ROYAUME DU MAROC

Faculté des Sciences et Techniques

Faculté des Sciences et Techniques

Département d'informatique

Béni-Mellal

N° d'ordre : --/2018

Centre d'Études Doctorales « Sciences et Techniques »

Formation Doctorale « Mathématique et Physique appliquée »

THÈSE

Présentée par

Mohamed BINIZ

Pour obtenir le grade de

Docteur

Spécialité : Informatique

Contribution à l'Amélioration de l'Alignement des Ontologies

Soutenue le 24/02/2018, devant la commission d'examen :

Pr. **Lalla S. CHADLI**	Faculté des Sciences et Techniques, Béni Mellal	Présidente
Pr. **B. MINAOUI**	Faculté des Sciences et Techniques, Béni Mellal	Rapporteur
Pr. **M. OUKESSOU**	Faculté des Sciences et Techniques, Béni Mellal	Rapporteur
Pr. **A. RACHIDI**	Ecole Nationale de Commerce et de Gestion, Agadir	Rapporteur
Pr. **K. NAFIL**	Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Rabat	Examinateur
Pr. **M. FAKIR**	Faculté des Sciences et Techniques, Béni Mellal	Directeur de thèse
Pr. **R. EL AYACHI**	Faculté des Sciences et Techniques, Béni Mellal	Co-directeur

À mes parents, et à ma famille qui se sont beaucoup sacrifiés pour mes études

Cette thèse a été réalisée au sein du Laboratoire de Traitement de l'Information et Aide à la Décision (**TIAD**) de la Faculté des Sciences et Techniques Béni Mellal (**FSTBM**) sous la direction de professeur Mohamed FAKIR et l'encadrement du professeur Rachid EL AYACHI

Remerciements

Ce travail a été effectué au sein du Laboratoire de Traitement de l'Information et Aide à la Décision (TIAD) de la Faculté des Sciences et Techniques Béni-Mellal (FSTBM) sous la direction de professeur Mohamed FAKIR. Je tiens à lui écrouler ici toute ma reconnaissance pour sa disponibilité et ses précieux conseils.

J'assure très sincèrement de ma gratitude, le professeur Rachid EL AYACHI, qui a dirigé une partie de mon travail et m'a longuement encouragé. Je suis sensible à l'honneur qu'il me fait de participer au jury.

Madame le Professeur Lalla Saadia CHADLI d'avoir accepté d'être président du jury. Je lui adresse mes sincères remerciements.

Je remercie les professeurs Brahim MINAOUI, Mohamed OUKESSOU, et Ali RACHIDI pour avoir accepté de rapporter ma thèse, et pour leurs conseils et leurs remarques pertinentes.

Je remercie également Professeur Khalid NAFIL pour l'intérêt qu'il a porté à mon travail et d'avoir accepté de participer au jury en tant qu'examinateur.

Je remercie tous les membres du laboratoire de Traitement de l'Information et Aide à la Décision et spécialement le Professeur Mohamed BASLAM.

Merci beaucoup à tous mes amis qui de près ou de loin m'ont aidé et encouragé aux moments opportuns. Je les remercie pour tout le temps précieux que nous avons passé ensemble.

ملخص

الويب الدلالي هو امتداد للويب حيث يتم تعريف البيانات المتاحة بشكل دلالي. تسمح هذه الميزة للتطبيقات بالوصول إلى البيانات المتنوعة للعديد من المصادر، ومعالجة تلك البيانات، وتقاسم نتائجها مع التطبيقات الأخرى.

الأونطولوجيا توفر تصور واضح للعلاقات بين البيانات في الويب الدلالي، وهي تعمل كبروتوكول للاتصال وتبادل النتائج بين التطبيقات.

في الغالب نجد العديد من الأونطولوجيات لنفس مفاهيم مجال معين، فمن الطبيعي أن يكون هناك مفاهيم متداخلة بين هذه الأونطولوجيات. في مثل هذه الحالات، لإنشاء قابلية التشغيل البيني للبيانات وتقاسمها بين التطبيقات، من الضروري مواجهة تحدي عدم التجانس الموجود بين المفاهيم التي تشكل الأونطولوجيا. ومن أجل التعامل مع مشكلة عدم التجانس بين الأونطولوجيات، تم إنشاء مجال محاذاة الأنطولوجيا. الهدف منها هو تحديد التوافق بين الكيانات وتوليد محاذاة بينهما. لذلك، يتم تشكيل المحاذاة بواسطة مجموعة من التوافقات بين الكينونات.

الهدف العام من هذا العمل هو تطوير نظام أكثر فعالية لمحاذاة الأنطولوجيا. ولتحقيق هذا الهدف، يقترح هذا العمل نظاما يحتوي على ثلاث طرق مقترحة وهي: OMBWSD وDROM وOM Neural NSGA-II.

Résumé

Le Web sémantique est une extension du Web dans laquelle les données disponibles sont définies sémantiquement. Cette fonctionnalité permet aux applications d'accéder à des données provenant de diverses sources, de traiter ces données et de partager leurs résultats avec d'autres applications.

Les ontologies permettent le fonctionnement du Web sémantique, car elles fournissent la conceptualisation explicite et formelle requise pour les données, fonctionnant comme un protocole de communication et de partage des résultats entre les applications.

Dans les scénarios d'ontologies multiples représentant des concepts d'un même domaine, il est courant qu'il y ait chevauchement de concepts entre ontologies. Dans des situations comme celle-ci, pour établir l'interopérabilité des données et leur partage entre applications, il est nécessaire de faire face au défi de l'hétérogénéité qui existe entre les concepts qui composent les ontologies. Afin de traiter le problème de l'hétérogénéité entre les ontologies, le domaine d'alignement d'ontologies a été créé. Le but de l'alignement des ontologies est d'identifier les correspondances entre les entités et de générer un alignement entre elles. Par conséquent, un alignement est formé par un ensemble de correspondances.

L'objectif général de ce travail est de développer un système d'alignement des ontologies plus efficace. Pour atteindre cet objectif, ce travail propose un système qui contient trois méthodes : OMBWSD, DROM et OM-NEURAL-NSGA-II.

Abstract

The Semantic Web is an extension of the Web in which the available data is defined semantically. This feature allows applications to access data from a variety of sources, process that data, and share their results with other applications.

Ontologies enable the semantic Web to work because they provide the explicit and formal conceptualization required for the data, functioning as a protocol for communication and sharing of results between applications.

In multiple ontology scenarios representing concepts of the same domain, it is common for there to be overlapping concepts between ontologies. In situations like this, to establish interoperability of data and its sharing between applications, it is necessary to face the challenge of the heterogeneity that exists between the concepts that make up ontologies. In order to deal with the problem of heterogeneity between ontologies, the domain of ontology alignment has been created. The goal of ontology alignment is to identify the matches between entities and to generate an alignment between them. Therefore, an alignment is formed by a set of matches.

The overall goal of this work is to develop a more effective ontology alignment system. To achieve this goal, this work proposes a system that contains three methods: OMBWSD, DROM and OM-NEURAL-NSGA-II.

Table de matières

Table des figures

Liste des tableaux

Glossaire

AM	Alignment Matrix
DL	Description Logic
DL	Description logics
F_x	F_x-measure
IRI	Internationalized Resource Identifier
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
OAEI	Ontology Alignment Evaluation Initiative
OWL	Ontology Web Language
P	Precision
R	Recall
RDF	Resource definition framework
RDF-S	Resource definition framework schema
SEALS	Semantic Evaluation At Large Scale
TAM	Temporary Alignment Matrix
URI	Uniform Resource Identifier
XML	Extended markup language

INTRODUCTION GÉNÉRALE

1. Contexte et motivation

Aujourd'hui, le Web offre une immense variété de sources d'informations[1] structurées, semi-structurées et non structurées (bases de données, pages Web, documents, figures, etc.) interconnectées par un nombre énorme de liens.

Chaque seconde, différents agents humains ou artificiels essaient de comprendre les données, intégrant différentes sources de données pour répondre aux exigences des internautes ou des professionnels afin d'explorer et d'utiliser les données disponibles, les agents devraient être en mesure de comprendre le message qu'il transmet et de formuler des requêtes significatives. Cependant, comprendre le sens est une tâche qui ne peut pas être achevée que par un agent humain. Actuellement, les ordinateurs ne visualisent pas, et ne stockent pas que les données sans comprendre les connaissances qu'elles véhiculent. Les machines ne peuvent rien faire pour extraire la sémantique. Elles ne voient que des chaînes de symboles où les gens voient des mots et des phrases. La recherche avec les moteurs de recherche était principalement basée sur la correspondance de chaîne sans tenir compte de la sémantique de l'entrée.

Rendre l'information compréhensible par machine est un problème clé de nos jours - par exemple, expliquer à l'ordinateur ce qu'est le « jaguar ». Les termes doivent être considérés dans leur contexte puisqu'il arrive parfois que le même terme soit utilisé pour représenter différents concepts ; par exemple : le jaguar comme dans le nom de l'animal et le jaguar comme une marque d'une voiture. Avec le temps, les significations des termes changent et de nouvelles significations pour les termes existants apparaissent - par exemple, Ibn Sina comme un nom du lycée à Rabat et Ibn Sina comme un nom d'un hôpital à rabat. Ainsi, pour comprendre la signification voulue, les agents doivent utiliser des définitions correspondantes pour les termes d'une source d'information qu'ils utilisent.

