

وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR-ANNABA



جامعة باجي مختار – عنابة

Faculté des Sciences de L'Ingéniorat Année 2019-2020

Département d'Informatique

**THESE**

Présentée en vue de l'obtention  
du diplôme de Doctorat en Sciences

***Alignement D'ontologies Hétérogènes Basé  
Sur Les Raisonneurs***

Spécialité  
Informatique

par  
**Saida Gherbi**

Devant le jury

Pr Seridi Hassina	Présidente	Professeur	Univ. Annaba
Pr Mohamed tarek Khadir	Rapporteur	Professeur	Univ. Annaba
Pr BouFaïda Zizette	Examineur	Professeur	Univ. Constantine 2
Dr Bouramoul Abdelkrim	Examineur	MCA	Univ. Constantine 2
Pr Kholadi Mohamed	Examineur	Professeur	Univ. El oued



## *Remerciements*

Je remercie d'abord Dieu le tout puissant pour toute la grâce qu'il m'a accordé jusqu'à ce jour.

Je tiens à remercier vivement mon directeur de thèse, Pr. Med Tarek KHADIR, pour son encadrement, sa patience, ainsi que ses conseils, son aide précieuse et son soutien. Je le remercie de son temps qu'il a pris pour échanger et corriger mes travaux de recherches.

Je tiens aussi à remercier madame belleili Habiba, pour avoir accepté de diriger mes travaux de recherches pendant les trois premières années d'inscriptions, et ses conseils.

Mes remerciements vont à Pr Hassina Seridi, qui a accepté de présider le jury de ma soutenance ainsi qu'à tous les membres de jury: Pr. BouFaïda Zizette, Mr Bouramoul Abdelkrim et Pr. Kholadi Mohamed

J'adresse aussi mes sincères remerciements à toute ma famille, pour leur générosité et leur soutien moral. Ils m'ont toujours encouragé et poussé à m'accrocher dans les moments difficiles de cette thèse.

## *Dédicace*

*Je dédie ce travail A toute ma famille et tous mes amis*

# Résumé

La construction d'ontologies se base sur *la réutilisation* d'ontologies déjà existantes, car la conception d'ontologies à partir de zéro (from scratch) est un processus long, couteux et très laborieux, parallèlement, elle accentue le phénomène de l'hétérogénéité des ontologies. Dans ce contexte, l'alignement des ontologies est la solution pour réaliser plusieurs applications telles que l'intégration et le rapprochement de ces différentes structures.

Cette thèse présente une approche d'appariement d'ontologies appelée ONTMAT1, vise à fournir automatiquement des relations de correspondance entre les deux ontologies hétérogènes à travers divers calculs de similarité entre leurs entités en utilisant des règles d'inférence basées sur la logique de description. Tout d'abord, le problème de recherche des alignements les plus pertinents, cohérents et significatifs est abordé. Ensuite, une stratégie qui garantit la flexibilité et l'évolutivité de l'application grâce à une combinaison de différents appariements est suggérée. De plus, un raisonneur et un dictionnaire WordNet ont été utilisés pour résoudre les problèmes de correspondance habituels rencontrés avec les synonymes. Des exemples illustratifs sont ensuite présentés pour expliquer l'approche développée.

L'approche a été simulée et testée en ligne dans le cadre de la tâche de conférence de Ontology Alignment Evaluation Initiative (OAEI) 2019. Les résultats obtenus sont détaillés ici avec des remarques sur les résultats.

**Mots-clés:** appariement d'ontologies, WordNet, OWL, stratégies d'appariement, OAEI.

# Abstract

Ontology construction may be established on the reutilization of existing ontologies, as creating ontologies from scratch is a long, costly and laborious process. Moreover, it accentuates the ontology heterogeneity. In this context, the ontology matching is the key to realizing many applications as: integration and reconciliation of these different structures.

This thesis presents an ontology matching approach called ONTMAT1, which aims to automatically provide correspondence relations among both fundamentally heterogeneous ontologies through diverse similarity calculations between their entities using Description Logic based inference rules. First, the problem of searching for the most pertinent, coherent and significant alignments is addressed. Next, a strategy that ensures flexibility and scalability of the application through a combination of different matchers is suggested. Moreover, a reasoner and WordNet dictionary have been used to resolve usual matching problems encountered with synonyms among other things. Illustrative examples are then presented to support the developed approach.

The approach has been simulated and tested online under the Ontology Alignment Evaluation Initiative (OAEI) 2019 conference task. The obtained results are detailed here with remarks about the findings.

**Keywords:** ontology matching, WordNet, OWL, matching strategies, OAEI.

# ملخص

يتم إنشاء انطولوجيا جديدة على إعادة استخدام الأنطولوجيات الموجودة ، لأن إنشاء الأنطولوجيا من الصفر يعتبر عملية طويلة ومكلفة ومرهقة. علاوة على ذلك ، يبرز عدم التجانس بين الأنطولوجيات. في هذا السياق ، فإن مطابقة الأنطولوجيا هي المفتاح لتحقيق العديد من التطبيقات مثل: التكامل والتوفيق بين هذه الهياكل المختلفة.

تقدم هذه الأطروحة نهج مطابقة الأنطولوجيات الذي يسمى ONTMAT1 ، والذي يهدف إلى توفير علاقات المطابقة الأتوماتيكية بين كل الأنطولوجيات الغير متجانسة بشكل أساسي من خلال حسابات التشابه المتنوعة بين كياناتهم باستخدام قواعد الاستدلال على أساس الوصف المنطقي. أولاً ، يتم بعد ذلك تناول مشكلة البحث عن المحاذاة الأكثر صلة وتماسكاً وأهمية. ، يتم اقتراح إستراتيجية تضمن المرونة وقابلية التطوير للتطبيق من خلال مجموعة من المطابقات المختلفة. علاوة على ذلك ، تم استخدام قاموس منطقي و WordNet لحل مشاكل المطابقة المعتادة كالمترادفات و. يتم بعد ذلك تقديم أمثلة توضيحية لدعم النهج المطور.

تمت محاكاة هذا النهج واختباره عبر الإنترنت في إطار مؤتمر Ontology Alignment Evaluation Initiative 2019 (OAEI). النتائج التي تم الحصول عليها يتم عرضها بصفة مفصلة هنا مع ملاحظات حول النتائج.

**كلمات المفتاح:** مطابقة الأنطولوجيا ، WordNet ، OWL ، استراتيجيات المطابقة ، OAEI.

# Table des matières

Résumé	5
Abstract	6
ملخص	7
Table des matière	8
Table des figures	12
Liste des tableaux	13
<b>Introduction Générale</b>	<b>14</b>
1. Contexte de travail	15
2. Problématiques et objectifs de la thèse	16
3. Principales Contributions de la thèse	18
4. Organisation de la thèse	19
5. Publications	21
<b>Partie I : Etat de l'art</b>	<b>23</b>
<b>Chapitre I : Ontologie et la logique de description</b>	<b>24</b>
I.1 Le Web sémantique	25
I.2. Ontologie	25
I.2.1. Définition	25
I.2.2. Composants d'une ontologie	26
I.2.3. Principes de construction d'une ontologie	27
I.2.4. Langages d'ontologies	28
I.2.5. Langage de représentation des ontologies OWL	29
I.3. Logique de description	31
I.3.1. Sémantique de la logique de description	31
I.3.2. Syntaxe de la logique de description	31
I.3.2.1. Niveau terminologique (TBox)	32
I.3.2.2. Niveau factuel (ABox)	32
I.3.3. Inférence	32
I.3.3.1. Inférence au niveau terminologique	33
I.3.3.2. Inférence au niveau factuel	33
I.3.4. Raisonnement	33

I.3.4.1. Raisonnement basé sur la subsomption structurelle	34
I.3.4.2. Raisonnement basé sur l'algorithme de tableaux	35
I.4. Moteurs d'inférence	35
I.4.1. Racer	36
I.4.2. Librairie Pellet	37
I.5. Conclusion	37
<b>Chapitre II : Problèmes d'appariement et les travaux connexes</b>	<b>39</b>
II.1. Terminologies de la mise en correspondance d'ontologies	41
II.2. Problèmes de l'alignement	42
II.1.2. Niveaux d'hétérogénéités entre les ontologies	42
II.2.2. Modes de l'interopérabilité sémantique	45
II.2.2.1. Mode intégré	46
II.2.2.2 Mode Fédéré	46
II.2.2.3 Mode Unifié	47
II.3. Processus d'appariement (Matching)	48
II.3.1 Description du processus d'alignement des ontologies	48
II.3.2. Techniques d'appariement	49
II.3.2.1. Similarité	49
II.3.2.2. Méthodes terminologiques	52
II.3.2.2.1. Approche syntaxique	52
II.3.2.2.2. Approche linguistique	53
II.3.2.3. Méthodes structurelles	55
II.3.2.4. Méthodes extensionnelles	57
II.3.2.5. Méthodes sémantiques	58
II.3.2.6. Méthodes probabilistes	60
II.4. Stratégies d'appariement	62
II.4.1. Partitionnement des ontologies et la recherche d'un espace d'élagage	63
II.4.1.1. Partitionnement des ontologies	63
II.4.1.2. Recherche d'un espace d'élagage	64
II.4.2. Agrégation de l'alignement	64
II.4.2.1. Pondération	65
II.4.2.2. Election	69

II.4.2.3. Argumentation	70
II.4.3. Méthodes d'apprentissage	70
II.5. Etude et comparaison des approches d'appariement	71
II.5.1. Etude des approches d'appariement participées à l'OAEI	72
II.5.2. Comparaison des approches d'appariement participées à l'OAEI	79
II.5.2.1. Comparaison des caractéristiques externes	80
II.5.2.2. Comparaison des techniques de bases	81
II.5.2.3. Comparaison des stratégies de matching	86
II.6. Conclusion	87
<b>Partie II : Contribution</b>	<b>89</b>
<b>CHAPITRE III : Inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe</b>	<b>90</b>
III.1. Modèle d'Ontologie de Karlsruhe	92
III.2. Matching des ontologies	92
III.3. Phase 1 de l'algorithme : Alignement des concepts à partir d'instances	94
III.4. Phase 2 de l'algorithme : Alignement des relations binaires	97
III.4.1. Restriction	97
III.4.2. Matching des relations binaires	98
III.5. Phase 3 de l'algorithme : Appariement à l'aide de raisonneur et WordNet	101
III.5.1. Matching des concepts de la base At	101
III.5.2. Matching des concepts en utilisant un raisonneur et WordNet	102
III.6. Evaluation et expérimentation	106
III.6.1. Implémentation et paramétrage	107
III.6.2. Evaluation de l'alignement de référence de crisp	108
III.7. Conclusion	108
<b>Chapitre IV : ONTMAT1: Matching d'ontologies en utilisant un raisonneur et une propriété de restriction</b>	<b>109</b>
IV.1. Architecture d'ONTMAT1	111
IV.2. Phase 1: Prétraitement des ontologies	113

IV.3. Phase 2: Affectation de poids à la restriction de propriété	115
IV.4. Phase 3: Similarité structurelle entre les concepts	117
IV.5. Phase 4: Agrégation des similarités	119
IV.6. Expérimentation et évaluation	120
IV.6.1. Implémentation et paramétrage	121
IV.6.2. Données et critères d'évaluation	122
IV.6.2.1. Evaluation de l'alignement de référence de crisp	123
IV.6.2.1. Résultats d'estimations basée sur la version incertaine d'alignement de référence	127
IV.7. Conclusion	129
<b>Conclusion générale et perspectives</b>	<b>131</b>
 <b>Annexes</b>	
<b>Annexe A</b>	135
A.1 Eclipse	136
<b>Annexe B</b>	137
B.1. Ontologies de la piste de conference	138
B.2. Résultats d'alignement de ONTMAT1	138
<b>Bibliographie</b>	<b>140</b>

# Table des figures

Figure II.1. Dimensions de l'hétérogénéité au niveau conceptuel	44
Figure II.2. Exemple de relations sémantiques	45
Figure II.3. Mode intégré [Elbyed M., 2009]	46
Figure II.4. Mode fédéré	47
Figure II.5. Mode Unifié	48
Figure II.6. Techniques de matching	51
Figure II.7. Exemple de graphe d'un réseau bayésien	61
Figure III.1. Ontologie O1	93
Figure III.2. Ontologie O2	93
Figure III.3. Application de la phase 1 sur O <sub>1</sub> et O <sub>2</sub>	96
Figure III.4 Application de la phase 2 sur O <sub>1</sub> et O <sub>2</sub>	100
Figure III.5 Application de matching des concepts de la base At sur O <sub>1</sub> et O <sub>2</sub>	101
Figure III.6. Application de matching des concepts en utilisant un raisonneur et WordNet sur O <sub>1</sub> et O <sub>2</sub>	105
Figure III.7. Interface de ONTMAT	107
Figure IV.1. Architecture de ONTMAT1	113
Figure IV.2. Eclipse Java EE IDE de Développeurs de Web	121
Figure IV.3 Instantané de ONTMAT1	122
Figure IV.4. Résultats d'évaluation établis sur la version incertaine de l'alignement de référence	130

## Liste des tableaux

Tableau II.1 Variantes du terme « enzyme activity»	53
Tableau II.2: Caractéristiques externes des principaux systèmes de matching participés dans les compagnies d'OAEI en 2017 et 2019	80
Tableau II.3. Comparaison selon les techniques de base optées par des systèmes de matching participés dans les compagnies d'OAEI en 2017 et 2019	83
Tableau II.4. Comparaison en fonction des stratégies de matching les divers systèmes participés dans les compagnies d'OAEI en 2017 et 2019	86
Tableau III.1. Résultats basés sur les alignements de référence de crisp de OAEI 2017	107
Tableau IV.1. Concepts des ontologies	112
Tableau IV.2. Résultats de similarités terminologiques	118
Tableau IV.3. Résultats de ONTMAT1 dans OAEI 2019.	124
Tableau IV.4. Estimation des résultats basée sur la version incertaine de l'alignement de référence	128
Tableau B.1. Les ontologies utilisées dans les teste de conference	138
Tableau B.2. Résultats des tests de ONTMAT1 en dessous de la piste de conference de OAEI 2019	139

# Introduction Générale

# *Introduction générale*

## **Sommaire**

1. Contexte de travail - - - - -	15
2. Problématiques et objectifs de la thèse - - - - -	16
3. Principales Contributions de la thèse - - - - -	18
4. Organisation de la thèse - - - - -	19
5. Publication - - - - -	21

## **1. Contexte de travail**

Le web traditionnel contient une quantité volumineuse de pages, mais une grande partie d'entre elles est dans un format lisible et compréhensible seulement pour l'homme. Pour pallier cette insuffisance du Web, les chercheurs ont conçu la vision du Web sémantique [Berners-Lee et al., 2001]. L'objectif principal de ce dernier est de bien exploiter cette quantité importante de données et de la rendre interopérable.

Contrairement au web classique qui fait une création des hyperliens entre les documents, le web de données fait une création des liens sémantiques entre les données de ces documents pour concevoir un seul espace de données global. Les liens entre ces données présentent un facteur clé dans plusieurs tâches intéressantes telles que l'interrogation, l'intégration des données, et la publication.

Ces liens sémantiques ont été conçus à l'aide des ontologies décrivant la structure et la sémantique des données. Les ontologies ont été définies comme une description explicite d'une conceptualisation [Gruber, 1995]. Ces ontologies permettent à l'utilisateur d'organiser l'information en taxonomie des concepts, chacun avec leurs attributs, et décrivent des relations entre ces concepts. Diverses applications utilisent les ontologies pour représenter les connaissances d'un domaine donné tel que: les services de web sémantique, l'apprentissage à distance, l'intégration des bases de données, les systèmes peer-to-peer, commerce, les réseaux sociaux, etc. [Fensel, 2014].

L'ontologie peut simplifier la communication en offrant des notions précises pouvant former et échanger les informations. La représentation de différents domaines par diverses descriptions et interprétations peut donner plusieurs ontologies [Euzenat & Shvaiko, 2007]. Par conséquent, il n'y avait pas d'ontologie universelle partagée, optée par tous les concepteurs d'un domaine donné. Ces ontologies ont été utilisées par plusieurs systèmes et applications.

Les tentatives d'interopérabilité et de communication entre les systèmes du Web sémantique et les autres applications exploitant ces ontologies, consistent à les réconcilier. Cette réconciliation consiste à identifier des liens de correspondance entre les entités des ontologies, et a engendré l'appariement d'ontologies.

## 2. Problématiques et objectifs de la thèse

L'appariement d'ontologies est devenu un élément primordial du web sémantique, où plusieurs outils d'alignements d'ontologies automatiques ou semi-automatiques ont été proposés (E.g. [Euzenat & Shvaiko, 2007; Shvaiko, Euzenat, Jiménez-Ruiz, Cheatham, & Hassanzadeh, 2018]). L'appariement d'ontologies est défini dans [Euzenat & Shvaiko, 2007] comme un processus de recherche de correspondance entre les différentes entités des ontologies à apparier.

Toute ontologie conçue donne une définition formelle de types, de propriétés, et d'interrelations des entités. La définition de la même entité par plusieurs noms dans les différentes ontologies peut causer l'hétérogénéité terminologique. Cette dernière est engendrée à cause de la synonymie (E.g. Paper vs. Article) (Euzenat et Shvaiko, 2007).

L'évolution du web sémantique et la croissance rapide des instances ontologiques par rapport aux classes et propriétés nous ont donné l'idée d'aligner les entités à l'aide de ses instances.

Vu que le raisonnement humain a du mal à comprendre le sens des termes avant de les placer ou de les lire dans un contexte défini, la recherche des voisins des concepts déjà alignés et la déduction de la relation d'alignement entre eux peuvent améliorer l'efficacité de notre proposition.

Deux ontologies peuvent décrire des parties distinctes ou chevauchées d'un domaine donné [Berners-Lee et al., 2001], ce qui peut provoquer l'hétérogénéité sémantique et structurelle (Euzenat, J. 2001). Pour pallier ce problème d'hétérogénéité, des techniques d'appariement structurelles et sémantiques peuvent être utilisées ensemble. Ces techniques peuvent être appliquées sur les concepts, les propriétés des ontologies afin d'incrémenter l'efficacité d'appariement d'ontologie.

La majorité de méthodes terminologiques et structurelles ont été établies sur des mesures de similarités et di-similarités entre les entités (les classes et les propriétés). Une fois ces valeurs de similarités et di-similarités entre les entités sont disponibles, l'alignement entre ces entités doit être calculé à l'aide d'une ou de plusieurs stratégies de matching [Euzenat & Shvaiko, 2013].

Dans la littérature, plusieurs stratégies de matching ont été proposées. Le partitionnement des ontologies est l'une de ces stratégies, implémentée dans Falcon-AO (Hu, Qu, & Cheng, 2008), TaxoMap (Hamdi, et al, 2010). L'élagage de l'espace de recherche est une autre technique optée par AROMA (David, 2011), Anchor-Flood (Hanif, & Aono, 2009), LogMap (Jiménez-Ruiz, 2019), etc. Ces méthodes ignorent des parties de l'ontologie afin d'incrémenter le rappel, ce qui peut influencer négativement sur le résultat final.

L'agrégation des similarités est une autre stratégie de matching ; y compris : *Max*, *Min*, *Moyenne*, et *pondérée* [Euzenat et Shvaiko, 2013]. La stratégie de Max/Min retourne la similarité maximale/minimale des entités alignées, tandis que la stratégie pondérée définit une somme pondérée entre les alignements des entités. La stratégie de moyenne est une instance de la stratégie pondérée où tous les poids ont eu la même valeur.

La stratégie de Max/Min choisit une valeur extrême parmi les valeurs de similarités fournies, qui peut être très optimiste surtout dans le cas de comparaison contradictoire (e.g. Max), où très pessimiste (e.g. Min). Cependant la stratégie de moyenne traite les similarités de manière égale et ne différencie pas entre elles.

En outre, la valeur de similarités calculée permet dans la plupart des systèmes de matching (e.g. Logmap [Jiménez-Ruiz, 2019], AgreementMaker [Faria et al., 2019]) de fournir uniquement les relations d'équivalence et de disjonction ce qui ne répond pas aux besoins des systèmes sur le web sémantique, tels que les systèmes d'interopérabilité et d'intégration.

Les problématiques auxquelles nous cherchons à concevoir des solutions dans le cadre de cette thèse se résument en:

- Comment peut-on détecter la relation entre des entités décrites par différents termes ayant le même sens.
- Comment peut-on trouver les voisins d'un concept donné, et découvrir l'alignement des entités à partir de matching existant.
- Comment peut-on combiner les différentes similarités découvertes par les diverses techniques de matching.
- Comment peut-on déduire des relations autres que l'équivalence et la disjonction, entre les entités qui vont aligner.

### 3. Principales Contributions de la thèse

En prenant en considération les objectifs mentionnés ci-dessus, les principales contributions de cette thèse sont énumérées dans ce qui suit :

1. Afin de mieux montrer l'utilité des systèmes proposés et participés dans la compagnie d'OAEI (Ontology Alignment Evaluation Initiative des années 2017 et 2019) avec nos systèmes proposés, les techniques et les stratégies de matching des systèmes participés dans cette compétition, vont être étudiées et comparées.
2. Nous cherchons dans cette thèse à résoudre le problème de synonymie en proposant d'exploiter un raisonneur avec une connaissance externe comme WordNet [Sabou, D'Aquin, Motta, 2006; Giunchiglia, Shvaiko, Yatskevich, 2004]. Ainsi, inférer l'alignement des concepts ayant des relations de voisinage avec les concepts déjà alignés. De plus, les entités dans cette thèse ont aligné non seulement par la relation d'équivalence mais aussi par les relations de subsomption et une nouvelle relation proposée appelée relation flou (fuzzy relation). [Gherbi et Khadir, 2016; Gherbi et Khadir, 2020]
3. Dans l'objectif d'augmenter l'efficacité du système proposé ONTMAT (ONTologies MATching), les caractéristiques de l'interprétation sémantique de la logique de description ont été exploitées. Ces caractéristiques permettent de prouver l'influence de l'alignement des instances sur l'alignement de leurs concepts. Ainsi, les caractéristiques d'inférence et de raisonnement ont été utilisées pour prouver la déduction

de l'alignement des concepts à partir de l'alignement de ses propriétés de restriction. [Gherbi et Khadir, 2016].

### 4. Organisation de la thèse

De nos objectifs détaillés dans la section précédente découle naturellement l'organisation de cette thèse qui est structurée en quatre chapitres :

Les deux premiers chapitres constituent l'état de l'art de la thèse :

- Le chapitre 1 introduit le Web Sémantique, et des notions générales de représentation des connaissances, qui serviront tout au long de cette thèse. En particulier, ce chapitre décrit les ontologies d'une manière très générale, puis détaille la syntaxe, la sémantique, l'inférence et le raisonnement des logiques de description.
- Le chapitre 2 présente tout d'abord la terminologie de la mise en correspondance des ontologies, puis les problèmes et les besoins qui ont engendré le processus d'appariement. Une analyse et une description bien détaillée du processus de matching et des propriétés caractérisant ce processus ont été abordées par la suite. Ainsi, une recherche sur les techniques et les solutions existantes résolvant le problème d'appariement d'ontologies, est présentée. Les stratégies de matching les plus utilisées dans la littérature et applicable pour résoudre le problème de déduction de l'alignement ont aussi illustré. De plus, il étudie et compare les systèmes avec lesquels notre processus a été met en compétition.
- Le troisième chapitre est consacré à la présentation de notre première contribution. Il s'agit d'un algorithme proposant une nouvelle méthode de matching. Cet algorithme utilise une description spécifique de l'ontologie basée sur la logique de description afin de mieux expliquer ses différentes étapes. L'approche commence par l'alignement des instances des concepts afin de déduire les relations entre ces derniers. Ces relations vont être stockées dans deux bases, une est appelée une base d'appariement d'ontologies  $Ao$  et l'autre une base temporaire d'appariement d'ontologies  $At$ . Ces bases seront des entrées de la deuxième phase du traitement, où les relations de  $At$  vont valider et les concepts voisins d' $Ao$  vont aligner en utilisant les méthodes terminologiques, les techniques de raisonnement, et le dictionnaire WordNet. Une

implémentation de cet algorithme par le système ONTMAT est validée sous OAEI 2017.

- Une extension du système ONTMAT par l'introduction d'un poids pour automatiser le calcul de mapping est appelée ONTMAT1. Le chapitre quatre, présente la contribution de ONTMAT1. Ce dernier exploite les propriétés de restriction des concepts, un raisonneur et la ressource de données WordNet, pour avoir plus de sémantique et de fiabilité au niveau du processus de matching. Cette approche automatique d'appariement d'ontologies est implémentée en quatre parties. La première partie compare les noms des individus et des concepts en appliquant une technique terminologique utilisant un raisonneur Pellet, une connaissance complémentaire sous forme de WordNet, et une mesure de similarité nommée n-gram. Les similarités obtenues ont enregistré dans deux matrices, une pour les individus et l'autre pour les concepts. La deuxième partie suggère un poids basé sur la sémantique de restriction des propriétés pour évaluer automatiquement l'impact des similarités structurelles sur la similarité finale. Cette dernière est estimée dans la troisième partie en exploitant les restrictions des propriétés. Enfin, les similitudes terminologiques et structurelles ont été agrégées par le poids proposé et les relations floues est décrites.

La dernière partie du chapitre présente l'ensemble des évaluations réalisé dans le cadre de cette thèse afin de prouver l'efficacité des solutions proposées dans la découverte des correspondances entre les entités des ontologies à apparié. L'étape d'évaluation contient l'expérimentation et l'estimation de différentes techniques de calcul de similarité utilisées par le système. L'expérimentation est réalisée sur les Benchmark de OAEI 2019 (Ontology Alignment Evaluation initiative) afin d'étudier l'évaluation de l'impact de poids à base de sémantique sur le processus d'appariement. L'étape d'évaluation inclut aussi une phase de comparaison des résultats obtenus par notre système avec ceux obtenus par les autres systèmes présentés dans l'état de l'art et participés dans la compagnie [Gherbi et Khadir, 2020; Gherbi et Khadir, 2019].

- Une conclusion générale des travaux réalisés et de ce que pourra être effectué en perspective, sera exhibée à la fin de ce mémoire.

### 5. Publications

Notre étude a été fructifiée par la publication des articles suivants:

1. [Gherbi et al., 2013], Gherbi S., Belleili H. and Khadir M., 2013. BRMAP : Un outil d'Alignement des ontologies. 7ème édition de la conférence maghrébine sur les avancés des systèmes décisionnels.
2. [Gherbi et Khadir, 2016], Gherbi, S, Khadir, M. T. , (2016), Inferred ontology concepts alignment using instances and an external dictionary, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology(SEIT-2016)/Affiliated-Workshops.  
doi:<http://dx.doi.org/10.1016/j.procs.2016.04.145>.  
URL:<http://www.sciencedirect.com/science/article/pii/S1877050916301752>,  
Procedia Computer Science 83 648-652 (Elsevier 2016).
3. [Gherbi et Khadir, 2017], Gherbi, S., & Khadir, M. T. (2017). ONTMAT: results for OAEI 2017. In OM@ ISWC (pp. 166-170).
4. [Gherbi et KHADIR, 2019], Saida Gherbi and MedTarek. ONTMAT1: Results for OAEI 2019. *Ontology Matching*, 2019, p. 164.
5. [Gherbi et Khadir, 2020], Gherbi, S. and Khadir, M.T. (2020) 'ONTMAT1: ontology matching using a reasoner and property restriction', *Int. J. Web Engineering and Technology*, Vol. 15, No. 2, pp.119–139.

Partie I:

Etat de l'art

## ***Chapitre I :***

# ***Ontologie et la logique de description***

# *Chapitre I*

## *Ontologie et la logique de description*

### **Sommaire**

I.1 Web sémantique . . . . .	25
I.2. Ontologie . . . . .	25
I.2.1. Définition . . . . .	25
I.2.2. Composants d'une ontologie . . . . .	26
I.2.3. Principes de construction d'une ontologie . . . . .	27
I.2.4. Langages d'ontologies . . . . .	28
I.2.5. Langage de représentation des ontologies OWL . . . . .	29
I.3. Logique de description . . . . .	31
I.3.1. Sémantique de la logique de description . . . . .	31
I.3.2. Syntaxe de la logique de description . . . . .	31
I.3.2.1. Niveau terminologique (TBox) . . . . .	32
I.3.2.2. Niveau factuel (ABox) . . . . .	32
I.3.3. Inférence . . . . .	32
I.3.3.1. Inférence au niveau terminologique . . . . .	33
I.3.3.2. Inférence au niveau factuel . . . . .	33
I.3.4. Raisonnement . . . . .	33
I.3.4.1. Raisonnement basé sur la subsomption structurelle . . . . .	34
I.3.4.2. Raisonnement basé sur l'algorithme de tableaux . . . . .	35
I.4. Moteurs d'inférence . . . . .	35
I.4.1. Racer . . . . .	36
I.4.2. Librairie Pellet . . . . .	37
I.5. Conclusion . . . . .	37

Ce chapitre est consacré à la description des différents termes utilisés le long de ce mémoire afin de le mieux comprendre. Nous commençons par définir le Web Sémantique fondé aussi bien sur les ontologies que sur un langage de représentation des ontologies appelé OWL. Ce dernier est basé sur des techniques de la logique de description qui seront exhibées

avec les techniques d'inférence et de raisonnement incluses dans le système proposé. Le moteur d'inférence pellet et ses différentes caractéristiques seront présentés par la suite.

### I.1 Web sémantique

Le Web Sémantique est une notion due à [Berners-Lee et al., 2001] au sein de W3C<sup>1</sup> qui a vu le Web comme un large espace d'échange de ressources entre les êtres humains et des machines acceptant une exploitation qualitativement supérieure, de grands volumes d'informations et de services variés. Le Web sémantique, effectivement, est une infrastructure pour agréer l'exploitation de connaissances formalisées en plus du contenu informel du Web syntaxique. Cette infrastructure doit admettre d'abord de localiser, d'identifier et de transformer des ressources de manière puissante et saine tout en assurant l'esprit d'ouverture du Web avec sa divergence d'utilisateurs. Ainsi, elle doit se poser sur un certain niveau de consensus portant par exemple sur les langages de représentation ou sur les ontologies utilisées. En outre, elle doit contribuer à garantir, le plus automatiquement possible l'interopérabilité et les transformations entre les différents formalismes et les différentes ontologies. De plus, elle doit servir la mise en œuvre de calculs et de raisonnements complexes tout en soumettant des assurées supérieures sur leur validité. Elle doit aussi fournir des mécanismes de protection (droits d'accès, d'utilisation et de reproduction), ainsi que des mécanismes acceptant de qualifier les connaissances afin de croître le niveau de confiance des utilisateurs [Charlet et al, 2003].

### I.2. Ontologie

#### I.2.1. Définition

Ontologie est un terme emprunté d'une branche de la philosophie appelée métaphysique traitant l'étude de l'être. Elle est classée de plusieurs façons à l'aide de certains critères tels que le champ d'application et le niveau d'abstraction [Vesselin , 2011]:

1. *Ontologie de base* : concept sur lequel se base le développement des autres types.
2. *Ontologie de domaine* (le plus fréquemment rencontré) : expose un champ de connaissances pour répondre à des objectifs spécifiques dans un domaine donné

---

<sup>1</sup> <http://www.w3.org/>

comme les domaines scientifiques, les technologies de l'information ou les langages naturels.

3. *Ontologie d'interface* : relie des domaines de spécialités différentes entre eux.
4. *Ontologie de processus* : décrit les modalités de transformations ou d'évolutions à travers les entrées et les sorties en fonction de la contrainte du temps ; par exemple les mouvements financiers, les réactions physiques ou cognitives.

Dans le domaine de l'IA, diverses définitions d'ontologies sont données, mais celle la plus optée dans la majorité des recherches et cette thèse est donnée par Tom Gruber [Gruber, 1995] « *Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée* », avec la signification des termes suivants:

- *Formelle* : désigne qu'une ontologie doit être compréhensible par la machine qui doit avoir l'habileté d'interpréter la sémantique de l'information fournie;
- *Explicite* : indique que le type de concepts utilisés et les contraintes sur leur utilisation doivent être explicitement déterminés ;
- *Conceptualisation* : se réfère à un modèle abstrait de certains phénomènes dans le monde qui assimile les concepts adéquats de ce phénomène ;
- *Partagée* : indique que l'ontologie supporte la connaissance consensuelle, et n'est pas restreinte à certains individus mais elle est acceptée par un groupe.

### I.2.2. Composants d'une ontologie

Une ontologie peut être décrite comme un ensemble bien structuré de concepts et de relations entre ces concepts destinés à symboliser les objets du monde sous une forme compréhensible aussi bien par l'être humain que par les machines. Les connaissances traduites par une ontologie sont modélisées à l'aide de plusieurs constituants de base qui sont principalement des : concepts, relations, fonctions, axiomes, instances [Gruber, 1993 ; Gomez-Perez, 2000].

**Les concepts** : aussi appelés termes ou classes sont des notions (ou objets) permettant la description d'une tâche, une fonction, une action, une stratégie ou un processus de raisonnement, etc. Ils peuvent être abstraits ou concrets, élémentaires ou composés, réels ou fictifs. Habituellement, les concepts sont organisés en taxonomie. Une taxonomie est une

hiérarchie de concepts (ou objets) reliés entre eux en fonction de critères sémantiques particuliers.

*Les relations* : représentent les liens existant entre les concepts d'un domaine donné et traduit le type d'interaction entre ces concepts. Elles sont formellement prescrites comme tout sous-ensemble d'un produit de  $n$  ensembles, c'est-à-dire:  $R : c_1 \times c_2 \times c_3 \times \dots \times c_n$ . Des exemples de relations binaires sont : sous-concept-de, sorte-de, etc. Ces relations permettent de tenir la structuration ainsi que l'interaction entre les concepts, ce qui accepte de montrer une grande partie de la sémantique de l'ontologie.

*Les propriétés* (ou attributs) : sont des restrictions des concepts ou des relations.

*Les fonctions* : sont des cas spécifiques de relations dans lesquelles le  $n^{\text{ième}}$  élément de la relation est défini d'une manière unique à partir des  $n-1$  éléments précédents. Formellement, les fonctions sont définies de ce fait comme suit:  $F : c_1 \times c_2 \times c_3 \times \dots \times c_{n-1} \rightarrow c_n$

*Les axiomes* de l'ontologie définissent la sémantique des termes (classes, relations), leurs propriétés et toutes contraintes quant à leur interprétation. Ils sont décrits à l'aide de formules bien formées de la logique du premier ordre en utilisant les prédicats de l'ontologie.

*Les instances* : permettent de représenter les occurrences.

### I.2.3. Principes de construction d'une ontologie

Afin d'obtenir une ontologie susceptible de répondre aux objectifs de l'ontologie, le processus de construction d'une ontologie doit respecter les critères de bases suivants [Gruber,1993] :

*La clarté et l'objectivité* : l'ontologie doit offrir le sens des termes définis en fournissant des définitions objectives ainsi que de la documentation associée en langage naturel.

*L'exhaustivité* : une définition exposée par une condition nécessaire et suffisante est préférable à une définition formulée seulement par une condition nécessaire ou par une condition suffisante.

*La cohérence* : afin de pouvoir exprimer des inférences cohérentes avec les définitions.

*L'extensibilité monotone maximale* : les nouveaux termes, qu'ils relèvent de la langue générale ou d'une langue de spécialité devraient être introduits dans l'ontologie sans emporter de modifications dans les définitions existantes.

*L'intervention ontologique minimale* : L'ontologie devrait spécifier le moins possible le sens de ses termes, en phase de modélisation, de façon à ce que les parties impliquées dans l'ontologie aient les mains libres pour spécialiser et instancier l'ontologie à leur façon.

### I.2.4. Langages d'ontologies

Une ontologie doit être formulée en utilisant un langage donné et, de ce fait, les langages suivants ont été développés:

- **KIF** : [GINSBERG et al, 1994] est un langage fondé sur les prédicats du premier ordre avec des extensions pour symboliser des définitions et des méta-connaissances. La logique du premier ordre est un langage de bas niveau pour la formulation d'ontologies. Dans le serveur d'édition d'ontologies, une extension du langage KIF, ONTOLINGUA<sup>2</sup> est utilisée.
- **F-Logic** [Kifer et al, 1995] (La logique de Frame) est un langage de bases de données orienté objet combinant l'expressivité des langages de bases de données déductives et la richesse de modélisation des modèles orientés objet.
- **RDF et RDF Schéma** : Le W3C a admis le langage RDF (Resource Description Framework) comme un des formalismes standards de représentation de connaissances sur le Web<sup>3</sup>. Le RDF décrit des ressources Web en termes de ressources, propriétés et valeurs en utilisant la syntaxe de standard XML (Extended Markup Language<sup>4</sup>). Une ressource peut être présentée comme une page Web (identifiée par son URI, United Resource Identifier) ou une partie de page (identifiée par une balise). Les propriétés dissimulent les notions d'attributs, relations ou aspects et aident à montrer une caractéristique d'une ressource en spécifiant sa valeur. Les valeurs peuvent être des ressources ou des littéraux. RDF dispose d'une sémantique

---

<sup>2</sup> Ontolingua, développe au Knowledge Systems Laboratory de l'université de Stanford.

<http://www.ksl.stanford.edu/software/ontolingua/>

<sup>3</sup> <http://www.w3.org/RDF/>

<sup>4</sup> <http://www.w3.org/XML/>

formelle semblable à celle d'un fragment de la logique du premier ordre [BAGET, J. , 2004].

- **DAML + OIL** : Dans la vue d'utiliser des ontologies sur le Web, le langage RDF-S a été enrichi par la contribution du langage OIL (Ontology Interchange Language<sup>1</sup>). Ce langage utilise la syntaxe de RDF-S pour exprimer une sémantique à travers le modèle des frames en offrant des nouvelles primitives permettant de définir des classes à l'aide de mécanismes ensemblistes issus de la logiques de description (union des classes, intersection des classes, complémentaire d'une classe). Il permet aussi d'affiner les propriétés de RDF-S en contraignant la cardinalité ou en restreignant la portée [Fensel et al, 2000].

