keywords Les mots-clés en anglais