Les sources d'information représentent différents domaines, points de vue et applications envisagées. Ils se chevauchent souvent. Pour des applications différentes, par exemple : l'intégration de données et la communication d'agents, il est souvent nécessaire de connaître la relation entre les données disponibles à partir de sources séparées ou entre différentes versions d'une même source. Pour comprendre ces relations, les agents doivent comprendre la signification des données utilisées.

Le grand nombre de sources d'information à la disposition des agents sont souvent dans des états différents, ils peuvent couvrir un domaine thématique partiellement ou peut-être pas à jour ; fournissant ainsi des informations incomplètes pour un domaine. La combinaison de données provenant de différentes sources, qui ont été développées pour servir des applications différentes, peut conduire à une représentation incohérente d'un domaine. En conséquence, les agents peuvent utiliser des données incomplètes, incohérentes et erronées comme entrée pour leurs algorithmes.

Ces problèmes ont entraîné l'évolution du Web 2.0 vers le Web sémantique, où les machines peuvent comprendre et traiter des données sans interaction humaine. En tant que résultat, la vision du Web sémantique est en train de se concrétiser ; il y a quelques années Wikipédia a introduit dbpedia[2] et SemanticPedia permettant des capacités de recherche sémantique dans ces contenus. La Wikipédia n'est pas la seule organisation qui utilise les technologies du web sémantique. Time Inc., Elsevier, et la Bibliothèque du Congrès ont tous également des systèmes de production construits en utilisant des technologies web sémantiques.

Le développement rapide des technologies sémantiques influence de plus en plus tous les aspects de notre vie.

Le concept du Web sémantique englobe un ensemble de technologies qui permettent aux ordinateurs de comprendre les données qu'ils stockent. C'est une extension du Web, pas son remplacement. Cette vision a d'abord été introduite par Tim Berners-Lee [3]. À travers plusieurs exemples, la publication illustre un monde où les agents intelligents explorent le Web, collectent et intègrent des informations pertinentes ; à partir de diverses sources de données afin de remplir des tâches compliquées sans l'aide humaine. En revanche, les machines actuelles ne peuvent effectuer que des tâches simples précisément spécifiées à l'avance. Comme ils ne comprennent pas la signification des données qu'ils collectent, ils ne peuvent pas combiner la production de tâches multiples dans un seul résultat fonctionnel et tirer des conclusions (les humains doivent le faire).

Pour illustrer le concept du Web sémantique, considérez l'exemple d'une tâche sophistiquée, comme l'achat d'un produit via l'internet. L'achat comprend différents aspects, tels que :

- La disponibilité du produit à acheter dans un magasin.

- Les produits choisis doivent correspondre au critère personnel et doivent être compatibles avec les préférences personnelles et les restrictions - le temps de livraison, les frais de livraison, l'état de produit (neuf ou occasion).
- Le prix doit être le minimum possible pour même qualité d'un produit, en comptant le change de devise et les taxes de dédouanement.

2. Difficulté et défi

La vision du Web sémantique exige un système qui peut changer les données et réutiliser les données échangées avec leurs significations voulues. Ceci est appelé l'interopérabilité sémantique qui est très fastidieuse et difficile à réaliser entre les différents systèmes d'information. En outre, les erreurs sont inévitables dans une sorte distribuée et hétérogène de l'environnement comme dans les différentes applications de web, qui dispose de plusieurs milliards de pages.

L'hétérogénéité de l'information se produit, en général, au niveau de la syntaxe, la structure et la sémantique.

Au premier niveau, l'hétérogénéité syntaxique est le problème de l'hétérogénéité la plus simple, provoqué par l'utilisation de différents formats de données. Pour résoudre ce problème des formats normalisés tels que XML, RDF/RDFS et OWL, ont généralement été utilisés pour exprimer les données d'une manière uniforme qui rend le traitement automatique de l'information partagée plus facile.

Au deuxième niveau, ce qui est structurel, l'hétérogénéité se produit à la suite de l'information de façon structurée, même dans des environnements syntaxiquement homogènes. En effet, la normalisation du format n'a pas en soi de surmonter une telle hétérogénéité structurelle. Par exemple, une source peut modéliser une bibliothèque des livres, mais les classer en quelques catégories seulement, tandis qu'un autre peut faire des distinctions à grain très fin entre les types de livres selon des critères tels que leur auteur, le genre, etc.

Enfin, le troisième niveau, l'hétérogénéité sémantique, se produit par exemple lorsque deux ontologies ne partagent pas la même interprétation de l'information, aussi, par exemple, lorsque vous essayez de dire la même chose de différentes façons.

Actuellement, plusieurs études ont été faites dans ce domaine, généralement sur les systèmes qu'utilisent des techniques différentes pour calculer ou trouver les similitudes entre les concepts

ou termes dans les différentes ontologies sources. Un grand nombre de ces approches sont discutées ultérieurement. D'autre part, la plupart de ces approches d'alignement ignorent certains aspects réalistes cruciaux de l'alignement de l'ontologie. De ce fait, plusieurs limites majeures se présentent, à savoir :

- La majorité des systèmes d'alignements existants reposent sur un processus d'alignement, qui n'exploite pas la totalité des informations des entités contenues dans les ontologies. Ces descripteurs offrent des degrés d'expressivité d'influence sur la qualité de l'alignement issue.
- Développement des méthodes d'alignement multilingues.
- Traiter les ontologies de grandes tailles, et de différents domaines.

3. Contribution

01/01/2018	BINIZ, Mohamed, EL AYACHI, Rachid. Optimisions Ontology Alignment by using Neural NSGA-II. Journal of Electronic Commerce in Organizations (JECO) Volume 16, Issue 1. https://www.igi-global.com/article/optimizing-ontology-alignments-by-using-neural-nsga-ii/196179
29/03/2017-31/03/2017	The Third International Conference on Business Intelligence (CBI'17) Faculté des Sciences et Technique Béni Mellal, Morocco " An Ontology Alignment Hybrid Method Based on Decision Rules".
29/03/2017-31/03/2017	The Third International Conference on Business Intelligence (CBI'17) Faculté des Sciences et Technique Béni Mellal, Morocco " Optimizing Ontology Alignment by using Neural NSGA-II".

29/03/2017-31/03/2017 The Third International Conference on Business Intelligence (CBI'17) Faculté des Sciences et Technique Béni Mellal, Morocco " An OWL-DL Ontology Alignment System".

01/01/2017 : BINIZ, Mohamed, EL AYACHI, Rachid, et FAKIR, Mohamed. Ontology Matching Using BabelNet Dictionary and Word Sense Disambiguation Algorithms. Indonesian Journal of Electrical Engineering and Computer Science, 2017, vol. 5, no 1.

http ://iaescore.com/journals/index.php/IJEECS/article/view/6064/pdf

06/12/2016-08/12/2016 : The International Arab Conference on Information Technology (ACIT), Faculté polydisciplinaire Béni Mellal, Morocco " Word Boundary Detection in Tifinagh using MaxEnt and n-gram algorithms".

http ://acit2k.org/ACIT/images/stories/year2014/month1/proceeding/79.pdf

28/11/2016-29/11/2016 : 7e Conférence Internationale sur les Technologies d'Information et de Communication pour l'Amazighe, Institut Royal de la Culture amazighe Rabat, Morocco » Détection automatique de fin des phrases pour la langue amazighe ».

http ://event.ircam.ma/data/papers2016/8.pdf

23/04/2015-25/04/2015 : Second international Conference on Business Intelligence (CBI'15) Faculté des Sciences et Technique Béni Mellal, Morocco « Alignement des ontologies en utilisant le dictionnaire WordNet et les algorithmes de désambiguïsation ».

24/11/2014-25/11/2014 :	6e Conférence Internationale sur les Technologies d'Information et de Communication pour l'Amazighe, Institut Royal de la Culture amazighe Rabat, Morocco » Managing Semantic Web Evolution in the Organization".

4. Organisation du manuscrit

La thèse est structurée comme suit : le premier chapitre donne un contexte sur les ontologies et fournit plus de détails sur l'alignement des ontologies. À la fin de ce chapitre, plusieurs systèmes d'alignement d'ontologies sont donnés. Le deuxième chapitre introduit notre système intégré avec ses trois méthodes (OMBWSD, DROM et OM-NEURAL-NSGA-II) ainsi que leurs algorithmes et leur flux de travail. Les mesures d'évaluation, les expériences réalisées avec le système, une comparaison de notre système avec les autres systèmes et une discussion de leurs résultats sont données au troisième chapitre.

CHAPITRE 1 : ALIGNEMENT DES ONTOLOGIES

1.1 Introduction

Ce chapitre présente les concepts de base du Web sémantique, définit la structure et les principales applications de l'ontologie et fournit un aperçu des langages d'ontologie utilisés pour l'exprimer sur le Web. Tous les termes pertinents sont expliqués pour fournir une compréhension de base des ontologies et d'alignement qui sont à la base de ce travail.

Il présente également diverses méthodes d'alignement. Enfin, il expose un aperçu de l'état de l'art et quelques systèmes d'alignement d'ontologies.

Parmi les objectifs du Web est de créer une source de référence pour des informations sur plusieurs sujets, tandis que le Web sémantique est conçu pour créer un réseau de signification.

La fondation des vocabulaires et une communication efficace sur le Web sémantique sont une ontologie.

D'après Gruber[4] : » Ontologie fournit une spécification formelle et explicite d'une conceptualisation partagée d'un domaine »

Par conséquent, il facilite le partage des connaissances sur les systèmes distribués ; en d'autres termes, il permet aux systèmes ou aux applications de coopérer.