Le langage DAML (Darpa Agent Markup Language<sup>5</sup>) est une extension de la syntaxe de RDF [HENDLER et al, 2001] en offrant les primitives usuelles d'une représentation à base de frames.

Les langages DAML et OIL ont été fusionné parce que l'intégration d'OIL permet aussi bien les inférences avec les logiques de description que les calculs de liens de subsomption.

L'intégration de RDF/RDF-S avec DAML+OIL a engendré le langage standard de représentation de connaissances pour le Web sémantique appelé OWL (Web Ontology Language) [ELBYED M., 2009].

### **I.2.5. Langage de représentation des ontologies OWL**

OWL est un langage de représentation des ontologies sur le Web, recommandé par le World Wide Web Consortium depuis 2004. Ce langage décrit le domaine d'une ontologie en termes de classes, propriétés et individus, ainsi que des descriptions des caractéristiques de ces objets peuvent être incluses [BECHHOFER ,2009]. Il se constitue de trois sous-langages fournissant différents degrés d'expressivité: OWL Lite, OWL DL, OWL Full. Ainsi, OWL a bien bénéficié des résultats de recherche de la Logique de Description (DL) [Baader et al 2003a, Horrocks et al 2004]. La sémantique d'OWL, est décrite de telle sorte que les gros fragments (expressions) du langage peuvent être directement formulés en DL [Baader et al 2003b].

---

<sup>5</sup> <http://www.ontoknowledge.org/oil/>

Le langage **OWL Lite** est conçu pour répondre aux besoins de la hiérarchie de classification et de la fonctionnalité des contraintes simples de cardinalité 0 ou 1. Cette cardinalité est une relation fonctionnelle, par exemple : une personne a une adresse. Toutefois, cette personne peut avoir un ou plusieurs prénoms [ELBYED M., 2009], donc, OWL Lite est insuffisant pour satisfaire cette situation et nous devons passer au sous langage suivant.

Le langage **OWL DL** offre une expressivité maximale couplée avec la complétude de calcul (cela assure toutes les inférences) et la décidabilité du système de raisonnement (i.e. tous les calculs seront achevés dans un intervalle de temps déterminé). Ce langage intègre toutes les structures d'OWL avec certaines restrictions, comme la séparation des types: une classe ne peut pas être à la fois un individu ou une propriété. Il est nommé DL car il répond aux différentes caractéristiques de la logique de descriptive (Description Logic : DL) [ELBYED M., 2009]. Ce langage est le plus favorisé dans notre thèse à cause de son mécanisme de raisonnement basé sur la logique de description et sa syntaxe supportée par XML. La description logique a des avantages dont les principaux sont [Zimmermann,2008]:

1. La capacité d'exploiter le raisonnement automatique en se centralisant sur la sémantique.
2. La description logique permet l'ajout automatique d'une nouvelle classe dans le bon endroit de la taxonomie tout en détectant l'héritage et en évitant la contradiction dans les définitions des classes. Cette propriété simplifie la localisation des concepts qui sont mis à jour.
3. Assurer un certain compromis entre l'expressivité et la calculabilité. L'expressivité d'un langage de formalisation de l'ontologie est de permettre de diffuser au mieux le sens tourné vers l'utilisateur /calculabilité : Admettre des inférences tournées vers la machine.
4. Les autres formalismes (RDF, RDF (S)) ne permettant pas les raisonnements car ils sont limités.

Le langage **OWL Full** assure une expressivité maximale. Il a une compatibilité complète avec RDF/RDFS mais avec un haut niveau de capacité de description qui ne garantit pas la complétude et la décidabilité des calculs liés à l'ontologie [ELBYED M., 2009].

### I.3. Logique de description

La logique de description est une famille de langages permettant l'écriture des phrases ou des formules représentant des connaissances données, à l'aide d'une syntaxe formulée, et une sémantique permettant de les interpréter. Elle contient des connaissances implicites qui peuvent être explicitées par des inférences [Gruber, 1993]. Par conséquent, un système de représentation des connaissances, fondés sur la logique de description est capable de réaliser certains types de raisonnement.

#### I.3.1. Sémantique de la logique de description

Les langages de représentation des connaissances de la logique de description sont caractérisés par la sémantique exprimant la manière dont les symboles doivent être interprétés et comment les formules contraignant ces interprétations. L'interprétation permet d'associer aux symboles d'une signature des éléments d'un certain "univers" ou domaine d'interprétation, les contraintes définissant leurs satisfactions [Zimmermann, 2008].

Une interprétation est constituée d'un domaine d'interprétation  $\Delta^I$  et d'une fonction d'interprétation  $I$ . Le domaine d'interprétation se compose d'un ensemble d'individus. La fonction d'interprétation assigne à chaque concept atomique  $C$  un ensemble tel que :  $C^I \in \Delta^I$ , et à chaque rôle atomique  $R$  une relation binaire  $R^I \subseteq \Delta^I \times \Delta^I$ .

En d'autres termes, une interprétation  $I$  d'une signature  $S$  d'un langage  $L$  est un couple  $\langle I, \Delta \rangle$  où  $\Delta$  est un ensemble non vide ( $\Delta_t$ ) indicé par les types de symboles de  $L$ , et  $I$  est un ensemble de fonctions  $I_t : S_t \rightarrow \Delta_t$  appelé une fonction d'interprétation. Delta «  $\Delta$  » est appelé le domaine d'interprétation de  $I$ . Une interprétation d'une ontologie  $O$  est une interprétation de la signature de  $O$ . Étant donnée en logique de description une signature  $S = (C, R, U)$ , une interprétation  $I$  de  $S$  associe à chaque concept  $C$  un ensemble  $C^I \subseteq \Delta$ , à chaque rôle  $R$  une relation  $R^I \subseteq \Delta \times \Delta$  et à chaque individu  $a$  un élément  $a^I \subseteq \Delta$  [Baader et al, 2003c].

#### I.3.2. Syntaxe de la logique de description

Un langage de représentation de connaissances doit utiliser une certaine syntaxe afin de pouvoir construire les "phrases" ou les assertions à partir d'un ensemble de symboles et de

règles de construction. La logique de description représente les connaissances d'une ontologie en deux niveaux de description : le niveau terminologique (TBox) et le niveau factuel (ABox). Le niveau terminologique utilise les concepts pour exprimer les ensembles d'entités du monde, et les relations entre ces entités sont représentées par des rôles. Le niveau factuel exprime des faits sur des individus définissant les entités elles-mêmes [Zimmermann, 2008].

### I.3.2.1. Niveau terminologique (TBox)

Une TBox possède des axiomes terminologiques de la forme  $C \equiv D$  ou  $C \subseteq D$  (Le côté gauche de l'exemple suivant présente un TBox), par exemple :  $Homme \equiv Humain \sqcap Male$ . Le premier sert à exprimer des relations d'équivalence entre les concepts, alors que le second permet d'énoncer des relations d'inclusion. Une interprétation  $I$  satisfait un axiome  $C \subseteq D$  si et seulement si  $C^I \subseteq D^I$ . Une interprétation  $I$  satisfait un axiome  $C \equiv D$  si et seulement si  $C^I = D^I$ . Une interprétation satisfait une TBox (est un modèle de TBox) si et seulement si l'interprétation satisfait tous les axiomes de la TBox [Baader et al, 2003c].

### I.3.2.2. Niveau factuel (ABox)

Une ABox comprend un ensemble d'assertions sur les individus : (1) des assertions d'appartenance et (2) des assertions de rôle. Toute ABox, doit être associée à une TBox, par exemple, l'individu *Anes* est associé au concept *Homme* dans  $Homme(Anes)$ , car les assertions s'expriment en termes de concepts et de rôles de la TBox.

Une ABox désigne des individus dans ses assertions par des noms qu'elle leur donne. Ce document utilise le terme individu nommé (nominal ou individual name, en anglais) pour référer à ces noms. La suite de ce document représente par les lettres  $a, b$  les individus nommés. Une fonction d'interprétation assigne à chacun de ces noms  $a$ , un individu  $a^I$  tel que  $a^I \in \Delta^I$ .

### I.3.3. Inférence

L'inférence s'effectue au niveau terminologique ou factuel. Ainsi, l'hypothèse de nom unique (HNU) est optée généralement par les moteurs d'inférence de LD, c'est-à-dire que pour tout individu nommé  $a$  et  $b$ ,  $a^I \neq b^I$ . Chaque assertion d'appartenance d'une ABox (notée  $C(a)$ ), assure que pour cette ABox, il existe un individu nommé  $a$ , membre du concept  $C$  de la TBox associée. Une interprétation satisfait une assertion d'appartenance  $C(a)$  si et seulement si  $a^I \in C^I$  [Baader et al 2003a].

### I.3.3.1. Inférence au niveau terminologique

Quatre principaux problèmes d'inférence se présentent au niveau terminologique [Baader et al, 2003c] :

- **Satisfiabilité** : Un concept  $C$  d'une terminologie  $T$  est satisfiable si et seulement s'il existe un modèle  $I$  de  $T$  tel que  $C^I \neq \phi$ .
- **Subsomption** : Un concept  $C$  est subsumé par un concept  $D$  pour une terminologie si et seulement si  $C^I \subseteq D^I$  pour tout modèle  $I$  de  $T$ .
- **Équivalence** : Un concept  $C$  est équivalent à un concept  $D$  pour une terminologie si et seulement si  $C^I \equiv D^I$  pour chaque modèle  $I$  de  $T$ .
- **Disjonction (disjointness)** : Des concepts  $C$  et  $D$  sont disjoints par rapport à une terminologie  $T$  si et seulement si  $C^I \cap D^I = \phi$  ; pour chaque modèle  $I$  de  $T$ .

### I.3.3.2. Inférence au niveau factuel

Le niveau factuel comprend quatre principaux problèmes d'inférence [Baader et al., 2003a] :

- **Cohérence** : une ABox  $A$  est cohérente par rapport à une TBox  $T$  si et seulement s'il existe un modèle  $I$  de  $A$  et  $T$ .
- **Vérification d'instance** : consiste à vérifier par inférence si une assertion  $C(a)$  est vraie pour tout modèle  $I$  d'une ABox  $A$  et d'une TBox  $T$ .
- **Vérification de rôle** : admet une vérification par inférence si une assertion  $R(a, b)$  est vraie pour tout modèle  $I$  d'une ABox  $A$  et d'une TBox  $T$ .
- **Problème de récupération** : pour une ABox  $A$ , un concept  $C$  d'une terminologie  $T$ , inférer les individus  $a_1^I, \dots, a_n^I \in C^I$  pour tout modèle  $I$  de  $T$ .

### I.3.4. Raisonnement

Le raisonnement sur les connaissances utilise une notion appelée satisfaction dont l'interprétation ne permet de relier que les symboles aux "choses" de l'univers de discours modélisés par le domaine d'interprétation. La satisfaction sert à relier les formules aux

interprétations satisfaisant ces formules, et valide la connaissance formulée par eux. Par exemple, soient les concepts  $C$  et  $D$  en logique de description et la formule  $C \sqsubseteq D$ . Cette dernière est satisfaite par une interprétation  $I$  si et seulement si l'interprétation de  $C$  par  $I$  est un sous-ensemble de l'interprétation de  $D$  par  $I$  [Zimmermann, 2008].

D'ailleurs, les deux grandes classes d'algorithmes de raisonnement de LD (algorithmes de subsomption de type normalisation/comparaison et algorithmes de vérification de satisfiabilité à base de tableaux) correspondent aux façons de réduire respectivement les problèmes d'inférence décrits dans la section I.3.3.1 aux problèmes de subsomption et de satisfiabilité [Baader et al., 2003 a,b,c].

### I.3.4.1. Raisonnement basé sur la subsomption structurelle

La comparaison structurelle entre les expressions des concepts, est l'idée principale des algorithmes de calcul de subsomption. Par conséquent, si deux expressions de deux concepts sont conçues de sous-expressions, alors elles peuvent être comparées en comparant une sous-expression d'un concept avec toutes celles de l'autre [Ronal, 1984]. L'algorithme se réalise en deux phases : premièrement, les concepts sont réécrits dans une forme normale (étendre les concepts et factoriser les rôles), et ensuite leurs structures sont appréciées.

Le langage  $\mathcal{FL}$  de la logique de description est opté, en vue de simplifier le déroulement suivant de cet algorithme [Baader et al., 2003 a,b,c] :

- La description d'un concept (e.g.  $A \sqcap (B \sqcap C)$ ) peut être transformée en une forme normale (e.g.  $A \sqcap B \sqcap C$ ) en utilisant l'associativité, la commutativité, l'idempotence de  $\sqcap$ .
- Soient  $C = C_1 \sqcap C_2 \sqcap \dots \sqcap C_m$  et  $D = D_1 \sqcap D_2 \sqcap \dots \sqcap D_n$ ,  $D$  subsume  $C$  si et seulement si pour chaque  $D_i$ , il existe un  $C_j$  avec :
  - Si  $D_i$  est un concept atomique, alors  $D_i = C_j$ .
  - Si  $D_i$  est un concept de forme  $\forall P.D'$ ,  $C_j = \forall P.C'$  (le même rôle atomique  $P$ ), alors  $C' \sqsubseteq D'$ .

### I.3.4.2. Raisonnement basé sur l'algorithme de tableaux

Cette technique de raisonnement s'appuie sur le calcul de tableaux pour la logique des prédicats du premier-ordre [Smullyan, 1968]. Son objectif primordial est l'adaptation des calculs aux analyses de la complexité. Les structures de tableaux acquises en raisonnant avec un langage de LD donné sont précieusement analysées, et les vérifications redondantes dans les tableaux sont écartées afin d'accorder une limite supérieure étroite sur la complexité de la méthode.

L'algorithme de tableaux dans une LD essaye de démontrer la satisfaisabilité d'une expression de concept  $C$  en prouvant l'existence d'un modèle - une interprétation  $I = (\Delta^I, \cdot^I)$  dans laquelle  $C^I \neq \emptyset$  [Hollunder et al., 1990]. Le calcul de tableaux exige de vérifier : *si une formule donnée  $F$  est une conséquence logique d'une théorie donnée  $H$* . Par la suite, il essaye de construire, avec les règles de propagation adéquates, le modèle le plus générique de  $H$  où  $F$  est faux. Si le modèle est construit avec succès, alors la réponse est NON (vue que  $F$  n'est pas une conséquence logique de  $H$ , donc la méthode est complète) ; si le modèle construit n'a pas de succès, alors la réponse est OUI (vu qu'il n'existe pas de modèle où  $H$  avec  $F$  donnent faux, donc  $F$  est réellement une conséquence logique de  $H$  et la méthode est correcte).

### I.4. Moteurs d'inférence

Les moteurs d'inférence actuels tirent souvent profit du fait que les quatre types de problèmes d'inférence peuvent être réduits à des problèmes de subsomption ou à des problèmes de satisfiabilité. Ces réductions impliquent que les moteurs d'inférence de LD ne requièrent souvent qu'un seul algorithme d'inférence pour raisonner au niveau terminologique [Baader et al, 2003c].

**Réduction vers la subsomption** : pour deux concepts  $C$  et  $D$  on a :

1.  $C$  est non satisfiable  $\Leftrightarrow C$  est subsumé par  $\perp$  ;
2.  $C$  et  $D$  sont équivalents  $\Leftrightarrow C$  est subsumé par  $D$  et  $D$  est subsumé par  $C$
3.  $C$  et  $D$  sont disjoints  $\Leftrightarrow C \sqcap D$  est subsumé par  $\perp$ .

**Réduction vers la satisfiabilité** :

1.  $C$  est subsumé par  $D \Leftrightarrow C \sqcap \neg D$  est insatisfiable;
2.  $C$  et  $D$  sont équivalents  $\Leftrightarrow$   
( $C \sqcap D \neg D$ ) et ( $\neg C \sqcap D$ ) sont insatisfiables
3.  $C$  et  $D$  sont disjoints  $\Leftrightarrow C \sqcap D$  est insatisfiable.

De nombreux moteurs d'inférences gratuits ou commerciaux ont été conçu, tels que Racer, Pellet, Fact, Fact++, Hermit, F-OWL et Howlet, etc. Ces moteurs d'inférence sont compatibles avec le langage OWL que nous avons opté. Certains moteurs d'inférence ne raisonnent qu'au niveau terminologique (c.à.d. au niveau des concepts et des propriétés) alors que les moteurs Racer et Pellet assurent aussi un raisonnement au niveau des instances des concepts (c.à.d. les individus particuliers) (Sirin et al., 2007).

### I.4.1. Racer

Racer [Haarslev et Möller, 2001] est le moteur d'inférence le plus connu et le plus utilisé à cause de ses performances et sa stabilité. Il est commercialisé par Racer Systems GmbH & Co. KG de l'université de Hambourg. Racer possède son propre langage de requête nRQL (new Racerpro query Language) pour interroger les ontologies sur ABox et TBox. Ce moteur d'inférence avait les avantages et les inconvénients suivants:

#### *Les avantages :*

- La disponibilité de la documentation sur Racer, provenant des concepteurs et des utilisateurs.
- Racer utilise un mécanisme d'abonnement à un concept permettant de détecter la création de nouvelles instances de ce concept.
- Racer peut ajouter des assertions et des individus dans les ABox après le chargement de l'ontologie.
- Racer utilise des règles de SWRL.

#### *Les inconvénients :*

- Racer considère que tous les propriétés sur les datatypes sont fonctionnelles (pas de valeurs multiples pour un datatype property)
- Racer n'autorise pas l'utilisation de type de données utilisateur (type défini par l'utilisateur), car il avait ces propres types de données et il permet de faire une conversion avec les types de base.
- Racer est un produit commercial, il n'y avait pas de version libre d'utilisation. Cependant il est probable d'obtenir une licence gratuite dans le cadre de la recherche scientifique.

### I.4.2. Librairie Pellet

Pellet [Sirin et al, 2007], [Clark and Parsia, 2009] est une librairie open-source Java implémentant des raisonnements applicables aux ontologies OWL. Elle est utilisable en conjonction avec Jena et les API OWL. Parmi les fonctionnalités offertes par Pellet, on peut citer:

- La vérification de la consistance d'une ontologie ;
- La classification pour les concepts des taxonomies ;
- Les tests de subsumption ;
- L'analyse et la réparation des ontologies qui implique une vérification des documents OWL dans la syntaxe RDF, avec le but d'identifier et de corriger les fautes de syntaxe.
- Le raisonnement sur les types de données.

Pellet est un raisonneur puissant basé sur l'algorithme des tableaux, et le seul qui prenne en charge tous les classes énumérées et les types de données définis par l'utilisateur [Mindswap, 2003], [Sirin et al, 2007].

Pour résoudre le problème d'URI désignant les nouveaux types, Pellet adopte la solution proposée par DAML+OIL qui se fonde sur les étapes suivantes:

1. la construction d'un nouvel URI à partir de l'URI du document contenant le schéma XML et le nom local du type simple [Mindswap, 2003].
2. Le document identifié par l'URI et qui contient le schéma XML sera analysé dans le but de localiser le type dont l'attribut name contient le nom de la définition.
3. En conséquence, OWL pourra utiliser des types de données simples définis au premier niveau d'un document XML Schéma, en les référençant par l'URI du document et le nom local du type [Mindswap, 2003].

La nouvelle version, Pellet 2.0 est compatible avec le langage OWL 2 [E. Sirin et al, 2007] et intègre diverses techniques d'optimisation, y compris pour les nominaux, les requêtes conjonctives, le raisonnement progressif, etc. [Sirin et al, 2007].

### I.5. Conclusion

Dans ce chapitre nous avons présenté la notion de l'ontologie constituant le noyau du Web sémantique, ainsi que le langage OWL permettant sa représentation. Ce langage se fonde sur

## **Chapitre I : Ontologie et la logique de description**

---

la logique de description dont la sémantique et la syntaxe permettant une description des classes, propriétés et instances, des ontologies.

En particulier, les règles d'inférence et les techniques de raisonnement exploitées dans nos propositions ont illustré. Ainsi, deux moteurs d'inférences ont été exhibé desquels nous allons choisir Pellet comme il sera expliqué dans la section IV.2.

***Chapitre II :***  
***Problèmes d'appariement***  
***et les travaux connexes***

## *Chapitre II*

### *Problèmes d'appariement et les travaux connexes*

#### Sommaire

II.1. Terminologies de la mise en correspondance d'ontologies . . . . .	41
II.2. Problèmes d'appariement . . . . .	42
II.2.1. Niveaux d'hétérogénéités entre les ontologies . . . . .	42
II.2.2. Modes de l'interopérabilité sémantique . . . . .	45
II.2.2.1. Mode intégré . . . . .	46
II.2.2.2. Mode Fédéré . . . . .	46
II.2.2.3 Mode Unifié . . . . .	47
II.3. Processus d'appariement (Matching) . . . . .	48
II.3.1 Description du processus d'alignement des ontologies . . . . .	48
II.3.2. Techniques d'appariement . . . . .	49
II.3.2.1. Similarité . . . . .	49
II.3.2.2. Méthodes terminologiques . . . . .	52
II.3.2.2.1. Approche syntaxique . . . . .	52
II.3.2.2.2. Approche linguistique . . . . .	53
II.3.2.3. Méthodes structurelles . . . . .	55
II.3.2.4. Méthodes extensionnelles . . . . .	57
II.3.2.5. Méthodes sémantiques . . . . .	58
II.3.2.6. Méthodes probabilistes . . . . .	60
II.4. Stratégies d'appariement . . . . .	62
II.4.1. Partitionnement des ontologies et la recherche d'un espace d'élagage . . . . .	63
II.4.1.1. Partitionnement des ontologies . . . . .	63
II.4.1.2. Recherche d'un espace d'élagage . . . . .	64
II.4.2. Agrégation de l'alignement . . . . .	64
II.4.2.1. Pondération . . . . .	65
II.4.2.2. Election . . . . .	69
II.4.2.3. Argumentation . . . . .	70

II.4.3. Méthodes d'apprentissage - - - - -	70
II.5. Etude et comparaison des approches d'appariement - - - - -	71
II.5.1. Etude des approches d'appariement participées à l'OAEI - - - - -	72
II.5.2. Comparaison des approches d'appariement participées à l'OAEI - - - - -	79
II.5.2.1. Comparaison des caractéristiques externes - - - - -	80
II.5.2.2. Comparaison des techniques de bases - - - - -	81
II.5.2.3. Comparaison des stratégies de matching - - - - -	86
II.6. Conclusion - - - - -	87

Dans ce chapitre, nous commençons tout d'abord par la présentation des terminologies de la mise en correspondance puis nous expliquons les divers problèmes d'alignement qui ont engendré le processus d'appariement tels que les niveaux d'hétérogénéités des ontologies et les différentes approches d'interopérabilité. Par la suite, nous nous intéresserons aux différentes techniques et stratégies de matching. Ainsi, nous illustrons les systèmes et les travaux de recherche liés au processus d'appariement d'ontologies. De plus, une étude et une comparaison des outils et systèmes existants traitant le matching d'ontologies sont présentées. Cette étude est fondée sur les trois éléments de l'hétérogénéité, les modes d'interopérabilité et les méthodes de matching.

### II.1. Terminologies de la mise en correspondance d'ontologies

Dans cette section, nous présentons les différentes terminologies les plus utilisées dans ce domaine pour désigner une mise en correspondance. Ainsi, la signification des termes liés à la problématique de mapping d'ontologie à savoir : l'alignement d'ontologies, les mappings, le matching, le matcher, la fusion et l'intégration d'ontologies.

**Alignement d'ontologies:** L'alignement d'ontologies consiste à établir des liens de correspondances entre les entités des deux ontologies originales [Euzenat et Shvaiko, 2007]. Il est appliqué habituellement quand les ontologies sources appartiennent à des domaines complémentaires.

**Mappings (Correspondances):** Les mappings sont des processus d'extraction des relations existantes entre les éléments de deux représentations (ontologies, schémas de bases de données, etc.), spécifiant une similarité relative selon une mesure donnée [Klein et al, 2001].

**Matching (Appariement):** Le matching définit un ensemble de fonctions permettant d'indiquer des « correspondances » entre termes [Xu et al, 2003, He et al, 2003].

**Matchers (Les méthodes d'alignement) :** Un matcher est une fonction calculant la distance entre deux entités en utilisant par exemple des mesures de similarités. Les matchers présentés par l'ensemble de fonctions (e.g. les fonctions similarités/dissimilarités) composent le processus de matching[Elbyed M., 2009].

**Fusion d'ontologies :** La fusion des ontologies est un processus rassemblant en une seule ontologie les connaissances de deux ou plusieurs ontologies sources existantes et différentes décrivant le même sujet ou appartenant au même domaine d'application. L'ontologie engendrée contient les informations de toutes les ontologies sources.

**Intégration d'ontologies :** L'intégration d'ontologies permet de construire une nouvelle ontologie qui n'est pas destinée à remplacer les autres (les ontologies originales peuvent être utilisées par ailleurs, mises à jour, ou modifiée, etc.).

## **II.2. Problèmes d'appariement**

Le processus d'appariement a été généré pour résoudre les diverses insuffisances et problèmes d'hétérogénéité et interopérabilité des ontologies comme il sera détaillé dans les sections (II.2.1, II.2.2), respectivement.

### **II.2.1. Niveaux d'hétérogénéités entre les ontologies**

Une ontologie est une conceptualisation des objets existant dans un domaine, de leurs propriétés et des relations reliant ces objets. La structure d'une ontologie représente les connaissances d'un domaine sous format informatique utilisable par les différentes applications. L'ontologie est devenue un élément primordial dans toutes les applications basées sur des connaissances.

Cependant, la représentation explicite des connaissances sous forme d'une ontologie dépend toujours d'assumptions implicites comme les objectifs des concepteurs, leurs connaissances remettant l'objectivité de la représentation d'un objectif difficile à réaliser. Ces connaissances implicites ont engendré diverses formes d'hétérogénéités entre les ontologies, même entre les ontologies décrivant le même domaine.

Il existe plusieurs classifications d'hétérogénéité entre les ontologies [Giunchiglia et Walsh., 1992]; [Benerecetti et al., 2000]; [Klein et al, 2001]; [Euzenat, 2001]; [Corcho, 2004]; [Hameed et al., 2004]; [Ghidini et Giunchiglia., 2004]. Dans ce qui suit, nous présentons une classification [Euzenat et al., 2007] qui récapitule les principales formes d'hétérogénéités en quatre niveaux, à savoir :

*1. Le niveau syntaxique* : Il correspond à toutes les formes d'hétérogénéité relatives au choix du format de représentation. En effet, il existe différents langages représentant les ontologies (OWL, KIF, RDFS ...) et chaque langage est basé sur une syntaxe distincte.

*2. Le niveau terminologique* : ce niveau intervient pour affecter des noms aux entités (classes, propriétés, relations ...) constituant une ontologie. Nommer une entité revient à lui raccorder un objet linguistique à partir d'un langage public. Des exemples de ce type d'hétérogénéité :

- Différents noms utilisés pour indiquer une même entité (synonymie)
- Le même nom utilisé pour désigner deux entités différentes (polysémie)
- Au contraire, le même terme peut symboliser différents concepts ; l'homonymie est un problème qui requiert bien souvent l'intervention humaine.
- Des mots provenant de distinctes langues (Français, Anglais, Italien ...), utilisés pour définir une même entité.
- Des variations syntaxiques du même mot (distinctes prononciations, abréviations, utilisation des préfixes et des suffixes ...)
- Finalement, l'encodage des données au sein de l'ontologie distingue bien souvent que ce soit pour les dates, les unités (monnaie, distances, ...).

*3. Le niveau conceptuel*: Les différences à ce niveau peuvent être restreintes en trois aspects :

- *La couverture* : la diversité entre deux ontologies peut être au niveau de la portée de la couverture du domaine décrit. Elles peuvent couvrir des parties distinctes (du monde réel ou d'un domaine) ou des parties qui se chevauchent, par exemple: une ontologie sur le sport couvre le sport de la course de chevaux qu'une autre ignorerait complètement.

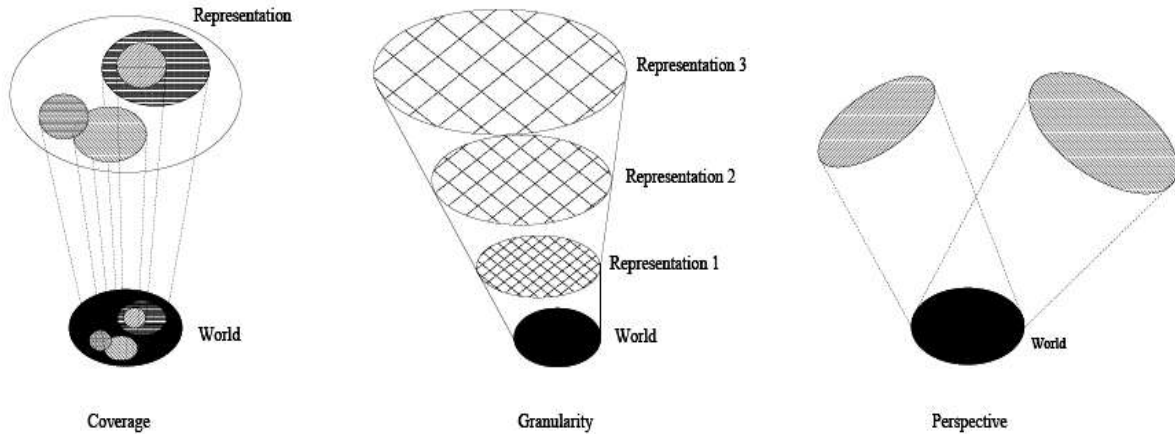


Figure II.1. Dimensions de l'hétérogénéité au niveau conceptuel [Euzenat et Shvaiko., 2007]

- *La granularité* : deux ontologies peuvent décrire les mêmes entités avec des niveaux de détail distincts, par exemple : une ontologie concernée par la comptabilité va considérer le concept générique du document alors qu'une ontologie décrivant le domaine des bibliothèques va différencier entre les différents types de documents : romans, biographies, manuscrits ...
- *Le niveau perspective* : un domaine peut être décrit par deux ontologies selon deux points de vue différents. Par exemple : le concept de la chaleur chez un Canadien sera forcément différent du même concept chez un Nigérien.

La figure II.1 est une description graphique de ces trois dimensions, à travers lesquelles une ontologie peut distinguer d'une autre ontologie au niveau conceptuel :

4. *Le niveau sémiotique ou pragmatique*: ce type d'hétérogénéité intervient lorsqu'une ontologie est interprétée par différentes personnes ou différentes communautés. Ces diverses interprétations sont souvent liées au choix du formalisme de représentation des connaissances, comme les clauses de la logique du premier ordre et les représentations hiérarchiques des classes, sont-ils équivalents, est-ce qu'ils véhiculent la même connaissance ?

En résumé, comprendre les distinctes formes d'hétérogénéité des ontologies est fondamental pour la réussite de l'alignement de ces dernières, car il est très critique de réaliser

un alignement entre des entités, en se basant uniquement sur les liens sémantiques, les autres dimensions doivent être prises en compte, par exemple dans la figure II.2: [zanobini, 2003].

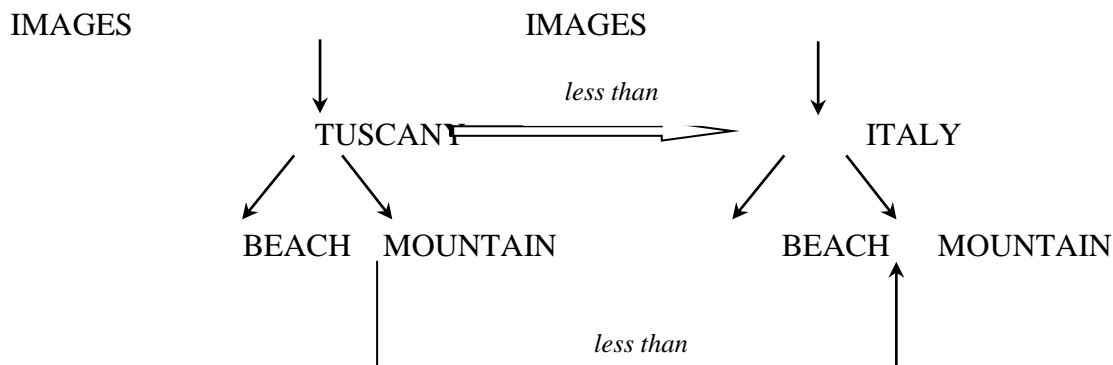


Figure II.2. Exemple de relations sémantiques [zanobini, 2003]

le concept « Mountain » dans la première hiérarchie décrit les images de montagnes en toscane alors que dans la seconde hiérarchie le concept « Mountain » présente les images de montagnes en Italie. Cela ne signifie pas qu'ils ne sont pas liés, ils ne sont pas équivalents mais, en tenant compte de la structure, il semble que le premier concept est moins général que le second (la divergence, ici, est d'ordre conceptuelle parce qu'elle est liée à la couverture).

Ces hétérogénéités ont rendu le choix d'une ontologie spécifique ou l'exploitation de plusieurs d'entre elles une tâche fastidieuse, ce qui requiert une comparaison entre eux afin de passer de l'une à l'autre ou de les intégrer, en générant le plus automatiquement possible des relations ou appariements entre les concepts de deux ontologies, comme il sera illustré dans la prochaine section (II.2.2).

### II.2.2. Modes de l'interopérabilité sémantique

Chaque organisation peut concevoir sa (ses) propre(s) ontologie(s), soit en produisant une nouvelle ontologie, soit en réutilisant des ontologies existantes. Pour trouver un accord entre toutes ces ontologies, il faut donc assurer l'interopérabilité sémantique qui est définie dans la littérature en trois modes ([Wache et al, 2002], [Noy et al, 2000],[Bouquet et al, 2003]) : le mode intégré, le mode fédéré et le mode unifié.

### II.2.2.1. Mode intégré

Il est aussi nommé «médiation centralisée», «ontologie globale» ou «ontologie unique ou simple», consiste à s'accorder sur une ontologie unique à laquelle toutes les sources d'information sont reliées (Figure II.3). Cela revient souvent à fusionner les ontologies existantes en une seule. Ainsi, il impose un consensus sur le vocabulaire utilisé, sa sémantique et syntaxe, et la granularité de l'ontologie, etc.

Ce type est utile lorsqu'on dispose d'une ontologie globale offrant un vocabulaire partagé pour la spécification sémantique, ou lorsqu'il existe plusieurs ontologies pour un même domaine avec une granularité proche qui sont réalisées dans une même optique. Elle n'est pas faisable quand l'une des ontologies qu'on veut intégrer contient des données hétérogènes et/ou évolue de façon indépendante : on doit alors modifier régulièrement l'ontologie globale, et les autres sources de données.

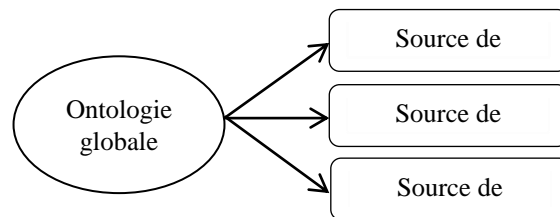


Figure II.3 Mode intégré [Elbyed M., 2009]

### II.2.2.2 Mode Fédéré

Parfois nommé «médiation décentralisée», «médiation distribuée» ou «ontologies multiples». Dans ce mode, chaque source d'information peut être décrite par diverses ontologies selon le point de vue d'une communauté ou d'un individu [Bouquet et al, 2003], et à les faire correspondre l'une à l'autre (Figure II.4).

L'avantage ici, est que chaque ontologie source peut être définie indépendamment des autres sources ou des autres ontologies. De plus, l'évolution autonome et les mises à jour fréquentes des ontologies ont offert une plus grande flexibilité à ce mode. Cependant l'absence d'un vocabulaire commun conduit à une grande difficulté au moment de comparaison des différentes ontologies sources. Pour pallier ce problème, un formalisme de représentation additionnelle représentant le mapping inter-ontologies est fourni. Ce dernier permet d'identifier sémantiquement les termes correspondants aux différentes ontologies

sources. Ce mapping est difficile à déduire à cause de nombreux problèmes tels que l'hétérogénéité sémantique, la synonymie et l'homonymie mais surtout l'ambiguïté due à un manque d'information [Doan, 2002]. La plupart des systèmes de mapping existants jusqu'à présent dans la littérature requièrent une grande intervention des experts du domaine. Nous nous apportons la lumière dans le cadre de ce travail de thèse à cette problématique.

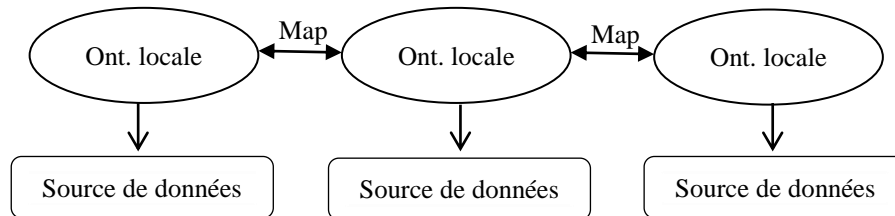


Figure II.4. Mode fédéré.

### II.2.2.3 Mode Unifié

Ce mode est également appelé «hybride» ou «ontologie de plus haut niveau», est un compromis entre les deux modes précédents, consiste à établir des correspondances entre les ontologies locales (une ontologie par source), et établir des correspondances entre chacune d'elles et une ontologie de plus haut niveau nommée globale ou complémentaire (figure II.5).

Ce mode est intéressant quand les ontologies se conforment à un certain standard ; ainsi, les ontologies locales soient décrites à l'aide d'un vocabulaire commun (qui peut être une ontologie) comprenant les termes fondamentales du domaine : de nouveaux termes, plus complexes pourront être produits à partir de la combinaison des premiers, l'aide d'opérateurs spécifiques. Ce mode nécessite cependant de commencer par concevoir un vocabulaire commun, ainsi que des règles de combinaison des termes.

L'avantage de ce mode est que de nouvelles sources peuvent être insérées sans avoir besoin de modifier le vocabulaire commun. Il maintient ainsi l'acquisition et l'évolution des ontologies. De plus, il faut un consensus au préalable, ou construire soi-même les diverses ontologies correspondantes aux différentes sources.