1.2 Web sémantique

Le Web sémantique est distribué et hétérogène, a amené l'évolution du Web à un niveau plus élevé. Il existe deux visions du futur dans la conception du Web, la première étant d'améliorer sa convivialité en tant que moyen de collaboration et la seconde pour s'assurer que son contenu peut être compris par les machines. La fourniture de données d'annotation facilitera ce deuxième but.

Tim Berners-Lee[5], qui a inventé le WWW et a travaillé sur le Web sémantique, déclare que ce dernier est une extension du Web actuel, dans lequel l'information est bien définie, permettant aux ordinateurs et aux personnes de coopérer. Ainsi, le Web sémantique se distingue par une représentation plus significative de l'information pour les humains et les ordinateurs, fournissant une description de son contenu et de ses services sous une forme lisible par machine ; de plus, il permet aux services d'être automatiquement annotés, découverts, publiés, annoncés et composés. Cela facilite l'interopérabilité et le partage des connaissances sur le

Web. Son objectif principal est donc de rendre l'information sur le Web accessible et compréhensible par les humains et les ordinateurs.

En fait, le Web sémantique et les services Web sont considérés comme un ensemble de ressources identifié par l'URI. La différence entre eux est que les services Web utilisent HTTP pour afficher le contenu d'une page, tandis que le Web sémantique tente de créer la lisibilité de la machine en représentant sémantiquement les données ou les informations dans les ressources. De nombreux outils et applications de technologies Web sémantiques sont récemment disponibles.

Les couches d'architecture de web sémantique[6] représentées dans la figure 1 sont décrites brièvement ci-dessous :

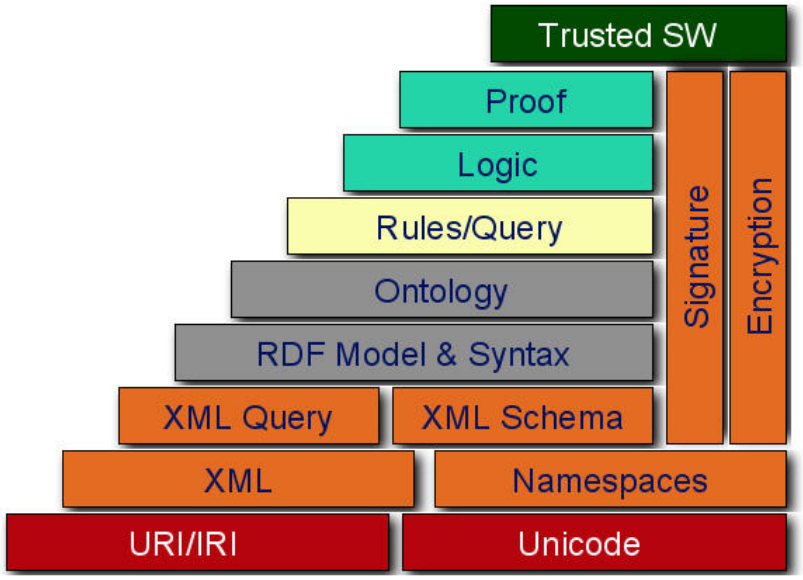

Figure 1 : Architecture en couches du web sémantique

- URI et Unicode : pour identifier et localiser les ressources sur le Web, un système uniforme d'identificateurs (URI) est utilisé. L'URI, qui est considéré comme le fondement du Web, est utilisé pour donner un nom unique à chaque ressource. Unicode est la norme pour la représentation du personnage informatique.
- Extensible Markup Language (XML)[7] est un langage de balisage, ce qui signifie qu'il est lisible par machine et possède son propre format. Il est largement connu dans la communauté WWW, car il a un format de texte flexible et a été conçu pour décrire les données et relever les défis de l'édition électronique à grande échelle et de l'édition électronique ; il joue un rôle important dans l'échange de différents types de données sur le Web. En fait, c'est à la base d'un nombre croissant d'activités de développement de

logiciels. Chaque document commence par une déclaration d'espace de noms utilisant l'espace de noms XML.

- Le Resource Description Framework (RDF)[8] est la première couche du Web sémantique. RDF est un cadre pour l'utilisation et la représentation des métadonnées et la description de la sémantique des informations sur les ressources sur le Web d'une manière accessible à la machine. Il utilise des URI pour identifier les ressources Web et pour décrire les relations entre ces ressources, en utilisant un modèle de graphique. Tout en décrivant les classes de ressources et les propriétés entre elles, en utilisant le schéma RDF (qui est un langage de modélisation simple), il fournit également un cadre de raisonnement simple pour inférer des types de ressources.
- Vocabulaire de l'ontologie, est un langage qui fournit un vocabulaire et une grammaire communs pour les données publiées, ainsi qu'une description sémantique des données utilisées pour préserver les ontologies et pour les garder prêts à s'intéresser. L'ontologie signifie décrire la sémantique des données, fournissant un moyen uniforme de permettre la communication par laquelle différentes parties peuvent se comprendre.
- Logique et preuve[9] : dans le Web sémantique, la construction des systèmes suit une logique qui tient compte de la structure de l'ontologie. Un raisonneur pourrait être utilisé pour vérifier et résoudre les problèmes de cohérence et la redondance de la traduction de concepts. Un système de raisonnement est utilisé pour faire de nouvelles inférences.
- Trust[9] est la dernière couche du Web sémantique. Cette composante concerne la fiabilité de l'information sur le Web afin de garantir l'efficacité de sa qualité.

1.3 Ontologie

Les ontologies, qui sont utilisées pour soutenir l'interopérabilité et la compréhension commune entre les différentes parties, sont un élément clé dans la résolution du problème de l'hétérogénéité sémantique, permettant ainsi une interopérabilité sémantique entre différentes applications et services Web.

Récemment, les ontologies sont devenues un sujet de recherche populaire dans de nombreux domaines, y compris le commerce électronique, la gestion des connaissances, l'ingénierie des connaissances et le traitement du langage naturel. Les ontologies fournissent une compréhension commune d'un domaine qui peut être communiqué entre les personnes et des systèmes d'applications hétérogènes et largement répandus. En fait, ils ont été développés dans les communautés de recherche d'intelligence artificielle (AI) pour faciliter le partage et la

réutilisation des connaissances. L'objectif d'une ontologie est de parvenir à une connaissance commune et partagée qui peut être transmise entre les personnes et entre les systèmes d'application. Ainsi, les ontologies[10] jouent un rôle important dans l'interopérabilité entre les organisations et sur le Web sémantique, car elles visent à saisir les connaissances de domaine et leur rôle est de créer une sémantique explicitement de manière générique, en fournissant la base d'un accord au sein d'un domaine. L'ontologie est utilisée pour permettre l'interopérabilité entre des applications Web de différentes zones ou de différentes vues sur une zone. Pour cette raison, il est nécessaire d'établir des mappages parmi les concepts d'ontologies différentes pour capturer la correspondance sémantique entre eux. Cependant, établir une telle correspondance n'est pas une tâche facile.

Étant donné qu'il existe de nombreuses définitions de l'ontologie, la présente recherche présente d'abord certaine de ces définitions qui ont été données à partir de différentes perspectives, puis explorent en profondeur les aspects de ces définitions qui sont liés au sujet d'une enquête. L'utilisation principale du mot « médecine » est dans la discipline de la philosophie, où cela signifie « l'étude ou la théorie de l'explication de l'être » ; il définit ainsi une entité ou un être et sa relation avec son activité et son activité dans son environnement. Dans d'autres disciplines, telles que l'ingénierie logicielle et l'IA, elle est définie comme « une spécification explicite formelle d'une conceptualisation partagée ». Les fondements de cette définition sont les suivants :

- Toute connaissance (par exemple, le type de concepts utilisés et les contraintes sur leur utilisation) dans l'ontologie doit avoir une spécification explicite.
- Une ontologie est une conceptualisation, ce qui signifie qu'elle a un concept universellement compréhensible.
- Shared indique un accord sur la signification dans de tels domaines. En d'autres termes, une ontologie devrait prendre en compte les connaissances consensuelles acceptées par les communautés.

1.4 Représentation d'une ontologie

L'ontologie comprend quatre composantes principales : les concepts, les instances, les relations et les axiomes. La présente recherche adopte les définitions[11] suivantes de ces composantes ontologiques :

- Un concept (également appelé classe ou terme) est un groupe, un ensemble ou une collection abstraite d'objets. C'est l'élément fondamental du domaine, représente habituellement un groupe, ou une classe dont les membres partagent des propriétés communes. Ce composant est représenté dans des graphiques hiérarchiques, de sorte qu'il ressemble à des systèmes orientés objet. Le concept est représenté par une « super classe », représentant la classe supérieure ou la « classe parente », et une « catégorie » qui représente la classe subordonnée ou appelée « classe enfant ». Par exemple, une université pourrait être représentée comme une classe avec de nombreuses sous-classes, telles que les facultés, les bibliothèques et les employés.
- Une instance (un individu) est le composant d'une ontologie qui représente un objet ou un élément spécifique d'un concept ou d'une classe. Par exemple, « Jordan » pourrait être une instance de la classe « Pays arabes » ou simplement « pays ».
- Une relation (également connue sous le nom de fente) est utilisée pour exprimer des relations entre deux concepts dans un domaine donné. Plus précisément, il décrit la relation entre le premier concept, représenté dans le domaine, et le second, représenté dans la gamme. Par exemple : « l'étude » pourrait représentée comme une relation entre le concept « personne » (qui est un concept dans le domaine) et « université » ou « collégialité » (ce qui est un concept dans la gamme).
- Un Axiome est utilisé pour imposer des contraintes sur les valeurs des classes ou des instances, de sorte que les axiomes sont généralement exprimés en utilisant des langages basés sur la logique tels que la logique de premier ordre ; ils sont utilisés pour vérifier la cohérence de l'ontologie.

1.5 Structure d'une ontologie

En général, la structure d'une ontologie[11] est décrite comme une

$$O = (C, R, I, H^C, H^R)$$

Sachant que :

- C Représente un ensemble de concepts (instances de type rdf :Class "). Ces concepts sont organisés avec une hiérarchie de subsomption correspondante H^C .
- R représente un ensemble de relations qui relient les concepts les uns aux autres (instances de rdf :Property). Ri $\in$ R et Ri $\rightarrow$ C $\times$ C.

- H^C représente une hiérarchie conceptuelle sous la forme d'une relation (une relation binaire correspondant à -rdfs : subClassOf). $H^C \subseteq C \times C$, où HC (C1, C2) indique que C1 est un sous-concept de C2.
- H^R représente une hiérarchie de relation sous la forme d'une relation HR $\subseteq$ R × R, où H^R (R1, R2) indique que R1 est un sous-rapport de R2 (rdfs : subPropertyOf).
- I est l'instanciation des concepts dans un domaine particulier (rdf : type).

1.6 Alignement des ontologies

L'alignement de l'ontologie[11] est le processus ou la méthode de création d'un lien cohérent entre deux ou plusieurs ontologies en les mettant en accord mutuel. Les alignements d'ontologie sont utilisés pour décrire clairement comment les concepts dans les différentes ontologies sont logiquement liés. Cela signifie que des axiomes supplémentaires illustrent la relation entre les concepts dans différentes ontologies sans changer la signification dans les ontologies originales.

En fait, l'alignement de l'ontologie est utilisé comme prétraitement pour la fusion de l'ontologie et l'intégration de l'ontologie. Il existe de nombreuses définitions différentes de l'alignement de l'ontologie, en fonction de son application et de ses résultats escomptés. Les exemples de définitions comprennent les éléments suivants :

- L'alignement de l'ontologie est utilisé pour établir des correspondances entre les ontologies sources et pour déterminer l'ensemble des concepts se chevauchant, des concepts qui ont un sens similaire, mais ont des noms ou une structure différents et des concepts propres à chacune des sources ».
- « L'alignement de l'ontologie est le processus consistant à intégrer deux ou plusieurs ontologies, ce qui le rend cohérent ».
- Donné deux ontologies O_1 et O_2 ; le mappage d'une ontologie sur un autre signifie que chaque entité (concept C, relation R ou instance) dans l'ontologie O_1, essaie de trouver une entité correspondante (c'est-à-dire en utilisant des algorithmes correspondants) dans l'ontologie O_2 ».

1.7 Enquête sur l'alignement des ontologies

1.7.1 Problème d'alignement d'ontologie

L'alignement des ontologies est un champ qui détermine la correspondance d'une paire d'ontologies O_1 et O_2. Par conséquent, en donnant une paire d'ontologies, la tâche

correspondante est de trouver un alignement entre ces ontologies. Selon la définition d'Euzenat [11], d'autres paramètres peuvent étendre la définition de l'alignement (voir Figure 2), à savoir :

- L'utilisation d'une entrée d'alignement A, qui doit être étendue
- Les paramètres correspondants, comme le poids ou les seuils.
- Ressources externes, telles que les connaissances communes et le domaine spécifique des thésaurus.

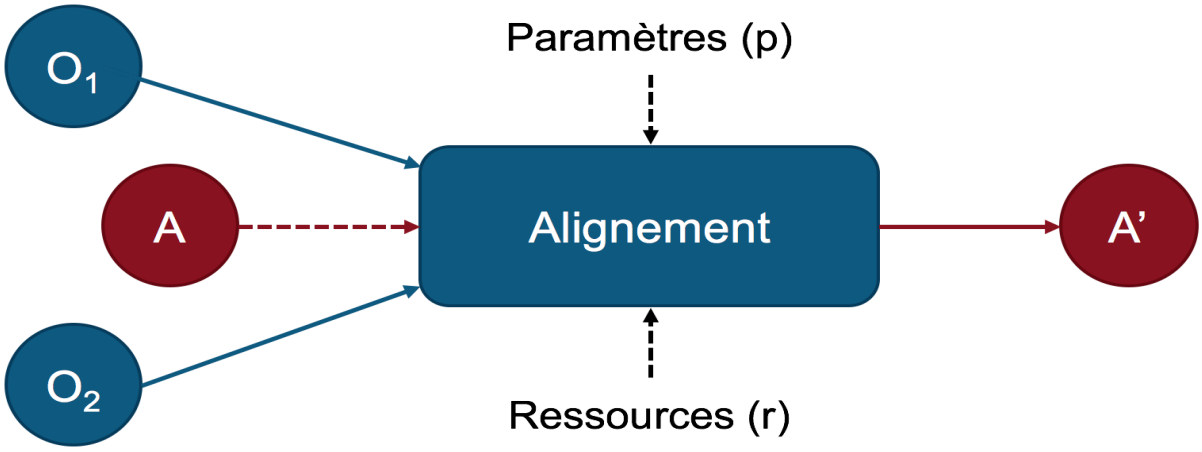

Figure 2 : processus d'alignement

Étant donné deux ontologies, Euzenat a mis l'alignement par 4-tuple : <id, E1, E2, R>, où :

- id est un identifiant pour la correspondance de données.
- E_1 et E_2 sont des entités, par exemple, les classes et les propriétés de la première et de la deuxième ontologie, respectivement.
- R est une relation entre E_1 et E_2, par exemple : équivalente, plus générale, disjointe.
- Les alignements peuvent avoir des cardinalités différentes : un à un, un à plusieurs, plusieurs à un, ou plusieurs à plusieurs.

1.7.2 Méthodes d'alignement

1.7.2.1 Méthodes de similarités

La similarité sémantique est considérée comme une similitude topologique en mathématiques, où elle est associée à une fonction appelée fonction de similarité. La valeur de cette fonction est souvent comprise entre 0 et 1. Dans la littérature, selon le schéma d'euznat pour calculer la

similitude, il existe deux méthodes : méthodes de base (Méthodes de terminologie, méthodes linguistiques, méthodes structurales, méthodes d'extension et méthodes sémantiques) et méthodes de combinaison (somme pondérée, produits pondérés, etc.).

Dans cet article, nous n'expliquons que les méthodes utilisées comme suit :

1.7.2.2 Méthodes terminologiques

1.7.2.2.1 Similarité de Jaro :

La similitude de Jaro[12] entre deux chaînes de caractère, est définie par :

$$Simj(s,r) = \frac{1}{3}\left(\frac{m}{|s|} + \frac{m}{|r|} + \frac{m-t}{m}\right) \quad (1)$$

Avec :

- $|s|$ et $|r|$ Sont respectivement les longueurs des chaînes de caractères s et r.
- m est le nombre de caractères correspondants.
- t est le nombre de transpositions.

1.7.2.2.2 Similarité de Jaro-winkler :

La métrique Jaro-winkler[12] produit la similitude entre deux chaînes en fonction du nombre et de l'ordre des caractères communs.

$$SimJW(s,r) = Simj(s,r) - l.p(1 - Simj(s,r)) \quad (2)$$

Sachant que :

- l est la longueur du préfixe commun (4 caractères maximum).
- p (p=0.1) est un coefficient qui favorise les chaînes avec un préfixe commun.

1.7.2.2.3 Similarité de Levenshtein

La distance Levenshtein[12] entre deux chaînes est le coût minimum des opérations d'édition qui doivent transformer une chaîne en une autre. Chaque opération à une fonction de coût associé, sous la forme la plus simple, chacun a coûté. La distance de Levenshtein peut résoudre

très bien les erreurs typographiques des variations de noms et la distance peut être transformée en similitude en soustrayant la distance normalisée de 1 :

$$Levein(c1, c2) = 1 - \frac{lev(c1, c2)}{\max\{|c1|, |c1|\}} \quad (3)$$

Sachant que :

- C1 et c2 : sont les deux valeurs comparées.
- Lev (c1, c2) : la distance (coût) entre c1 et c2.
- | C1 | et | c2 | : la longueur de c1 et c2 respectivement.

1.7.2.3 Méthode structurelle (Similarity Flooding)

L'inondation de similarité[13] est un algorithme qui prend deux graphiques en entrée et produit comme sortie un mappage entre les nœuds correspondants des graphiques.

- Dans une première étape, nous transformons des ontologies dans un graphique G dans lequel les sommets sont des paires de concepts et dont les noyaux existent entre deux nœuds, s'il existe une relation dans les deux ontologies entre les nœuds des deux paires. En fait, l'algorithme original d'inondation de similarité n'associe que les concepts dont les bords ont la même étiquette.
- Dans une deuxième étape, nous affectons le poids w aux bords, qui sont généralement 1/n, où n est le nombre de bords sortants.
- En troisième étape, nous assignons une similitude initiale σ^0 pour chaque nœud.
- En quatrième étape, nous calculons σ^{i+1} pour chaque nœud avec la formule suivante :

$$\sigma^{i+1}(x, x') = \sigma^i(x, x') + \sum_{(\langle y,y\prime\rangle,p,\langle x,x\prime\rangle)} \sigma^i(y, y') \times w(\langle y, y'\rangle, p, \langle x, x'\rangle) \quad (4)$$

Sachant que :

- $\boldsymbol{\sigma^{i+1}(x, x')}$ est la valeur de similarité entre deux entités (x, x') dans l'itération i+1.