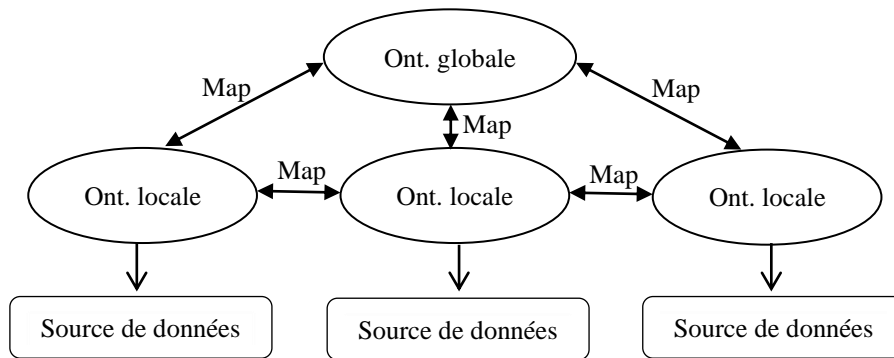


Figure II.5 Mode Unifié

### II.3. Processus d'appariement (Matching)

Le processus d'appariement (matching) exprime l'alignement des termes des ontologies à appairer. L'alignement consiste à mettre en correspondance sémantique les entités des structures [Euzenat et al., 2007] en mesurant la similarité/dissimilarité entre eux. Ces structures peuvent être des schémas XML, des bases de données, ou des ontologies. Les liens permettant d'assurer les correspondances sémantiques comprennent les relations d'équivalence ( $\equiv$ ), de généralisation/spécialisation ( $\supseteq, \subseteq$ ), de chevauchement ( $\cap$ ) ou encore d'incompatibilité ( $\perp$ ). L'évaluation de ces liens peut être booléenne ou par le biais des autres mesures telles que : les mesures de similarité, les mesures symboliques....

Le présent travail, se situe dans le cadre de l'alignement et l'appariement des ontologies qui se différencie à plusieurs égards de l'alignement des schémas. En effet, une description détaillée du processus de l'alignement, et des méthodes utilisées dans la littérature pour mesurer la similarité/dissimilarité telles que: les méthodes terminologiques, structurelles, extensionnelle, sémantiques ou probabiliste auront lieu dans la section (II.3.2.).

#### II.3.1 Description du processus d'alignement des ontologies

Le processus d'alignement est une fonction  $f$ , qui à partir d'une paire d'ontologies, un alignement en entrée (optionnel), un ensemble de paramètres (ex : paramètres de pondération, seuils ...) et un ensemble de ressources externes (ex : thésaurus, lexique, etc.), détermine un alignement entre les entités de ces deux ontologies. L'alignement peut avoir plusieurs cardinalités: 1:1(one to one), 1:m (one-to-many), n:1 (many-to-one) or n:m (many to many). La fonction  $f$  est définie par le 4-tuple  $\langle id, e_1, e_2, r \rangle$  [Euzenat, 2004], où :

- $id$  : est un identifiant unique désignant chaque correspondance découverte ;

- $(e_1, e_2)$  : ce sont les entrées du processus de matcher, définissent les entités des deux ontologies à appairer (e.g., classes, propriétés) ;
- $r$  : c'est la sortie du processus d'alignement, elle présente la relation qui peut être entre les entités (e.g. équivalence) des ontologies à appairer.

### II.3.2. Techniques d'appariement

Les différentes techniques de comparaison utilisées dans le processus de matching, présentées dans cette partie et illustrées dans la figure II.6, sont extraites de [Euzenat et Shvaiko, 2013]. Ces techniques sont organisées selon le type d'entités à comparer et la méthode utilisée pour évaluer la similarité qui sera définie dans la prochaine partie (II.3.2.1).

#### II.3.2.1. Similarité

Les domaines de la psychologie et des mathématiques ont défini la similarité de différentes manières [Lê, 2006]. Dans notre document la similarité optée est vue comme celle de la similarité topologique en mathématiques. Cette similarité est une fonction dont la valeur est plus grande quand les deux points sont plus proches (contrairement à la distance, qui est définie comme une mesure de dis-similarité : plus les points sont proches, plus la distance est petite). La notion de cette fonction peut être changée selon les approches et les propriétés désirées. La valeur de cette fonction est généralement comprise entre 0 et 1, ce qui admet des éventualités d'interprétation probabiliste de la similarité.

La similarité utilisée dans notre contexte est appelée similarité sémantique. Elle est définie en associant à des documents des termes ou des entités une métrique fondée sur la similitude de leurs significations ou de leurs contenus sémantiques. La similarité est la quantité qui reflète la puissance du rapport entre deux objets ou deux caractéristiques. Effectivement, ceci peut être effectué en déterminant une similarité topologique, ou en servant des ontologies pour définir une distance entre les termes [Lê, 2006].

Il y a d'autres notions qui ont parfois utilisé au lieu de la similarité telle que la dis-similarité et la distance. La distance mesure la dis-similarité de deux entités, elle est opposée de la similarité : si la valeur de la fonction de similarité de deux entités est grande, la distance entre elles, est petite et vice-versa. Les valeurs de similarité et dis-similarité notées  $S$  et  $DS$

respectivement, sont souvent intégrées dans des formules plus complexes ce qui a nécessité une normalisation de ces valeurs et leurs sommes est  $S + DS = 1$ .

La similarité, la dis-similarité, la distance et la normalisation ont défini dans [Euzenat et al., 2004] comme suit.

**Définition 1 (Similarité).** La similarité appliquée sur les ontologies  $O$ ,  $S : O \times O \rightarrow R$  est une fonction d'une paire d'entités à un nombre réel formulant la similarité entre ces deux entités, telle que:

- $\forall a, b \in O, S(a, b) \geq 0$  (positivité)
- $\forall a, b, c \in O, S(a, a) \geq S(b, c)$  et  $S(a, a) = S(a, b) \Leftrightarrow a = b$  (autosimilarité ou maximalité)
- $\forall a, b \in O, S(a, b) = S(b, a)$  (symétrie)
- $\forall a, b, c \in O, S(a, b) = S(b, c) \Rightarrow S(a, b) = S(a, c)$  (transitivité)
- $\forall a, b \in O, S(a, b) < \infty$

**Définition 2 (Dis-similarité).** La dis-similarité  $DS : O \times O \rightarrow R$  est définie comme une fonction d'une paire d'entités à un nombre réel formulant la dis-similarité entre ces deux entités, telle que:

- $\forall a, b \in O, DS(a, b) \geq 0$  (positivité)
- $\forall a, b, c \in O, DS(a, a) \leq DS(b, c)$  et  $DS(a, a) = 0$  (minimalité)
- $\forall a, b \in O, DS(a, b) = DS(b, a)$  (symétrie)
- $\forall a, b \in O, DS(a, b) < \infty$  (finitude)

**Définition 3 (Distance).** La distance  $D : O \times O \rightarrow R$ , est définie comme une fonction de la dis-similarité assurant la définitivité et l'inégalité triangulaire :

- $\forall a, b \in O, D(a, b) = 0 \Leftrightarrow a = b$  (définitivité)
- $\forall a, b, c \in O, D(a, b) + D(b, c) \geq D(a, c)$  (inégalité triangulaire)

**Définition 4 (Normalisation).** Une mesure est appelée une mesure normalisée si les valeurs calculées par cette mesure appartiennent à un intervalle de 0 à 1. Ces valeurs calculées sont des valeurs normalisées. Les fonctions du calcul sont des fonctions normalisées et notées  $f$ .

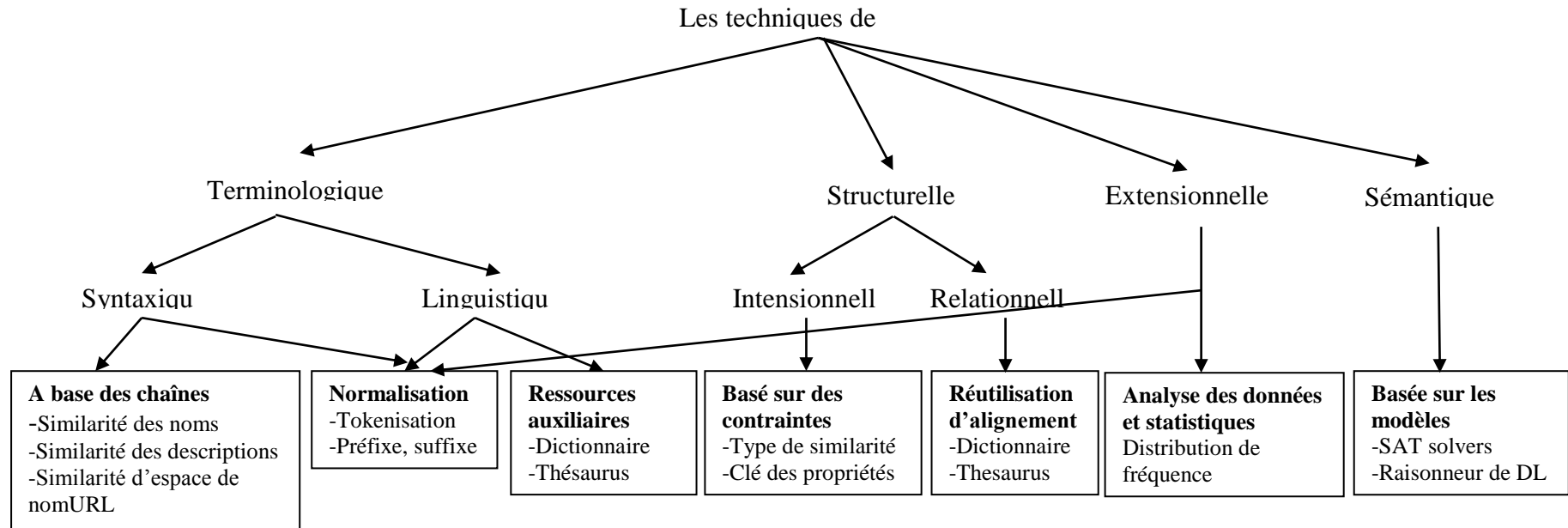


Figure II.6 Techniques de matching

### **II.3.2.2. Méthodes terminologiques**

Ces méthodes consistent à calculer la valeur de similarité entre les entités textuelles, telles que les noms, les étiquettes, les commentaires, etc. Elles peuvent encore être divisées en deux approches: l'une contient des méthodes comparant les termes en se basant sur les caractères contenus dans ces termes, appelée l'approche syntaxique et l'autre utilise certaines connaissances, nommée l'approche lexicale ou linguistique.

#### **II.3.2.2.1. Approche syntaxique**

L'approche syntaxique réalise la correspondance à travers les mesures de dis-similarité des chaînes de caractères. Ces mesures analysent la structure des chaînes de caractères, l'ordre des caractères dans la chaîne, et la fréquence d'apparitions d'un caractère dans cette chaîne ; et ne prennent pas en compte la signification des termes. Par exemple quand on compare les deux termes « appareil » et « Appareil », une grande valeur de similarité sera retournée (jusqu'à 1), mais une petite valeur, voire la valeur 0 sera le résultat de la comparaison des deux termes « appareil » et « machine ».

Les chaînes fournies aux formules calculant la similarité, ont été traitées par la normalisation textuelle du domaine de traitement automatique de la langue naturelle (TALN), qui diffère de la normalisation définie dans la sous-section précédente. Cette normalisation a trois types, le premier est appelé normalisation des caractères, il convertit toutes les lettres majuscules dans une chaîne de caractères en leurs formes minuscules ou vice-versa. Par exemple, la chaîne de caractères « MachineS » sera convertie à « machine » et ensuite, elle est examinée comme égale à l'autre chaîne de caractères « machines ».

Le second type est nommé normalisation des espaces. Il consiste à substituer toutes les séquences consécutives des espaces, des tabulations, des retours de chariot (les caractères CR) aperçues dans une chaîne de caractères par un seul caractère d'espace. Par exemple, l'expression « La machine » est normalisée à « la machine ».

Suppression des signes diacritiques ou des accents (aigus, graves...) représentant le dernier type de normalisation, substitue des caractères avec des signes diacritiques par caractères correspondants sans signes diacritiques. Par exemple, le mot « Saïda » est remplacé par le mot « Saida » sans changer la signification du mot, une ville algérienne. Cependant, certaines suppressions modifieront la signification du terme : « là » (adverbe de lieu) et « la »

(article). La suppression des chiffres, élimination des ponctuations, élimination des mots vides (les mots contenant peu d'informations tels que : « est », « un » ...), suppression des affixes (préfixes, suffixes), extension des abréviations, tokenisation, et lemmatisation (passer au singulier, à l'infinifitif pour les verbes, au masculin pour les adjectifs...), sont aussi des normalisations de troisième type [Lê, 2006].

Dans la littérature, il existe diverses mesures calculant la valeur de similarité ou la distance entre deux chaînes de caractères comme la similarité de Jaccard, la distance de Hamming, la distance de Levenshtein... Certaines sont implémentées en Java<sup>6</sup>, ou en C<sup>7</sup>. [Lê, 2006].

### II.3.2.2.2. Approche linguistique

Cette approche réalise la correspondance entre les termes en utilisant les méthodes linguistiques intrinsèques ou extrinsèques. Ces méthodes se fondent principalement sur les propriétés expressives et productives de la langue naturelle [Maynardet Ananiadou, 1999].

#### *Méthodes intrinsèques*

Les méthodes intrinsèques peuvent exploiter des propriétés linguistiques internes des termes telles que les propriétés morphologiques ou syntaxiques (e.g synonymie, hyponymie, etc.). Par exemple, le Tableau II.1 (extrait de [Euzenat et al., 2004]) illustre des variantes possibles du terme « enzyme activity » qui peut être un label d'une entité référencée par plusieurs termes (synonymie) ou par plusieurs variantes d'un même terme.

Tableau II.1 Variantes du terme « enzyme activity » (extrait de [Euzenat et al., 2004])

Type	Sous-type	Exemple
Morphologique	Inflexion Dérivation Flexionnel-Dérivationnel	enzyme activities enzymatic activity enzymatic activities
Syntaxique	Insertion Permutation Coordination	enzyme amidolytic activity activity of enzyme enzyme and bactericidal activity

<sup>6</sup> <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>

<sup>7</sup> [http://www.monkey.org/~jose/software/libdistance/distance\\_3.html](http://www.monkey.org/~jose/software/libdistance/distance_3.html)

Morpho-syntaxiques	Dérivation-Coordination Inflexion- Permutation	enzymatic and bactericidal activity activity of enzymes
Sémantique		fermentation
Multilingue	French Vietnamien	activité d'enzyme sf lèn men

Ces méthodes exploitent les variantes linguistiques (lexème) pour chercher la forme canonique ou représentative d'un mot ou d'un terme (lemme). Par exemple, la comparaison de deux mots « ran » et « running » par la mesure de similarité donne un 0 (c-a-d. ils sont dissemblables), mais la valeur de la même mesure pour les lemmes de ces mots sera égale à 1, ce qui implique que « ran » et « running » sont similaires [Lê, 2006].

La recherche du lemme d'un mot peut être réalisée à l'aide d'un dictionnaire, d'un stemmer ou d'une lemmatisation. Le stemmer est un programme automatique, léger et efficace, détermine la forme radicale à partir d'une forme infléchie ou dérivée d'un mot donné. Les radicaux (stems) découverts par les stemmers ne sont pas toujours similaires à la racine morphologique du mot. Ce programme assure l'association des mots similaires à un même radical, même si ce radical n'est pas une racine de mot valide. Un stemmer pour le français, par exemple, la racine "mainten" est associées aux chaînes de caractères «maintenaient», « maintenait », « maintenant », ou « maintenir » [Lê, 2006].

Julie Beth Lovins a écrit et publié en 1968 le premier stemmer [Lovins, 1968]. Ensuite en 1980, Martin Porter a développé un autre [Porter, 1980]. Ce dernier processus a connu un grand succès, et est devenu l'algorithme standard exploité dans la langue anglaise pour chercher des radicaux. Il est implémenté dans divers langages de programmation. Le framework Snowball<sup>8</sup> permet l'implémentation des algorithmes de stemmers, leurs variantes ou bien des algorithmes de stemmers pour d'autres langues (par exemple le russe, le danois l'allemand, le français...), est ainsi créé par Porter.

La lemmatisation est un algorithme plus complexe pour déterminer le radical exact d'un mot. Elle consiste à définir la partie de discours (catégorie lexicologique) d'un mot, et l'application des règles de normalisation différentes pour chaque partie du discours. Cette

<sup>8</sup> <http://snowball.tartarus.org/>

approche exige la compréhension de la grammaire d'une langue, des règles différentes... Elle est donc compliquée et difficile à implémenter [Lê, 2006].

### *Méthodes extrinsèques*

Ces méthodes exploitent des ressources externes telles que des vocabulaires, des dictionnaires ou WordNet<sup>9</sup> afin d'évaluer la similarité entre deux termes. La similarité est déduite à l'aide des liens sémantiques déjà existant dans ces ressources externes comme des liens synonymes (pour l'équivalence), des liens hyponymes/ hypernymes (pour la subsomption). Par exemple, grâce aux ressources des synonymes, « appareil » et « machine » sont similaires. Typiquement, un système lexicologique est utilisé dans WordNet pour trouver des relations telles que la synonymie entre des termes. Ainsi, les liens sémantiques existant dans WordNet ont été exploités pour évaluer la distance sémantique entre ces termes et déduire l'existence d'une relation entre eux [Lê, 2006].

### **II.3.2.3. Méthodes structurelles**

Ces méthodes se composent de méthodes structurelles internes et externes, les premières déduisent la similarité de deux entités en comparant les structures internes des entités (e.g., intervalle de valeur, cardinalité d'attributs, etc.). Les méthodes structurelles externes exploitent la structure externe des entités (e.g. leurs pères, fils, etc.) afin de les aligner.

### *Méthodes structurelles internes*

Les informations des structures internes ont été exploitées par ces méthodes pour évaluer la similarité entre deux entités. Ces informations concernent les attributs de l'entité, telles que les informations du co-domaine des attributs, la cardinalité des attributs, les caractéristiques des attributs (la transitivité, la symétrie), ou les autres types de restriction sur les attributs [Lê, 2006].

Plusieurs méthodes basées sur les contraintes à propos de ces éléments ont été proposées pour évaluer la similarité entre deux éléments de deux schémas de base de données. L'algorithme Cupid [Madhavan et al., 2001] est l'une des méthodes offertes dans ce sens,

---

<sup>9</sup> <http://wordnet.princeton.edu/>

fondé sur la compatibilité des attributs, des types de données pour chercher des correspondances entre des éléments.

L'approche de [Valtchev, 1999] est une autre méthode qui propose une mesure de similarité entre deux types de données. Cette mesure se fonde sur la différence entre deux tailles de types, la taille d'un type définie comme la cardinalité de l'ensemble de valeurs qu'il développe, et la taille de leur généralisation commune dont la description dépend du type (il est défini par un ensemble pour les types énumérés, et un intervalle pour les types ordonnés...).

### *Méthodes structurelles externes*

Les méthodes structurelles externes sont fondées sur les relations entre les entités elles-mêmes, qui sont généralement des relations de subsomption (is\_a ou spécialisation) ou de méréologie (part-whole). Ces relations permettent de considérer les entités dans des hiérarchies et la similarité entre elles est déduite de l'analyse de leurs emplacements dans ces hiérarchies. L'idée fondamentale de cette approche est que si deux entités sont similaires, leurs voisines pourraient également être d'une façon ou d'une autre similaires. Cette proposition peut être utilisée de diverses façons en regardant les relations avec d'autres entités dans des hiérarchies [Euzenat et Shvaiko, 2007]. Deux entités peuvent être vues comme similaires si :

- Leurs super-entités directes (ou toutes leurs super-entités) sont similaires ;
- Leurs sœurs (ou toutes leurs sœurs, qui sont les entités possédant la même super entité directe avec les entités en question) sont auparavant similaires;
- Leurs sous-entités directes (ou toutes leurs sous-entités) sont auparavant similaires;
- Leurs descendants (entités dans le sous-arbre disposant pour racine d'entité en question) sont déjà similaires.
- Toutes (ou presque toutes) leurs feuilles (les entités de même type, qui n'ont aucune sous-entité, dans le sous-arbre disposant pour racine l'entité en question) sont aussi similaires.
- Toutes (ou presque toutes) les entités dans les chemins de la racine aux entités en question sont aussi similaires.

En fait, l'application de ces méthodes sur des hiérarchies différentes au niveau de granularité peut retourner des faux résultats. Par exemple, l'entité « Personne » dans une hiérarchie peut avoir deux sous-entités « Enfant » et « Adulte », mais dans une autre hiérarchie, la même entité « Personne » peut être divisée en deux autres sous-entités « Femme » et « Homme », la déduction que « Enfant » et « Femme » ou « Enfant » et « Homme » sont similaires, est fautive dans tous les cas.

Plusieurs mesures ont été proposées dans cette approche pour comparer les entités des hiérarchies à mettre en correspondance, telles que la mesure récursive de la dis-similarité topologique structurelle qui se fonde sur la distance de chemin le plus court dans un graphe [Valtchev et Euzenat, 1997]. En outre, la notion « upward cotopy » notée par  $UC(c, H) = \{c' \in H ; c \leq c'\}$ , définie l'ensemble d'entités qui sont super-entités de l'entité  $c$  dans la hiérarchie  $H$ , et permet de déduire une mesure de similarité inspirée de la distance de Jaccard [Mädche et Staab, 2002].

### II.3.2.4. Méthodes extensionnelles

Ces méthodes exploitent les extensions des entités définies par leurs ensembles des instances afin de découvrir la similarité entre eux.

Les distances de Hamming ou de Jaccard adaptées pour les ensembles des instances, présentées dans les définitions dans [Lê, 2006], ont été exploitées pour traiter le cas où les ensembles des instances partagent une partie commune. Principalement, la mesure de Hamming compte le nombre d'éléments distincts entre deux ensembles à comparer et la mesure de Jaccard donne le rapport entre l'intersection des ensembles et leur union.

Ces deux mesures fournissent la similarité de deux entités basée sur la similarité entre les deux ensembles de leurs instances. Cette similarité se focalise sur la comparaison exacte des éléments dans deux ensembles. Cependant, ces mesures ne sont plus applicables dans le cas où les ensembles des instances ne partagent aucune partie commune, (le résultat retourné aura la valeur 1, c.-à-d. les entités à comparer sont toujours différentes).

Une autre mesure est définie dans [Valtchev, 1999] où la similarité entre deux ensembles est la similarité moyenne des paires des éléments dans Pairing. Ce dernier est défini par l'ensemble de correspondances ayant la somme maximale de toutes les similarités des paires

dans l'ensemble. L'évaluation de l'ensemble  $\text{Pairing}(S, T)$  est une optimisation qui maximise le total des similarités des paires des éléments de Source (S) et Target(T).

Une technique d'analyse multidimensionnelle dans le domaine de la statistique a permis d'évaluer la similarité entre deux ensembles [Cox, 1994]. Cette technique propose que si deux entités ont des distances très similaires à toutes autres entités, elles doivent être très similaires. Les entités dans des ensembles sont indiquées par des vecteurs dont la valeur d'une dimension est représentée par la similarité de l'entité en question avec une autre entité dans les deux ensembles. La similarité entre deux ensembles est la valeur du cosinus de deux vecteurs moyens de ces deux ensembles.

### **II.3.2.5. Méthodes sémantiques**

Les méthodes sémantiques permettent d'assurer la complétude, la cohérence et/ou l'incohérence de l'alignement fourni par les techniques discutées auparavant. Par conséquent, l'entrée de ces méthodes doit être des alignements qui peuvent être fournis par des matchers linguistiques, syntaxiques et structurels. Les matchers sémantiques produisent de nouvelles relations de mapping à partir de ces alignements, et extraient le maximum de relations de correspondance existant entre les ontologies à faire correspondre, tout en utilisant les relations sémantiques produites par lui-même.

L'approche sémantique peut comparer les interprétations des entités en basant sur des modèles de la logique de description tels que la satisfiabilité propositionnelle (SAT). Ainsi, les techniques de déduction et d'inférence peuvent être exploitées pour déduire les relations de matching entre entités.

#### ***Techniques de satisfiabilité propositionnelles***

Les techniques de satisfiabilité propositionnelle peuvent être exploitées pour examiner la validité d'un ensemble de formules propositionnelles qui est construit en traduisant des relations auparavant connues et des relations à examiner entre des entités vers des formules propositionnelles [Giunchiglia et al., 2004; Giunchiglia et Shvaiko, 2003; Bouquet et al., 2003].

[Giunchiglia and Shvaiko, 2003; Shvaiko ,2006] ont proposé d'appliquer les techniques de SAT suivant les étapes en dessous :

1. Construire une théorie ou un domaine de connaissances (axiomes) pour les ontologies d'entrée sous forme d'une conjonction des axiomes disponibles. La théorie peut être construite en utilisant le mapping fourni par les précédentes techniques de matching.
2. Mettre les axiomes de cette théorie sous la forme suivante:

$$axiome \Rightarrow r(c, c')$$

Où  $c, c'$  sont des concepts des ontologies à aligner, et  $r$  est la relation d'alignement offerte par les autres techniques de matching. Cette dernière peut être représentée par : l'équivalence ( $\equiv$ ), subsumption ( $\subseteq, \supseteq$ ), ou la disjonction ( $\perp$ ).

3. Les techniques de satisfiabilité propositionnelle exploitées dans cet algorithme assument qu'une formule est vrai si et seulement si toutes ses propositions sont vrai, et une formule propositionnelle est valable si et seulement si sa négation est insatisfiable. Ces techniques peuvent être ainsi utilisées pour faire une vérification complète de toutes les correspondances possibles en assurant un calcul de la fermeture déductive de l'alignement initial.

### *Techniques de déduction de la logique de description*

Le principe de ces techniques est basé sur la proposition suivante : deux concepts peuvent être similaires si leurs voisinages sémantiques le sont, sachant que le voisinage sémantique d'un concept est l'ensemble des concepts avec lesquels il est lié par une relation sémantique dans une même ontologie [Elbyed M., 2009]. Ces concepts similaires se trouvent dans la base de mapping fourni par les autres méthodes d'alignement et seront enrichis par des méthodes déductives fondées sur les techniques de la logique de description.

Le voisinage d'un concept permet de définir son contexte et comprendre son interprétation. En outre, les techniques de la logique de description (telles que le test de subsumption) peuvent être exploitées pour évaluer des relations sémantiques entre ces concepts, telles que l'équivalence (la similarité est égale à 1 et symbolisée par " $\equiv$ "), la subsumption (la similarité est de 0 à 1 et notée par " $\subseteq$ " ou " $\supseteq$ ") ou l'exclusion (la similarité est égale à 0 et notée par " $\perp$ "), et admettent donc de déduire la similarité de deux concepts [Lê, 2006].

Afin de mieux comprendre cette technique on utilise l'exemple en dessous extrait de [Euzenat et Shvaiko, 2007]. Les concepts *Micro-company* et *SME* des ontologies à aligner ont été présentés en logique de description par les deux formules suivantes:

$$\begin{aligned} \textit{Micro} - \textit{company} &\equiv \textit{company} \sqcap_{\leq 5} \textit{employée} \\ \textit{SME} &\equiv \textit{Firm} \sqcap_{\leq 10} \textit{associée} \end{aligned}$$

La première signifie qu'une *Micro-company* est une *company* dont le nombre d'employées ne doit pas dépasser 5.

La deuxième définit *SME* comme une *Firm* dont le nombre maximum des associées est 10.

Les écrivains de cet exemple ont supposé que les alignements initiaux offerts par les autres techniques de matching, comportent les formules suivantes:

$$\begin{aligned} \textit{Company} &\equiv \textit{Firm} \\ \textit{associée} &\subseteq \textit{employée} \end{aligned}$$

Ils expriment qu'une *Company* est l'équivalent d'une *Firm*, et *associée* est une sous propriété de *employée*. Ce qui implique que *Micro - company* est une sous classe de *SME*:

$$\textit{Micro} - \textit{company} \subseteq \textit{SME}$$

### II.3.2.6. Méthodes probabilistes

Les deux techniques probabilistes, présentées dans cette thèse ont été bien détaillées dans [Euzenat et Shvaiko, 2013]. Ces techniques peuvent améliorer les résultats des mesures de similarités ou d'alignements offerts en entrée. L'une de ces techniques peut affecter une interprétation probabiliste à l'alignement, et l'autre exploite l'interprétation sémantique des alignements et des ontologies.

#### *Réseau bayésien*

Le réseau bayésien est une méthode probabiliste pour modéliser les causes et les effets. Il est composé d'un graphe acyclique orienté contenant des nœuds et des arcs, et des tables de probabilité conditionnelle. Ce réseau bayésien a été modélisé de plusieurs manières dans le domaine de matching des ontologies. Par exemple, les deux ontologies peuvent être

modélisées en deux réseaux bayésiens et le matching entre ces derniers est un processus d'inférence de preuve entre ces deux réseaux [Pan et al. 2005]. Dans [Mitra et al. 2005] ce réseau est exploité pour améliorer les résultats de matching en dérivant des alignements oubliés.

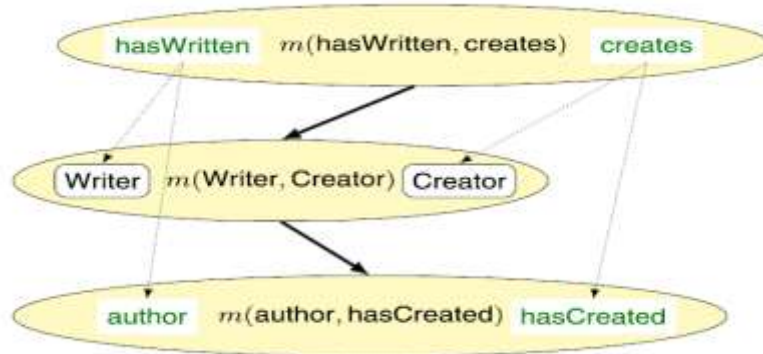


Figure II.7. Exemple de graphe d'un réseau bayésien [Euzenat et Shvaiko, 2007]

[Mitra et al. 2005] ont proposé de représenter les alignements entre les paires de classes ou de propriétés par des nœuds (variables) dans un graphe (voir la figure II.7) de réseau bayésien, et l'influence d'un nœud sur un/des autres nœud(s) est modélisée par des arcs. Ces arcs ont été illustrés par des flèches solides dans ce graphe, par contre les flèches en pointillés exposent les relations entre les entités dans les ontologies en question. Par exemple, la correspondance entre les propriétés  $hasWritten \in O_1$  et  $creates \in O_2$ , affecte la correspondance entre les concepts qui ont le rang  $Writer \in O_1$  et  $Creator \in O_2$ , qui à leurs tour affecte la correspondance entre  $author \in O_1$  et  $hasCreated \in O_2$ . Les tables de correspondance conditionnelle ont été générées en exploitant les métarègles génériques comme la règle de probabilité de distribution dans laquelle le nœud fils est influencé par son père. Dans l'exemple de la figure II.7, si les deux concepts  $Writer$  et  $Creator$  appartiennent, et il y a une relation appelée  $hasWritten$  entre  $Writer$  et  $Text$  dans  $O_1$ , et ainsi une relation nommée  $author$  reliant  $Creator$  au  $Work$  dans  $O_2$  alors la probabilité d'appariement entre  $Text$  dans  $O_1$  et  $Work$  dans  $O_2$ , peut être incrémentée.

### Réseaux de Markov et les réseaux logiques de Markov

Les réseaux de Markov [Pearl 1988], et les réseaux logiques de Markov [Richardson and Domingos 2006] sont des réseaux probabilistes structurés exploités dans le domaine d'appariement des ontologies par [Albagli et al. 2012] et [Niepert et al. 2010] respectivement.

*Les réseaux de Markov* sont similaires aux réseaux bayésien, symbolisés par:  $N = \langle V, E \rangle$ . Les nœuds de ce réseau représentent les variables ou les appariements sont symbolisés par  $V$ , et les dépendances statistiques entre ces nœuds sont représentées par des arcs nommés  $E$ . Ces réseaux permettent des distributions de probabilité sur les événements, qui sont représentés par des variables. Contrairement aux réseaux bayésien, les réseaux de Markov sont non orientés. La distribution de probabilité dans ce réseau, est définie par les fonctions potentielles  $p$  sur les cliques  $C$ , c'est à dire des ensembles de nœuds, tels que chaque paire est connectée par au moins un bord. Les tables potentielles ( $P_c$ ) peuvent être associées par chaque sous graphe complet dans le réseau ce qui correspond aux probabilités conditionnelles dans les réseaux bayésien. La distribution conjointe des probabilités d'événements définies par les réseaux de Markov est le produit de tous les potentiels comme écrite dans l'équation II.1, sachant que  $Z$  est une constante normalisée, nommée fonction de partition.

$$P(N) = \frac{1}{Z} \prod_{C \in \text{cliques}(N)} P_c(C) \quad (\text{II.1})$$

*Les réseaux logiques de Markov* unifient les réseaux de Markov et la logique du premier ordre. La structure de graphe de ces réseaux est un ensemble de formules de la logique de premier ordre avec des poids. Le fondement de ces formules est obtenu en remplaçant chaque variable dans la formule par une constante. La possibilité qu'une formule soit vraie s'est élevée ainsi que son poids. Par conséquent, le poids représente les difficultés qui ont permis l'atteint de l'interprétation de la formule. Ce qui permet de traiter les cas de violation de formule en mettant une basse potentialité aux formules représentant des correspondances.

## II.4. Stratégies d'appariement

Les techniques de base exhibées dans la section II.3 ont représenté les fondements des stratégies d'alignement. L'alignement est déduit une fois que la similarité (ou la dissimilarité) entre les entités est évaluée. Cependant la déduction de l'alignement final requiert des traitements globaux tel qu'une composition d'alignement, une agrégation de similarité, un calcul global de similarité, des méthodes d'apprentissage et des méthodes probabilistes... [Euzenat et Shvaiko, 2013].

### II.4.1. Partitionnement des ontologies et la recherche d'un espace d'élagage

Le processus de matching peut traiter des ontologies contenant des dizaines, des centaines, ou des milliers d'entités. Afin de mieux appairier ces ontologies, il est possible de partitionner ces grandes ontologies en des petites ontologies, et ensuite aligner les différentes entités de ces ontologies (section II.4.1.1), ou ignorer des parties de ces ontologies en passant par le processus d'élagage qui sera expliqué dans la section II.4.1.2.

#### II.4.1.1. Partitionnement des ontologies

Il est fréquemment utile de partitionner les ontologies dans des parties afin de réaliser un appariement de chacune. Chaque partie est alignée indépendamment des autres parties et à la fin tous ces alignements vont être agrégés pour former le dernier résultat de match. Ce processus regroupe l'ensemble des entités dans des parties exhaustives et non chevauchées. En effet, certaines parties d'une ontologie peuvent être ignorées parce qu'ils n'ont pas des parties équivalentes dans l'autre ontologie, et des autres parties peuvent être appariées provisoirement avec plusieurs parties.

Le principe de regroupement est l'une des techniques de partitionnement inspiré du domaine de l'interconnexion traitant un nombre volumineux de données. Il est retenu par cet algorithme pour améliorer le processus de matching en évitant le produit cartésien au moment de comparaison de toutes les entités des ontologies. En effet le processus ne compare que les entités des parties regroupées. Techniquement, le regroupement est réalisé en faisant correspondre les entités à un index multidimensionnel dont les entités similaires sont positionnées les unes à côté des autres. Une ontologie peut être partitionnée en prenant en considération l'autre ontologie à l'aide de la technique de modularisation développée par [[Stuckenschmidt et al. 2009] et [Ghazvinian et al. 2011]].

Les techniques de partitionnement et d'élagage peuvent aussi exploiter la notion d'ancre. Ces ancres ont obtenu en recherchant les entités qui ont des nœuds bien appariés dans la fonction de hachage. Pour cela, au lieu de comparer tous les nœuds avec une complexité de  $O(n^2)$ , il est possible de construire un index de toutes les entités d'une même ontologie à l'aide d'une table de hachage, et de vérifier l'existence de l'index correspondant avec une complexité de  $O(2n)$  dans l'autre ontologie.

### **II.4.1.2. Recherche d'un espace d'élagage**

Les techniques d'élagage comparent les parties des ontologies partitionnées au préalable. Ce principe est opté par plusieurs systèmes de matching tel que : Lily (voir la section II.5.1.), Anchor-Flood (voir la section II.5.1.), LogMap.

Lily propose une technique de réduction des ancres permettant de déduire les comparaisons à éliminer [Wang et al. 2011]. Cette technique quand elle met en correspondance deux classes, les ancres positives interdisent la comparaison des sous classes de l'une avec les super classes de l'autre, et les ancres négatives interdisent la comparaison de voisinage d'une classe avec le voisinage de la classe non appariée avec elle où la valeur de similarité entre eux est très faible.

Anchor-Flood commence par la désignation des ancres, et comparera par la suite le voisinage des ancres appariées. Ce voisinage compose ce qu'on appelle un segment représenté par l'ensemble de parents, des grand parents, des fils, des petits fils, et des frères, etc.... Cet algorithme compare uniquement les entités appartenant aux segments ancrés, débute par les ancres et se répand dans le voisinage jusqu'à ce que toutes les entités sont atteintes [Hanif and Aono, 2009]. Les paires des entités à apparié sont ceux enrichis par la même opération (ascendant, descendant, frères).

LogMap commence par l'indexation des entités en exploitant les techniques terminologiques telles que les méthodes linguistiques appliquées sur les labels et URI. Il extrait les candidats appariés de chaque paire d'entités indexées ensemble. Exclusivement, ces candidats de correspondance vont être pris en considération pour l'appariement, par conséquent l'élagage doit être assez fort pour limiter les comparaisons. En plus, l'exploitation des méthodes sémantiques a aussi permis de partitionner l'ontologie dans des modules, et le processus de matching ne compare que les entités appartenant au même module.

### **II.4.2. Agrégation de l'alignement**

Il est éventuel de calculer diverses similarités entre les objets des entités de chaque ontologie. Ces similarités peuvent être obtenues à partir de leurs noms, des super classes, des instances et de leurs propriétés respectives. Agréger ces similarités est devenu nécessaire afin de définir une seule similarité entre les entités. Il existe plusieurs types d'agrégation, desquels nous allons illustrer par la suite : la pondération qui fait une combinaison arithmétique des

similarités en affectant un poids à chaque alignement ; l'élection qui réalise un vote sur les appariements des entités ; et l'argumentation qui sélectionne les correspondances en fonction de similarités argumentées.