- $\boldsymbol{\sigma^i(x, x')}$ est la valeur de similarité entre deux entités $(\mathrm{x}, \mathrm{x}')$ dans l'itération i.
- $\boldsymbol{\sigma^i(y, y')}$ est la valeur de similarité entre deux entités $(\mathrm{y}, \mathrm{y}')$ dans l'itération i.
- $\boldsymbol{w(\langle y, y'\rangle, p, \langle x, x'\rangle)}$ est le poids de l'arc sortant des entités$(\mathrm{y}, \mathrm{y}')$ vers aux entités $(\mathrm{x}, \mathrm{x}')$.

- Dans une cinquième étape, nous normalisons tous σ^{i+1} en divisant par la plus grande valeur.
- Sixième étape, si aucune similitude ne change plus que le seuil ϵ, ou après un nombre d'étapes préfixées, arrêtez-vous, passent à la quatrième étape.

1.7.2.4 Méthode linguistique

1.7.2.4.1 Similarité de Lesk et de Lesk amélioré

Lesk[14] a proposé un algorithme de désambiguïsation de sens de mot très simple, qui considère la similitude entre deux sens comme le nombre de mots en commun dans leurs définitions. Dans la version originale, il ne prend pas en compte l'ordre des mots dans les définitions (sac de mots[15]). La similitude de Lesk s'exprime par l'équation (5).

$$\mathrm{LESK(CLASS1, CLASS2)} = \mathrm{DEFINITION(CLASS1)} \cap \mathrm{DEFINITION(CLASS2)} \qquad (5)$$

Pederson et Banerjee[16] ont proposé un Lesk amélioré, appelé « Adapted Lesk » et défini par l'équation (7), en fonction de deux étapes. La première étape est l'incorporation des définitions du sens lié par les relations taxonomiques de BabelNet dans la définition d'un sens donné. Dans la deuxième étape, il calcule le chevauchement entre les définitions des mots en considérant non seulement le chevauchement entre les définitions des deux sens, mais aussi les définitions des relations R : hyperonymes (has-kind), hyponymes (type de) méronymes (partie-de) des abréviations (à la main), mais aussi par des troponymes attribuer des relations (similaires et aussi à voir). Pour s'assurer que la mesure est symétrique, les auteurs proposent de regrouper les évaluations de récupération entre les définitions des relations paires $\Re$.

Soit ψ la série de connexions pour calculer la récupération. Un ensemble A est défini par l'équation (6) :

$$\Re = \{(\mathrm{R1}, \mathrm{R2}) | \forall (\mathrm{R1}, \mathrm{R2}) \in \psi^2, (\mathrm{R1}, \mathrm{R2}) \in \Re^2 \Rightarrow (\mathrm{R1}, \mathrm{R2}) \in \Re^2\} \qquad (6)$$

Par conséquent, le score est calculé comme la somme de chevauchement entre les définitions de paires de relations :

$$\mathrm{ADLESK}(\mathrm{C1},\mathrm{C2}) = \sum_{\forall(\mathrm{R1},\mathrm{R2})\in\Re^2} (|\mathrm{DEFINITION}(\mathrm{R1}(\mathrm{C1})) \cap \mathrm{DEFINITION}(\mathrm{R2}(\mathrm{C2}))|)^2 \qquad (7)$$

1.7.2.4.2 Similarité de Wu & Palmer

La similarité de Wu et Palmer[17] mesurent la profondeur de deux concepts dans la taxonomie donnée WordNet[18] et la profondeur de leur ancêtre commun le plus bas (Lowest Common Subsume [LCS]) et les combinent pour créer un score de similarité :

$$WUP(C1, C2) = \frac{2 * depth\ (LCS)}{depth(C1) + depth\ (C2)} \qquad (8)$$

Sachant que :

- **depth(LCS)** est la profondeur de l'ancêtre commun le plus bas entre deux classes et la racine.
- **depth(C1)** est la profondeur de la racine à la classe 1.
- **depth(C2)** est la profondeur de la racine à la classe 2.

La figure 3 représente la taxonomie de l'université.

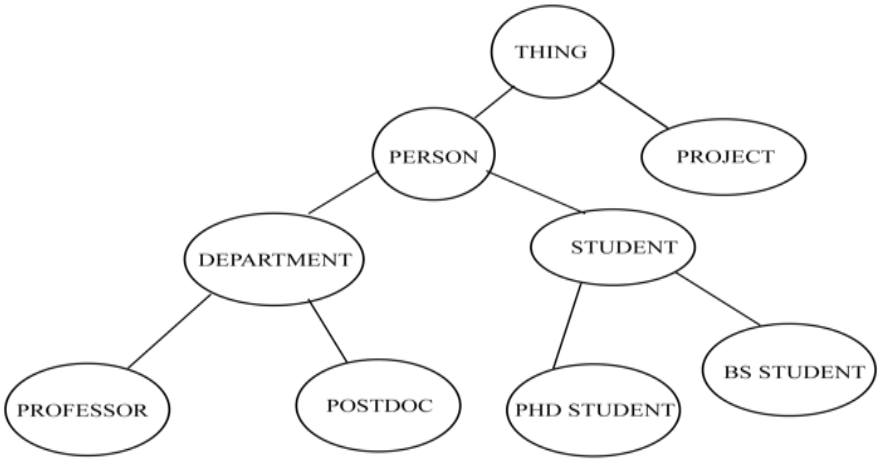

Figure 3 : Ontologie d'une université

D'après la taxonomie de la figure 3, on considère les concepts PROFESSEUR, BS-ÉTUDIANT et POSTDOC, sachant que le LCS du PROFESSEUR et BS-ÉTUDIANT est

PERSONNE et que le LCS du PROFESSEUR et du POSTDOC est DÉPARTEMENT, nous avons :

$$\mathrm{WUP}(\mathrm{PROFESSOR}, \mathrm{BS\ STUDENT}) = \frac{2 * \mathrm{depth}(\mathrm{PERSON})}{\mathrm{depth}(\mathrm{PROFESSOR}) + \mathrm{depth}(\mathrm{BS\ STUDENT})} = \frac{2 * 1}{3 + 3}$$

$$= 0.33$$

Et

$$\mathrm{WUP}(\mathrm{PROFESSOR}, \mathrm{POSTDOC}) = \frac{2 * \mathrm{depth}(\mathrm{DEPARTMENT})}{\mathrm{depth}(\mathrm{PROFESSOR}) + \mathrm{depth}(\mathrm{POSTDOC})} = \frac{2 * 2}{3 + 3}$$

$$= 0.66$$

Par conséquent, PROFESSOR et POSTDOC sont plus proches l'un de l'autre que BS-STUDENT.

1.7.2.4.3 Similarité de Leacock et Chodorow

Leacock et Chodorow[19] ont utilisé une seule relation (hyponymie) et ont changé la formule de la longueur du chemin pour refléter le fait que l'arc inférieur de l'hyponymie hiérarchique correspond à la plus petite distance sémantique. La similarité entre les deux concepts Class1 et Class2 est donnée par l'équation (9) :

$$\mathrm{LCH}(\mathrm{CLASS1}, \mathrm{CLASS2}) = -\log\left(\frac{\text{shorter path length}(\mathrm{CLASS1}, \mathrm{CLASS2})}{2 * \mathrm{depth}}\right) \qquad (9)$$

D'après la taxonomie de la figure 3, si on considère les concepts PROFESSOR, POSTDOC et BS-STUDENT, nous avons :$\mathrm{LCH}(\mathrm{PROFESSOR}, \mathrm{BS\ STUDENT}) = -\log\left(\frac{4}{2*3}\right) = 0.176$

Et

$$\mathrm{LCH}(\mathrm{PROFESSOR}, \mathrm{POSTDOC}) = -\log\left(\frac{2}{2 * 3}\right) = 0.4771$$

Par conséquent, PROFESSOR et POSTDOC sont plus proches l'un de l'autre que BS-STUDENT.

1.7.2.4.4 Similarité de Resnik

La mesure de resnik[20] renvoie le contenu informationnel (IC) du sous-dénominateur commun le plus bas (LCS) de deux concepts donnés, elle est calculée par l'équation (10) :

$$\begin{aligned} \mathrm{RES}(\mathrm{CLASS1}, \mathrm{CLASS2}) &= \mathrm{IC}\big(\mathrm{LCS}(\mathrm{CLASS1}, \mathrm{CLASS2})\big) \\ &= -\log\Big(\mathrm{P}\big(\mathrm{LCS}(\mathrm{CLASS1}, \mathrm{CLASS2})\big)\Big) \end{aligned} \qquad (10)$$

On considère les concepts de PROFESSEUR, BS-ÉTUDIANT et POSTDOC ; sachant que le LCS du PROFESSEUR et de l'ÉTUDIANT BS est PERSONNEL et que le LCS du PROFESSEUR et du POSTDOC est DÉPARTEMENT, et si nous supposons que IC (PROFESSEUR) = IC (BS STUDENT) = 0,4, IC (POSTDOC) = 0,5, IC (PERSONNE) = 0,3 et IC (DÉPARTEMENT) = 0,5, nous avons :

$$\begin{aligned} \mathrm{RES}(\mathrm{PROFESSOR}\,, \mathrm{BS\ STUDENT}) &= \mathrm{IC}\big(\mathrm{LCS}(\mathrm{PROFESSOR}, \mathrm{BS\ STUDENT})\big) = \mathrm{IC}(\mathrm{PERSON}) \\ &= 0.3 \end{aligned}$$

Et

$$\begin{aligned} \mathrm{RES}(\mathrm{PROFESSOR}, \mathrm{POSTDOC}\,) &= \mathrm{IC}\big(\mathrm{LCS}(\mathrm{PROFESSOR}, \mathrm{POSTDOC}\,)\big) = \mathrm{IC}(\mathrm{DEPARTMENT}) \\ &= 0.5 \end{aligned}$$

1.7.2.5 Méthode extensionnelle : similarité basée sur les instances

Pour la similarité basée sur les instances, par exemple, nous utilisons le modèle d'espace vectoriel [21] (VSM), il s'agit d'un modèle algébrique pour représenter des documents texte (et tous les objets, en général) en tant que vecteurs d'identifiants, tels que les termes d'index

Les documents et les requêtes sont représentés sous forme de vecteurs :

- $d_j = \{w_{j1}, w_{j2}, w_{j3}, w_{j4}, \ldots, w_{jn}\}$.
- $d_q = \{w_{q1}, w_{q2}, w_{q3}, w_{q4}, \ldots, w_{qn}\}$.

Sachant que :

- w_{ji} est le poids du terme dans la position i et le document j.

Dans le modèle d'espace vectoriel classique, les mots-clés d'un document sont attribués à des poids qui reflètent que certains mots sont meilleurs pour discriminer les documents que d'autres. De même, dans notre approche, les annotations ont un poids qui reflète la pertinence

de l'instance est considéré comme le document de sens. Les poids sont calculés automatiquement par une adaptation de l'algorithme TF-IDF[22], en fonction de la fréquence d'occurrence des instances dans chaque document. Plus précisément le poids $w_{j,i}$ de l'instance Ii pour le document dj est calculé comme suit :

$$W_{j,i} = \frac{Occur_{j,i}}{max_k Occur_{j,k}} \times log \frac{N}{n_i} \quad (11)$$

Sachant que :

- $Occur_{j,i}$ est le nombre d'occurrences de I_i en d_j
- $max_k Occur_{j,k}$ est la fréquence de l'instance la plus répandue dans d_j.
- n_i est le nombre de documents annotés avec I_i.
- N est le nombre total de documents dans l'espace de recherche.

Après la modélisation des documents utilisant le modèle d'espace vectoriel et les algorithmes TF-IDF, la similitude est calculée par l'algorithme Cosine en utilisant l'équation suivante :

$$VSM(d_j, d_q) = \frac{d_j \times d_q}{|d_j| \times |d_q|} = \frac{\sum_i w_{j,i} \times w_{q,i}}{\sqrt{\sum_i w_{j,i}^2} \times \sqrt{\sum_i w_{q,i}^2}} \quad (12)$$

1.8 Systèmes existants d'alignement d'ontologie

1.8.1 AgreementMakerLight

AgreementMakerLight[23][24] est un système automatique, il dispose d'une interface utilisateur sophistiquée ; et de fournir un ensemble de stratégies d'évaluation. Il a été conçu pour gérer les ontologies à grande échelle. Le système gère les ontologies dans XML, RDF Schema[25] et OWL. Il est structuré en deux modules : le calcul de la similarité et la sélection de l'alignement. Le système combine des correspondants utilisant trois couches :

- Les correspondants de la première couche comparant des caractéristiques de conception, telles que des étiquettes, des commentaires et des instances, qui sont représentées comme des vecteurs TF-IDF utilisés avec une métrique de similarité cosinusoïdale et d'autres correspondances de chaînes.

- La deuxième couche utilise les propriétés structurelles de l'ontologie. Il comprend deux correspondants : l'héritière de similarité descendante (DSI) et l'appariement de la similarité entre frères et sœurs (SSC)[26].
- Dans la troisième couche, une combinaison linéaire pondérée est calculée sur la base des résultats des deux premières couches, dont les résultats sont filtrés sur la base de seuils.

1.8.2 RIMOM

RIMOM[27] est un système dynamique d'alignement d'ontologie. Il se concentre sur la combinaison de stratégies d'appariement multiple par la minimisation du risque de décision bayésienne. Le système estime quantitativement les caractéristiques de similarité pour chaque tâche d'appariement. Ces caractéristiques permettent de sélectionner et de combiner dynamiquement les méthodes d'appariement multiples.

Deux méthodes d'appariement sont utilisées : la similarité terminologique (Édit distance) et la similarité structurelle (Similarity d'inondation). Si les deux ontologies ont un fort facteur de structure de similarité, le système utilise un processus de propagation de similarité pour affiner les alignements trouvés et trouver de nouveaux alignements qui ne peuvent pas être trouvés en utilisant d'autres stratégies.

1.8.3 ALIN

ALIN[28] est un système d'alignement d'ontologies, spécialisé dans l'alignement interactif d'ontologies, basé principalement sur des techniques d'appariement linguistique, utilisant le WordNet comme ressource externe. Après avoir généré un ensemble initial de correspondances (appelé ensemble de correspondances candidates, correspondant aux correspondances sélectionnées pour recevoir le retour de l'expert), des interactions sont faites avec l'expert, et à chaque interaction, l'ensemble des correspondances candidates est modifié. La modification de l'ensemble des correspondances candidates se fait par l'utilisation de l'analyse structurelle des ontologies et de l'utilisation des anti-patterns d'alignement. Les interactions se poursuivent jusqu'à ce qu'il n'y ait plus des correspondances candidates. ALIN a été construit avec un accent particulier sur la base de correspondance interactive d'OAEI 2017.

1.8.4 LogMap et ces variantes

LogMap[29], [30] est un système de mise en correspondance d'ontologies hautement évolutive qui implémente les principes de cohérence et de localité. LogMap prend également en charge

l'interaction de l'utilisateur (en temps réel) pendant le processus de correspondance, ce qui est essentiel pour les cas d'utilisation nécessitant des mappages très précis. LogMap est l'un des rares systèmes de correspondance d'ontologies ; qui peut efficacement associer des ontologies sémantiquement riches contenant des dizaines de milliers de classes. Il intègre des techniques sophistiquées de raisonnement et de réparation pour minimiser le nombre d'incohérences logiques. Il fournit un support pour l'intervention de l'utilisateur pendant le processus d'appariement.

LogMapLt est une variante « légère » de LogMap, qui applique uniquement d'une manière essentielle des techniques (efficaces) de correspondance de chaînes.

LogMapBio inclut une extension pour utiliser BioPortal en tant que fournisseur (dynamique) d'ontologies médiatiques au lieu de s'appuyer sur quelques ontologies présélectionnées.

1.8.5 XMap

eXtended Mapping (XMap)[31], [32] est l'un des principaux systèmes d'appariement d'ontologies pour des ontologies à grande échelle en s'appuyant sur la notion de contexte afin de traiter l'ambiguïté lexicale ainsi qu'une approche de diviser pour régner.

Une mesure de similarité sémantique a été définie en utilisant UMLS[33] et WordNet pour fournir un degré de synonymie entre deux entités provenant d'ontologies différentes, en explorant à la fois leurs contextes lexicaux et structurels. La traduction en plusieurs langues est basée sur le traducteur Microsoft. Ce système stocke localement tous les résultats de traduction de Microsoft Bing Translator[34] dans des fichiers de dictionnaire. Le traducteur ne sera interrogé que si aucune traduction enregistrée n'est trouvée afin de gagner du temps et d'éviter de surcharger le serveur.

1.8.6 WikiV3

WikiV3[35] est un système qui exploite des bases de connaissances externes, dans ce cas Wikipédia. Il utilise l'API MediaWiki[36] et recherche les pages qui correspondent à une ressource donnée. Lors de l'exploration des liens inter langues de Wikipedia[37], le système est également capable de trouver les correspondances entre des ontologies de langues différentes. Ces liens pointent d'une page Wikipedia à une page correspondante dans Wikipedia avec une langue différente.

Wikidata[2, p. 574] est un projet distinct qui permet de construire une base de connaissances éditée en collaboration. Une partie de ce projet consiste à centraliser les liens inter langues. Ainsi, le texte de Wikipédia est utilisé pour mieux correspondre aux entités de Wikidata que de simplement utiliser le texte disponible dans Wikidata.

Pour chaque ressource de la première ontologie, une liste de concepts Wikidata correspondants est générée. Une ressource peut être une classe, une propriété de type de données ou une propriété d'objet. Ils sont tous gérés séparément pour garantir qu'aucune correspondance entre différents types de ressources n'est générée. De la même manière, une liste d'identifiants Wikidata (WID) est créée pour la deuxième ontologie. S'il y a au moins un WID d'une liste dans l'ontologie 2 apparaissant dans une liste de WID dans l'ontologie 1, alors un mappage est créé.

1.8.7 POMap

Une ontologie peut modéliser un domaine particulier ainsi que les relations entre ses entités afin d'assurer sa réutilisation. Plusieurs ontologies décrivant le domaine similaire sont générées et utilisées par différentes parties et terminologies. Malgré la standardisation de la représentation du langage ontologique, le problème d'hétérogénéité apparaît. Par conséquent, il est important de surmonter cette hétérogénéité pour assurer la réutilisabilité de diverses ontologies. En effet, de nombreux chercheurs ont proposé et développé de nombreux systèmes d'alignement des ontologies automatique L'alignement d'ontologie est le processus consistant à trouver un ensemble de correspondances sémantiques entre les entités de deux ou plusieurs ontologies représentant un domaine similaire. Par conséquent, ces systèmes utilisent une variété de stratégies reposant sur la combinaison de plusieurs techniques telles que : syntaxique, sémantique et structurelle. POMap[38] poursuit une composition séquentielle au cours de ces trois techniques d'appariement. POMap explore toutes ces trois techniques afin d'assurer une correspondance de haute qualité. POMap emploie un aligneur sémantique. Ensuite, il utilise un comparateur syntaxique, qui suit une stratégie tout contre tout. Après, l'aligneur structurel prend en entrée les alignements générés à partir d'aligneur sémantique et d'aligneur syntaxique afin de trouver de nouveaux alignements. La composition séquentielle adoptée vise à tailler l'espace de recherche utilisé par l'aligneur structurel. Cet aligneur structurel est composé de deux sous-matcheurs structurels : les frères et sœurs et les sous-classes.