### II.4.2.1. Pondération

La structure des objets (classes ou individus) est fréquemment associée par des relations, ce qui implique la possibilité de calculer la similarité entre des entités liées aux autres objets. Par exemple, la similarité entre deux classes est obtenue à partir de leurs noms, la similarité de leurs super classes, la similarité de ses instances, et la similarité de ses propriétés. Par conséquent, toutes ces similarités hétérogènes doivent être agrégées dans une seule valeur, et ça peut être assuré à l'aide des normes triangulaires, des Distances multidimensionnelles et des sommes pondérées, l'agrégation floue et la moyenne pondérée.

#### *Norme triangulaire*

La norme triangulaire est utilisée comme un opérateur de conjonction dans un calcul incertain. Elle est définie comme une fonction  $T$  de  $D \times D \rightarrow D$ , où  $D$  est un ensemble ordonné par  $\leq$ , contient une borne supérieure  $S$ , et satisfait les conditions suivantes:

$$T(x, S) = x \quad (\text{condition limite})$$

$$x \leq y \Rightarrow T(x, z) \leq T(y, z) \quad (\text{monotonie})$$

$$T(x, y) = T(y, x) \quad (\text{commutativité})$$

$$T(x, T(y, z)) = T(T(x, y), z) \quad (\text{associativité})$$

Elle s'est représentée par les fonctions typiques:  $\min(x, y)$ ,  $x \times y$ , et  $\max(x + y - 1, 0)$ . Les valeurs fournies par ces fonctions sont normalisées si les valeurs fournies par les mesures sont normalisées. Ces normes triangulaires exigent une borne supérieure parmi l'ensemble des candidats à agréger, peuvent être exposées sous forme de combinaison de ces trois fonctions sur l'intervalle unitaire [Hájek 1998].

Une autre norme triangulaire pour agréger les différentes dimensions  $n$  est le produit pondéré des objets, définie par la formule suivante :

$$\delta(x, x') = \prod_{i=1}^n \delta(x_i, x'_i) \quad (\text{II.1})$$

Où  $x, x'$  sont des objets,  $\delta(x_i, x'_i)$  est la dis-similarité de la paire d'objets de la  $i^{\text{ème}}$  dimension, et  $w_i$  est le poids de la  $i^{\text{ème}}$  dimension.

Contrairement aux agrégateurs multidimensionnels, les valeurs de diverses dimensions dans les normes triangulaires sont dépendantes, et la valeur d'une dimension peut être remplacée par la valeur d'une autre dimension. Par exemple, le fait d'avoir un 0 dans l'une des dimensions implique un résultat final égale un 0 [Euzenat et Shvaiko, 2013].

### *Distances multidimensionnelles et sommes pondérées*

Dans le cas où la différence entre certaines propriétés doit être agrégée, la distance de Minkowski est la plus utilisée. Contrairement aux précédentes, ces mesures conviennent bien aux dimensions indépendantes et tendent à équilibrer les valeurs entre ces dimensions. La distance de Minkowski entre la paire d'objets  $x, x'$  est définie comme suit :

$$\delta(x, x') = \sqrt[p]{\sum_{i=1}^n \delta(x_i, x'_i)^p} \quad (\text{II.2})$$

où  $\delta(x_i, x'_i)$  est la dis-similarité entre la paire d'objets dans la  $i^{\text{ème}}$  dimension.

La distance de Minkowski est appelée la distance Euclidienne quand ( $p = 2$ ), les distances de Manhattan, a.k.a. City-blocks quand ( $p = 1$ ) et la distance de Chebichev si ( $p = \infty$ ). Ces distances sont utilisées pour agréger les mesures de dimensions indépendantes.

Ces distances peuvent être pondérées afin de donner plus d'importance à certaines dimensions. Elles peuvent être normalisées en divisant leurs résultats par la distance maximale possible, ce qui lui rend non linéaire si  $p = 1$ . Ceci est un inconvénient majeur lorsque cette distance est exploitée autant qu'une fonction dans une autre distance (Valtchev1999).

Une agrégation linéaire peut être affinée en ajoutant des poids à cette somme, considère que les valeurs à agréger n'ont pas la même importance. Par exemple, la similarité des propriétés est plus importante que la similarité des commentaires. La fonction d'agrégation

utilise un ensemble de poids  $w_1, \dots, w_n$  correspondant à une catégorie d'entités (des classes, des propriétés), est définie comme suit:

$$\delta(x, x') = \sqrt[p]{\sum_{i=1}^n w_i \times \delta(x_i, x'_i)} \quad (\text{II.3})$$

Où  $x, x'$  sont des objets,  $\delta(x_i, x'_i)$  est la dis-similarité entre  $x_i, x'_i$  dans la  $i^{\text{ème}}$  dimension, et  $w_i$  est le poids de cette dimension.

La somme pondérée peut être considérée comme une généralisation de la distance de Manhattan avec une dimension pondérée, et correspond également à la moyenne pondérée avec des poids normalisés. En outre, elle peut utiliser un ensemble de poids  $w_C^P$  dépend de la catégorie de l'objet  $C$  et de type de la valeur calculé  $P$ . Ce type de mesure est normalisé, si toutes les valeurs sont ainsi, en ayant  $\sum_{i=1}^n w_i = 1$ .

### *Agrégation floue et moyenne pondérée*

Les opérateurs d'agrégation floue ont exploité pour assimiler les quantités homogènes en préservant la structure des domaines agrégés. Ces opérateurs sont définis par une fonction  $f: D^n \rightarrow D$ , où  $D$  est un ensemble ordonné par  $\leq$  et contient une limite supérieure ( $S$ ) satisfaisant les conditions suivantes avec  $\forall x, x_1, \dots, x_n, y_1, \dots, y_n \in D^n$ :

$$\begin{aligned} f(x, \dots, x) &= x && (\text{idempotence}) \\ \forall x_i, y_i, x_i \leq y_i &\Rightarrow f(x_1, \dots, x_n) \leq f(y_1, \dots, y_n) && \text{monotonie croissante} \\ f \text{ est une fonction continue} &&& \text{continuité} \end{aligned}$$

$\min$  est aussi une fonction d'agrégation floue, sachant que l'agrégation est ordonnée comme suit :  $f(x, y) \geq \min(x, y) \geq x \times y \geq \max(x + y - 1, 0)$ .

Un exemple typique d'une fonction d'agrégation floue est la moyenne pondérée représentée par la fonction suivante :

$$\delta(x, x') = \frac{\sum_{i=1}^n w_i \times \delta(x_i, x'_i)}{\sum_{i=1}^n w_i} \quad (\text{II.4})$$

Où  $x, x'$  sont des objets,  $\delta(x_i, x'_i)$  est la dis-similarité entre  $x_i, x'_i$  dans la  $i^{\text{ème}}$  dimension, et  $w_i$  est le poids affecté à la  $i^{\text{ème}}$  dimension.

Une fonction moyenne simple est une fonction dont tous les poids sont égaux. La moyenne pondérée est normalisée si ses valeurs sont ainsi. En fait, la somme pondérée normalisée est aussi une moyenne pondérée.

Les fonctions d'agrégation floue sont conseillées lors de l'agrégation des résultats d'algorithmes concurrents (efficaces par rapport à certains aspects et non par rapport à d'autres). En outre, elles sont très utiles dans les algorithmes d'apprentissage pour connaître les poids de la mesure (voir Sect. II.4.3). (Gal et al. 2005) préfèrent ces fonctions plus que les normes triangulaires pour l'agrégation des mesures de confiance.

### ***Somme pondérée adaptative harmonique***

La notion de la somme pondérée adaptative harmonique est conçue par [Mao et al., 2010] pour pondérer les différents matchers. Elle donne les poids les plus élevés aux mesures les plus discriminantes. Ces mesures sont enregistrées dans des cellules d'une matrice, et par la suite un opérateur calcule le ratio des cellules dont la valeur est la plus petite dans la même ligne et la même colonne sur la plus grande valeur.

La Somme pondérée adaptative harmonique des mesures entre les deux objets  $x, x'$  appartenant respectivement aux ensembles d'objets  $o, o'$  comparés par  $n$  mesures, est calculée comme suit :

$$\delta(x, x') = \sum_{i=1}^n h(\delta_i) \times \delta_i(x, x') \quad (\text{II.5})$$

où  $\delta_i(x, x')$  est la mesure de dis-similarité de la paire d'objets de la  $i^{\text{ème}}$  mesure et  $h$  est définie comme suit :

$$h(\delta_i) = \frac{|\{(e, e') \in o \times o'; \wedge \forall y \in o' \setminus \{e'\}, \sigma(e, y) > \sigma(e, e')\}|}{\min(|o|, |o'|)}$$

La normalisation de cette mesure est la division des résultats par la somme des poids.

### *Moyenne pondérée commandée*

Un autre opérateur d'agrégation appelé moyenne pondérée commandée [Yager,1988], donne plus d'importance aux valeurs les plus élevées. Par conséquent, au moment d'agrégation des résultats d'appariement, cet opérateur ne retient que les grandes valeurs de matches en négligeant les méthodes appliquées. La fonction  $f$  de la moyenne pondérée commandée de  $D^n \rightarrow D$ , où  $D$  est un ensemble ordonné par  $\leq$  et contient une limite supérieure ( $S$ ) satisfaisant les conditions suivantes avec  $\forall x, x_1, \dots, x_n, \in D^n$ , et  $n$  est la dimension dont laquelle les objets ont été analysés, est définie comme suit :

$$f(x_1, \dots, x_n) = \sum_{i=1}^n w_i \times x'_i$$

Où  $(w_1, \dots, w_i)$  est un ensemble de poids appartenant à  $[0,1]$  avec  $\sum_{i=1}^n w_i$ , et  $x'_i$  est l'élément le plus grand de l'ensemble  $\{x_1, \dots, x_n\}$ .

Cet opérateur est commutative, monotone et idempoten. Ainsi, les fonctions de moyenne min et max sont des cas spéciaux de la moyenne pondérée commandée [Euzenat et Shvaiko, 2013].

### **II.4.2.2. Election**

L'agrégation des alignements peut considérer que chaque appariement est une source d'information indépendante et que la décision d'inclure une correspondance dans un alignement est un vote de privilège sur cette correspondance. Cette décision peut être prise en appliquant une élection simple de majorité, ou en prenant en compte la confiance associée aux correspondances comme poids dans la méthode d'élection de majorité pondérée. Cette dernière peut également définir un seuil pour les poids distribués au lieu de la majorité (pour plus de détails, voir [Euzenat et Shvaiko, 2013]).

En fait, toute technique d'élection [Taylor, 2005] peut être appliquée à l'agrégation d'alignement et toutes les mesures sommatives dévisagées précédemment (somme pondérée, moyenne pondérée, moyenne pondérée ordonnée) en ajoutant un seuil.

### II.4.2.3. Argumentation

L'argumentation est une technique permettant de trouver un accord entre les parties en échangeant des arguments favorables ou défavorables dans une position donnée. Dans la mise en correspondance d'ontologies, elle peut avoir deux rôles:

- négociation d'un alignement entre deux agents, et voir si chacun d'entre eux accepte les arguments de l'autre,
- la négociation multi-agent basée sur la négociation quantitative à la place des arguments, peut être vue comme une autre technique d'agrégation entre deux techniques d'alignements.

L'argumentation permettant aux agents de fournir des arguments opposés et de choisir les arguments dépendant de leurs préférences, est exploitée dans l'appariement des ontologies pour trouver les arguments entre les alignements [Trojahn et al., 2011]. Les arguments peuvent aussi exprimer selon le système d'argumentation basé sur la valeur [Laera et al., 2006 ; Bench-Capon, 2003]. En outre, les théories de la logique de description [Dung, 1995; Amgoud et al, 2000] ont exploité en donnant un ensemble d'arguments et des préférences des agents afin de définir un alignement entre deux parties.

Plusieurs résultats ont obtenu en utilisant les divers raffinements d'argumentations comme : l'argumentation basée sur la valeur qui permet d'avoir différentes préférences entre les arguments [Laera et al., 2006] ; ainsi l'argumentation basée sur la grande valeur et l'argumentation basée sur la valeur du vote qui a construit respectivement des préférences sur les confidences et des votes entre agents [Trojahn et al, 2008].

### II.4.3. Méthodes d'apprentissage

La plupart des méthodes de cette section (II.4.3.) se basent sur l'exploitation des exemples d'apprentissage dans la première phase de l'algorithme et une deuxième phase appelée phase de classification. Ces exemples peuvent être positifs ou négatifs, extraient respectivement des alignements corrects ou incorrects en utilisant les méthodes d'alignement manuelles [Euzenat et Shvaiko, 2013]. Dans la première phase, ces alignements ont exploités pour apprendre à réaliser l'alignement. Les alignements appris ont utilisés dans l'alignement de nouvelles ontologies dans la seconde phase. L'alignement produit peut être enrichi par la suite dans le jeu d'apprentissage des alignements.

Les méthodes d'apprentissage les plus connues sont : l'apprentissage de Bayes, les réseaux de neurones, la machine à vecteurs de support, et les arbres de décision.

L'apprentissage de Bayes [Good, 1965] est un algorithme d'induction probabiliste. Il est exploité par diverses approches d'appariement autant que classificateur [Doan et al., 2004; Straccia and Troncy, 2006; Lambrix and Tan, 2006; Nandi and Bernstein, 2009; Spiliopoulos et al., 2010; Esposito et al., 2010; Tournaire et al., 2011).

Plusieurs réseaux de neurones ont exploité dans des tâches différentes de processus d'appariement des ontologies. Par exemple, la découverte de correspondance entre les attributs le long de la classification [Li and Clifton, 1994; Esposito et al., 2010] ou l'apprentissage des paramètres d'appariement comme le poids [Ehrig et al., 2005; Mao et al. 2010; Gracia et al., 2011 ; Djeddi and Khadir, 2013].

Les machines à vecteurs de support (en anglais support vector machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destiné à résoudre des problèmes de classification ou de régression [Cristianini and Shawe-Taylor, 2000]. SVMs ont exploitées autant que classificateurs dans les approches d'appariement [Ehrig et al., 2005; Spiliopoulos et al., 2010; Tournaire et al., 2011].

Les arbres de décision peuvent exploiter par les systèmes de matching des ontologies autant qu'un découvert de correspondances entre les entités [Xu and Embley, 2003; Duchateau et al., 2009; Spiliopoulos et al., 2010; Tournaire et al., 2011], et de paramètres d'apprentissage, e.g., thresholds, pour adopter automatiquement la tâche de matching fournie [Ehrig et al., 2005; Duchateau et al., 2008].

### **II.5. Etude et comparaison des approches d'appariement**

Cette section étudie et compare les systèmes qui ont des résultats significatifs dans les derniers challenges de OAEI (Ontology Alignment Evaluation Initiative) [Shvaiko et al., 2018, Alsayed et al., 2019] comme ALIN, ALOD2Vec, AML, DOME, FCAMapX, Holontology, KEPLER, Lily, LogMap, LogMapLt, SANOM, Wiktionary.

### II.5.1. Etude des approches d'appariement participées à l'OAEI

Les systèmes à étudier dans cette thèse ont participé dans les compagnies d'OAEI 2018 et/ou 2019 avec nos propositions, et vont expliquer dans les paragraphes suivants:

#### *ALIN*

**ALIN** utilise six métriques linguistiques [Algergawy et al.,2018; Da Silva et al.,2019] intégrées dans les interfaces d'application intégrée Lin, Symmetrics API [Surhone, 2010] (Jaccard, Jaro-Winkler, et n-Gram) et HESML API [Lastra-Daz, 2017] (Resnick, Jiang-Conrath, et Lin). En outre les méthodes syntaxiques de ce système utilisent Stanford CoreNLP API incluant WordNet. En outre, ce système exploite les alignements des classes acceptés par les experts pour aligner ses sous classes [Da Silva,2017; Da Silva,219].

ALIN adopte la stratégie d'agrégation appelée la somme pondérée avec l'intervention d'un expert pour choisir le bon alignement. Les valeurs des métriques citées auparavant sont comparées avec celle de la somme pondérée, et les métriques dont les valeurs sont supérieures à celle de cette somme seront suggérées à un expert pour choisir le bon alignement.

#### *ALOD2Vec*

Le système de matching **ALOD2Vec** [Portisch et Paulheim, 2018] récupère les descriptions textuelles de tous les éléments des ontologies à apparier. Ensuite, il utilise un filtre pour ajouter des correspondances de chaînes simples à l'alignement final. Les étiquettes restantes sont liées et comparées aux concepts de l'ensemble de données externes en utilisant la similarité de cosinus, et la meilleure solution est ajoutée à l'alignement final. L'ensemble de données externes exploité est nommé *WebIsALOD*, et il est disponible sous le lien LOD (Linked Open Data)<sup>10</sup>. Afin de mieux exploité cette source, elle est transformé en graphe de données RDF reliées par des relations d'hyponymie. Chaque élément de ce graphe est représenté par un vecteur extrait de la base de données à l'aide de modèles du langage neuronal (RDF2Vec).

---

<sup>10</sup> <http://webisa.webdatacommons.org/>

La similarité de cosinus entre les vecteurs des labels liés aux concepts est exploitée par le système ALOD2Vec pour fournir les alignements de ce système. Ces vecteurs ont calculé à l'aide d'un modèle du langage neuronal [Algergawy et al.,2018].

### *Agreement Maker (AML)*

**Agreement Maker (AML)** est un système automatique de matching basé sur l'utilisation de WordNet et une ontologie biomédicale autant que des connaissances complémentaires [Algergawy et al.,2018 ; Algergawy et al.,2019]. Son algorithme est structuré en trois étapes:

- La première étape se charge de la comparaison des labels, des commentaires et des instances à l'aide de la similarité de cosinus, et les mesures basées sur les chaînes telles que : edit distance et substrings.
- La deuxième étape exploite la structure des propriétés dans les ontologies à appairer afin de comparer les descendants des concepts qui ont une forte similarité dans l'étape précédente de cet algorithme.
- La troisième étape applique un seuil avec la cardinalité de l'alignement favori sur les résultats des deux étapes précédentes pour calculer le résultat final.

### *DOME (Deep Ontology MatchEr)*

**DOME (Deep Ontology MatchEr)** [Hertling & Paulheimn, 2018 ; Hertling & Paulheimn, 2019] exploite principalement l'approche de Doc2Vec [Le et al., 2014] qui s'active seulement quand il y a assez de texte à traiter. Doc2Vec symbolise tous les types de textes (phrases, paragraphes, les documents) comme un vecteur de caractéristiques de taille fixe. Ce vecteur est conçu pour prédire les mots apparaissant dans le document (les commentaires, URI des concepts). En d'autres termes, le vecteur représente le sens et la sémantique du concept lors de l'apprentissage de textes. En revanche, les labels et les segments de chaque source ont aligné à l'aide des similarités basées sur le string et la tokenisation comme indiqué dans le tableau II.3.

### *FCAMapX*

**FCAMapX** est un système automatique de matching des ontologies, basé sur l'analyse formelle des concepts, et le modèle mathématique d'analyse des individus et des structures

des concepts. Ce système est un successeur de FCA-MAP qui génère progressivement trois types de contextes formels et d'extraits de mapping à l'aide des réseaux [Zhao et al, 2018]. Premièrement, le contexte formel basé sur les ancres représente le principe de matching lexical entre les ontologies, décrit la manière de partage des jetons lexicaux entre les noms, les classes, les libellés et les synonymes. Deuxièmement, la validation structurelle positive et négative des résultats lexicaux est faite à l'aide du contexte formel basé sur des relations décrivant le type de la relation entre les classes et les ancres (taxonomiques, paronomiques et disjonction). Troisièmement, le contexte basé sur des relations positives peut être utilisé pour fournir les appariements structurels dérivés [Chen et al., 2018 ; Chen et al., 2019].

### *Holontology (Holistic ontology matcher)*

**Holontology (Holistic ontology matcher)** est un système modulaire basé sur le système LPHORM (Linear Program for Holistic Ontology Matching) [Megdiche et al., 2016], peut assurer la mise en correspondance de plusieurs ontologies. Ce système traite le problème d'appariement des ontologies comme un problème d'optimisation combinatoire au niveau du schéma. Ce problème est modélisé par un programme linéaire étendant le problème d'appariement des graphes à pondération maximale avec des contraintes linéaires (cardinalité correspondante, contraintes structurelles et de cohérence) selon les étapes suivantes [Roussille et al., 2018] :

- La première étape de l'Holontology consiste à charger les ontologies d'une manière indépendantes. Ensuite, traduire chaque ontologie en une structure interne conservant sa structure originale fondée sur les trois blocs de base: classes et propriétés (objet et données). Chacune d'entre elles est représentée sous la forme d'un nœud, dépendant de ses sous-type, qui sont sous classés dans: *ClassNode*, *ObjectPropertyNode* ou *DataPropertyNode*. Ces nœuds sont ensuite enrichis à l'aide de *AnnotationData*.
- La deuxième étape étend le titre en noms propres à l'aide des hiérarchies des ontologies déjà chargées, et les méthodes de stockage des références en Java à la place de traducteur et des matrices.
- La troisième étape calcule le produit cartésien entre toutes les entités de même type (classes et propriétés) afin de construire un programme similarité linéaire.
- La quatrième étape mène à la création de nouvelles relations en calculant la similarité tendue (exemple : déterminer si une propriété d'objet est passive ou active par rapport

à ses classes). Plus précisément, pour chaque propriété, l'algorithme propose de calculer sa polarité à l'aide d'une analyse profonde, par ex. *Author reviews Paper* (actif) vs. *Paper reviewed by Author* (passif), ce qui permet l'appariement de *reviews* et *reviewed by reversed*.

- La cinquième étape applique une combinaison de plusieurs mesures de similarité (la correspondance exacte, Levenstein, Jaccard et Lin) extraites de la construction *AnnotationData* des étapes précédentes et des contraintes basées sur les ontologies.
- La sixième étape convertit les contraintes données en une forme pouvant être utilisée par un solveur. Le solveur CPLEX est choisi à cause de sa capacité d'optimisation et de gestion des seuils probables.
- Les résultats fournis par le solveur ont été convertis en un alignement exporté en format RDF dans la septième étape.
- Dans la huitième étape (facultative), combine le résultat de l'alignement et exécute les étapes cinq à sept pour renforcer les résultats obtenus.

### *Kepler*

Le système de matching des ontologies **Kepler** se base sur l'exploitation de techniques terminologique, la stratégie de partitionnement, et un traducteur pour l'appariement du multilinguisme [Kachroudi et al., 2018]. L'algorithme de cette approche est implémenté en six modules présentés comme suit:

- Le module d'analyse et de prétraitement extrait les entités et leurs descriptions de l'ontologie, et les organise dans une liste.
- Partitionnement de l'ontologie dans des segments qui vont être alignés par la suite.
- La traduction dans Kepler est assurée par les traducteurs Microsoft Bing<sup>11</sup> et WordNet, qui traduisent l'ontologie de la langue d'entrée vers la langue anglaise.
- La stratégie d'indexation est exploitée afin de traiter le problème du coût de calcul de similarité entre toutes les entités des ontologies. Par conséquent, une stratégie de recherche efficace est implémentée grâce à des index construits à partir des deux ontologies en entrée afin de permettre une recherche plus rapide.

---

<sup>11</sup> <https://www.bing.com/translator>

- Le module d'identification des candidats permet de trouver les entités communes entre les index en utilisant une stratégie d'interrogation. Cette tâche consiste à interroger des documents dans un vecteur d'espace contenant un ensemble d'entités ontologiques et leurs synonymes obtenus via WordNet. Pour ce faire, une technique fondée sur la recherche terminologique et les aspects sémantiques est implémentée.
- Le module de filtrage est constitué de deux sous-modules complémentaires, chacun étant responsable d'une tâche spécifique afin d'affiner l'ensemble des correspondances principalement identifiées. Le premier filtre et élimine les redondances possibles entre ses candidats. Le deuxième filtre élimine les candidats faux positifs, qu'appartiennent à deux alignements différents.
- La génération d'alignements est le module qui fournit les résultats de correspondance en format RDF.

### *Lily*

**Lily** [Wu et al., 2019] est un système universel d'appariement des ontologies, basé sur plusieurs techniques spécifiques comme: l'extraction d'un graphe sémantique pour évaluer le sens réel des éléments des ontologies et découvrir des clés de matching plus crédibles. Cette extraction exploite le modèle de circuit électrique et ainsi les nouvelles règles de calcul de conductivité pour améliorer la qualité de ce graphe.

Par ailleurs, le calcul des similarités littérales est appliqué sur les documents de description sémantique (SDD). Le document de description sémantique d'un concept contient des informations de la hiérarchie de la classe, des propriétés, et des instances. Le document de description sémantique d'une propriété comporte les hiérarchies, les domaines, les rangs, les restrictions, et les instances reliées. Ensuite, il combine ces similarités à l'aide des poids expérimentiels.

Lily a intégré les Méta-thésaurus UMLS afin de pouvoir détecter les concepts synonymes et améliorer les performances du système dans le domaine biomédical qui avait une syntaxe spécifique. En outre, le document virtuel est une notion conçue pour les ontologies dont les concepts ont influencé par ses voisins [Qu et al., 2006]. Ce document comporte les descriptions locales *URIref* d'un concept et ses voisins qui ont un reflet sur ce concept.

Ainsi, ce système exploite une stratégie pour le matching des ontologies à grande échelle, appelée la réduction des ancrs [Wang et al. 2011] mentionnée à la section (II.4.1.2). De plus,

un post-traitement est assuré à l'aide d'une technique de débogage de mapping des ontologies, qui consiste à détecter des appariements erronés et redondants [Wang et al., 2008].

### *LogMap*

**LogMap** est une famille des systèmes évolutifs d'appariement d'ontologies composée de LogMapLt et LogMapBio [Jiménez-Ruiz et al., 2017; Jiménez-Ruiz, 2019]. LogMapLt "lightweight" est un variant de LogMap, basé essentiellement sur les techniques de matching de string. LogMapBio inclut une extension permettant d'utiliser BioPortal [Fridman et al., 2009; Ghazvinian et al. 2009] comme un médiateur dynamique d'ontologies à la place de quelques ontologies présélectionnées [Chen et al, 2014].

Ce system se repose sur des éléments différents tels que : l'indexation lexicale, la logique basée sur le module d'extraction, le raisonnement propositionnel de horn, le suivi des axiomes, la réparation locale, et l'indexation sémantique. Le premier élément utilise un index sauvegardant l'information lexicale des ontologies en entrée afin de calculer d'une manière efficace l'ensemble d'appariement préalable, ce dernier est calculé à l'aide de matcher de string [Jiménez-Ruiz, et al., 2012] et des ressources externes comme WordNet ou UMLS-lexicon [Jiménez-Ruiz, et al., 2011].

Le deuxième élément exploite les techniques de modularisation des ontologies pour réduire la taille des ontologies en entrées. Il a une influence critique sur la faisabilité de la détection de l'insatisfaction et la réparation critique. Par conséquent, les modules d'ontologie avec une bonne compréhension de la sémantique des propriétés peuvent être bien calculés et ils sont plus utiles et généralement plus petits que ceux de l'ontologie en entrée [Dowling et Gallier, 1984].

Le raisonnement propositionnel de horn est utilisé pour encoder les modules pertinents des ontologies en entrée et les candidats potentiels d'appariements. Par ailleurs, l'algorithme classique de Dowling-Gallier de la satisfiabilité propositionnelle de Horn (Shvaiko et al., 2018) qui détecte efficacement les classes insatisfaisantes n'est pas complètement implémenté par LogMap. Cet algorithme est étendu par l'élément de suivi des axiomes pour suivre tous les appariements qui peuvent impliquer l'insatisfaction d'une classe. Cette application réalise une réparation locale de l'insatisfaction à la volée et inspecte uniquement le premier plan de réparation disponible [Agrawal et al, 1989].

L'indexation sémantique se fonde sur le schéma d'étiquetage d'intervalle [Christophides, et al., 2003] qui optimise la structure des données pour le stockage des graphes acycliques dirigés (directed acyclic graphs: DAG). Les DAG réduisent le coût de réponse aux requêtes taxonomiques [Christophides et al., 2003; Nebot, et Berlanga, 2009] en indexant efficacement la représentation propositionnelle de Horn des modules de l'ontologie et les appariements. En particulier, cet index sémantique répond à de nombreuses requêtes d'implication comme une opération de recherche d'index des ontologies d'entrée et des mappages calculés sans avoir besoin de raisonnement. De plus, il complète l'utilisation de l'encodage propositionnel pour identifier et réparer les classes non satisfaisantes (Dowling et Gallier, 1984).

### *SANOM*

**SANOM (Simulated annealing-based ontology matching)** est un système d'appariement des ontologies basé sur la simulation de recuit [Shvaiko et al., 2019]. Une bonne simulation de recuit (simulated annealing (SA)) a permis la découverte de concepts communs des ontologies à aligner. Un alignement potentiel est modélisé comme un état de SA afin de produire un appariement plus fiable entre les ontologies pendant l'évolution. Cette évolution nécessite une fonction de fitness afin d'évaluer la qualité des solutions intermédiaires au problème d'appariement d'ontologie.

Une fonction de fitness utilise des méthodes de simulation lexicale et structurelle pour estimer la finesse d'un alignement. Mais avant d'appliquer ces métriques, le système implémente un prétraitement basé sur la tokenisation (voir la section II.3.2.2.1.), une liste de mots appelée Glasgow utilisé pour détecter les termes qui ont plusieurs sens, le stemming (voir la section II.3.2.2.2.),

Le Maximum de la métrique Jaro-Winkler avec un seuil de 0.9, et de la mesure de Wu et Palmer avec un seuil de 0.95 ont retenu dans l'évaluation lexicale.

La méthode structurelle exploitée est celle de dérivation de nouveaux appariements des voisins des alignements déjà découverts par la précédente méthode. En outre, les propriétés sont considérées similaires s'ils ont eu des domaines et/ou rangs analogues. De plus, l'outil TF-IDF est exploité pour calculer la finesse des noms des propriétés, des domaines et des rangs. Enfin, l'appariement de propriétés est la moyenne de deux plus grandes valeurs de

finesses, fournies par TF-IDF. Les classes qui ont des propriétés similaires sont aussi considérées similaires.

### *Wiktionary*

**Wiktionary** est une application d'appariement d'ontologies fondée sur l'exploitation de dictionnaire Wiktionary<sup>12</sup> comme une source externe de connaissance complémentaire. Wiktionary est une grande ressource de connaissance lexicale construite en collaboration en ligne. Ce dictionnaire est disponible dans plusieurs langues, et il est organisé comme Wikipédia mais avec des connaissances plus larges.

L'appariement dans ce système est un appariement au niveau des éléments, basé sur les étiquettes utilisant les versions publiques de Wiktionary présentées en DBnary [Sérasset, 2015] et RDF. L'ensemble de données en DBnary utilise une extension du modèle LEMON [McCrae et al., 2012] pour décrire 8 langues de Wiktionary et les intègre dans un seul graphe en RDF.

Vu que le dictionnaire Wiktionary a un traducteur, ce système peut faire des appariements monolingues et multilingues. La version monolingue fait l'appariement des ontologies monolingue en reliant les labels aux concepts du dictionnaire à l'aide de la similarité de cosinus, puis vérifiant si ces concepts sont similaires dans la source externe. Le système multilingue utilise les données de la version anglaise du Wiktionary pour faire la correspondance multilingue à l'aide du système de matching COMS [Lin et al., 2010, Lin et al., 2011].

Ce système assure l'appariement des instances existantes, à l'aide de l'indexation des chaînes de caractères sans utiliser le dictionnaire pour améliorer ses performances. Une confiance plus élevée est attribuée aux instances appartenant aux classes déjà alignées.

### **II.5.2. Comparaison des approches d'appariement participées à l'OAEI**

La comparaison des approches d'appariement peut être réalisée sur plusieurs volets. Le premier volet admet de les comparer en s'appuyant sur leurs caractéristiques externes. Ces caractéristiques englobent essentiellement les entrées et les sorties des méthodes. Le second

---

<sup>12</sup> <https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary>

volet étudie les techniques fondamentales exploitées par les systèmes de matching afin d'évaluer la similarité. Ces techniques utilisent les éléments de l'ontologie pour produire un alignement. Le troisième volet expose les stratégies d'appariement exploitées par les systèmes de matching. Ces stratégies démontrent potentiellement la démarche suivie par l'approche pour déterminer le résultat de correspondance final entre les divers composants des deux ontologies à apparier.

### II.5.2.1. Comparaison des caractéristiques externes

Les caractéristiques externes des systèmes décrites dans la section (II.5.1.) et illustrées dans la table II.2 sont le type de composition des alignements utilisé, la technique de sélection exploitée et éventuellement les posts-traitements exécutés. La composition admet la combinaison de diverses techniques d'alignement ou certains niveaux d'information. La méthode de sélection rassemble le seuillage ou l'optimisation (maximisation ou minimisation) locale de matching.

La majorité des systèmes de matching repose sur une composition de plusieurs résultats d'alignement à l'aide d'une ou de plusieurs techniques de composition comme : la somme pondérée, le contexte basé sur des relations positives, et le poids expérimental, optés par ALIN, FCAMapX, et Lily, respectivement.

Les systèmes Dome ; Holontology reposent sur la similarité de cosinus ; le produit cartésien et les méthodes de Levenstein, de Jaccard et Lin; respectivement. Ainsi, ils ont implémenté une sélection à l'aide d'un seuil. Les systèmes AML et LogMap exploitent uniquement une sélection à base de seuil.

Le post-traitement est implémenté sous forme d'un filtrage, et une technique de débogage de mapping, par les systèmes Kepler et Lily, respectivement.

*Tableau II.2: Caractéristiques externes des principaux systèmes de matching participés dans les compagnies d'OAEI en 2017 et 2019*

<b>Système</b>	<b>Composition</b>	<b>Sélection</b>	<b>Post-traitement</b>
ALIN	Somme pondérée	-	-
AML		Seuil	
DOME	La similarité de cosinus	Seuil	-

FCAMapX	Le contexte basé sur des relations positives	-	-
Holontology	Le produit cartésien Levenstein, Jaccard et Lin	-Seuil probable (Solveur CPLEX)	
KEPLER			Le filtrage
Lily	Poids expérimental	-	La technique de débogage de mapping
LogMap		Seuil	
SANOM	Le maximum	Seuil	

### II.5.2.2. Comparaison des techniques de bases

La table II.3 illustre une étude comparative exposant les principales techniques exploitées par chaque méthode d'alignement. Ces techniques rassemblent les méthodes terminologiques, structurelles, extensionnelles et sémantiques, ou des nouvelles techniques proposées dans l'algorithme du système.

Les techniques terminologiques regroupent les techniques syntaxiques et les techniques linguistiques. Les méthodes syntaxiques incluses dans ALIN sont les distances d'édition, grams, Jaccard, Jaro-Winkler, , Resnick, Jiang-Conrath, et Lin. Les correspondances de Jaccard et Lin ont intégré dans Holontology avec la correspondance exacte, et celle de Levenstein. La similarité de cosinus est utilisée par ALOD2Vec et AML qui exploitent aussi les similarités de edit distance et Substrings etc....

Dome utilise une similarité basée sur string. FCAMapX a introduit le contexte formel basé sur les jetons. Lily a inclus des mesures littérales appliquées sur les documents de description sémantique (SDD) de concept.

Les techniques linguistiques s'appuient fréquemment sur le thésaurus WordNet. Certains systèmes utilisent avec WordNet une autre connaissance complémentaire comme: l'ontologie biomédical (AML), le traducteur Microsoft Bing(Kepler), ou UMLS Meta-thésaurus(Lily), UMLS- lexicon et BioPortal (LogMap).

Les techniques structurelles consistent à rechercher de nouvelles relations à partir d'un alignement d'entrée (généralement proposé par une approche syntaxique) ou à confirmer des relations déjà découvertes. Cette technique considère que deux entités sont en relation si leurs voisins respectifs sont également en relation. Quelques autres techniques suggèrent une approche structurelle générique. Cette dernière prend tous les types de relations et les traite un par un d'une manière indépendante. La prise en compte de la structure interne (attribut) est fréquemment effectuée par le biais d'une table de compatibilité entre les types de données des attributs.

Le système holontology a introduit les méthodes de stockage des références en Java, et a calculé la polarité des propriétés.

LogMap a implémenté un suivi des axiomes, le raisonnement propositionnel de horn, l'algorithme classique de Dowling-Gallier

Les différentes techniques sémantiques: Doc2Vec, le solveur CPLEX (logique de description), la recherche sémantique, et le graphe sémantique, sont optées par les systèmes : ALOD2Vec, Holontology, Kepler, et Lily, respectivement. En outre, LogMap a introduit : le schéma d'étiquetage d'intervalle, l'encodage propositionnel.

Les techniques extensionnelles sont généralement les techniques terminologiques appliquées sur les instances ou les individus des concepts tels que : la similarité de cosinus, la distance d'édition, et les sous chaînes (substrings) utilisées dans AML.

## Chapitre II : Problèmes d'appariement et les travaux connexes

Tableau II.3. Comparaison des techniques de base optées par des systèmes de matching participés dans les compagnies d'OAEI en 2017 et 2019

Système	Techniques terminologiques		Techniques structurelles		Techniques sémantiques	Techniques Extensionnelle
	Syntaxique	linguistique	Internes	Externes		
ALIN	Jaccard, Jaro-Winkler, n-Gram, Resnick, Jiang- Conrath, and Lin.	WordNet (Synonymie)	-	Aligner les subsumés en basant sur les avis des experts.	-	-
ALOD2Vec	La similarité de Cosinus	RDF data utilisé comme des connaissances complémentaires	-		-Doc2Vec	-
AML	- la similarité de Cosinus -Edit distance -Substrings ....	-WordNet -Ontologie biomédicale	-	Aligner les subsumés des concepts qui ayant une forte similarité	-	- la similarité Cosinus -Edit distance -Substrings ....
DOMÉ	Similarité basée sur string et tokenisation	-	-	-	-	-

FCAMapX	Le contexte formel basé sur les jetons	WordNet (Synonymie)		Le contexte formel basé sur les relations (taxonomiques, partonomiques et disjonction)		
Holontology	La correspondance exacte, Levenstein, Jaccard et Lin.	-		-Les méthodes de stockage des références en Java - Calculer la polarité des propriétés	- Le solveur CPLEX( logique de description)	-
KEPLER		-WordNet (Synonymie) - Le traducteur Microsoft Bing -Traduire toutes les ontologies en anglais	-Organiser les entités des ontologies dans une liste,	-	La Recherche sémantique	
Lily	Mesures littérales Appliquées sur SDD de concept	UMLS Meta-thésaurus (synonyme des	Les Mesures littérales Appliquées sur		Le Graphe sémantique	

## Chapitre II : Problèmes d'appariement et les travaux connexes

		termes médicaux)	SDD de propriété			
LogMap	Appariement à base de string	-WordNet -UMLS-lexicon - BioPortal (LogMapBio)		- Un suivi des axiomes -Le raisonnement propositionnel de horn - L'algorithme classique de Dowling-Gallier	-Le schéma d'étiquetage d'intervalle -L'encodage propositionnel	
SANOM	-Wu et Palmer -Jaro-Winkler	WordNet avec la métrique de Jaro-Winkler	TF-IDF(propriétés)	Aligner les voisins des appariements déjà découverts		
Wiktionary	La similarité de Cosinus	Wiktionary (synonymie)				Indexation des chaînes de caractères

### II.5.2.3. Comparaison des stratégies de matching

Le tableau II.4 recense les systèmes présentés dans la section (Sect II.5.1), qui ont optés une stratégie d'appariement et/ou un type de prétraitement dans leurs algorithmes. Les techniques de prétraitement: l'alignement API, le modèle du langage neuronal, la traduction l'ontologie en une structure interne, l'extraction des entités et leurs descriptions dans une liste, et l'extraction d'un graphe sémantique pour chaque ontologie à aligner sont introduites par ALIN, ALOD2VEC, Holontology, KEPLER, et Lily, respectivement.

Les différentes stratégies d'alignements adoptées par les différents systèmes sont : l'agrégation(ALIN) ; le partitionnement et l'indexation (KEPLER) ; le poids pour l'agrégation, la réduction des ancrs, et la recherche d'un espace d'élagage (Lily) ; la réduction des ancrs (LogMap).

*Tableau II.4. Comparaison des stratégies de matching des systèmes participés dans les compagnies d'OAEI en 2017 et 2019*

<b>Système</b>	<b>Prétraitement</b>	<b>Technique</b>
ALIN	Alignment API [David et al, 2011]	L'agrégation,
ALOD2VEC	le modèle du langage neuronal,	
Holontology	Traduire l'ontologie en une structure interne.	
KEPLER	L'extraction des entités et leurs descriptions dans une liste.	- La partitionnement, - L'indexation.
Lily	-L'extraction d'un graphe sémantique pour chaque ontologie à aligner	- Le poids pour l'agrégation - La réduction des ancrs - rechercher un espace d'élagage
LogMap		La réduction des ancrs
RiMOM	Classification	Le réseau bayésien naïf
SANOM	-La tokenisation -La liste de Glasgow -Le stemming	

### II.6. Conclusion

Nous avons exposé dans ce chapitre le cadre technique de notre travail. Nous avons décrit les terminologies de mise en correspondance des ontologies, et quelques problèmes autres que le Web Sémantique qui a engendré le processus de matching, tel que les niveaux d'hétérogénéités entre les ontologies et les modes d'interopérabilité sémantique. Ainsi, nous avons présenté le processus de matching et les différentes techniques d'appariement terminologique, structurelle, sémantique, probabiliste, etc. Ces dernières techniques peuvent être agrégées pour avoir le résultat final en utilisant une ou plusieurs stratégies de matching.

Les systèmes de matching utilisant une ou plusieurs techniques ou stratégies de matching présenté auparavant ont étudiées et comparées entre eux. La plupart de ces systèmes ont participé dans les compagnies d'OAEI 2017 et/ou 2019 avec nos outils proposés dans ce document ONTMAT, et ONTMAT1.

**Partie II :**

**Contribution**

## ***Chapitre III :***

***Inférer l'alignement des concepts à l'aide  
d'instances et de dictionnaire externe***

## *Chapitre III*

### *Inférer l’alignement des concepts à l’aide d’instances et de dictionnaire externe*

#### **Sommaire**

III.1. Modèle d’Ontologie de Karlsruhe . . . . .	91
III.2. Matching des ontologies . . . . .	91
III.3. Phase 1 de l’algorithme : Alignement des concepts à partir d’instances . . .	93
III.4. Phase 2 de l’algorithme : Alignement des relations binaires . . . . .	96
III.4.1. Restriction . . . . .	96
III.4.2. Matching des relations binaires . . . . .	97
III.5. Phase 3 de l’algorithme : Appariement à l’aide de raisonneur et WordNet .	99
III.5.1. Matching des concepts de la base At . . . . .	99
III.5.2. Matching des concepts en utilisant un raisonneur et WordNet . . . .	100
III.6. Evaluation et expérimentation . . . . .	104
III.6.1. Implémentation et paramétrage . . . . .	105
III.6.2. Evaluation de l’alignement de référence de crisp . . . . .	106
III.7. Conclusion . . . . .	106

Afin de mieux expliquer le processus de matching proposé dans ce chapitre, une autre description de l’ontologie basée sur la logique de description est présentée dans la première section. Cette description est maintenue car notre algorithme implémenté dans le système ONTMAT [Gherbi et Khadir, 2016, Gherbi et Khadir,2017] exploite l’inférence de l’alignement des concepts à l’aide de leurs instances. Les sections 3,4 et 5 expliquent les différentes étapes de cet algorithme.

### **III.1. Modèle d'Ontologie de Karlsruhe**

L'ontologie décrite par le Modèle de Karlsruhe [Stumme et al., 2003] est la plus appropriée pour exprimer le processus de matching expliqué dans ce chapitre. Elle est définie comme un tuple  $O_i := (D, A, KB, Lex)$  où :

- $D$  représente la base de l'ontologie ;
- $A$  est le L-axiom de système ;
- $KB$  est la base de connaissance ;
- $Lex$  est le lexique utilisé.

### **III.2. Matching des ontologies**

Les méthodes de matching évaluent la similarité et/ou la relation entre les concepts et les relations des ontologies à appairer. Ces concepts  $C$  (*owl:Class*) et relations  $B$  peuvent être présentés comme une structure  $D := (C, <_C, F, B, <_B)$  de l'ontologie  $O$  où :

- $<_C$  : est la hiérarchie des concepts ou la taxonomie représentée par un ordre partiel sur  $C$  ; correspond à l'ensemble de relation théorique  $Rel = \{\equiv, \subseteq, \supseteq, \perp\}$ , où:
  - Equivalence ( $\equiv$ ): est présentée en OWL par *owl:equivalentClass*, *owl:equivalentProperty*, et *owl:sameAs* pour les concepts, les relations, et les instances [Ehrig, 2007].
  - Moins général ( $\subseteq$ ): *rdfs:subClassOf*
  - Plus général ( $\supseteq$ ): *rdfs:subClassOf*
  - disjonction ( $\perp$ ): *owl:differentFrom*.
- La fonction de signature  $F: B \rightarrow C \times C$  restreint les modèles de relation binaire ; avec  $F(B_1) = \{\text{dom}(B_1), \text{rang}(B_1)\}$  pour  $B_1 \in B$ ,  $\text{dom}(B_1)$  désigne le domaine et  $\text{rang}(B_1)$  le range, qui ont été traités comme des instances des concepts dans la première phase de l'algorithme et un concept dans la deuxième phase.
- La relation binaire est définie par l'ordre partiel  $<_B$  comme suit :

$$\begin{aligned} B_1 <_B B_2 \\ \text{Iff } \text{dom}(B_1) <_C \text{dom}(B_2) \\ \text{et } \text{ran}(B_1) <_C \text{ran}(B_2). \end{aligned} \tag{III.1}$$

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

---

Afin d'améliorer le processus d'appariement, nous commençons par la comparaison des instances. Cela nous permet de découvrir la relation entre ses concepts et les relations binaires reliant ces instances.

Ces instances (si elles existent) ont exprimé comme une structure  $KB := (C, B, I, i_C, i_B)$ ; où

- Les ensembles  $C$  et  $B$  sont les même présentés auparavant ;
- $I$  : est l'ensemble des instances
- $i_C: C \rightarrow 2^{T^{13}}$  est une fonction d'association, associe chaque concept comme  $C_1$  et  $C'_1$  à ses instances dans  $I$  ; et  $i_{B_1}: B_1 \rightarrow 2^T$ ; avec  $B_1 \subseteq i_{C'_1} \times i_{C_1}$  pour tout  $B_1 \in B$  associe les concepts en relations binaires à leurs instances.

Ensuite, les méthodes terminologiques ont utilisé pour comparer les noms (des instances, les relations et les entités dans les sections suivantes) présentés par  $Lex := (S_C, S_B, S_I)$ . Les trois ensembles  $S_C, S_B, S_I$ , ont symbolisé par  $A_i$  pour exprimer les noms des concepts, des relations et des instances respectivement. Cet identifiant est associé à un axiome par une relation l'association nommée  $x$  dans L-axiom  $A := (A_i, x)$ .

Afin d'illustrer le processus de matching, les deux ontologies  $O_1$  et  $O_2$  décrivant *Human* et *Person* ont été présenté sous forme de hiérarchie graphique dans les figures Figure III.1. et Figure III.2, respectivement.

Les rectangles indiquent les concepts, les octogones désignent les propriétés, les instances ont dessiné par des ellipses et les relations hiérarchiques par des flèches solides. La flèche entrante à la relation vient de son domaine et celle sortante de son range.

---

<sup>13</sup>La notation  $2^T$  représente l'ensemble des ensembles de chaîne de caractères [Jérôme, 2007]

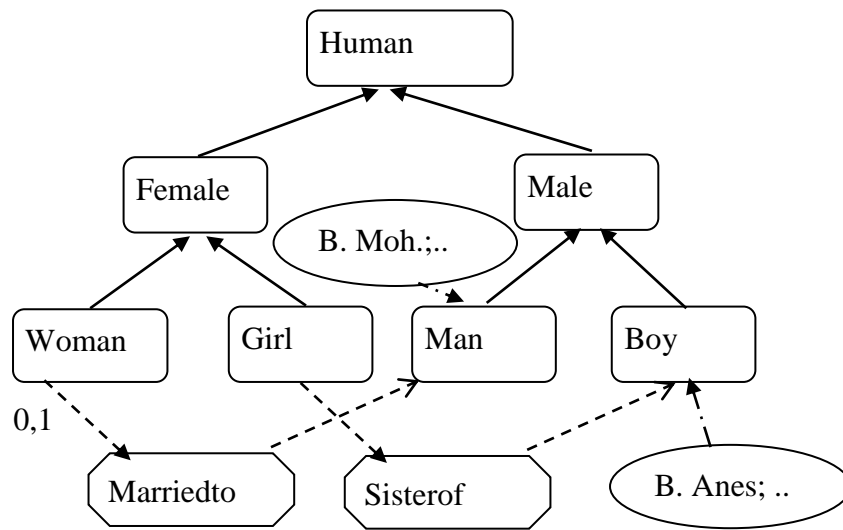


Figure III.1. Ontologie  $O_1$

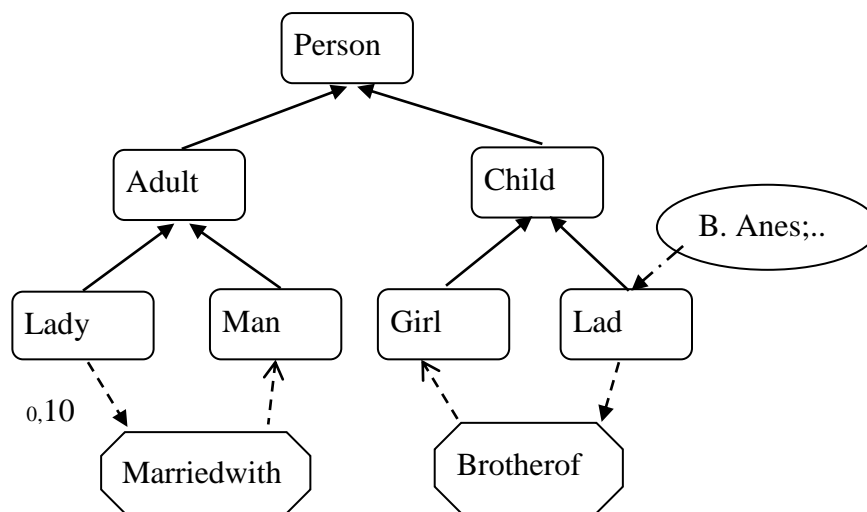


Figure III.2. Ontologie  $O_2$

### III.3. Phase 1 de l'algorithme : Alignement des concepts à partir d'instances

Dans cette phase nous comparons les instances afin de déduire les relations entre leurs concepts. Les chaînes de caractères  $S_{Ic1}$  de  $I_{c1}$  et  $S_{Ic2}$  de  $I_{c2}$  sont normalisées [Euzenat et Shvaiko, 2007] et comparées entre eux à l'aide de la métrique de similarité n-gram. Cette dernière contrôle la taille du lexique et maintient un seuil raisonnable pour chaque terme composé. Le résultat de comparaison peut être l'un des cas suivants :

1.  $S_{Ic1} = S_{Ic2}$

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

---

2.  $S_{Ic1} \subseteq S_{Ic2}$
3.  $S_{Ic1} \supseteq S_{Ic2}$
4.  $S_{Ic1} \neq S_{Ic2}$

La normalisation transforme un nom de classe en un ensemble d'unités lexicales appelées tokens. Un nom est découpé en certains tokens grâce à la ponctuation, à la casse (majuscule), aux symboles spéciaux, aux chiffres, par exemple, le nom de classe "AirCanada " est découpé en deux tokens "Air " et "Canada". Elle contient aussi une expansion de token : les abréviations, les acronymes sont dilatés, par exemple le token "WS " est élargi à " Web ", "Sémantique". Cette expansion est réalisée grâce à un dictionnaire externe, dont chaque entrée est une paire composée d'un token (abréviation ou acronyme) et d'un ensemble de mots qui correspondent au token. Ce dictionnaire est soit établi spécialement pour le domaine des ontologies à aligner, soit il s'agit d'un dictionnaire général contenant des termes communs et ce dernier est le dictionnaire adopté pour notre approche. Les tokens dans l'ensemble de tokens ont rendu minuscules afin d'être comparés antérieurement.

La déduction de la relation entre  $C_1$  et  $C_2$  est effectuée en utilisant les caractéristiques de la logique de description DL décrivant l'ontologie comme il a été expliqué dans la section 3 du chapitre I. La connaissance de base de DL est exprimée par la paire:  $\langle T, A \rangle$ , où :

- $T$  est la terminologie de box (TBox), contient un ensemble fini de définition de concepts et de rôle. La définition des concepts est exprimée par les axiomes terminologiques de la forme  $C_1 \subseteq C_2$ ,  $C_1 \sqsubseteq C_2$ ,  $C_1 \equiv C_2$ , ou  $C_1 \perp C_2$ ; où  $C_1$  et  $C_2$  sont des concepts atomiques.
- $A$  est une connaissance affirmative T(Box), décrit les individus en les nommant et les spécifiant à ses concepts et rôles.

Plusieurs ABoxes peuvent être associées au même Tbox [Fournier et Philippe, 2005] à l'aide de la fonction d'association illustrée dans la section III.2.

La signification de chaque élément d'ontologie est assurée par une interprétation  $I$ , consistant en un ensemble non vide  $\Delta^I$ , appelé le domaine d'interprétation qui est composé d'un ensemble d'individus ou d'instances. Ainsi, une fonction d'interprétation assignant un ensemble d'individus  $A^I \subseteq \Delta^I$  à chaque concept atomique  $A$ , et un ensemble de paires

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

d'individus  $B^I \subseteq \Delta^I \times \Delta^I$  [Baader et al., 2003] à chaque relation binaire atomique  $B$ . Les domaines d'interprétation des figures III.1 et III.2 sont les suivants :

- $\Delta_1 = \{B. Moh, B. Anes, G. Billel, G. Hichem\}$  of  $O_1$
- $\Delta_2 = \{B. Anes, G. Billel\}$  of  $O_2$

La fonction d'association  $i_C$ , associant à  $O_1$  les concepts  $C_1, C'_1$  nommés respectivement *Man* et *Boy* aux ensemble d'instances nommés  $I_{C_1} = \{B. Moh, G. Billel, G. Hichem\}$  et  $I_{C'_1} = \{B. Anes\}$ . Cette fonction associe aussi les concepts  $C_2, C'_2$  appelés *Man* et *Lad* aux ensembles  $I_{C_2} = \{G. Billel\}$  et  $I_{C'_2} = \{B. Anes\}$ . Ensuite, l'alignement des concepts est assuré à l'aide des règles suivantes:

1. Les deux concepts  $C_1, C_2$  sont équivalents ( $C_1 \equiv C_2$ ), Si  $C_1^I = C_2^I$  pour toutes les interprétations  $I$  ( $I_{C_1} = I_{C_2}$ ).
2. Un concept  $C_1$  est subsumé par un concept  $C_2$  si et seulement si  $C_1^I \subseteq C_2^I$  ( $C_1 \supseteq C_2$ ), i.e. Si  $I_{C_1} \subseteq I_{C_2}$  ( $I_{C_1} \supseteq I_{C_2}$ ).
3. Les concepts  $C_1, C_2$  ( $C_1 \perp C_2$ ) sont disjoint, Si  $C_1^I \cap C_2^I = \emptyset$  ( $\emptyset$  désigne l'ensemble vide), i.e., Si  $I_{C_1} \neq I_{C_2}$ .

Ainsi, à partir de 1. nous pouvons écrire, Si  $I_{C_1} = I_{C_2}$  alors  $C_1 \equiv C_2$ . La relation d'équivalence  $\equiv$  permet l'ajout de deux relations ( $C_1 \subseteq C_2$ ) et ( $C_1 \supseteq C_2$ ), et les deux relations ( $C_1 \subseteq C_2$ ) et ( $C_1 \supseteq C_2$ ), permettent de dériver la relation ( $C_1 \equiv C_2$ ). Ces règles de dérivation d'appariement sémantique ont justifié et prouvé par inférence.

Les résultats d'alignement des concepts obtenus sont symbolisés sous la forme :  $(C_1, C_2, Rel)$ , et permet d'initialiser la base des ontologies appariées ( $Ao$ ). Cette dernière sera l'une des entrées de la prochaine phase de l'algorithme avec les entités des ontologies à aligner. Dans le cas de manque d'instances, un expert doit intervenir pour construire  $Ao$ .

Le deuxième point nous permet d'écrire: Si  $I_{C_1} \in I_{C_2}$  ( $I_{C_2} \in I_{C_1}$ ) alors  $C_1$  de  $O_1$  ( $C_2$  of  $O_2$ ) peut être subsumé par  $C_2$  de  $O_2$  ( $C_1$  of  $O_1$ ) ou pas et vice-versa. Cela est dû aux insuffisances dans la description de l'ontologie en termes d'instances.

Vu la possibilité de manque d'instances, le dernier point peut donner l'ensemble vide dans l'intersection des ensembles d'instances. En effet, une ontologie d'appariement temporaire ( $At$ ) est construite et elle sera traitée par la suite.

## Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

Les deux figures III.1 et III.2 nous permettent de déduire l'équivalence des ensembles  $I_{C_1'}$  et  $I_{C_2'}$  en appliquant la mesure de n-gram sur le couple  $(\{B.Anes\}, \{B.Anes\})$ ; ce qui implique l'équivalence entre les concepts *boy*, *lad*. En effet le tuple  $(Boy, Lad, \equiv)$  va initialiser la base de  $A_0$  et il sera illustré dans la figure III.3, mais le second tuple  $(Man, Man, \supseteq)$  sera le premier élément de  $A_t$  puisque  $I_{C_1} \supseteq I_{C_2}$ , où l'instance  $\{G.Billel\}$  de  $I_{C_2}$  appartient à  $I_{C_1}$ .

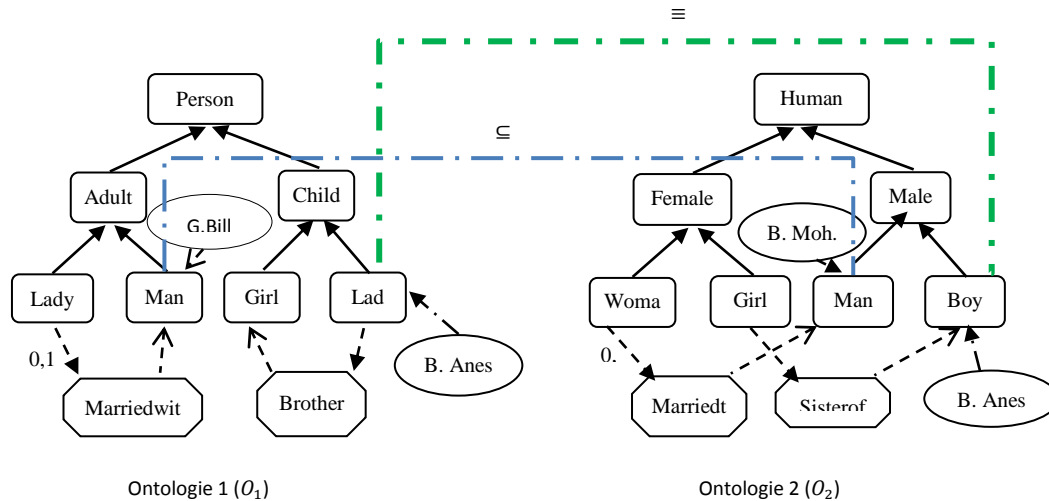


Figure III.3. Application de la phase 1 sur  $O_1$  et  $O_2$

### III.4. Phase 2 de l'algorithme

Cette section permet l'inférence de nouvelles relations à partir de celle de la section III.3 et l'alignement des relations binaires. Afin de prouver les résultats de ce dernier alignement nous définissons la restriction des concepts dans DL. Ensuite, ces relations sont comparées en basant sur la construction précédente.

#### III.4.1. Restriction

La restriction est le rôle exprimant une relation entre les concepts caractérisés par une restriction [Baader et al., 2003]. Elle peut être déclarée comme une propriété d'un concept en utilisant la syntaxe suivante en OWL-DL (par exemple le rôle : *Sisterof*):

```
<owl:Restriction>
<owl:onPropertyrdf:resource='auto#Sisterof' />
</owl:Restriction>
```

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

Supposons que la restriction  $R_1$  exprime le lien de concept  $C_1$  au concept noté  $C_1'$  de  $O_1$ , et  $R_2$  représente le lien de concept  $C_2$  au concept  $C_2'$  de  $O_2$ . Ils sont interprétés dans les équations (III.2) et (III.3) comme suit:

$$R_{1C_1}^{C_1'} = \{(x, y) \in \Delta^1 x \Delta^1 | (x, y) \in R_1^I \wedge x \in C_1^I \wedge y \in C_1^I\} \quad (\text{III.2})$$

$$R_{2C_2}^{C_2'} = \{(a, b) \in \Delta^1 x \Delta^1 | (a, b) \in R_2^I \wedge a \in C_2^I \wedge b \in C_2^I\} \quad (\text{III.3})$$

Le premier type de restriction de rôle  $R_{1C_1}^{C_1'}$ , interprète l'ensemble d'instances  $x$  de  $C_1'$  qui sont en relation avec les instances  $y$  appartenant à l'ensemble d'instances de  $C_1$ , et le deuxième rôle de restriction symbolisé par  $R_{2C_2}^{C_2'}$  décrit l'ensemble d'instances  $a$  de  $C_2'$  qui ont des relations avec l'ensemble d'instances  $b$  de  $C_2$ .

De plus, les relations binaires peuvent avoir des nombres de restrictions notés  $n_1$  et  $n_2$  ( $n_1$  et  $n_2$  sont des entiers positifs) restreignant les ensembles de cardinalités de  $R_1$  et  $R_2$  respectivement comme suit:

$$(\min_1 R_1)^I = \{y \in \Delta^I | |\{x | (x, y) \in R_1^I\}| \geq \min_1\} \quad (\text{III.4})$$

$$(\max_1 R_1)^I = \{y \in \Delta^I | |x| |\{(x, y) \in R_1^I\}| \leq \max_1\} \quad (\text{III.5})$$

$$(\min_2 R_2)^I = \{b \in \Delta^I | |\{a | (a, b) \in R_2^I\}| \geq \min_2\} \quad (\text{III.6})$$

$$(\max_2 R_2)^I = \{b \in \Delta^I | |\{a | (a, b) \in R_2^I\}| \leq \max_2\} \quad (\text{III.7})$$

Où,

- “|. |” Symbolize la cardinalité d'un ensemble dans chaque équation
- La restriction de au moins  $\min_1 R_1$  ( $\min_2 R_2$ ) designe la limite inférieure sur le nombre d'instances  $y(b)$  qui ont une relation binaire  $B_1(B_2)$  avec l'instance  $x(a)$  de  $C_1(C_2)$  (notée en OWL-DL par:

$$\langle owl:minCardinality \rangle \mathbf{min}_1 \langle /owl:minCardinality \rangle$$

- la restriction de au plus  $\max_1 R_1$  ( $\max_2 R_2$ ) indique la limite supérieure et elle est présentée en OWL-DL par:

$$\langle owl:maxCardinality \rangle \mathbf{max}_1 \langle /owl:maxCardinality \rangle$$

### III.4.2. Matching des relations binaires

Afin de comparer les relations binaires des concepts de  $Ao$  comme de  $C_1$  et  $C_2$  dans ses ontologies, nous supposons que :

- Le concept  $C_1$  est relié avec le concept  $C'_1$  de  $O_I$  par la relation binaire  $B_I$ , restreint par  $R_{1C_1}^{C'_1}$  mentionné dans l'équation (III.2);
- $R_{2C_2}^{C'_2}$  désigne la restriction de la relation binaire  $B_2$ , permet de relier  $C_2$  et  $C'_2$  dans  $O_2$ .
- Les concepts  $C_1$  et  $C_2$  peuvent être reliés et alignés de premier temps par l'une des relations de l'ensemble  $Rel_I = \{\equiv, \subseteq, \supseteq, \perp\}$ , mais après une itération la relation binaire  $B_i$  sera ajoutée à ce dernier ensemble et on aura :  $Rel_2 = \{\equiv, \subseteq, \supseteq, \perp, B_i\}$ .
- Il n'y avait pas de nombre de restriction, ce qui nous permet d'écrire:  $y = I_{C_1}$  et  $b = I_{C_2}$ .

Premièrement, Si  $C_1 \equiv C_2$  alors  $y = b$ , et quand on remplace:  $y$  par  $b$  et  $C_1$  par  $C_2$  dans l'équation (III.2), ainsi  $b$  par  $y$  et  $C_2$  par  $C_1$  dans l'équation (III.3); nous pouvons déduire les équations (III.8) et (III.9):

$$R_{1C_2}^{C'_1} = \{(x, b) \in \Delta^1 x \Delta^1 \mid (x, y) \in R_1^I \wedge x \in C_1^I \wedge b \in C_2^I\} \quad (III.8)$$

$$R_{2C_1}^{C'_2} = \{(a, y) \in \Delta^1 x \Delta^1 \mid (a, b) \in R_2^I \wedge a \in C_2^I \wedge y \in C_1^I\} \quad (III.9)$$

L'équation (III.8) signifie que les instances  $x$  de  $C'_1$  sont en relation  $B_1$  avec les instances  $b$  de  $C_2$  restreint par  $R_1$ , ce qui agrée l'insertion de tuple  $(C_2, C'_1, B_1)$  dans  $Ao$ . De plus,  $(C_1, C'_2, B_1)$  sera ajouté à cette base puisque de l'équation (III.9) nous avons les instances  $a$  de  $C'_2$  ont une relation avec ceux de  $C_1$  noté  $y$ . Ainsi, à partir de l'équation (III.1) et si les concepts  $C'_1$  et  $C'_2$  existent dans  $Ao$ , une des relations suivantes peut les relier (la relation  $(\equiv)$  peut être remplacée par  $(\subseteq, \supseteq)$ ):

- Si  $C'_1 \subseteq C'_2$  alors  $B_1 \subseteq B_2$
- Si  $C'_1 \supseteq C'_2$  alors  $B_1 \supseteq B_2$

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

En second lieu, Si  $C_1 \subseteq C_2 (C_1 \supseteq C_2)$  alors  $y \subseteq b (y \supseteq b)$ , cela signifie que l'ensemble d'instances de  $y(b)$  appartenant à l'ensemble  $b(y)$ . Ce qui implique que les instances de l'ensemble  $x(a)$  ont une relation avec les instances de l'ensemble  $b(y)$ , et ils ont aussi la même relation avec ceux de  $y(b)$  appartenant à  $b(y)$ .

De plus, les deux tuples  $(C_2, C'_1, B_1)$  et  $(C_1, C'_2, B_2)$  seront ajoutés aux matrices d'alignement des concepts par des relations binaires sources  $A_{BS}$  et des relations binaires cibles  $A_{BT}$  (target), respectivement.

Ainsi, le premier tuple  $(B_1, B_2, \subseteq)$  sera ajouté à la matrice des relations binaires  $A_{BR}$ , Si  $(C'_1 \subseteq C'_2)$ ; sinon c'est le deuxième tuple  $(B_1, B_2, \supseteq)$  qui sera ajouté à  $A_{BR}$  Si  $(C'_1 \supseteq C'_2)$  [Gherbi et Khadir, 2017]. Cependant, Si  $C'_1 \supseteq C'_2$  est dans le premier tuple ( $C'_1 \subseteq C'_2$  dans le deuxième tuple), alors, nous supposons l'ajout de la relation flou (fuzzy relation :&)  $(C'_1, C'_2, \&)$  à  $A_o$  et l'ensemble de relations redevient  $Rel_3 = \{\equiv, \subseteq, \supseteq, \perp, B_i, \&\}$ .

Néanmoins, l'existence de nombre de restrictions permet de faire ce traitement: Si  $(C_1 \equiv C_2 \vee C_1 \subseteq C_2 \vee C_1 \supseteq C_2)$  alors les mêmes résultats cités auparavant vont maintenir. Le remplacement de  $y$  par  $b$  dans les équations (III.4) et (III.5) quand  $C_1 \subseteq C_2$  et  $b$  par  $y$  dans les équations (III.6) et (III.7) quand  $C_1 \supseteq C_2$ ; nous donne, respectivement les équations (III.10), (III.11) et (III.13), (III.14).

$$(\min_1 R_1)^I = \{b \in \Delta^I \mid |\{x \mid (x, b) \in R_1^I\}| \geq \min_1\} \quad (III.10)$$

$$(\max_1 R_1)^I = \{b \in \Delta^I \mid |x \mid \{(x, b) \in R_1^I\}| \leq \max_1\} \quad (III.11)$$

$$(\min_2 R_2)^I = \{y \in \Delta^I \mid |\{a \mid (a, y) \in R_2^I\}| \geq \min_2\} \quad (III.12)$$

$$(\max_2 R_2)^I = \{y \in \Delta^I \mid |\{a \mid (a, y) \in R_2^I\}| \leq \max_2\} \quad (III.13)$$

Les équations (III.10) ou (III.11) permettent de relier les instances de  $C_2$  aux instances de  $C'_1$  avec un nombre minimal  $\min_1$  ou un nombre maximal  $\max_1$ . Ainsi, les équations (III.12) ou (III.13) interprètent les instances de  $C_1$  qui ont des relations avec ceux de  $C'_2$  par les restrictions *au moins* ou *au plus*.

L'application de ces équations sur le concept "Lad" de  $O_2$  de la figure III.2 qui a la relation "brotherof" avec "Girl" et les nombres de restriction : *min et max* appartenant à  $(0, N)$ ; nous donne "Lad brotherof Girl". De plus,  $(Boy \equiv Lad)$  est dans  $A_o$ . En effet

## Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

(*Lad, Girl, brotherof*) sera inséré à la base de  $A_{BT}$ , et dessinée dans la figure III.4 par une ligne pointillée.

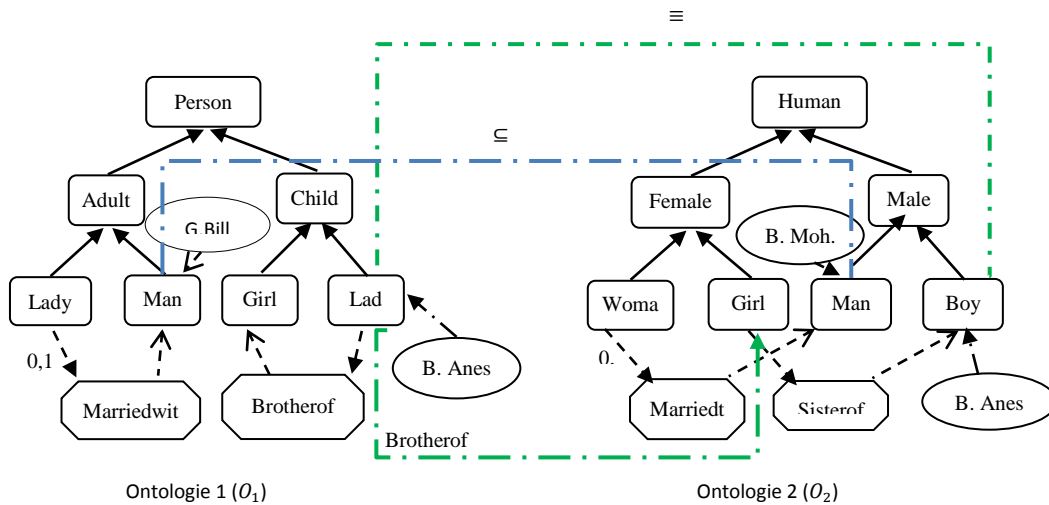


Figure III.4 Application de la phase 2 sur  $O_1$  et  $O_2$

### III.5. Phase 3 de l'algorithme : Appariement à l'aide de raisonneur et WordNet

Dans cette phase nous commençons par la validation des tuples de  $At$  et les insérer dans  $Ao$ . Ensuite, les méthodes terminologiques, les techniques d'inférence et de raisonnement sont utilisées avec WordNet pour offrir un appariement des ontologies.

#### III.5.1. Matching des concepts de la base $At$

Dans cette phase nous appliquons les instructions suivantes sur les noms des concepts afin de valider les tuples de  $At$  et les insérer dans  $Ao$ .

- Si  $((C_1 \subseteq C_2) \text{ or } (C_1 \supseteq C_2))$  et  $(S_{C_1} \equiv S_{C_2})$  alors  $C_1 \equiv C_2$
- Si  $(C_1 \subseteq C_2)$  et  $(S_{C_1} \subseteq S_{C_2})$  alors  $C_1 \subseteq C_2$
- Si  $(C_1 \supseteq C_2)$  et  $(S_{C_1} \supseteq S_{C_2})$  alors  $C_1 \supseteq C_2$
- Si  $(C_1 \supseteq C_2)$  et  $(S_{C_1} \subseteq S_{C_2})$  alors  $(C_1 \& C_2)$
- Si  $(C_1 \subseteq C_2)$  et  $(S_{C_1} \supseteq S_{C_2})$  alors  $(C_1 \& C_2)$
- Si  $(C_1 \perp C_2)$  et  $(S_{C_1} \text{ Rel } S_{C_2})$  alors  $C_1 \text{ Rel } C_2$

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

Les trois premières contraintes impliquent la suppression des alignements  $C_1 \subseteq C_2$  et/ou  $C_1 \supseteq C_2$  de  $At$  et l'insertion de ces derniers dans  $Ao$ , s'ils n'existent pas, et de la première contrainte  $C_1 \equiv C_2$  sera ajouté à  $Ao$ . La même chose pour  $C_1 \subseteq C_2$  et  $C_1 \supseteq C_2$ .

La quatrième et la cinquième contraintes signifient que  $C_1, C_2$  se chevauchent parce que nous pouvons déduire  $(C_1 \supseteq C_2)$  ou  $(C_1 \subseteq C_2)$ . Il est donc possible de les supprimer de  $At$ , et ajouter  $C_1 \& C_2$  à  $Ao$  sous forme de tuple écrit auparavant.

Cependant dans la dernière contrainte la relation  $Rel_1$  mentionnée dans la section III.4.2 exceptée  $\{\perp\}$ , sera maintenue et  $C_1 Rel C_2$  sera ajouté à  $Ao$ . Vu le manque d'instance cette relation exclue peut être fausse.

L'application de la première contrainte de matching des concepts de la base  $At$  sur les exemples illustratifs des ontologies  $O_1$  et  $O_2$  des figures III.1 et III.2, nous permet de déduire la relation d'équivalence entre  $Man$  et  $Man$  de  $O_1$  et  $O_2$ , respectivement. Ensuite, la suppression de cette dernière relation de  $At$  et son ajout à  $Ao$ , comme il sera illustré dans la figure III.5.

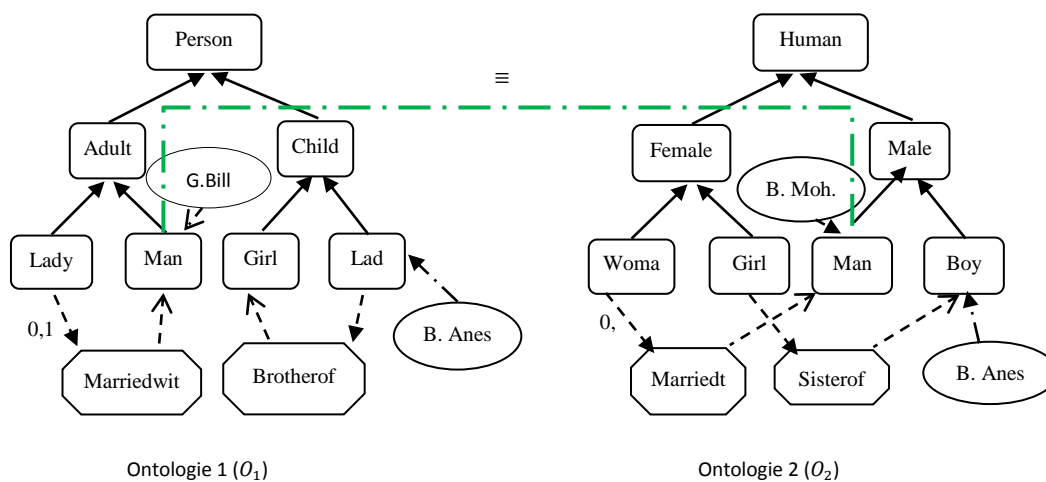


Figure III.5 Application de matching des concepts de la base  $At$  sur  $O_1$  et  $O_2$

#### III.5.2. Matching des concepts en utilisant un raisonneur et WordNet

Les méthodes terminologiques vont être exploitées pour comparer les noms des concepts voisins des concepts  $C_1$  et  $C_2$  alignés et existés dans  $Ao$ . Ce voisinage est cherché dans les ontologies originales. Cette recherche est faite parce que les concepts voisins ont plus de

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

---

chances d'être similaires [Euzenat et Shvaiko, 2007]. De plus, les noms des concepts comme:  $S_{C'_1}$  de  $C'_1$  et  $S_{C'_2}$  de  $C'_2$ , sont atomiques dans le langage OWL DL et apparaissent au plus une seule fois dans la partie gauche de l'axiome.