1.8.8 SANOM

SANOM[39] est un système d'alignement d'ontologies basé sur l'énergie qui essaie de trouver l'alignement le plus possible par la minimisation d'une fonction d'énergie prédéfinie.

Pour définir la fonction d'énergie pour un alignement donné, nous devons traiter chaque correspondance dans l'alignement. Pour ce faire, trois mesures de similarité différentes sont prises en compte. Pour chaque correspondance dans l'alignement, la somme négative de toutes les mesures de similarité est considérée comme l'énergie ; par conséquent, l'alignement avec l'énergie minimale implique des concepts plus similaires. Dans ce qui suit, les mesures de similarité potentielles sont passées en revue avec la fonction d'énergie.

1.8.9 ONTMAT

ONTology MATching[40] (ONTMAT) est un outil d'alignement d'ontologies, visant à aligner des entités OWL, participant pour la première fois à OAEI (piste de conférence).

ONTMAT utilise une méthode terminologique basée sur le dictionnaire WordNet, qui est exploitée comme une connaissance de base pour fournir un ensemble de relations entre les noms individuels des ontologies sources et cible. Ensuite, si le nom n'existe pas dans WordNet, l'approche gère la mesure n-gramme au lieu du dictionnaire. De plus, à partir de cet ensemble de relations individuelles, on peut déduire la relation d'équivalence ou de subsomption entre leurs concepts. Les concepts équivalents sont enregistrés dans une matrice d'alignement (AM), et les concepts liés par des relations de subsomptions sont enregistrés dans une matrice d'alignement temporaire (TAM).

De plus, les éléments TAM et les concepts voisins d'AM sont comparés ; en utilisant les rôles d'inférence avec les techniques terminologiques citées précédemment et l'alignement retenu sera ajouté à AM. Les concepts voisins sont ceux liés par des relations hiérarchiques ou binaires avec des concepts AM. Ici, on aligne d'abord les concepts voisins parce qu'ils ont plus de chance d'être similaires, et après on alignera les autres concepts en utilisant les mêmes techniques. Ensuite, les techniques d'inférence sont appliquées sur AM pour aligner les relations binaires.

1.8.10 KEPLER

Le système KEPLER[41] exploite, outre les techniques classiques, une ressource externe, c'est-à-dire un traducteur pour le traitement du multilinguisme. Cette méthode met en œuvre une stratégie d'alignement qui vise à exploiter toute la richesse des ontologies utilisées.

L'idée principale de la méthode KEPLER est d'exploiter l'expressivité du langage OWL pour déduire la similarité entre les entités de deux ontologies données. Les entités sont décrites en utilisant des primitives OWL avec leurs sémantiques. Nous pouvons alors considérer l'ontologie comme un graphe sémantique où les entités sont des nœuds connectés par des liens qui sont des primitives OWL. Ces liens ont des primitives sémantiques spécifiées. En effet, si deux ontologies d'un même domaine sont similaires, leurs graphes sémantiques sont également les mêmes.

1.8.11 LYAM++

LYAM++ (Yet Another Matcher - Light)[42] est un système d'appariement d'ontologies entièrement automatique basé sur l'utilisation de sources externes. LYAM++ ne s'appuie pas sur la traduction automatique pour l'appariement d'ontologies inter langues. Au lieu de cela, il utilise le réseau sémantique multilingue ouvert à usage général BabelNet afin de recréer le contexte sémantique manquant dans le processus d'appariement.

Le processus de LYAM++ se compose de quatre composants principaux : un aligneur terminologique, un module de sélection de cartographie et, enfin, un aligneur structurel.

1.8.12 YAM-BIO

YAM-BIO[43] peut être vu comme une extension de YAM++[44] avec l'utilisation de mappages existants comme connaissances de base pour améliorer l'appariement des ontologies biomédicales.

Le flux de travail de YAM-BIO contient trois étapes principales. Le premier consiste à découvrir la correspondance directe entre les ontologies source et cible en utilisant YAM++. La deuxième étape tente de trouver des correspondances pour les concepts sources qui n'ont pas été mis en correspondance en composant des mappages existants. La troisième étape consiste à traiter l'union des alignements produits par les étapes précédentes.

1.8.13 CroMatcher

CroMatcher[45] est un système d'appariement d'ontologies dans lequel le processus d'appariement est effectué automatiquement. Il prend en charge la correspondance entre les ontologies exprimées par OWL ; qui est recommandée par le W3C en tant que standard international pour la représentation des ontologies. Il y a plusieurs aligneurs de base de chaîne et de structure dans le système de CroMatcher. Chaque aligneur de base détermine la similarité entre les entités en utilisant des informations obtenues à partir d'un ou plusieurs composants des ontologies comparées, par conséquent les résultats de correspondance obtenus par tous les coupleurs de base doivent être agrégés afin d'obtenir les meilleurs résultats d'appariement final. Les coupleurs basiques de chaînes, ainsi que les coupleurs basiques structurels, sont liés par une composition parallèle de coupleurs basiques. Tout d'abord, les coupleurs de base de chaîne sont exécutés. Les résultats obtenus par les coupleurs basiques sont automatiquement agrégés en utilisant l'agrégation pondérée. Ces résultats agrégés sont ensuite utilisés dans l'exécution des coupleurs structurels comme valeurs initiales de correspondances entre entités. De nouveau, les résultats obtenus par les coupleurs basiques structurels sont agrégés en utilisant l'agrégation pondérée. Avant l'alignement final, les résultats agrégés des coupleurs de chaînes et les résultats agrégés des coupleurs structurels sont agrégés en utilisant l'agrégation pondérée. Finalement, la méthode finale d'alignement final est exécutée afin de sélectionner les correspondances appropriées entre les entités des ontologies comparées à partir des résultats d'appariement agrégés.

1.8.14 Lily

Lily[46] est un système d'appariement d'ontologies capable de résoudre certains problèmes liés aux ontologies hétérogènes. Il peut traiter les ontologies normales, les ontologies informatives faibles, le débogage d'alignement d'ontologie et la mise en correspondance d'ontologies, à la fois à grande échelle et à échelle normale.

Le principe de base des stratégies de correspondance de Lily est d'utiliser les informations utiles correctement et efficacement. Lily combine plusieurs techniques d'appariement efficaces pour faciliter les alignements. Il existe quatre stratégies de correspondance principales : (1) Le GOM (Generic Ontology Matching) est utilisé pour les tâches de correspondance courantes avec les ontologies de taille normale. (2) La correspondance d'ontologie à grande échelle (LOM) est utilisée pour les tâches de correspondance avec les ontologies de grande taille. (3) Le débogage

de mappage d'ontologie est utilisé pour vérifier et améliorer les résultats d'alignement. (4) L'accord d'ontologie est utilisé pour améliorer la performance globale.

Le processus d'appariement comporte principalement trois étapes : (1) le prétraitement, lorsque Lily analyse les ontologies et prépare les informations nécessaires pour les étapes ultérieures. Pendant ce temps, les ontologies seront généralement analysées, dont les caractéristiques, ainsi que les ensembles de données étudiés, seront utilisées pour déterminer les paramètres et les stratégies. (2) Calcul de similarité, quand Lily utilise des méthodes spéciales pour calculer les similitudes entre les éléments d'ontologies différentes. (3) Post-traitement, lorsque les alignements sont extraits et affinés par le mappage du débogage.

1.8.15 CroLOM

Cross-Lingual Ontology Matching[47], [48] (CroLOM) System est fondé sur des approches basées sur des ressources externes (c'est-à-dire la traduction) pour aligner des ontologies inter linguistique. CroLOM suit les phases suivantes :

1. **Extraction et normalisation :** CroLOM extrait d'abord les entités des ontologies d'entrée. Ensuite, il utilise la PNL pour normaliser les entités décrites dans différentes langues naturelles.

2. **Traduction :** Une fois les entités normalisées, CroLOM utilise le traducteur Yandex pour traduire en anglais les entités décrites dans différentes langues naturelles en tant que langage pivot. Après la traduction, CroLOM utilise pour la deuxième fois l'étape de normalisation, afin d'éliminer les mots d'arrêt de la langue anglaise des étiquettes d'entités.

3. **Calcul de similarité :** Une fois la traduction et la standardisation effectuées, CroLOM applique d'abord une conversion de cas en convertissant toutes les entités en minuscules puis passe à l'étape de calcul de similarité.

4. **Identification de l'alignement :** Enfin, CroLOM applique un filtre pour sélectionner les correspondances candidates qui possèdent la valeur de similarité maximale dans chaque ligne de produit cartésien entre entités. Ensuite, il applique une seconde un filtre pour identifier les correspondances qui possèdent une valeur de similarité supérieure à un seuil donné.