Les concepts voisins sont ceux qui ont une des relations de l'ensemble  $Rel_2$  mentionné dans la section III.4.2, par exemple:  $C'_1$  est un voisin de  $C_1$  dans  $O_1$  si et seulement si  $C_1 Rel_2 C'_1$ , et le voisin de  $C_2$  dans  $O_2$  est  $C'_2$  SSi  $C_2 Rel_2 C'_2$ . Nous alignons maintenant les concepts voisins par une relation binaire  $B_i$  avec  $C_1$  et  $B_j$  avec  $C_2$ , comme suit:

- Si  $((C_1 \equiv C_2) \text{ or } (C_1 \subseteq C_2))$  et  $(C_1 B_i C'_1)$  alors  $C_2 B_i C'_1$
- Si  $((C_2 \equiv C_1) \text{ or } (C_2 \subseteq C_1))$  et  $(C_2 B_j C'_2)$  alors  $C_1 B_j C'_2$

Ces contraintes permettent l'insertion de  $(C_2, C'_1, B_i)$  et/ou  $(C_1, C'_2, B_j)$  dans  $Ao$ . De plus, quand on déduit que  $(C'_1 \subseteq C'_2$  ou  $C'_1 \supseteq C'_2)$ , les relations  $(B_i \subseteq B_j$  ou  $B_i \supseteq B_j)$  sont aussi ajoutées à  $Ao$ .

Ensuite, nous apparions les concepts qui ont une des relations hiérarchiques  $Rel_1$  avec  $C_1$  et  $C_2$  en utilisant les méthodes terminologiques. Ces dernières méthodes souffrent de plusieurs inconvénients tel que :

- Synonymie : signifie l'existence d'une seule entité décrite par divers noms.
- Homonyme : nomme des entités différentes par le même nom.

Par conséquent, nous ne pouvons pas assumer avec certitude que deux entités sont différentes parce qu'elles ont des noms différents, ou qu'elles sont équivalentes si elles ont le même nom. Pour résoudre ces problématiques, des connaissances externes sous forme de ressources de données telles que WordNet ont exploité [Miller, 1995].

WordNet<sup>14</sup> est une base de données anglaise open source, se base sur des termes appelés des synsets. Un Synset est un ensemble de mots synonymes qui est considéré comme un concept. Ces concepts peuvent être reliés par des relations conceptuelles sémantiques et lexicales comme les structures d'hyponyme (super-concept/sous-concept) et de méronyme (partie d'une relation). Une brève description de chaque synset est assurée par un "gloss". Plusieurs synsets dissimilaires représentent les formes d'un mot par des significations

---

<sup>14</sup><http://wordnet.princeton.edu/wordnet/>

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

---

distinctes et des gloss différents. Ainsi, la signification de chaque forme dans WordNet est unique.

Dans ce document, nous supposons que WordNet est organisé hiérarchiquement comme  $W(S, \leq, A_g, g)$ , où  $S$  est un ensemble de synsets  $\{s_1, s_2, \dots, s_i\}$  ( $i$  est un nombre positif), et la fonction  $A_g$  associe le gloss  $g$  à tout synset. De plus, les relations  $\leq$  entre les synsets  $s_1, s_2$ , peuvent être présentées dans les relations logiques suivantes:

- 1)  $s_1 \subseteq s_2$  ; signifie que  $s_1$  est un hyponyme ou méronyme de  $s_2$ ;
- 2)  $s_1 \supseteq s_2$ ; exprime que  $s_1$  est un hyperonyme ou holonyme de  $s_2$ ;
- 3)  $s_1 \equiv s_2$ ; signifie que  $s_1$  et  $s_2$  appartenant au même synset sont similaires ;
- 4)  $s_1 \perp s_2$ : lorsque  $s_1$  et  $s_2$  sont des frères dans la partie de la hiérarchie, ils sont reliés par une relation d'antonymie.

Afin de découvrir la relation entre les concepts  $C'_1$  et  $C'_2$ , WordNet est exploité comme une base de connaissances complémentaires dans les systèmes BRMAP (Background Reasoner MAPping) [Gherbi et al., 2013]. Ce dernier est fondé sur une approche combinant les méthodes d'appariement terminologiques, structurelles et les connaissances complémentaires en intégrant le raisonneur Pellet. Cet outil est implémenté en trois phases, la première phase d'alignement calcule les valeurs de similarité entre les concepts de l'ontologie source et cible en exploitant les mesures terminologiques et structurelles. La deuxième phase d'ancrage apparie chaque concept non aligné à la phase précédente avec les concepts de l'ontologie complémentaire et donne des ancres. La dernière phase de dérivation s'appuie sur l'ontologie complémentaire, le raisonneur et les ancres pour définir des appariements.

Les entrées de raisonneur sont: les noms des concepts de l'ontologie source  $O_1$ , notés respectivement par  $S_{C'_1}$  et/ou le commentaire  $S_{m_1}$  (s'il existe), et ceux de l'ontologie cible  $O_2$ , symbolisés par  $S_{C'_2}$  pour le nom et/ou par  $S_{m_2}$  pour le commentaire (s'il existe). Les sorties vont être les relations entre les concepts en entrée. Les commentaires sont traités comme une option de cet outil.

Les noms comparés aux synsets en utilisant la mesure de n-gram, peuvent produire les points suivants :

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

---

- $S_{C'_1} \equiv s_1$  et  $S_{C'_2} \equiv s_2$ , s'il y a des synsets  $(s_1, s_2)$  équivalents aux noms en entrée  $(S_{C'_1}, S_{C'_2})$ ;
- $S_{C'_1} \perp s_1$  et  $S_{C'_2} \perp s_2$ , s'il n'y a pas de synsets équivalents à l'un de paire  $(S_{C'_1}, S_{C'_2})$ .

De plus, les commentaires  $S_{m_1}, S_{m_2}$ , vont être comparés aux glosses  $g_1, g_2$  des synsets  $s_1, s_2$  par la même mesure de similarité.

Une fois les ancres de la source et de la cible appartiennent au même synset, i.e.  $(s_1 \equiv s_2)$ ; alors les noms des concepts sont égaux et nous pouvons écrire:

- Si  $(S_{C'_1} \equiv s_1$  et  $S_{C'_2} \equiv s_2$  et  $s_1 \equiv s_2)$  alors  $S_{C'_1} \equiv S_{C'_2}$ ;

Cependant, si les synsets sont différents  $(s_1 \not\equiv s_2)$ , la distance entre eux est mesurée en utilisant la structure de hypernyme, incluse dans le raisonneur Pellet comme suit:

- Si  $((S_{C'_1} \equiv s_1)$  et  $(s_1 \subseteq s_2)$  et  $(S_{C'_2} \equiv s_2))$  alors  $(S_{C'_1} \subseteq S_{C'_2})$
- Si  $((S_{C'_1} \equiv s_1)$  et  $(s_1 \supseteq s_2)$  et  $(S_{C'_2} \equiv s_2))$  alors  $(S_{C'_1} \supseteq S_{C'_2})$
- Si  $((S_{C'_1} \equiv s_1)$  et  $(s_1 \perp s_2)$  et  $(S_{C'_2} \equiv s_2))$  alors  $(S_{C'_1} \perp S_{C'_2})$

Les trois dernières contraintes donnent  $(S_{C'_1} \subseteq S_{C'_2})$ ,  $(S_{C'_1} \supseteq S_{C'_2})$ ,  $(S_{C'_1} \perp S_{C'_2})$ , et permettent de déduire  $(C'_1 \subseteq C'_2)$ ,  $(C'_1 \supseteq C'_2)$ , et  $(C'_1 \perp C'_2)$ , respectivement. Ces dernières relations présentent les sorties du système et ils vont être ajoutés à  $Ao$ . Si les concepts  $C'_1, C'_2$  insérés ne sont pas disjoint  $(C'_1 \perp C'_2)$ , et ils ont des relations binaires avec les concepts existant dans la base, donc, ces dernières relations vont ajouter à la base comme il a été expliqué dans la section III.4.

Si l'un des concepts ajouté avait une relation binaire avec un autre concept dans son ontologie alors l'autre concept en relation sera ajouté avec cette relation binaire à  $Ao$ , s'il n'existait pas.

L'exemple de  $O_1$  de la figure III.1 comporte  $(Boy \subseteq Male)$  et  $(Man \subseteq Male)$ ;  $Ao$  contient "*Boy, Man*", et le voisin de ces derniers concepts est "*Male*". Ainsi, l'exemple de  $O_2$  dans la figure III.2 avait  $(Lad \subseteq Child)$  et  $(Man \subseteq Adult)$ ; les concepts "*Lad, Man*" sont inclus dans  $Ao$ , et "*Child, Adult*" sont les voisins de ces concepts. Selon l'approche présentée

### Chapitre III : inférer l'alignement des concepts à l'aide d'instances et de dictionnaire externe

auparavant, “Child,Adult” vont être comparés séparément avec “Male” en utilisant la métrique de n-gram pour trouver les synsets similaires à eux dans WordNet:

1. Le synset équivalent au “Male” est: ” **male**, male person (a person who belongs to the sex that cannot have babies)”;
2. Le synset équivalent au “Child” est: ”**child**, kid, youngster, minor, shaver, nipper, small fry, tiddler, tike, tyke, fry, nestling (a young person of either sex) "she writes books for children"; "they're just kids"; "tiddler' is a British term for youngster"”;
3. Le synset équivalent au “Adult” est: “**adult**, grownup (a fully developed person from maturity onward)”;

Le système BRMAP permet l'inférence des relations entre les synsets “Male” et “Child”, ainsi “Male” et “Adult”. Supposons que le système peut fournir  $(Male \subseteq Child)$  et  $(Male \subseteq Adult)$ , alors les tuples suivants vont ajouter à  $A_0$  et illustrer dans la figure III.6 :

- $(Male, Child, \subseteq)$  puisque  $(Male \equiv \mathbf{male} \text{ et } \mathbf{male} \subseteq Child)$ .
- $(Male, Adult, \subseteq)$  puisque  $(Male \subseteq \mathbf{adult} \text{ et } \mathbf{adult} \subseteq Adult)$ .

De la même manière, les relations entre les autres concepts des ontologies  $O_1$  et  $O_2$  vont être déduites.

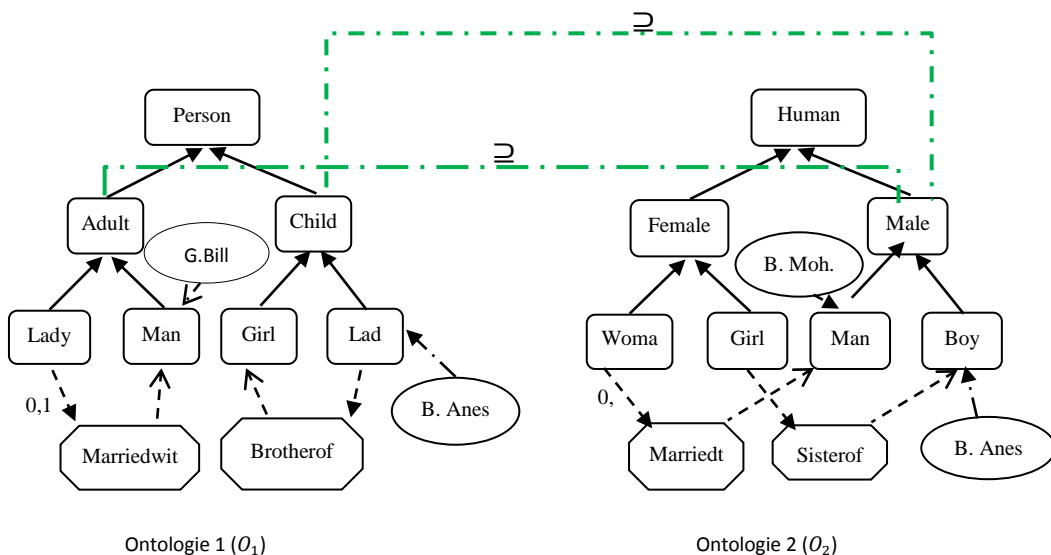


Figure III.6 Application de matching des concepts en utilisant un raisonneur et WordNet sur  $O_1$  et  $O_2$

### III.6. Evaluation et expérimentation

L'approche présentée au-dessus est implémentée par un système appelé ONTMAT [Gherbi et Khadir, 2017] dont l'interface est illustrée dans la figure III.4. Ce système est évalué sous la compagnie OAEI (Ontology Alignment Evaluation Initiative) de matching d'ontologie 2017<sup>15</sup>. Cette compagnie est choisie puisqu'elle évalue les techniques de matching d'ontologie, et estime leurs points fort et faible. Cependant, OAEI teste les appariements des entités en utilisant uniquement la relation d'équivalence, et ONTMAT offre des autres types de relations comme : la relation floue (&), et la relation binaire entre les concepts.

#### III.6.1. Implémentation et paramétrage

ONTMAT est implémenté en Java sous la Version: 3.5.0 de Eclipse SDK, comme un plugin de Protégé. De plus, le système BRMAP présenté dans la section III.5.2, et la mesure de similarité n-gram ont été exploitées pour fournir les relations terminologiques qui peuvent être entre les noms des entités. Les techniques d'inférences ont aussi implémenté pour dériver la relation entre les relations binaires.

ONTMAT [Gherbi et Khadir, 2017] a appliqué le test de conférence d'OAEI 2017 pour évaluer ses performances. La section IV.6.2 du quatrième chapitre explique en détail l'ensemble de données utilisées et les critères d'évaluation.

---

<sup>15</sup><http://oaei.ontologymatching.org/2017/>

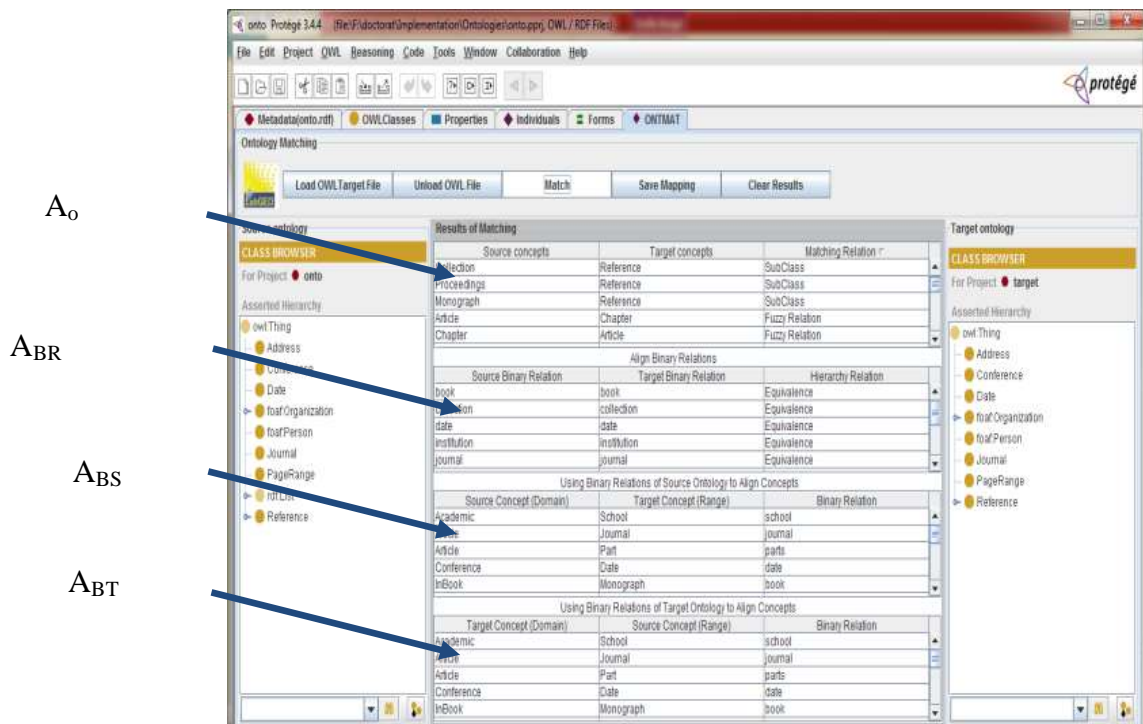


Figure III.7. Interface de ONTMAT

### III.6.2. Evaluation selon l'alignement de référence de crisp

Le tableau III.1 contient les résultats d'évaluation d'ONTMAT en appliquant toutes les variantes combinaisons d'évaluation des alignements de référence de crisp. Il existe des mesures de précision, rappel, F1-mesure, F2-mesure et F.5-mesure, calculées pour le seuil fournissant la moyenne la plus élevée F1-mesure évalué pour chaque appariement. Le F1-mesure est la moyenne harmonique de précision et de rappel. Le F2-mesure (pour bêta = 2) donne au rappel une valeur plus que la précision, et le F.5- mesure F0,5 (pour bêta = 0,5) évalue la précision plus que le rappel.

Tableau III.1. Résultats basés sur les alignements de référence de crisp de OAEI 2017<sup>16</sup>, [Gherbi et Khadir, 2017].

Évaluat.	Precision	Recall	F1-meas	F2-meas	F.5-meas
ra1-M1	0.11	0.52	0.18	0.3	0.13
ra1-M3	0.06	0.07	0.11	0.19	0.43

<sup>16</sup> <http://oaei.ontologymatching.org/2017/conference/eval.html#ra1-M1>

<b>ra2-M1</b>	0.11	0.13	0.18	0.28	0.46
<b>ra2-M3</b>	0.06	0.07	0.1	0.19	0.39
<b>rar2-M1</b>	0.11	0.13	0.18	0.29	0.49
<b>rar2-M3</b>	0.06	0.07	0.1	0.19	0.41

### III.7. Conclusion

La théorie dans ce chapitre a démontré avec succès que l'alignement des concepts par des relations autres que la relation l'équivalence telle que: la relation floue, les relations de subsomption et la relation binaire, est possible. De plus, la déduction de l'alignement des concepts à partir de l'alignement de ses instances. Ainsi, la relation binaire peut être également alignée à l'aide d'une relation hiérarchique. Cependant OAEI 2017 aient certaines limites en termes d'évaluation, i.e. n'évalue que les relations d'équivalence découvertes entre les entités.

Les résultats offerts par ONTMAT ne sont pas bons car il offre la possibilité de choisir entre les alignements des concepts déduits à partir de l'alignement d'instances et ceux offert par les méthodes terminologiques. Ainsi, cet outil peut négliger les résultats produits par l'une les étapes 1 et 3 de l'algorithme. Ce qui nous a amené à proposer un autre outil appelé ONTMAT1 [Gherbi et Khadir, 2020]. Ce dernier utilise le raisonneur pellet, le dictionnaire WordNet, et un poids qui se calcule automatiquement à l'aide des propriétés de restriction pour combiner les diverses similarités. Le chapitre IV de cette thèse explique en détail l'outil ONTMAT1.

*Chapitre IV :*

***ONTMAT1: Matching d'ontologies***

*en utilisant un raisonneur*

*et une propriété de restriction*

## Chapitre IV

### *ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction*

#### Sommaire

IV.1. Architecture d'ONTMAT1	111
IV.2. Phase 1: Prétraitement des ontologies	113
IV.3. Phase 2: Affectation de poids à la restriction de propriété	115
IV.4. Phase 3: Similarité structurelle entre les concepts	117
IV.5. Phase 4: Agrégation des similarités	119
IV.6. Expérimentation et évaluation	120
IV.6.1. Implémentation et paramétrage	121
IV.6.2. Données et critères d'évaluation	122
IV.6.2.1. Evaluation de l'alignement de référence de crisp	123
IV.6.2.2. Résultats d'estimations basée sur la version incertaine d'alignement de référence	127
IV.7. Conclusion	129

Dans ce chapitre nous présentons une approche automatique de matching d'ontologie appelée ONTMAT1 [Gherbi et Khadir, 2020]. Les similarités entre les entités ont été calculées à l'aide de méthodes terminologiques, le dictionnaire WordNet et le reasoner Pellet. Pour résoudre le problème de combinaison de diverses similarités calculées auparavant, un poids basé sur la sémantique de la propriété de restriction et ses propriétés des individus et de datatype est évalué. Ce poids estime les performances des similarités structurelles et linguistiques calculées.

## **ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction**

---

L'objectif de ce poids est de pallier les inconvénients des stratégies d'agrégation illustrées dans la section 2, et de poids nommé *harmony*. Ce dernier [Mao et al., 2010] exprime une similarité de valeur considérable comme une similarité plus fiable et lui affecte un poids plus élevé lors de l'agrégation, et filtre les similarités de petites valeurs comme des fausses correspondances. Cet *harmony* dépend des similarités de chaîne de caractères, structurelle et il n'a aucune base sémantique. Cependant, le poids introduit se base sur la sémantique des classes et apprécie la qualité des similarités.

La suite de ce chapitre est organisée comme suit: dans la section 1, nous présentons les principes et l'architecture du système proposé. Nous abordons les quatre étapes de ce système dans les sections 2,3,4,5. Nous discutons dans la section 6 l'implémentation, le paramétrage, les données et les critères d'évaluation d'ONTMAT1.

### **IV.1. Architecture d'ONTMAT1**

L'architecture du système de matching proposé est illustrée dans la figure IV.1. Cette dernière, montre que l'algorithme suggéré peut aligner les individus, les concepts, en utilisant les méthodes des chaînes de caractères, linguistique, structurelle, et sémantique, ainsi que le reasoner et le dictionnaire WordNet, dans la première phase. Ensuite dans la deuxième phase de l'algorithme, les méthodes structurelles et sémantiques basées sur les restrictions des propriétés offrent un poids qui estime l'impact des propriétés *d'individu* et de *datatype* sur les classes. La troisième phase évalue les similarités entre les noms des propriétés de restriction et des propriétés des données en utilisant un reasoner, WordNet, et *n*-gram pour fournir les similarités structurelles entre les concepts. La dernière phase agrège les différentes similarités en une seule valeur en exploitant le poids proposé, et propose une nouvelle relation appelée la relation floue (*fuzzy relation*).

Le tableau IV.1 présente des parties de deux concepts *product* et *volume* extraites de deux ontologies décrites en OWL: *culture-shop.owl* and *library.owl*, respectivement (Euzenat et Shvaiko, 2013). Ces ontologies vont être exploitées par la suite afin de montrer le déroulement de différentes phases de l'algorithme proposé.

*Tableau IV.1. Concepts des ontologies [Gherbi et Khadir, 2020]*

<i>culture-shop.owl</i>	<i>library.owl</i>
<code>&lt;owl:Class rdf:ID="Product"&gt;</code>	<code>&lt;owl:Class rdf:ID="Volume"&gt;</code>
...	...
<code>&lt;owl:Restriction&gt;</code>	<code>&lt;owl:Restriction&gt;</code>
<code>&lt;owl:onProperty</code>	<code>&lt;owl:onProperty rdf:resource="#author"</code>
<code>rdf:resource="#name"/&gt;</code>	<code>/&gt;</code>
<code>&lt;owl:maxCardinality</code>	<code>&lt;owl:minCardinality</code>
<code>rdf:datatype="&amp;xsd;nonNegativeInteger"&gt;1&lt;/owl:maxCardinality&gt;</code>	<code>rdf:datatype="&amp;xsd;nonNegativeInteger"&gt;1&lt;/owl:minCardinality&gt;</code>
<code>&lt;/owl:Restriction&gt;</code>	<code>&lt;/owl:Restriction&gt;</code>
...	...
<code>&lt;owl:Restriction&gt;</code>	<code>&lt;owl:Restriction&gt;</code>
<code>&lt;owl:onProperty</code>	<code>&lt;owl:onProperty rdf:resource="#title"</code>
<code>rdf:resource="#writer"</code>	<code>/&gt;</code>
<code>/&gt;&lt;owl:minCardinality</code>	<code>&lt;owl:minCardinality</code>
<code>rdf:datatype="&amp;xsd;nonNegativeInteger"&gt;1&lt;/owl:minCardinality&gt;</code>	<code>rdf:datatype="&amp;xsd;nonNegativeInteger"&gt;1&lt;/owl:minCardinality&gt;</code>
<code>&lt;/owl:Restriction&gt;</code>	<code>&lt;/owl:Restriction&gt;</code>
<code>...&lt;/owl:Class&gt;</code>	<code>...&lt;/owl:Class&gt;</code>

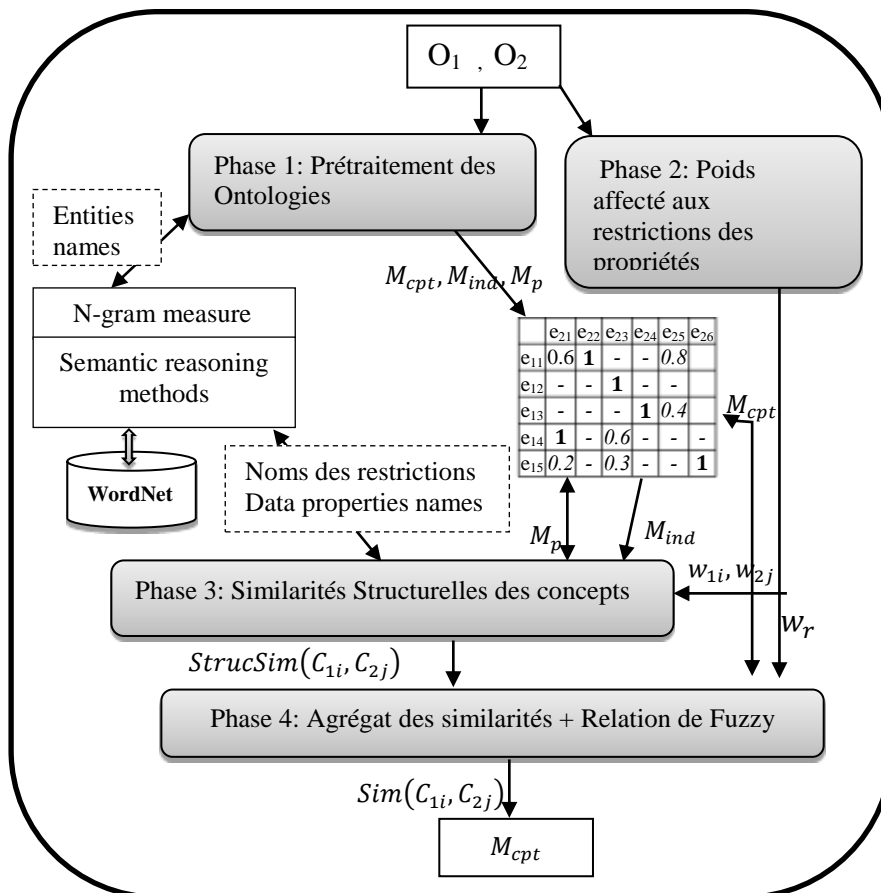


Figure IV.1. Architecture de ONTMAT1 [Gherbi et Khadir, 2020]

## IV.2. Phase 1: Prétraitement des ontologies

Dans cette phase, le système exploite des techniques terminologiques pour comparer les noms d'instances et de concepts d'ontologies à apparier vu que chaque entité en ontologie est étiquetée (Ngan et al., 2009) par un nom unique pour chaque concept. Les résultats obtenus ont ensuite enregistré dans une matrice des individus ( $M_{ind}$ ) et une matrice des concepts ( $M_{cpt}$ ). Pour simplifier l'explication de ce système, les noms des entités des ontologies  $O_1$  et  $O_2$  (instances et classes) ont représenté respectivement par  $e_{1i}, e_{2j}$  ( $i$  et  $j$  sont des entiers positifs). Les lignes de chaque matrice représentant les noms des entités des ontologies sources et les colonnes, indiquent ceux de l'ontologie cible.

Les techniques de normalisation telles que: la lemmatisation (Euzenat & Shvaiko, 2007), ont appliqué sur chaque nom d'entité de la matrice, accompagné par une mesure de  $n$ -gram

estimant les similarités entre les entités. Cette mesure est une séquence adjacente de  $n$  caractères de séquence de texte (Essayeh & Abed, 2015). Le système ONTMAT1 a considéré que  $n = 3$  où:

$$TriGram(e_{1i}, e_{2j}) = \frac{TriGram(e_{1i}) \cap TriGram(e_{2j})}{\min(|e_{1i}|, |e_{2j}|) - 2} \quad (IV.1)$$

L'utilisation des métriques pour aligner les entités peut poser plusieurs inconvénients. Afin de pallier les inconvénients de synonymie et d'homonymie, le dictionnaire WordNet et un reasoner Pellet ont exploité.

Le reasoner Pellet a été choisi car il est une librairie open-source Java implémentant des raisonnements applicables aux ontologies OWL. Cette librairie est exploitable en conjonction avec Jena et les API OWL. Il permet de faire des tests de subsomption et d'autres fonctionnalités comme il a été expliqué dans la section I.4.

Pellet est un reasoner puissant basé sur l'algorithme des tableaux, et le seul qui prenne en charge tous les classes énumérées et les types de données définis par l'utilisateur [Mindswap, 2003], [Sirin et al, 2007].

Les noms des entités  $e_{1i}$  et  $e_{2j}$  vont être les entrées de WordNet et Pellet, et la sortie sera la relation entre ces entités. Les noms des entités vont être comparés avec les synsets en utilisant la mesure de similarité  $n$ -gram (où  $n=3$ ) et le résultat produit est :  $e_{1i} \equiv s_1$  and  $e_{2j} \equiv s_2$  ; Si les synsets ( $s_1, s_2$ ) sont équivalents aux entrées ( $e_{1i}, e_{2j}$ ).

Une fois les synsets équivalents à la source et à la cible appartiennent au même synset (i.e.  $s_1 \equiv s_2$ ), donc les noms des concepts sont égaux et nous pouvons écrire:

- Si ( $e_{1i} \equiv s_1$  et  $e_{1j} \equiv s_2$  et  $s_1 \equiv s_2$ ) alors  $e_{1i} \equiv e_{1j}$ .

Étant donné les concepts équivalents  $e_{1i}$  et  $e_{1j}$  ( $e_{1i} \equiv e_{1j}$ ), l'alignement entre eux est considéré comme égal à 1.0. La moyenne pondérée est utilisée pour agréger cette valeur avec la valeur enregistrée dans leur matrice par la mesure de tri-gram, et la similarité obtenue est attribuée à l'intersection entre ces entités dans cette matrice. Sinon, la valeur existante est

conservée. La similitude attribuée à l'intersection entre ces entités dans chaque matrice est notée :  $TermSim(e_{1i}, e_{2j})$ .

### IV.3. Phase 2: Affectation de poids à la restriction de propriété

La restriction de propriété est un type spécial de description d'une classe. Elle peut spécifier la fonction d'une propriété appliquée sur une instance d'une classe spécifique. Elle permet aussi de définir les propriétés à l'intérieur d'un contexte particulier d'une classe de restrictions [Hebeler et al., 2011]. Ainsi, l'appariement basé sur le contexte est exprimé dans les catégories syntaxique et sémantique [Euzenat et Shvaiko, 2013]. Par conséquent, dans cette phase, un poids estimant l'impact sémantique de la restriction de propriété (propriétés d'objet et propriétés datatype) sur la similarité structurel de concept est proposé. Ce poids est représenté par  $w_r$  dans Eq. (IV.2).

Le sous langage OWL-DL d'OWL (Ontology Web Language) certifié par (W3C)<sup>17</sup>, est opté dans cette recherche pour décrire notre algorithme de matching des ontologies. Ce langage est choisi parce qu'il distingue deux types de restrictions de propriété: les contraintes de valeur et les contraintes de cardinalité. Une contrainte de valeur applique des contraintes sur la plage de propriété. Ces contraintes sur la classe  $C$  ou sur un objet  $o$  peuvent être représentées par les fonctions:  $allValuesFrom(C)$ ,  $someValuesFrom(C)$ , ou  $hasValue(o)$ . Une contrainte de cardinalité est présentée par  $maxCardinality(n)$  et  $minCardinality(n)$ , où  $n$  est le nombre de valeurs qu'une propriété peut prendre.

$allValuesFrom(C)$  est l'équivalent de quantificateur universel (pour tous  $\forall$ ) de la logique des prédicats, où pour chaque instance de  $C$ , chaque valeur de la propriété doit satisfaire les contraintes, donc nous pouvons affirmer que cette propriété a une forte influence sur la classe. Par conséquent, d'après sa sémantique, l'effet de cette restriction sur la classe est considéré comme «fort», et 1.0 est suggéré comme poids ( $w_{1i}, w_{2j}$  dans  $O_1, O_2$ , respectivement) affecté à  $allValuesFrom$  par le system.

$someValuesFrom(C)$  est l'équivalent au quantificateur existentiel dans la logique des prédicats de sorte que pour chaque instance de  $C$ , il existe au moins une valeur de la propriété

---

<sup>17</sup> <https://www.w3.org/>

## Chapitre IV : ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

---

qui satisfait la contrainte. L'impact de cette contrainte sur une classe donnée, peut être estimé par une moyenne, et la valeur 0.75 est assignée au  $w_{1i}$  dans  $O_1$  et  $w_{2j}$  dans  $O_2$ .

*hasValue(o)* lie une classe de restriction à une valeur  $o$ , qui peut être une valeur d'individu ou de données (data). Cette restriction décrit une classe de tous les individus pour lesquels la propriété concernée a au moins une valeur sémantique équivalente à  $o$  (elle peut avoir également des valeurs supplémentaires). L'impact de cette restriction est considéré comme faible et le poids 0.25 est affecté au  $w_{1i}$  dans  $O_1$  et  $w_{2j}$  dans  $O_2$ .

*owl:maxCardinality(n)* décrit une classe dont tous les individus ont au plus  $n$  valeurs différentes (individus ou valeurs de données) de la propriété concernée. Cette contrainte n'impacte que les  $n$  valeurs et pour cette raison, elle est considérée comme une contrainte faible. ONTMAT1 attribue une valeur de 0.25 aux poids  $w_{1i}, w_{2j}$  dans  $O_1, O_2$ , respectivement. De plus, il en est de même pour *minCardinality(n)* décrivant une classe pour tous les individus qui ont au moins  $n$  valeurs différentes de la propriété concernée.

$$w_r = \frac{\sum_{\substack{1 \leq i \leq m \\ 1 < j < p}} (1 - |w_{1i} - w_{2j}|)}{\max(m, p)} \quad (\text{IV.2})$$

Où :

- $m$  et  $p$  dénotent le nombre de restrictions dans la description d'une classe dans les ontologies source et cible respectivement.
- $w_{1i}, w_{2j}$  symbolisent les poids proposés affectés aux  $DataSim(D_{1i}, D_{2j})$  dans l'équation (IV.3) et  $IndSim(ind_{1i}, ind_{2j})$  dans l'équation (IV.4), fournissant les valeurs maximales de  $RSimData(D_{1i}, D_{2j})$  et  $RSimInd(ind_{1i}, ind_{2j})$ , respectivement.
- $max$  désigne la valeur maximale entre  $m$  et  $p$ .

L'application de l'équation (IV.1) sur l'exemple du Tableau 1 donne la valeur suivante de  $w_r$ :

$$w_r = \frac{1 - |0.25 - 0.25| + 1 - |0.25 - 0.25|}{2} = \mathbf{1}$$

#### IV.4. Phase 3: Similarité structurelle entre les concepts

La similarité structurelle entre les concepts dans ce chapitre est fondée sur la comparaison des restrictions des propriétés. Ces dernières peuvent être appliquées soit sur les propriétés de *datatype* (où, le type de données est la valeur de la propriété), ou la propriété de l'*object* (où, l'individu est la valeur de la propriété)<sup>18</sup>. La restriction est déclarée par le constructeur '*owl:Restriction*', et la propriété à laquelle se rapporte la restriction désignée par '*owl:onProperty*' (Hebeler et al., 2011).

Premièrement, les noms des restrictions (identifiés par '*owl:onProperty rdf:resource=*') des concepts  $(C_{1i}, C_{2j})$  à appairer, ont comparé à l'aide des méthodes terminologiques illustrées dans la section IV.2, et donnent  $RNameSim(Rest_{1i}, Rest_{2j})$ .

En second lieu, les mêmes méthodes terminologiques ont exploité pour mesurer les similarités entre les noms de la propriété *datatype* des concepts à aligner. Ensuite, la moyenne de ces similarités, dénotée par  $Datasim(D_{1i}, D_{2i})$  est calculée.

Finalement, les similarités entre les individus des concepts à appairer ont extrait de  $M_{ind}$ , pour calculer leur moyenne, dénoté  $IndSim(ind_{1i}, ind_{2i})$ .

Ensuite, 1 moins la différence entre les poids  $w_i$  and  $w_j$  (évaluent l'influence de la propriété sur les concepts dans Eq. (IV.2), est multiplié par  $DataSim(C_{1i}, C_{2i})$  et  $IndSim(C_{1i}, C_{2i})$ , comme il sera illustré dans les équations (IV.3) et (IV.4), respectivement.

$$RSimData(D_{1i}, D_{2j}) = \frac{RNameSim(Rest_{1i}, Rest_{2j}) + (1 - |w_{1i} - w_{2j}|) * DataSim(D_{1i}, D_{2j})}{2} \quad (IV.3)$$

$$RSimInd(ind_{1i}, ind_{2j}) = \frac{RNameSim(Rest_{1i}, Rest_{2j}) + (1 - |w_{1i} - w_{2j}|) * IndSim(ind_{1i}, ind_{2j})}{2} \quad (IV.4)$$

De plus, la similarité finale de la propriété *datatype* entre les concepts  $(C_{1i}, C_{2j})$  dans  $FinalDataSim(C_{1i}, C_{2j})$ , est la moyenne de meilleurs résultats fournis par  $RSimData(D_{1k}, D_{1l})$  (où  $k$  et  $l$  sont des entiers), comme il est illustré dans l'équation (IV.5).

---

<sup>18</sup> <https://www.w3.org/TR/owl-ref/>

## Chapitre IV : ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

L'autre similarité est basée sur la propriété *object*, dénotée par  $FinalIndSim(C_{1i}, C_{2j})$ , est la moyenne de meilleurs résultats fournis  $RestIndSim(ind_{1k}, ind_{2l})$ , comme il est présenté dans l'équation (IV.6). La similarité structurelle entre les concepts est ainsi la moyenne des résultats donnés par les équations (IV.5) et (IV.6), où  $d_{ind}(d_D)$  prend la valeur 1.0 s'il y avait des individus (le type data) dans la restriction. Sinon, la valeur 0 est appliquée à  $d_{ind}(d_D)$ .

$$FinalDataSim(C_{1i}, C_{2j}) = \frac{\sum_{k=0, l=0}^{m, p} \max(RSimData(D_{1k}, D_{2l}))}{\max(m, p)} \quad (IV.5)$$

$$FinalIndSim(C_{1i}, C_{2j}) = \frac{\sum_{k=0, l=0}^{m, p} \max(RSimInd(ind_{1k}, ind_{2l}))}{\max(m, p)} \quad (IV.6)$$

$$StrucSim(C_{1i}, C_{2j}) = \frac{d_D * FinalDataSim(C_{1i}, C_{2j}) + d_{ind} * FinalIndSim(C_{1i}, C_{2j})}{d_D + d_{ind}} \quad (IV.7)$$

Tableau IV.2. Résultats des similarités terminologiques [Gherbi et Khadir, 2020]

	<i>Volume</i>	<i>Author</i>	<i>title</i>	<i>nonNegativeInteger</i>
<i>Product</i>	0	/	/	/
<i>Name</i>	/	0	0	/
<i>Writer</i>	/	1	0	/
<i>nonNegativeInteger</i>	/	/	/	1

Le Tableau IV.2 présente les similarités terminologiques entre les noms des entités, les propriétés d'*object*, et les propriétés de *data* de l'exemple illustratif du tableau IV.1. L'alignement entre les propriétés d'*object* 'writer' et 'author', et les propriétés de *data* 'nonNegativeInteger' ont la plus grande valeur de similarité. Cependant, il n'offre aucune possibilité d'alignement entre les entités 'product' et 'volume' ; et l'autre nom de propriété d'*object*. Ainsi, ces résultats sont basés sur les équations (IV.3), (IV.5), et (IV.7); et les résultats suivants peuvent être obtenus:

## ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

---

$RSimData(writer, author)$

$$= \frac{RNameSim(writer, author) + (1 - |0.25 - 0.25|) * DataSim(nonNegativeInteger, nonNegativeInteger)}{2}$$
$$= \frac{1 + 1 * 1}{2} = 1$$

$$FinalDataSim = \frac{\max(RSimData(name, author), RSimData(name, title)) + \max(RSimData(writer, author) + RSimData(writer, title))}{2}$$
$$= \frac{0.5 + 1}{2} = 0.75$$

$$StrucSim(Product, Volume) = \frac{0+1*0.75}{0+1} = 0.75$$

### IV.5. Phase 4: Agrégation des similarités

La méthode d'agrégation de différentes similarités est optée dans cette phase de notre algorithme. Par conséquent, l'alignement d'une paire de concepts noté par  $Sim(C_{1i}, C_{2j})$  est définie par l'équation (IV.8), où, selon l'influence de la restriction sur les propriétés de *datatype* et *individual* sur la classe, les poids ( $w_i, w_j$ ) ont évalué comme il a été expliqué dans les sections IV.3 et IV.4.

L'impact de la similarité terminologique est ensuite amené à : 1 moins la moyenne structurelle des poids. La valeur de  $Sim(C_{1i}, C_{2j})$ , est assignée à l'intersection entre  $(C_{1i}, C_{2j})$  dans la matrice des concepts alignés pour présenter l'alignement final entre eux.

$$Sim(C_{1i}, C_{2j}) = (1 - w_r) * TermSim(C_{1i}, C_{2j}) + StructSim(C_{1i}, C_{2j}) \quad (IV.8)$$

En outre, le système développé propose une nouvelle relation appelée la relation flow (fuzzy) qui indique que le system proposé ne peut pas décider concernant la relation potentielle entre les entités à apparier.

Le problème de domination de la similarité structurelle dans l'équation (IV.2) apparait une fois la valeur de la similarité structurelle est plus grande que celle de la similarité terminologique, ce qui implique que cette dernière n'a eu aucune influence sur le résultat final.

Cette relation est affectée aux concepts dont la différence entre les valeurs calculées  $TermSim(C_{1i}, C_{2j})$  et  $StructSim(C_{1i}, C_{2j})$  dépasse un certain seuil. Ce seuil est calculé en fonction de l'expertise de l'application en OAEI. Les relations flow entre les concepts sont représentées par  $-1$  pour évaluer l'intersection entre les concepts dans la matrice  $M_{cpt}$  présentée dans la section IV.2.

Les similarités terminologiques ont une grande importance dans le domaine de matching d'ontologies. Cependant, une classe peut être décrite sans aucune propriété de restrictions et la valeur 0 peut être assignée à  $StructSim(C_{1i}, C_{2j})$  dans l'équation (IV.8).

La similarité entre les entités de l'exemple du tableau IV.1 était 0.75, et la similarité terminologique, était 0, donc 'Product' et 'Volume' sont reliés par la relation flow car la similarité structurelle avec  $w_r=1$  s'est imposée dans l'équation (IV.8).

$$Sim(Product, Volume) = (1 - 1) * 0 + 0.75 = \mathbf{0.75}$$

### IV.6. Expérimentation et évaluation

Proposer une nouvelle approche ne peut être valable tant qu'une évaluation objective appropriée, n'est pas effectuée. Ici, ONTMAT1 a été évalué dans le cadre de la campagne d'appariement d'ontologies OAEI 2019. Ces tests sont retenus dans ce travail car OAEI établit des normes de campagne d'évaluation visant à examiner les techniques d'appariement d'ontologie. Ainsi cette campagne évalue les points forts et faibles des systèmes de matching afin d'améliorer les techniques d'évaluation et de comparer les performances des systèmes participants.

De plus, OAEI teste les appariements des entités en utilisant uniquement les relations d'équivalence. Cependant, ONTMAT1 offre d'autres options en termes de relations d'alignement entre entités, appelées relations floues, qui ne peuvent pas être testées sous la plateforme de SEALS<sup>19</sup>.

---

<sup>19</sup> <http://oaei.ontologymatching.org/2019/seals/index.html>

## ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

### IV.6.1. Implémentation et paramétrage

La version d'ONTMAT1<sup>20</sup> testée sous la plateforme de SEAL est disponible sur le site github, est implémentée comme illustré dans la figure IV.2 en Java sous la version Mars.2 Release (4.5.2) d'Eclipse Java EE IDE de Développeurs de Web. De plus, le raisonneur Pellet et WordNet ont été utilisés pour inférer les relations entre les ancres des entités à mettre en correspondance et fournir un ensemble de relations entre les noms d'entités des ontologies à appairer. La mesure de n-gram a été utilisée pour comparer ces noms et identifier les similarités entre eux. La relation floue symbolisée par  $-1$ , a déduit les relations entre les noms des concepts. La figure IV.3 montre un imprime écran de ONTMAT1. [Gherbi et Khadir, 2019].

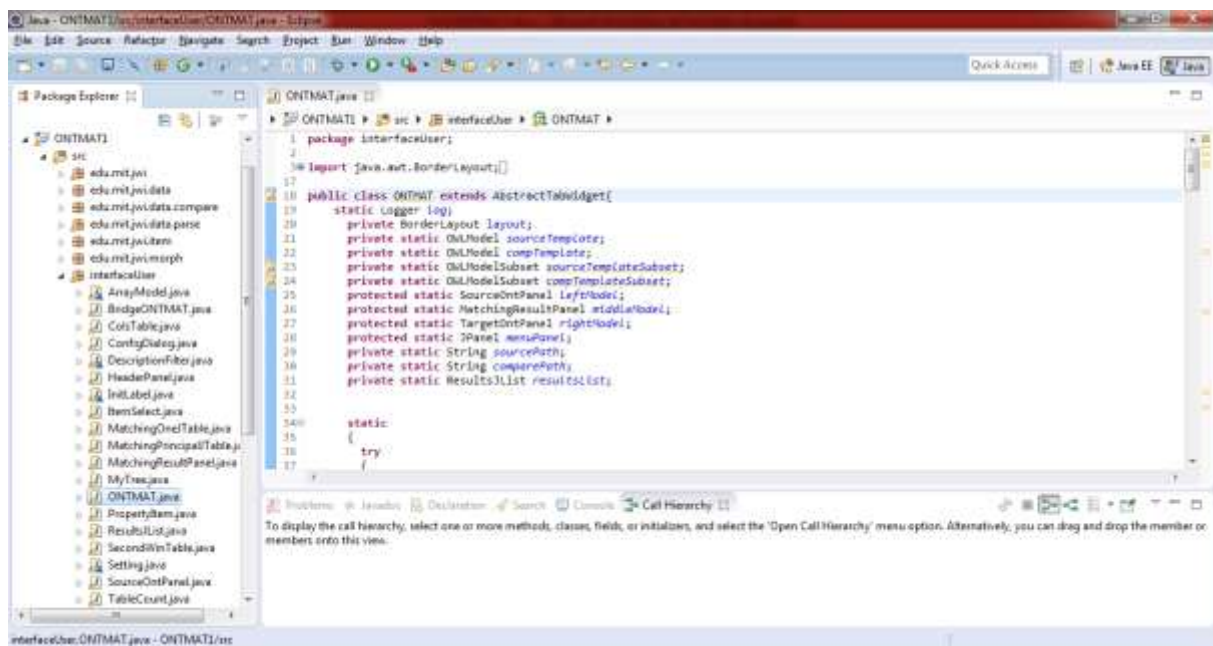


Figure IV.2. Eclipse Java EE IDE de Développeurs de Web

<sup>20</sup> <https://github.com/gherbi82/ONTMAT1>

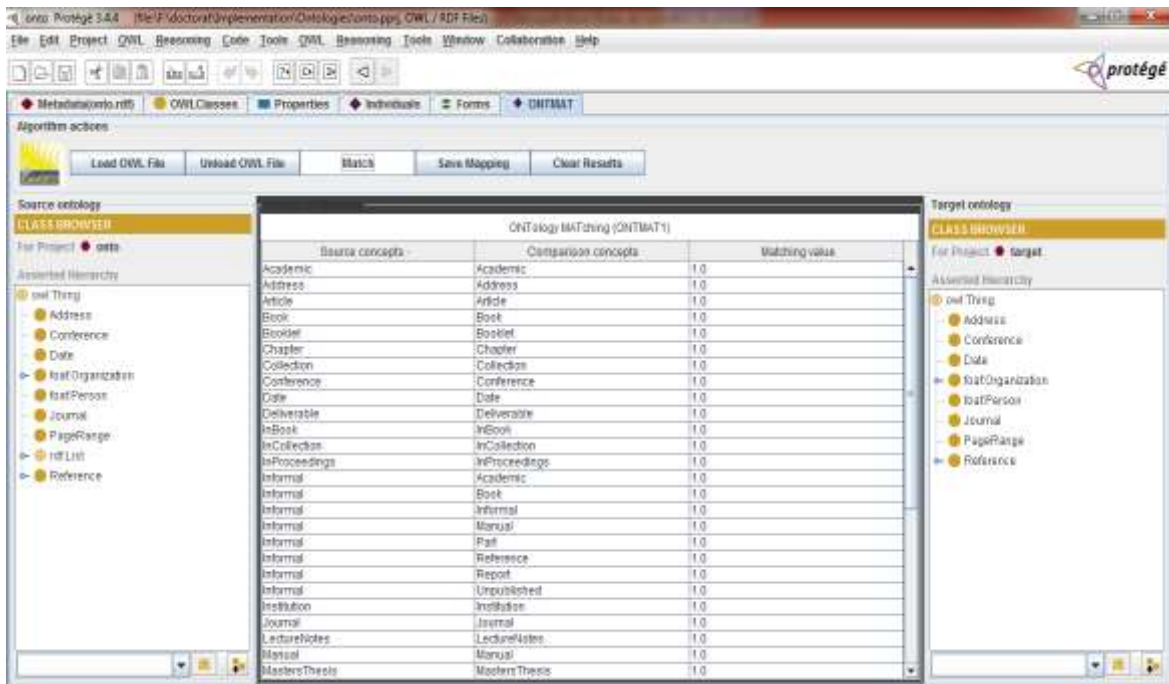


Figure IV.3 Instantané de ONTMAT1

### IV.6.2. Données et critères d'évaluation

Le teste de conférence<sup>21</sup> de la campagne OAEI 2019 a été appliqué pour évaluer ONTMAT1. Le but de ce test est de trouver des alignements au sein d'une collection de 16 ontologies hétérogènes décrivant différents domaines (Zamazal et Svátek, 2017). Cette tâche été fondée sur un alignement de référence précis (Cheatham et Hitzler, 2014) et une évaluation fondée sur la version incertaine de l'alignement de référence.

Afin d'évaluer les résultats d'expérimentation de ONTMAT1 et les autres outils d'alignement, nous avons besoin d'utiliser les mesures de Précision, Rappel et Fallout ou F-mesure [Do et al, 2002] qui sont des métriques largement utilisées pour apprécier la qualité des alignements obtenus. L'objectif capital de ces mesures est l'automatisation du processus de comparaison des méthodes d'alignement ainsi que l'estimation de la qualité des alignements produits. La première phase dans le processus d'appréciation de la qualité de l'alignement consiste à résoudre le problème manuellement. Le résultat obtenu manuellement est considéré comme l'alignement de référence et il sera assigné par un expert.

<sup>21</sup> <http://oaei.ontologymatching.org/2019/conference/index.html>

## ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

---

La comparaison du résultat de l'alignement de référence avec celui de l'appariement obtenu par la méthode d'alignement donne trois ensembles :  $N_{found}$ ,  $N_{expected}$  et  $N_{correct}$ . L'ensemble  $N_{found}$  indique les paires alignées par la méthode d'alignement. L'ensemble  $N_{expected}$  représente l'ensemble des couples appariés dans l'alignement de référence.

L'ensemble  $N_{correct}$  désigne l'intersection des deux ensembles  $N_{found}$  et  $N_{expected}$ . Il est l'ensemble des paires appartenant à la fois à l'alignement obtenu et l'alignement de référence. La précision est le rapport du nombre de paires pertinentes découvertes, i.e., " $N_{correct}$ ", rapporté au nombre total de paires, i.e., " $N_{found}$ ". Il renvoie ainsi, la partie des vraies correspondances parmi celles découvertes. Ainsi, la fonction précision (P) est déterminée par :

$$\text{Précision} = \frac{|N_{correct}|}{|N_{found}|} \quad (\text{IV.10})$$

Le *rappel* est le rapport du nombre de paires pertinentes découvertes, " $N_{correct}$ " ( $N_C$ ), rapporté au nombre total de paires pertinentes, " $N_{expected}$ " ( $N_e$ ). Il désigne ainsi, la part des vraies correspondances trouvées. La fonction *rappel* (R) est décrite par :

$$\text{Rappel} = \frac{N_{correct}}{N_{expected}} \quad (\text{IV.11})$$

La mesure *Fallout* (F) permet d'évaluer le pourcentage d'erreurs obtenu au cours du processus d'alignement. Elle est déterminée par le rapport des paires erronées, " $(N_{found} - N_{correct})$ ", rapporté au nombre total des paires trouvées, " $N_{found}$ " ( $N_f$ ),

$$\text{Fallout} = \frac{N_{found} - N_{correct}}{N_{found}} \quad (\text{IV.12})$$

### IV.6.2.1. Evaluation de l'alignement de référence de crisp

Dans l'alignement de référence de crisp, les valeurs de confiance de tous les appariements sont de 1.0, tandis que dans la version incertaine de l'alignement de référence, les valeurs de

## Chapitre IV : ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

---

confiance des alignements de référence de crisp contiennent 21 alignements (cas de test). Celles-ci correspondent à l'espace d'alignement complet de sept ontologies de l'ensemble de données (Cmt, ConfTool, Edas, Ekaw, Iasted, Sigkdd, Sofsem). Il s'agit d'un sous-ensemble du total de 17 ontologies à l'intérieur de ce test. Le nombre total de cas de test probables est donc de 120. Trois types d'alignements de référence sont utilisés ici:

- ra1: représente l'alignement de référence original.
- ra2: présente un alignement de référence détaillé. Il est généré comme une fermeture transitive, calculée sur l'alignement de référence d'origine (ra1). Afin de trouver un ensemble d'alignement de référence cohérent, les correspondances conflictuelles ont été examinées et déterminées par les évaluateurs. En conséquence, le degré d'exactitude et d'exhaustivité de ra2 peut être supérieur à celui de ra1, bien que les différences soient relativement limitées.
- rar2: a été utilisé pour la première fois en 2017. Il décrit une version sans violation de l'alignement de référence (ra2). Les principales correspondances violatrices ont été découvertes en utilisant l'approche adoptée par [Cheatham et Hitzler, 2014] et [Zamazal et Svátek, 2017], et soigneusement déterminées par un expert.

OAEI fournit trois types d'évaluation pour chaque alignement de référence:

- **M1** contient uniquement les classes,
- **M2** contient uniquement les propriétés,
- **M3** contient les classes et les propriétés (i.e.  $M3 = M1 + M2$ ).

Cependant, ONTMAT1 a testé uniquement l'appariement des classes dans cette version. Vu que les propriétés n'ont pas été traitées dans cette version d'évaluation de notre système, leurs performances globales ont été influencées négativement dans le d'évaluation M3. Les critères d'évaluation utilisés par la compagnie sont illustrés en détail dans le tableau IV.3.

*Tableau IV.3. Résultats d'ONTMAT1 dans OAEI 2019. [Gherbi et Khadir,2020]*

<i>Matching System</i>	<i>M1-ra1</i>			<i>M1-ra2</i>			<i>M1-rar2</i>		
	<i>P.</i>	<i>F.</i>	<i>R.</i>	<i>P.</i>	<i>F.</i>	<i>R.</i>	<i>P.</i>	<i>F.</i>	<i>R.</i>

## ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

ALIN	0.87	0.65	0.52	0.81	0.59	0.47	0.81	0.54	0.50
DOME	0.87	0.67	0.54	0.82	0.61	0.48	0.82	0.55	0.51
LogMap	0.84	0.73	0.64	0.79	0.67	0.58	0.79	0.64	0.61
LogMapLt	0.84	0.66	0.54	0.79	0.6	0.49	0.78	0.56	0.52
AML	0.83	0.76	0.7	0.79	0.71	0.65	0.76	0.68	0.66
<b>OntMat1</b>	<b>0.82</b>	<b>0.61</b>	<b>0.49</b>	<b>0.77</b>	<b>0.56</b>	<b>0.44</b>	<b>0.77</b>	<b>0.50</b>	<b>0.46</b>
Wiktionary	0.82	0.68	0.58	0.79	0.64	0.54	0.77	0.59	0.56
SANOM	0.78	0.77	0.76	0.8	0.73	0.67	0.77	0.70	0.69
Lily	0.59	0.61	0.63	0.58	0.57	0.56	0.60	0.55	0.54

Les résultats d'évaluation des systèmes ont présenté dans le tableau IV.3 [Algergawy et al., 2019]. Les systèmes ont été classés en ordre décroissant en fonction de précision (P.) pour comparer les performances d'ONTMAT1 et les divers systèmes de matching. La précision globale de ONTMAT1 (0.82) est comparable aux systèmes d'appariement d'ontologies les mieux classés : ALIN (0.87) [Da Silva et al., 2019], DOME (0.87) (Deep Ontology Match Er) [Hertling et Paulheimn, 2019], LogMap (0,84) [Jiménez-Ruiz , 2019] et AML (0.83) (Faria et al., 2019). La différence entre ONTMAT1 et les autres systèmes était inférieure ou égale à 0,05 en termes de rappel (recall).

De plus, en termes de f-mesure (F.) qui représente la moyenne harmonique de précision et de rappel, le système proposé atteint une f-mesure de  $F = 0.61$ . Ceci est également très proche de ALIN (0.65), DOME (0.67), LogMap (0.73) et AML (0.76). Ains,i la différence entre ONTMAT1 et les meilleurs appariements n'est pas vraiment significative (la différence de F. était inférieure ou égale à 0.15).

Les résultats obtenus et répertoriés dans le tableau IV.3 ont démontré que l'utilisation de la mesure de n-gram, le poids basé sur la sémantique de restriction de propriété dans une classe et le reasoner Pellet avec WordNet, ont produit des bons résultats de performance en termes de précision et de F-mesure. Cependant, les autres systèmes ont appliqué de nombreuses autres stratégies, telles que :

## Chapitre IV : ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

---

- l'indexation lexicale, l'extraction de modules logiques, le raisonnement de horn propositionnel, le suivi des axiomes, la réparation locale et l'indexation sémantique dans Logmap [Jiménez-Ruiz, 2019];
- un choix par un expert humain de résultat final parmi ceux fournis par les six métriques linguistiques, Jaccard, JaroWinkler, n-Gram, Resnick, Jiang-Conrath et Lin dans ALIN [Da Silva et al., 2019];
- et l'utilisation d'un seuil pour combiner les diverses mesures courantes telles que l'édition de distance et les sous-chaînes en AML (Faria et al., 2019).

L'intégration de toutes ces techniques prend beaucoup de temps en termes d'indexation, d'attente de l'approbation des experts et de calcul de différentes mesures, ce qui diminue l'efficacité de l'approche globale. À cet égard, ONTMAT1 utilise moins de temps car il ne met en œuvre que trois techniques. En général, l'application des méthodes de structure interne sur les restrictions de propriété permet d'avoir un résultat plus précis déterminant plus précisément les types de données correspondants. Cela améliore le temps d'exécution de matching d'ontologies en utilisant une méthode automatique de calcul et d'affectation de poids pour agréger les différentes similarités.

L'utilisation de n-gram uniquement pour mesurer les similarités n'était pas suffisante et nous proposons d'introduire d'autres mesures, telles que Jaccard, pour améliorer l'efficacité de notre système.

De plus, le rappel obtenu est acceptable mais il est insuffisant. Afin de l'améliorer, nous proposons l'équation (IV.9) qui ajoute 1.0 au numérateur de l'équation (IV.2). La valeur ajoutée permet de tenir en compte la similarité terminologique lors de l'évaluation de la force de la similarité structurelle. Une fois la similarité terminologique est prise en compte dans l'évaluation de  $w_r$ , la domination de la similarité structurelle sur la similarité terminologique dans l'équation (IV.8) se diminue. Cela peut améliorer les résultats globaux et aider à découvrir des alignements plus pertinents.

$$w_r = \frac{\sum_{1 \leq i \leq m(1 - |w_{1i} - w_{2j}|)} \mathbb{1}_{1 < j < p}}{\max(m, p) + 1} \quad (\text{IV.9})$$

$$w_r = \mathbf{0.67}$$

$$Sim(Product, Volume) = (1 - 0.67) * 0 + 0.75 = \mathbf{0.75}$$

Les résultats ci-dessus ont été obtenus lors de l'application de l'équation (IV.9) sur l'exemple illustratif du tableau IV.1, obtenant 0.75 comme une nouvelle similarité entre 'Product' et 'Volume' dans la matrice  $M_{cpt}$  définie à la section IV.2.

### IV.6.2.2. Résultats d'estimation basée sur la version incertaine d'alignement de référence

Il existe deux manières d'évaluation de systèmes d'alignement admettant ces alignements de référence «incertains», un est discret et l'autre continue. L'évaluation discrète traite tout alignement dans l'alignement de référence a une valeur de confiance de 0.5 ou plus comme un alignement complètement correct, et ceux dont la valeur est moins de 0.5 comme un alignement complètement incorrect. De même, l'alignement d'un système d'alignement est estimé comme un « oui » si la valeur de confiance est supérieure ou égale au seuil du système et un «non» dans le cas contraire. En principe, cela est identique à l'approche d'évaluation «sharp», sauf que certains appariements sont éliminés car moins de la moitié du groupe de crowd sourcing est en accord avec eux.

La stratégie d'évaluation continue punit un système d'alignement s'il échoue un appariement sur lequel la plupart des systèmes sont d'accord plus que s'il échoue un appariement plus controversé. Par exemple, si  $A = B$ , avec une confiance de 0.85, dans l'alignement de référence et qu'un algorithme d'alignement propose une correspondance avec une confiance de 0.40, alors le calcul pour un vrai positif est de  $0.85 * 0.40 = 0.34$  et le calcul pour un faux négatif est de  $0.85 \text{ à } 0.40 = 0.45$ .

Ci-dessous, le tableau IV.4 et la figure IV.4, extraites des résultats de l'OAEI 2019, illustrent le F-mesure, la précision et le rappel des différents systèmes d'alignement évalués à l'aide des métriques sharp (s), incertaine discrète (d) et incertaine continue (c).

## Chapitre IV : ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction

Tableau IV.4. Estimation des résultats basée sur la version incertaine de l'alignement de référence

System Matcher	Sharp			Discrete			Continuous		
	P.	F.	R.	P.	F.	R.	P.	F.	R.
ALIN	0.87	0.58	0.44	0.87	0.68	0.56	0.87	0.69	0.57
AML	0.84	0.74	0.66	0.79	0.78	0.77	0.80	0.77	0.74
DOME	0.78	0.59	0.48	0.78	0.68	0.60	0.78	0.65	0.56
Lily	0.59	0.56	0.53	0.67	0.02	0.01	0.59	0.32	0.22
LogMap	0.82	0.69	0.59	0.81	0.70	0.62	0.80	0.67	0.57
LogMapLt	0.73	0.59	0.50	0.73	0.67	0.62	0.72	0.67	0.63
<b>OntMat1</b>	<b>0.82</b>	<b>0.55</b>	<b>0.41</b>	<b>0.82</b>	<b>0.64</b>	<b>0.52</b>	<b>0.82</b>	<b>0.64</b>	<b>0.53</b>
SANOM	0.79	0.74	0.69	0.66	0.74	0.83	0.65	0.72	0.81
Wiktionary	0.69	0.61	0.54	0.81	0.68	0.58	0.74	0.69	0.64

Les résultats extraits de site Web des résultats de OAEI 2019 démontrent que des systèmes attribuent des valeurs de confiance nuancées aux correspondances plus qu'ils créent. Lors de la comparaison des performances des systèmes sur les alignements de référence incertains avec celles des alignements de référence sharp, tous les systèmes excepté Lily, ont effectué la même chose ou mieux en termes de F-mesure (le F-mesure de Lily a diminué presque à 0) dans le cas discret.

Les modifications de F- mesure des cas discrets ont étendu de -1 à 17% sur l'alignement de référence de sharp. Cela a été principalement déterminé par un rappel augmenté, qui est le résultat de l'existence de moins de correspondances «controversées» dans la version incertaine de l'alignement de référence.

Les F-measures des six systems, ALIN, AML, DOME, LogMap, LogMapLt, et SANOM, ont resté constants au moment d'évaluation par la version incertaine d'alignement de référence. Les performances qui ont diminué considérablement dans le cas discret sont ceux de Lily.

Les performances d'ONTMAT1 dans des situations discrètes et continues ont augmenté de 16% (de 0.55 à 0.64) en termes de F-mesure par rapport à l'alignement de référence – sharp-, qui est principalement dû à l'augmentation du rappel. Les performances de Wiktionary

## **ONTMAT1: Matching d'ontologies en utilisant un reasoner et une propriété de restriction**

---

peuvent être aussi considérablement améliorées, passant de sharp aux conditions discrètes et continues.

### **VI.7. Conclusion**

Dans ce chapitre nous avons présenté l'architecture d'une nouvelle approche de matching appelée ONTMAT1. L'algorithme de cette approche est implémenté en quatre phases. La première phase utilise la mesure de similarité tri-gram, le reasoner pellet et le dictionnaire WordNet pour appairer les noms des concepts, et les individus des ontologies. La deuxième phase de l'algorithme suggère un poids qui se calcule automatiquement en exploitant la sémantique des propriétés de restriction. La similarité structurelle entre les concepts dans la phase 3 consiste à aligner les noms des propriétés de restriction en utilisant les mêmes méthodes terminologiques de la première phase. Agréger les similarités terminologique et structurelle à l'aide de poids calculé dans la phase 2 permet d'avoir l'alignement final entre les concepts.

Enfin, une expérimentation d'ONTMAT1 sous la compagnie d'OAEI 2019 a été présentée. Ainsi, une évaluation de cet outil avec une comparaison des résultats obtenus avec ceux des outils participés dans cette compagnie a été illustrée.

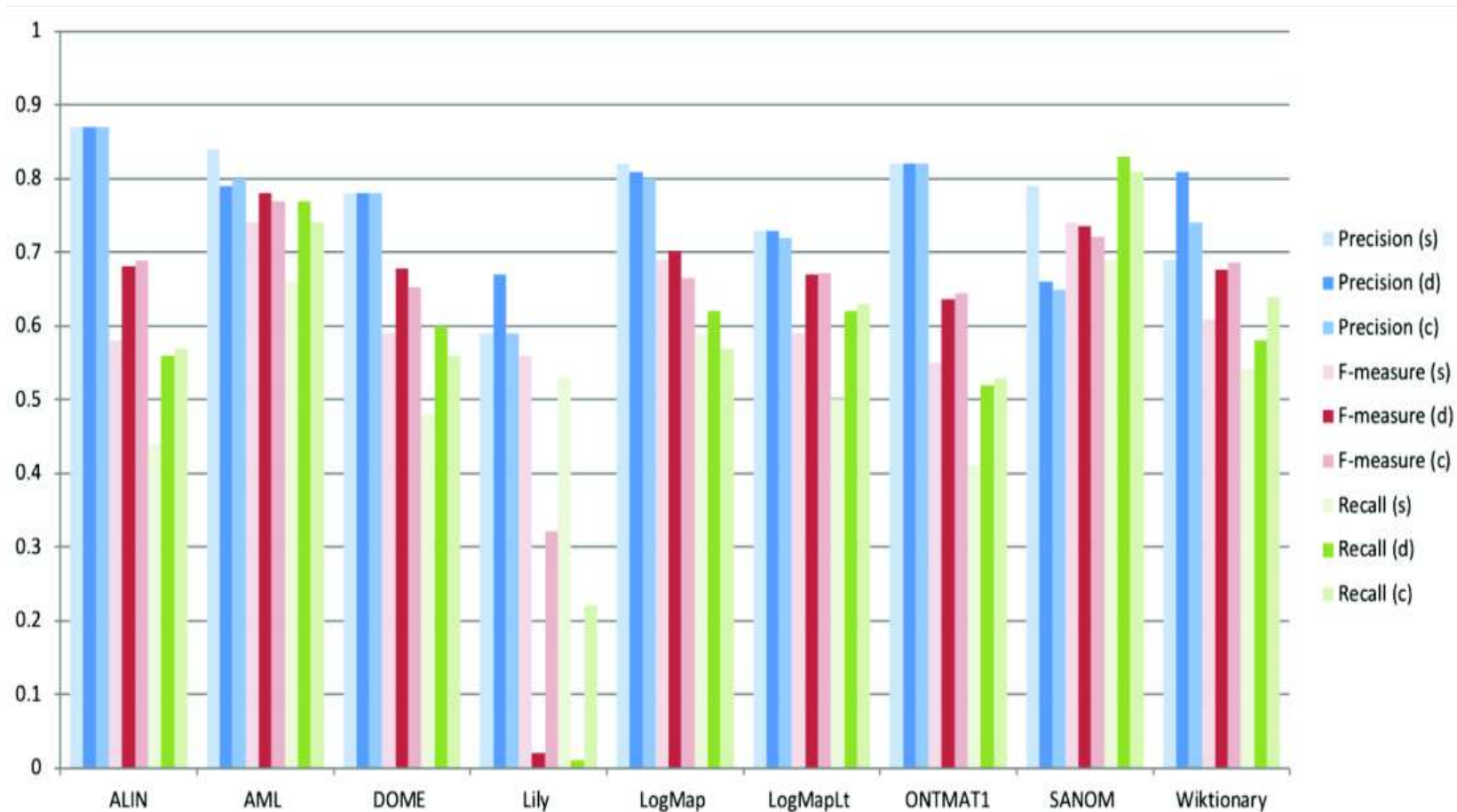


Figure IV.4. Résultats d'évaluation établis sur la version incertaine de l'alignement de référence [Shvaiko et al., 2019]

# *Conclusion Générale et perspectives*

## Conclusion générale

Les algorithmes de matching sont au cœur de nombreux domaines d'application, tel que le Web sémantique, l'intégration des bases de données, les réseaux sociaux, etc. Ainsi, la plupart des systèmes d'appariement proposés se fonde sur une combinaison des méthodes terminologiques et structurelles. L'objectif de cette thèse était d'étudier la possibilité d'introduire la sémantique dans la découverte de l'alignement des concepts en utilisant les raisonneurs, les techniques d'inférences et les propriétés de restriction. Le raisonneur est exploité avec le dictionnaire WordNet pour offrir les noms des entités synonymes dans les deux systèmes proposés ONTMAT et ONTMAT1.

Les techniques d'inférence sont appliquées par ONTMAT pour trouver les concepts voisins des concepts déjà alignés pour améliorer les performances du système. De plus, les instances des concepts ont été comparées entre eux afin de déduire la relation qui peut être entre leurs concepts.

Les propriétés de restriction permettent de calculer un poids sémantique automatisant l'agrégation des similarités structurelles et terminologiques dans ONTMAT1. Ce dernier et ONTMAT ont été testés respectivement sous OAEI 2019 et 2017.

En résumé les principales contributions et études réalisées dans cette thèse permettant de répondre aux objectifs initialement fixés :

1. L'étude de l'existant dans le domaine de matching nous a permis d'exhiber les différentes techniques de matching comme les techniques terminologiques, structurelles, sémantiques, extensionnelles, et probabilistes. L'utilisation de l'une de ses techniques ne répond pas aux besoins des domaines appliquant l'appariement des ontologies. Ce qui a conduit à la combinaison de quelques techniques pour augmenter les performances des systèmes de matching en utilisant une ou plusieurs stratégies de matching.
2. La plupart des systèmes participés avec les outils proposés dans les campagnes de OAEI 2017 et 2019 ont été étudiés et comparés entre eux.
3. La découverte de la synonymie entre les noms des concepts a été réalisée à l'aide du raisonneur Pellet, la métrique de n-gram et le WordNet.

4. L'influe positif des instances sur l'alignement des concepts a été prouvé [Gherbi et Khadir, 2016] mais la domination de cet alignement sur les résultats finals dans l'implémentation de ONTMAT a donné des résultats insatisfaisants pendant la participation dans OAEI 2017.
5. Les résultats insatisfaisants d'ONTMAT nous ont donné l'idée de calculer un poids basé sur la sémantique des propriétés de restriction pour combiner systématiquement les différentes similarités trouvées par les méthodes terminologiques et structurelles exploitées.
6. Les expérimentations montrent qu'ONTMAT1 [Gherbi et Khadir, 2019] améliore la précision, le rappel et fallout d'ONTMAT. Néanmoins, le nombre des faux mappings d'ONTMAT1 reste toujours moins que les autres systèmes participants dans OAEI 2019 [Gherbi et Khadir, 2020].

### Perspectives

Dans la continuité directe de notre travail de thèse, nos perspectives sont les suivants :

Le test de ONTMAT1 sous OAEI 2019 a prouvé que ce système est comparable à la plupart des systèmes les mieux classés sur la piste de conférence. Les chercheurs espèrent que l'OAEI considérera le type de relation proposé ici, comme la relation floue, pour augmenter le rappel.

Pour améliorer les performances d'ONTMAT1, les propriétés seront alignées et testées sous OAEI. Ainsi, plusieurs mesures seront utilisées pour comparer les noms des instances, des concepts et des propriétés. De plus, proposer une méthode d'agrégation et une autre stratégie afin de pouvoir combiner les alignements des instances et des concepts et avoir l'alignement final des concepts.

Adapter notre outil pour lire tous les types de fichiers et intégrer le traducteur pour tester l'outil sous d'autres pistes d'OAEI, comme la correspondance d'instance et MultiFarm, entre autres.

Les travaux futurs comprendront aussi l'exploration et la mise en œuvre de méthodes sémantiques de logique de description, telles que l'application de règles d'inférence pour évaluer l'impact d'une classe donnée sur leurs voisins.

# *Annexe A*

Nous décrivons ici les environnements de développement intégrés Eclipse Ganymede et Eclipse Java EE IDE de Développeurs de Web pour l'implémentation des outils d'alignement ONTMAT et ONTMAT1 respectivement.

Le développement de notre application est fait en Java qui offre plusieurs API tels que : OWL-API, Jena ; permettant de manipuler les ontologies et faire les tests en dessous de OAEI. De plus, il existe également plusieurs API de calcul de mesures de similarité réalisées en Java : n-gram,... Nous les réutilisons dans notre application avec d'autres bibliothèques comme Pellet.

## **A.1 Eclipse**

Eclipse est un environnement de développement intégré spécialement conçu pour le langage de programmation Java. Le logiciel est entièrement gratuit, open-source, mais est également extensible. Ainsi, la partie servant à développer en Java n'est qu'une partie des plug-ins qu'utilise Eclipse. En effet, d'autres plug-ins peuvent être utilisés afin de développer tous les langages et tous les formats de fichiers supportés. Il est ainsi possible de programmer en Java, en PHP, en XML, en HTML, en C#, ou encore en C++.

Grâce à son interface complète et accessible, Eclipse va permettre de développer des sites Web et logiciels en toute simplicité. En plus d'être doté de nombreuses performances, Eclipse est gratuit et open-source ce qui fait de lui un concurrent de taille face aux autres environnements de développement qui eux, sont généralement payant.

La base de cet environnement de développement intégré est l'Eclipse platform qui est composée de : Platform Runtime (qui démarre la plateforme et gère les plug-ins), SWT (la bibliothèque graphique de base de l'EDI), JFace (bibliothèque graphique de plus haut niveau), Eclipse Workbench (dernière couche graphique permettant de manipuler des composants).

Nous avons utilisé Eclipse SDK version 3.4.1 pour programmer notre outil d'alignement d'ontologie afin de pouvoir l'opter aux tests d'OAEI sous la plateforme de SEAL. Cependant, l'implémentation de notre outil sous la précédente version ne permet pas de faire les tests sous la nouvelle plateforme de OAEI appelé HOBBIT. Donc la nouvelle version ONTMAT1 est implémentée sous Eclipse Java EE IDE de Développeurs de Web afin de pouvoir faire les tests sous les deux plateformes SEAL et HOBBIT.

# *Annexe B*

## B.1. Ontologies de la piste de conference

Les ontologies utilisées dans les tests de la piste conference extraient de OAEI 2019<sup>22</sup> sont illustrées dans le tableau B.1. La première colonne contient le nom de l'ontologie, la deuxième, la troisième et la quatrième, colonnes comportent le nombre de classes, de Datatype Properties, et de Object Properties de chaque ontologie, respectivement. De plus, l'expressivité de la logique de description est écrite dans la dernière colonne.

Tableau B.1. Les ontologies utilisées dans les teste de conference

Name	Number of Classes	Number of Datatype Properties	Number of Object Properties	DL expressivity
<a href="#">Ekaw</a>	74	0	33	SHIN
<a href="#">Sofsem</a>	60	18	46	ALCHIF(D)
<a href="#">Sigkdd</a>	49	11	17	ALEI(D)
<a href="#">Iasted</a>	140	3	38	ALCIN(D)
<a href="#">Micro</a>	32	9	17	ALCOIN(D)
<a href="#">Confious</a>	57	5	52	SHIN(D)
<a href="#">Pcs</a>	23	14	24	ALCIF(D)
<a href="#">OpenConf</a>	62	21	24	ALCOI(D)
<a href="#">ConfTool</a>	38	23	13	SIN(D)
<a href="#">Crs</a>	14	2	15	ALCIF(D)
<a href="#">Cmt</a>	36	10	49	ALCIN(D)
<a href="#">Cocus</a>	55	0	35	ALCIF
<a href="#">Paperdyne</a>	47	21	61	ALCHIN(D)
<a href="#">Edas</a>	104	20	30	ALCOIN(D)
<a href="#">MyReview</a>	39	17	49	ALCOIN(D)
<a href="#">Linklings</a>	37	16	31	SROIQ(D)

## B.2. Résultats d'alignement d' ONTMAT1

Le tableau B.1 contient les résultats de différents tests de la piste de conference dans le système ONTMAT1. L'identifiant présenté dans la première colonne de ce dernier tableau désigne les ontologies participées dans le test et Run Time contenue dans la deuxième colonne donne le temps d'exécution de chaque paire d'ontologies. Les mesures d'évaluation précision, rappel et F-mesure ont présenté dans les colonnes 3, 4 et 5 du tableau B.2.

<sup>22</sup> <http://oaei.ontologymatching.org/2020/conference/index.html>

Tableau B.2. Résultats des tests de ONTMATI en dessous de la piste de conférence de OAEI  
2019

Test Case ID	Run Time	Precision	Recall	F-measure
cmt-conference	30524	0.667	0.267	0.381
cmt-confof	8069	0.8	0.25	0.381
cmt-edas	18923	1.0	0.538	0.7
cmt-ekaw	35535	1.0	0.455	0.625
cmt-iasted	49732	0.8	1.0	0.889
cmt-sigkdd	9388	1.0	0.583	0.737
conference- confof	14075	0.875	0.467	0.609
conference-edas	34741	0.875	0.412	0.56
conference- ekaw	24744	0.667	0.32	0.433
conference- iasted	45959	0.75	0.214	0.333
conference- sigkdd	16637	0.889	0.533	0.666
confof-edas	23118	0.889	0.421	0.571
confof-ekaw	16225	0.889	0.4	0.552
confof-iasted	30052	1.0	0.333	0.5
confof-sigkdd	11468	1.0	0.286	0.445
edas-ekaw	41996	0.75	0.391	0.514
edas-sigkdd	29088	1.0	0.4	0.571
ekaw-iasted	55506	1.0	0.5	0.667
ekaw-sigkdd	20379	1.0	0.636	0.778
iasted-sigkdd	38381	0.5	0.6	0.545
<b>Global</b>	<b>554540</b>	<b>0.823</b>	<b>0.423</b>	<b>0.559</b>

# **Bibliographie**

# Bibliographie

- [Agrawal et al, 1989], Agrawal, R., Borgida, A., Jagadish, H.V.: Agrawal, R., Borgida, A., & Jagadish, H. V. (1989, June). Efficient management of transitive relationships in large data and knowledge bases. In *ACM SIGMOD Record* (Vol. 18, No. 2, pp. 253-262). ACM.
- [Albagli et al. 2012], Albagli, S., Ben-Eliyahu-Zohary, R., Shimony, S.E.: Markov network based ontology matching. *J. Comput. Syst. Sci.* 78(1), 105–118 (2012) (pp. 143, 233)
- [Algergawy et al., 2018], Algergawy, A., Cheatham, M., Faria, D., Ferrara, A., Fundulaki, I., Harrow, I., ... & Lambrix, P. (2018, October). Results of the Ontology Alignment Evaluation Initiative 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)* (Vol. 2288, pp. 76-116).
- [Algergawy et al., 2019], Algergawy I., Faria D., Ferrara A., Fundulaki I, Harrow I, Hertling S., Jiménez-Ruiz E., Karam N, Khiat A, Lambrix P, Li H, Montanelli S, Paulheim H, Pesquita C, Saveta T, Shvaiko P, Splendiani A., Thiéblin E., Trojahn C., Vataščinová J, Zamazal O, and Zhou L, (2019). Results of the Ontology Alignment Evaluation Initiative 2019\*.
- [Alsayed et al., 2019], Alsayed Algergawy, Daniel Faria, Alfio Ferrara, Iriini Fundulaki, Ian Harrow, Sven Hertling, Ernesto Jiménez-Ruiz, Naouel Karam, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Stefano Montanelli, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Elodie Thiéblin, Cássia Trojahn, Jana Vataščinová, Ondřej Zamazal, Lu Zhou, 2019. Results of the Ontology Alignment Evaluation Initiative 2019 In *14th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2019)*
- [Amgoud et al, 2000], Amgoud, L., Parsons, S., Maudet, N., (2000): Arguments, dialogue and negotiation. In: Proc. 14th European Conference on Artificial Intelligence (ECAI), Berlin, Germany, pp. 338–342 (p. 171).
- [Baader et al, 2003a], Baader, F., Horrocks, I. et Sattler, U., 2003. Description logics as ontology languages for the semantic web. Dans Hutter, D. et Stephan, W. (éditeurs), *Festschrift in honor of Jörg Siekmann. Lecture Notes in Artificial Intelligence*. Springer-Verlag.
- [Baader et al, 2003b], Baader, F., 2003. Appendix 1 : Description logic terminology. Dans Baader, F., Calvanese, D., McGuinness, D., Nardi, D. et Patel-Schneider, P. (éditeurs), *The Description Logic Handbook : Theory, Implementation and Applications*. Cambridge University Press.
- [Baader et al., 2003], Baader F., Calvanese D., D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, 2003, *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge University Press, Cambridge, UK.
- [Baader et al, 2003c], Baader, F. et Nutt, W., 2003. Basic description logics. Dans Baader, F., Calvanese, D., McGuinness, D., Nardi, D. et Patel-Schneider, P. (éditeurs), *The Description Logic Handbook : Theory, Implementation and Applications*. Cambridge University Press, pp. 47100.
- [Baget, J. , 2004] ,BAGET, J. (2004). Homomorphismes d'hypergraphes pour la subsomption en RDF/RDFS. In Actes de la 10<sup>e</sup> conference Langages et Modeles a Objet (LMO'2004), pages 203–216.
- [Bechhofer ,2009], BECHHOFFER, 2009. Sean, OWL: Web ontology language. In : *Encyclopedia of database systems*. Springer US, p. 2008-2009.
- [Bench-Capon, 2003], Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.* 13(3), 429–448 (2003) (p. 170)

## Bibliographie

- [Benerecetti et al., 2000] Benerecetti M., Bouquet P. and Ghidini C., “*Contextual reasoning distilled*”, Journal of Theoretical and Experimental artificial Intelligence, 12(3): pp. 279–305, July 2000.
- [Berners-Lee et al., 2001], BERNERS-LEE Tim, HENDLER James & LASILLA Ora (2001). The Semantic Web, scientific American.
- [Bouquet et al, 2003], Bouquet, P., Magni, B., & Serafini, L. (2003). A SATBased Algorithm for Context Matching. In : 4<sup>th</sup> International and Interdisciplinary Conference, CONTEXT 2003. June 23-25, 2003, Stanford, CA, USA. (Springer, Éd.) pp. 66-79.
- [Cheatham et Hitzler, 2014], Cheatham M., & Hitzler, P: (2014); Conference v2.0: An Uncertain Version of the OAEI Conference Benchmark. International Semantic Web Conference (2): 33-48.
- [Chen et al, 2014], Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V., (2014): Extending an ontology alignment system with bioportal: a preliminary analysis. In: Poster at Int'l Sem. Web Conf. (ISWC).
- [Chen et al., 2018], Chen, G., & Zhang, S. (2018, December). FCAMapX results for OAEI 2018. In OM@ISWC (pp. 160-166).
- [Chen et al., 2019], Chang, F., Chen, G., & Zhang, S. (2019). FCAMap-KG Results for OAEI 2019. OM@ISWC, 2536, 138-145.
- [Christophides et al., 2003], Christophides, V., Plexousakis, D., Scholl, M., & Tourtounis, S. (2003, May). On labeling schemes for the semantic web. In *Proceedings of the 12th international conference on World Wide Web* (pp. 544-555). ACM.
- [Clark and Parsia, 2009], Clark and Parsia. Pellet : The open source owl dl reasoner. Pellet home page, 2009.
- [Corcho, 2004] Corcho O., “*A declarative approach to ontology translation with knowledge preservation*”, PhD thesis, Universidad Politecnica de Madrid, Madrid (ES), 2004.
- [Cox, 1994] Cox, T. et Cox, M. Multidimensional Scaling. Chapman and Hall, 1994.
- [Cristianini and Shawe-Taylor, 2000], Cristianini, N., Shawe-Taylor, J., (2000): Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (p. 177).
- [Da Silva, 2017], Da Silva, J., Baiao, F. A., Revoredo, K., & Euzenat, J. (2017, October). Semantic interactive ontology matching: Synergistic combination of techniques to improve the set of candidate correspondences.
- [David et al. 2007], David, J., Guillet, F., Briand, H., (2007): Association rule ontology matching approach. Int. J. Semantic Web Inf. Syst. 3(2), 27–49 (pp. 152, 252).
- [David et al, 2011], David, J., Euzenat, J., Schare, F., Trojahn dos Santos, C.: The alignment api 4.0. Semant. web 2(1) (January 2011) 3-10.
- [Dowling et Gallier, 1984], Dowling, W. F., & Gallier, J. H. (1984). Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *The Journal of Logic Programming*, 1(3), 267-284.
- [Declercq, 2010], Declercq, S. (2010). ONTOLOGIES ET WEB SÉMANTIQUE... L'ère du documentaliste a-t-elle sonné?. *Cahiers de la documentation–Bladen voor documentatie*, 2.
- [Djeddi and Khadir,2013], Djeddi, W. E., & Khadir, M. T. (2013). Ontology alignment using artificial neural network for large-scale ontologies. *International Journal of Metadata, Semantics and Ontologies* 16, 8(1), 75-92.
- [Do et al., 2002], Do, H.H. et Rahm, E. (2002) “Coma – a system for flexible combination of schema matching approaches”, Dans Proc. VLDB, pages 610–621.
- [Doan, 2002], Doan, A., Madhavan, J., & Domingos, P. (2002, May 7-11). Learning to Map between ontologies on the Semantic Web. In : the eleventh International World Wide Web conference (WWW2002), Honolulu, Hawaii, USA.

- [Doan et al., 2004], Doan, A.-H., Madhavan, J., Domingos, P., Halevy, A., (2004): Ontology matching: a machine learning approach. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 385–404. Springer, Berlin . Chap. 18 (pp. 173, 235)
- [Dung, 1995], Dung, P.M., (1995): On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2), 321–358 (p. 171)
- [Duchateau et al., 2008], Duchateau, F., Bellahsene, Z., Coletta, R., (2008): A flexible approach for planning schema matching algorithms. In: Proc. 16th International Conference on Cooperative Information Systems (CoopIS), Monterrey, Mexico. Lecture Notes in Computer Science, vol. 5331, pp. 249–264.
- [Duchateau et al., 2009], Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R., (2009): (not) Yet Another Matcher. In: Proc. 18<sup>th</sup> ACM Conference on Information and Knowledge Management (CIKM), Hong Kong, China, pp. 1537–1540.
- [Ehrig et al., 2005], Ehrig, M., Staab, S., Sure, Y., (2005): Bootstrapping ontology alignment methods with APFEL. In: Proc. 4th International Semantic Web Conference (ISWC), Galway, Ireland. Lecture Notes in Computer Science, vol. 3729, pp. 186–200.
- [Ehrig, 2007], Ehrig, M., 2007, “Ontology Alignment Germany Bridging the Semantic Gap”, by Marc Ehrig, University of Karlsruhe Germany, pages 35. 2007. Springer Science-i-Business Media, LLC.
- [Elbyed M., 2009], ELBYED M., 2009 .ROMIE, une approche d’alignement d’ontologies à base d’instances. Page 34.
- [Esposito et al., 2010], Esposito, F., Fanizzi, N., d’Amato, C., (2010): Recovering uncertain mappings through structural validation and aggregation with the MoTo system. In: Proc. 25th ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, pp. 1428–1432.
- [Essayeh et Abed, 2015], Essayeh, A., & Abed, M. (2015). Towards ontology matching based system through terminological, structural and semantic level. *Procedia computer science*, 60, 403-412.
- [Euzenat, 2001] Euzenat, J., “*Towards a principled approach to semantic interoperability*”, in Proc. IJCAI Workshop on Ontologies and Information Sharing, pp. 19–25, Seattle (WA US), 2001.
- [Euzenat, 2004], Euzenat J., “An API for ontology alignment”, in Proc. 3rd International Semantic Web Conference (ISWC), volume 3298 of LNCS, pp. 698–712, Hiroshima (JP), 2004.
- [Euzenat et al., 2004] Euzenat, J., Bach, T.L., Barrasa, J., Bouquet, P., Bo, J.D., Dieng-Kuntz, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Tessaris, S., Acker; S.V. et Zaihrayeu, I, 2004. State of the art on ontology alignment, deliverable 2.2.3, IST Knowledge web NoE, Knowledge web NoE, 80p., June 2004
- [Euzenat et al., 2007] Euzenat, J. and al., “*Heterogeneity in the semantic web, deliverable 2.2*”, Knowledge Web, December 2007.
- [Euzenat et Shvaiko, 2007], Euzenat, J., Shvaiko. P, 2007. “Ontology Matching”, Springer-Verlag Berlin Heidelberg.
- [Euzenat et Shvaiko, 2013], Euzenat, J., & Shvaiko, P. (2013). Ontology matching..
- [Fensel et al, 2000], Fensel, D., Horrocks, I., van Harmelen, F., & De, S. (2000). Oil in a nutshell. In Proceedings of European Knowledge Acquisition Workshop (EKAW’2000), volume 1937, pages 1–16. Springer-Verlag LNAI.
- [Fensel, 2014], Fensel Dieter. Ontologies, (2014): a silver bullet for knowledge management and electronic commerce. Springer, Heidelberg (DE), 2nd edition.
- [Faria et al, 2019], Faria, D., Pesquita, C., Tervo, T., Couto, F.M., Cruz, I. F., (2019), AML and AMLC Results for OAEI 2019.

## Bibliographie

- [Fridman et al.,2009], Fridman Noy, N., Shah, N.H., Whetzel, P.L., Dai, B., et al., (2009): BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research* 37, 170–173.
- [Fournier et Philippe, 2005], Fournier-Viger, Philippe (2005) "Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents". Mémoire de maîtrise (M.Sc.), Université de Sherbrooke, Sherbrooke, Canada.
- [Gal et al. 2005], Gal, A., Anaby-Tavor, A., Trombetta, A., Montesi, D. (2005): A framework for modeling and evaluating automatic semantic reconciliation. *VLDB J.* 14(1), 50–67 (p. 165)
- [Gal, 2011], Gal, A. (2011): *Uncertain Schema Matching*. Synthesis Lectures on Data Management. Morgan & Claypool, San Rafael.
- [Ghazvinian et al. 2009], Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N.H., Musen, M.A., (2009): What four million mappings can tell you about two hundred ontologies. In: *Int’l Sem. Web Conf. (ISWC)*.
- [Ghazvinian et al. 2011], Ghazvinian, A., Noy, N., Musen, M., (2011): From mappings to modules: using mappings to identify domain-specific modules in large ontologies. In: *Proc. 6th International Conference on Knowledge Capture (K-CAP)*, Banff, Canada, pp. 33–40 (p. 151).
- [Gherbi et al., 2013], Gherbi S., Belleili H. and Khadir M., 2013. BRMAP : Un outil d’Alignement des ontologies. 7ème édition de la conférence maghrébine sur les avancés des systèmes décisionnels.
- [Gherbi et Khadir, 2016], Gherbi, S, Khadir, M. T. , (2016), Inferred ontology concepts alignment using instances and an external dictionary, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops. doi:<http://dx.doi.org/10.1016/j.procs.2016.04.145>. URL <http://www.sciencedirect.com/science/article/pii/S1877050916301752>, *Procedia Computer Science* 83 648-652 (Elsevier 2016).
- [Gherbi et Khadir, 2017], Gherbi, S., & Khadir, M. T. (2017). ONTMAT: results for OAEI 2017. In *OM@ ISWC* (pp. 166-170).
- [Gherbi et KHADIR, 2019], Saida Gherbi and MedTarek. ONTMAT1: Results for OAEI 2019. *Ontology Matching*, 2019, p. 164.
- [Gherbi et Khadir, 2020], Gherbi, S. and Khadir, M.T. (2020) ‘ONTMAT1: ontology matching using a reasoner and property restriction’, *Int. J. Web Engineering and Technology*, Vol. 15, No. 2, pp.119–139.
- [Ghidini et Giunchiglia., 2004] Ghidini, C. and Giunchiglia, F., “*A semantics for abstraction*”, in *Proceedings ECAI*, 2004.
- [Ginsberg et al, 1994], GINSBERG, M. L. (1994). Knowledge Interchange Format: the KIF of death. *AI Magazine*, 12(3):57–63.
- [Giunchiglia et Walsh., 1992] Giunchiglia, F. and Walsh, T., “*A theory of abstraction*”, *Artificial Intelligence*, 56(2–3):323–390, Also available as DAI Research Paper No 516, Dept. of Artificial Intelligence, Edinburgh, 1992.
- [Giunchiglia et Shvaiko, 2003] Giunchiglia, F. et Shvaiko, P. Semantic matching. Dans *Proc. IJCAI 2003 Workshop on ontologies and distributed systems*, Acapulco (MX), pages 139–146, 2003.
- [Giunchiglia et al., 2004] Giunchiglia, F., Shvaiko, P. et Yatskevich, M. S-Match: an algorithm and an implementation of semantic matching. Dans *Proceedings of ESWS 2004*, Heraklion (GR), pages 61–75, 2004.
- [Good,1965], Good, I.J.: *The Estimation of Probabilities: an Essay on Modern Bayesian Methods*, (1965). MIT Press, Cambridge (p. 173).

- [Gracia et al., 2011], Gracia, J., Bernad, J., Mena, E., (2011): Ontology matching with CIDER: evaluation report for OAEI 2011. In: Proc. 6th International Workshop on Ontology Matching (OM) at the 10th International Semantic Web Conference (ISWC), Bonn, Germany, pp. 126–133.
- [Gruber, 1993] Gruber, T. R. (1993), A translation approach to portable ontology specifications, Knowledge Acquisition, 5((2): 199 – 220.
- [Gruber, 1995] Gruber, Tom R.. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies. 1995.
- [Gomez-Perez 2000], Gomez-Perez A., Développement récents en matière de conception, de maintenance et d'utilisation d'ontologies, 3èmes rencontre Terminologie et intelligence artificielle TIA, Vol 19, PP : 9-20.
- [Hájek 1998], Hájek, P., (1998): The Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (p. 161).
- [Hameed et al., 2004], Hameed, A., Preece, A. and Sleeman, D., “*Ontology reconciliation*”, in Steffen Staab and Rudi Studer, editors, Handbook on ontologies, chapter 12, pages 231–250. Springer Verlag, Berlin (DE), 2004.
- [Hanif and Aono, 2009], Hanif, M.S., Aono, M., (2009): An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. J. Web Semant. 7(4), 344–356 (pp. 152, 229).
- [Haarslev et Möller, 2001], V. Haarslev and R. Moller, 2001. Racer user's guide and reference manual version 1.6. Technical report, University of Hamburg, Computer Science Department.
- [He et al, 2003], He, B., Chang, K., & Han, J. (2003, December). Automatic Complex Schema Matching across Web Query Interfaces: A Correlation Mining Approach. Technical Report UIUCDCS-R-2003-2388, Department of Computer Science, UIUC., B., Chang, K., & Han, J. (2003, December). Automatic Complex Schema Matching across Web Query Interfaces: A Correlation Mining Approach. Technical Report UIUCDCS-R-2003-2388, Department of Computer Science, UIUC.
- [Hebeler et al., 2011], Hebeler, J., Fisher, M., Blace, R., & Perez-Lopez, A. (2011). Semantic web programming. John Wiley & Sons.
- [Hendler et al, 2001], HENDLER, J., & MCGUINNESS, D. (2001). The Darpa Agent Markup Language. <http://www.daml.org>.
- [Hertling & Paulheim, 2018], Hertling, S., & Paulheim, H. (2018, December). DOME results for OAEI 2018. In OM@ ISWC (pp. 144-151).
- [Hertling & Paulheim, 2019], Hertling, S., & Paulheim, H. (2019). DOME Results for OAEI 2019. OM@ ISWC, 2536, 123-130.
- [Hollunder et al., 1990], B. Hollunder, W. Nutt, and M. Schmidt-Schauß, 1990. Subsumption algorithms for concept description languages. In ECAI, pages 348–353.
- [Horrocks et al 2004] HORROCKS I., PATEL-SCHNEIDER P. F., BOLEY H., TABET S., GROSOFF B., DEAN M., « *SWRL : A Semantic Web Rule Language Combining OWL and RuleML* », W3C Member Submission, , 2004.
- [Jérôme, 2007], Jérôme, D., 2007. « AROMA : une méthode pour la découverte d'alignements orientés entre ontologies à partir de règles d'association » ; thèse de doctorat à l'École Polytechnique de l'Université de Nantes.
- [Jiménez-Ruiz, et al., 2011], Jiménez-Ruiz, E., Morant, A., & Grau, B. C. (2011). LogMap results for OAEI 2011. Ontology Matching, 163.
- [Jiménez-Ruiz, et al., 2012], Jiménez-Ruiz, E., Morant, A., & Grau, B. C. (2012). LogMap results for OAEI 2011. Ontology Matching, 163.

## Bibliographie

- [Jiménez-Ruiz et al., 2017], Jiménez-Ruiz, E., Grau, B. C., & Cross, V. (2017, December). LogMap family participation in the OAEI 2017. CEUR Workshop Proceedings.
- [Jiménez-Ruiz, 2019], Jiménez-Ruiz, (2019). LogMap Family Participation in the OAEI 2019\*.
- [Da Silva et al., 2019], Jomar da Silva, Carla Delgado, Kate Revoreda, and Fernanda Araujo Baião, 2019. ALIN Results for OAEI 2019.
- [Kachroudi et al., 2018], Kachroudi, M., Diallo, G., & Yahia, S. B. (2018, December). KEPLER at OAEI 2018. In OM@ ISWC (pp. 173-178).
- [Kifer et al, 1995], Kifer, M., Lausen, G., & Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741-843, 1995.
- [Klein et al, 2001], Klein, M., Gomez-Pérez, A., Gruninger, M., & Stuckenschmidt, H. (2001, August 4-5). Combining and relating ontologies : an analysis of problems and solutions. Workshop on Ontologies and Information Sharing (IJCAI-01).
- [Laera et al., 2006], Laera, L., Tamma, V., Euzenat, J., Bench-Capon, T., Payne, T., (2006): Reaching agreement over ontology alignments. In: Proc. 5th International Semantic Web Conference (ISWC), Athens, GA, USA. Lecture Notes in Computer Science, vol. 4273, pp. 371–384 (pp. 19, 170, 171, 362).
- [Lastra-Daz, 2017], Lastra-Daz, J.J., Garca-Serrano, A., Batet, M., Fernandez, M., Chirigati, F., (June 2017) : Hesml.Inf. Syst. 66(C) 97-118.
- [Lambrix and Tan, 2006], Lambrix, P., Tan, H.: SAMBO—a system for aligning and merging biomedical ontologies. *J. Web Semant.* 4(1), 196–206 (2006) (pp. 173, 252).
- [Le et al, 2014], Le, Q., Mikolov, T., (2014): Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188-1196.
- [Lê, 2006], Lê Bach Thanh. Construction d'un Web sémantique multi-points de vue. domain\_other. École Nationale Supérieure des Mines de Paris, 2006. English. <pastel-00001989>
- [Li and Clifton, 1994], Li, W.-S., Clifton, C., (1994): Semantic integration in heterogeneous databases using neural networks. In: Proc. 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, pp. 1–12.
- [Lin et al., 2010], Lin, F., Butters, J., Sandkuhl, K., Ciravegna, F., (2010): Context-based ontology matching: Concept and application cases. In: 10th IEEE International Conference on Computer and Information Technology, CIT 2010, Bradford, West Yorkshire, UK, June 29-July 1, 2010. pp. 1292{1298. IEEE Computer Society (2010). <https://doi.org/10.1109/CIT.2010.233>, <https://doi.org/10.1109/CIT.2010.233>
- [Lin et al., 2010], Lin, F., Krizhanovsky, A. (2011): Multilingual ontology matching based on wiktory data accessible via SPARQL endpoint. CoRR abs/1109.0732 (2011), <http://arxiv.org/abs/1109.0732>
- [Lovin, 1968] Lovins, J.B. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11: 22-31, 1968.
- [Mädche et Staab, 2002] Mädche, A. et Staab, S. Measuring similarity between ontologies. In Proc. Of the 13th Int. Conference on Knowledge Engineering and Management (EKAW-2002), Sigüenza, Spain, October 2002. Springer-Verlag.
- [Madhavan et al., 2001] Madhavan, J., Bernstein, P. et Rahm, E. Generic schema matching with cupid. Dans Proceedings of the 27th International Conference on Very Large Data Bases, pages 49–58. Morgan Kaufmann Publishers Inc., 2001.

- [**Mao et al., 2010**], Mao, M., Peng, Y., & Spring, M. (2010). An adaptive ontology mapping approach with neural network based constraint satisfaction. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1), 14-25.
- [**Maynard et Ananiadou, 1999**], Maynard, D.G. et Ananiadou, S. Term extraction using a similarity-based approach. In *Recent Advances in Computational Terminology*. John Benjamins, 1999.
- [**McCrae et al., 2012**], McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gomez-Pérez, A., Gracia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., Wunner, T., 2012: Interchanging Lexical Resources on the Semantic Web. *Language Resources and Evaluation* 46(4), 701-719 (Dec 2012). <https://doi.org/10.1007/s10579-012-9182-3>, <http://link.springer.com/10.1007/s10579-012-9182-3>
- [**Megdiche et al., 2016**], I. Megdiche, O. Teste, and C. Trojahn. An extensible linear approach for holistic ontology matching. In *Proceedings of the 15th International Semantic Web Conference (ISWC 2016)*, Kobe, Japan, October 18, 2016., pages 393–410. Springer International Publishing, 2016.
- [**Miller, 1995**], Miller, G., 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41., [66, 86, 87].
- [**Mindswap, 2003**], mindswap. Pallet, 2003.FF
- [**Mitra, 2005**], Mitra, P., Noy, N., Jaiswal, A. (2005): Ontology mapping discovery with uncertainty. In: *Proc. 4th International Semantic Web Conference (ISWC)*, Galway, Ireland. *Lecture Notes in Computer Science*, vol. 3729, pp. 537–547 (pp. 141, 142, 220)
- [**Nandi and Bernstein, 2009**], Nandi, A., Bernstein, P.: HAMSTER: using search clicklogs for schema and taxonomy matching. *Proc. VLDB Endow.* 2(1), 181–192 (2009) (pp. 173, 255)
- [**Nebot, et Berlanga, 2009**], Nebot, V., & Berlanga, R. (2009). Efficient retrieval of ontology fragments using an interval labeling scheme. *Information Sciences*, 179(24), 4151-4173.
- [**Niepert et al. 2010**], Niepert, M., Meilicke, C., Stuckenschmidt, H.: A probabilistic-logical framework for ontology matching. In: *Proc. 24th Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, USA, pp. 1413–1418 (2010) (pp. 143, 260)
- [**Noy et al., 2000**], Noy, N. F., & Musen, M. A. (2000). Prompt: algorithm and tool for automated ontology merging and alignment. In *Proceeding of Seventeenth National Conference on Artificial Intelligence AAAI*.
- [**Pan et al., 2005**], Pan, R., Ding, Z., Yu, Y., Peng, Y. (2005): A Bayesian network approach to ontology mapping. In: *Proc. 4th International Semantic Web Conference (ISWC)*, Galway, Ireland. *Lecture Notes in Computer Science*, vol. 3729, pp. 563–577 (pp. 141, 219).
- [**Pearl, 1988**], Pearl, J., (1988): *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco (p. 143).
- [**Porter, 1980**], Porter, M.F. An Algorithm for Suffix Stripping, *Program*, 14(3): 130-137, 1980.
- [**Portisch et Paulheim, 2018**], Portisch, J., Paulheim, H.: Alod2vec matcher. In: *OM@ISWC. CEUR Workshop Proceedings*, vol. 2288, pp. 132{137. [CEUR-WS.org](http://CEUR-WS.org) (2018)
- [**Qu et al., 2006**], Qu Y, Hu W, Cheng G, (2006): Constructing virtual documents for ontology matching[C] *Proc. International Conference on World Wide Web*. 23-31.
- [**Richardson and Domingos, 2006**], Richardson, M., Domingos, P, (2006):. Markov logic networks. *Mach. learn.* 62(1–2), 107–136 (p. 143).
- [**Ronal, 1984**], Ronal J. Brachman and Hector J. Levesque. The tractability of subsumption in frame-based description languages. In *Proc. of the 4th National Conference on Artificial Intelligence (AAAI-84)*, pages 34–37, 1984.

## Bibliographie

- [**Roussille et al. , 2018**], Roussille, P., Megdiche Bousarsar, I., Teste, O., & Trojahn, C. (2018). Holontology: results of the 2018 OAEI evaluation campaign. CEUR-WS: Workshop proceedings.
- [**Sabou et al.**], Sabou, M., D'Aquin, M., Motta, E., (2006), "Using the Semantic Web as Background Knowledge for Ontology Mapping", ISWC'06 Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA.
- [**Sérasset, 2015**], Sérasset, G., (2015): Dbnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. *Semantic Web* 6(4), 355-361. <https://doi.org/10.3233/SW-140147>, <https://doi.org/10.3233/SW-140147>.
- [**Shvaiko, 2006**], Shvaiko, P., Giunchiglia, F., Schorlemmer, M., McNeill, F., Bundy, A., Marchese, M., Yatskevich, M., Zaihrayeu, I., Ho, B., Lopez, V., Sabou, M., Abian, J., Siebes, R., Kotoulas, (2006), S.: Dynamic ontology matching: a survey. Deliverable 3.1, OpenKnowledge (p. 15)
- [**Shvaiko et Euzenat, 2012**], Shvaiko, P., Euzenat, J. ,2012. Ontology matching: state of the art and future challenges *IEEE Transactions on Knowledge and Data Engineering.Campaign - ISWC Ontology Matching Workshop*, Shanghai International Convention Center, Shanghai, China, November 7.
- [**Shvaiko et al., 2018**], Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., & Hassanzadeh, O. (2018). Ontology Matching: OM-2018: Proceedings of the ISWC Workshop. In 13th ISWC workshop on ontology matching (OM) (pp. 1-227). No commercial editor.
- [**Shvaiko et al., 2019**], Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., & Hassanzadeh, O. (2019). Ontology Matching: OM-2019: Proceedings of the ISWC Workshop.
- [**Sirin et al, 2007**], E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2), 2007.
- [**Smullyan, 1968**], R. M. Smullyan, 1968. First-order logic. Springer-Verlag, Berlin.
- [**Spiliopoulos et al., 2010**], Spiliopoulos, V., Vouros, G., Karkaletsis, V., (2010): On the discovery of subsumption relations for the alignment of ontologies. *J. Web Semant.* 8(1), 69–88.
- [**Stumme et al., 2003**], Stumme, Gerd, Ehrig, Marc, Handschuh, Siegfried, Hotho, Andreas, Maedche,Alexander, Motik, Boris, Oberle, Daniel, Schmitz, Christoph, Staab,Steffen, Stojanovic, Ljiljana, Stojanovic, Nenad, Studer, Rudi, Sure, York,Volz, Raphael, and Zacharias, Valentin (2003). The Karlsruhe view on ontologies. Technical report, Institut AIFB, Universitat Karlsruhe (TH), Karlsruhe, Germany.
- [**Straccia and Troncy, 2006**], Straccia, U., Troncy, R. (2006): Towards distributed information retrieval in the semantic web: query reformulation using the oMAP framework. In: Proc. 3rd European Semantic Web Conference (ESWC), Budva, Montenegro. *Lecture Notes in Computer Science*, vol. 4011, pp. 378–392 (pp. 173, 244).
- [**Stuckenschmidt et al., 2009**], Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.), 2009: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. *Lecture Notes in Computer Science*, vol. 5445. Springer, Berlin (p. 151)
- [**Surhone, 2010**], Surhone, L.M., Timpledon, M.T., Marseken, S.F.: SimMetrics. VDM Publishing (2010).
- [**Taylor, 2005**], Taylor, A. (2005): *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, Cambridge (p. 167)
- [**Tournaire et al., 2011**], Tournaire, R., Petit, J.-M., Rousset, M.-C., Termier, A., (2011): Discovery of probabilistic mappings between taxonomies: principles and experiments. *J. Data Semant.* XV, 66–101.
- [**Trojahn et al, 2008**] Trojahn, C., Moraes, M., Quesma, P., Vieira, R., (2008): A cooperative approach for composite ontology mapping. *J. Data Semant.* X, 237–263 (p. 171).

- [Trojahn et al. 2011], Trojahn, C., Euzenat, J., Tamma, V., Payne, T., (2011): Argumentation for reconciling agent ontologies. In: Elai, A., Kona, M., Orgun, M. (eds.) *Semantic Agent Systems*, pp. 89–111. Springer, New York. Chap. 5 (pp. 19, 169, 360)
- [Valtchev, 1999] Valtchev, (1999), P. Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets. Thèse d'informatique, Université Grenoble 1.
- [Valtchev et Euzenat, 1997] Valtchev, P et Euzenat, J, (1997). Dissimilarity measure for collections of objects et values. In P. Coen X. Liu et M. Berthold, editors, *Proc. 2nd Symposium on Intelligent Data Analysis.*, volume 1280, pages 259–272.
- [Vesselin , 2011] Vesselin Petrov, *Ontological Landscapes: Recent Thought on Conceptual Interfaces Between Science and Philosophy*, 137 ff p. (ISBN 3868381074, lire en ligne ), « Chapter VI: Process ontology in the context of applied philosophy », 2011.
- [Wache et al, 2002], Wache, H., Visser, U., & Scholz, T. (2002). Ontology Construction - An Iterative and Dynamic Task. In: Florida Artificial Intelligence Research Society Conference (FLAIRS), Pensacola, FL, USA. pp. 445-449.
- [Wang et al., 2008], Wang P., Xu B. 2008): Debugging Ontology Mapping: A Static Approach. *Computing and Informatics*, vol. 27(1), pp. 2136 (.
- [Wang et al. 2011], Wang, P., Zhou, Y., Xu, B. (2011): Matching large ontologies based on reduction anchors. In: *Proc. 22<sup>nd</sup> International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, pp. 2343–2348 (pp. 152, 229)
- [Wu et al., 2019], Wu, J., Pan, Z., Zhang, C., & Wang, P (2019). Lily Results for OAEI 2019★.
- [Xu et al, 2003], Xu, L., & Embley, D. (2003, October 20). Using Domain Ontologies to Discover Direct and Indirect Matches for Schema Elements. In : *Second International Semantic Web Conference (ISWC-03)*.
- [Yager ,1988], Yager, R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans. Syst. Man Cybern.* 18(1), 183–190 (1988) (pp. 165, 166)
- [Zamazal, et Svátek, 2017], Zamazal, O., Svátek, V. (2017). The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. *Web Semantics: Science*
- [Zanobini, 2003] Zanobini, S., *“Improving CtxMatch by means of grammatical and ontological knowledge in order to handle attributes”*, disponible sur <http://eprints.biblio.unitn.it/archive/00000554/>, April 2006.
- [Zimmermann, 2008], Zimmermann, A. (2008). *Sémantique des réseaux de connaissances: gestion de l'hétérogénéité fondée sur le principe de médiation* (Doctoral dissertation, Université Joseph-Fourier-Grenoble I).
- [Zhao et al, 2018], Zhao, M., Zhang, S., Li, W., & Chen, G. (2018). Matching biomedical ontologies based on formal concept analysis. *Journal of biomedical semantics*, 9(1), 11.