1.8.16 DKP-AOM

Le système DKP-AOM[49] suit une méthodologie en cinq étapes, il génère les modèles intermédiaires (OWL-DL Graphs) des ontologies sources et effectue un prétraitement sur les URI et les labels de concept. En second lieu, en utilisant ces graphes, le composant MatchManager effectue la tâche de premier niveau consistant à trouver les correspondances linguistiques, synonymiques et axiomatiques initiales entre les concepts. Pour cela, il construit d'abord l'espace de recherche basé sur des axiomes disjoints à l'intérieur des ontologies sources pour trouver les correspondances entre les ontologies.

Troisièmement, ConsistencyChecker possède de nombreux détecteurs qui effectuent la validation de chaque mappage trouvé dans l'étape initiale, de sorte que l'ontologie fusionnée reste cohérente avec la référence aux ontologies sources. Lorsque les mappages initiaux passent le test de cohérence, ConsistencyChecker transmet les alignements au raisonneur.

Quatrièmement, le raisonneur agrège la sortie de différentes mesures de similarité, résout les conflits et fusionne les mappages pour générer une ontologie fusionnée globale. Ce raisonneur est un composant du système DKP-AOM. Il met en œuvre divers modèles pour la fusion automatique d'ontologies sources en cas de différents types de conflits et de différences structurelles.

Enfin, il compile la sortie sous forme d'une ontologie globale fusionnée automatiquement ou d'une liste finale de correspondances cohérentes selon les besoins de l'utilisateur final. L'algorithme de fusion importe la première ontologie en tant qu'ontologie fusionnée et effectue ensuite plusieurs opérations pour construire les définitions combinées pour chacun des concepts à partir des ontologies sources. Chacune des définitions axiomatiques des ontologies sources est mise en correspondance, leur fusion est effectuée et les axiomes riches combinés sont ajoutés dans l'ontologie fusionnée. L'algorithme de fusion effectue la suppression des axiomes ou la réécriture de certains d'entre eux afin de préserver les conséquences souhaitées tout en supprimant ceux qui ne sont pas souhaités. La fusion de définitions axiomatiques aboutit à une ontologie fusionnée plus riche qui capture des définitions suffisantes à partir des ontologies sources. Enfin, il applique les critères de qualité et assure l'objectif ultime d'atteindre la satisfiabilité de l'ontologie fusionnée en vérifiant l'exactitude et la cohérence des concepts, des propriétés et des axiomes de l'ontologie générée.

1.8.17 DiSMatch

DisMatch[50] est un système expérimental conçu dans le but d'évaluer l'applicabilité d'un corpus axé sur le domaine d'état de l'art basé sur la mesure des relations sémantiques, à une tâche d'alignement d'une ontologie.

Pour une paire d'ontologies, DisMatch calcule la matrice de relation sémantique entre les étiquettes représentant leurs concepts. Il utilise ensuite cette matrice comme entrée pour l'algorithme classique de Similarité Flooding, afin d'intégrer les informations taxonomiques dans les résultats finaux.

Le flux de travail de DisMatch peut être décomposé selon les étapes suivantes :

- Prétraitement : extraction des taxonomies et des libellés des concepts.
- Affectation de représentations distributionnelles aux concepts des ontologies
- Calcul de la parenté sémantique pour les paires de concepts des ontologies respectives
- Calcul de la propagation de similarité compte tenu des taxonomies et des scores de parenté initiaux (SimFlood)
- Calcul des scores de similarité finaux
- Filtrage

1.8.18 FCA-Map

Étant donné deux ontologies, FCA-Map[51] construit des contextes formels et utilise les réseaux conceptuels dérivés pour regrouper les points communs entre les classes d'ontologie, respectivement au niveau lexical et structurel. Concrètement, FCA-Map procède étape par étape comme suit :

- Acquérir des ancrages lexicalement. Le contexte formel à base de jetons est construit, et à partir de son réseau conceptuel dérivé, un groupe d'ancrages lexicaux A à travers les ontologies peut être extraites.
- Validation des ancrages structurellement. Basé sur A, le contexte formel relationnel est construit, et à partir de son réseau conceptuel dérivé, des preuves structurelles positives et négatives d'ancrages peuvent être extraites. De plus, un alignement amélioré A' sans les incohérences entre les ancres sont obtenues.

- Découvrir des correspondances supplémentaires basées sur *A »*, le contexte formel basé sur la relation positive est construit, et à partir de son réseau conceptuel dérivé, des correspondances supplémentaires à travers les ontologies peuvent être identifiées.

1.8.19 LPHOM

LPHOM (Linear Program for Holistic Ontology Matching)[52] est un système holistique de mise en correspondance d'ontologies. Bien que le système ait été conçu pour faire face à l'appariement d'ontologies holistiques (c'est-à-dire en faisant correspondre plusieurs ontologies simultanément), il est également capable de traiter l'appariement d'ontologies par paires.

LPHOM traite le problème de l'appariement d'ontologies comme un problème d'optimisation combinatoire. Le problème est modélisé par un programme linéaire étendant le problème de correspondance de graphe à pondération maximale avec des contraintes linéaires (contraintes de cardinalité, de structure et de cohérence).

LPHOM est composé de quatre étapes principales :

- La première étape consiste à charger, aplatir et traduire l'ontologie.
- La deuxième étape consiste en la construction de matrices de similarité.
- La troisième étape consiste à construire le programme linéaire.
- La quatrième étape consiste à résoudre le programme linéaire à l'aide du solveur CPLEX[53].

1.8.20 PhenoMM, PhenoMF et PhenoMP variantes de PhenomeNET

PhenomeNET[54] est un système de priorisation des gènes de maladies qui inclut comme une de ses composantes une ontologie conçue pour intégrer les ontologies phénotypiques. Bien qu'il ne s'applique pas aux ontologies arbitraires correspondantes, PhenomeNET peut être utilisé pour identifier les phénotypes apparentés chez différentes espèces, y compris l'humain, la souris, le poisson-zèbre, le ver nématode, la mouche des fruits et la levure.

Le système d'alignement PhenomeNET se décline en trois versions, qui s'appuient sur trois versions différentes de l'ontologie PhenomeNET. PhenomeNET-Plain (PhenoMP) repose sur une ontologie simple qui n'utilise que les axiomes fournis par l'ontologie HP (Human Phenotype Ontology) et l'ontologie MP (Mammalian Phenotype Ontology). PhenomeNET-Map (PhenoMM) utilise des axiomes d'équivalence lexicale supplémentaires entre HP et MP

fournis par BioPortal. Enfin, PhenomeNET-Full (PhenoMF) s'appuie sur une version étendue de l'ontologie PhenomeNET avec des mappages d'équivalence aux ontologies DOID et ORDO[55] obtenus via BioPortal et le système d'appariement AML.

1.9 Conclusion

Cette partie a introduit la notion de web sémantique, de l'alignement d'ontologie, les caractéristiques externes des méthodes d'alignement. Enfin, elle a mentionné certains outils et les systèmes qui ont été développés pour résoudre les problèmes d'alignement des ontologies.

En résumé, l'objectif d'une ontologie est de parvenir à une connaissance commune et partagée qui peut être transmise entre les personnes et entre les systèmes d'application. Par conséquent, les ontologies jouent un rôle clé dans la réalisation de l'interopérabilité entre les organisations, parce qu'elles visent à capturer la connaissance du domaine et leur rôle est de créer de manière explicite la sémantique d'une manière générique, fournissant la base d'accord au sein d'un domaine. Par conséquent, les ontologies sont devenues un sujet de recherche populaire dans de nombreuses collectivités. En fait, l'ontologie est un composant principal de notre recherche ; c'est pourquoi la définition de la structure et des principales opérations et application de l'ontologie sont fournies.

L'objectif d'un système d'intégration de l'information est de fournir une vue uniforme de l'ensemble des sources d'information sur la même portée, mais créée indépendamment les uns des autres, qui peuvent être différenciés par formats, structures, modes d'accès ou les termes représentés. En effet, l'alignement d'ontologies joue un rôle important dans la résolution de l'interopérabilité dans les systèmes hétérogènes et dans de nombreux domaines d'application.

CHAPITRE 2 : CONTRIBUTIONS ELABOREES POUR L'ALIGNEMENT DES ONTOLOGIES

2.1 Introduction

L'alignement d'ontologies s'est vu devenir nécessaire pour la réduction de l'hétérogénéité. Les méthodes d'alignement opèrent généralement sur deux phases. La première phase assure la transformation des ontologies à aligner sous la forme d'une représentation permettant l'exploitation facile des informations contenues dans les ontologies. La deuxième phase est le processus de l'alignement proprement dit. Ce processus prend en charge les entités appartenant aux ontologies et cherche les correspondances qui peuvent exister entre les entités. Les méthodes proposées suivent cette démarche et utilisent leur propre mécanisme d'alignement, en exploitant des mesures de similarité et une démarche d'exploration des linguistiques ou des structures internes des ontologies à aligner.

Cette partie décrit en détail les principales composantes de notre système, ainsi que l'interaction entre eux. Au début, elle Il présente les stratégies multiples utilisées ; ensuite, elle montre comment le système peut regrouper des résultats différents qui sont produits par des méthodes différentes.

2.2 Description de la méthode OMBWSD

Le but de notre travail est de fournir un modèle d'intégration, flexible et puissant à la fois, capable d'unifier les différentes sources de données hétérogènes. Pour cela, la définition de notre architecture doit inclure un nombre minimal d'étapes à gérer.

Comme nous pouvons le voir sur la figure 4, l'approche se compose de trois étapes distinctes, à savoir : l'extraction, le prétraitement et l'alignement.

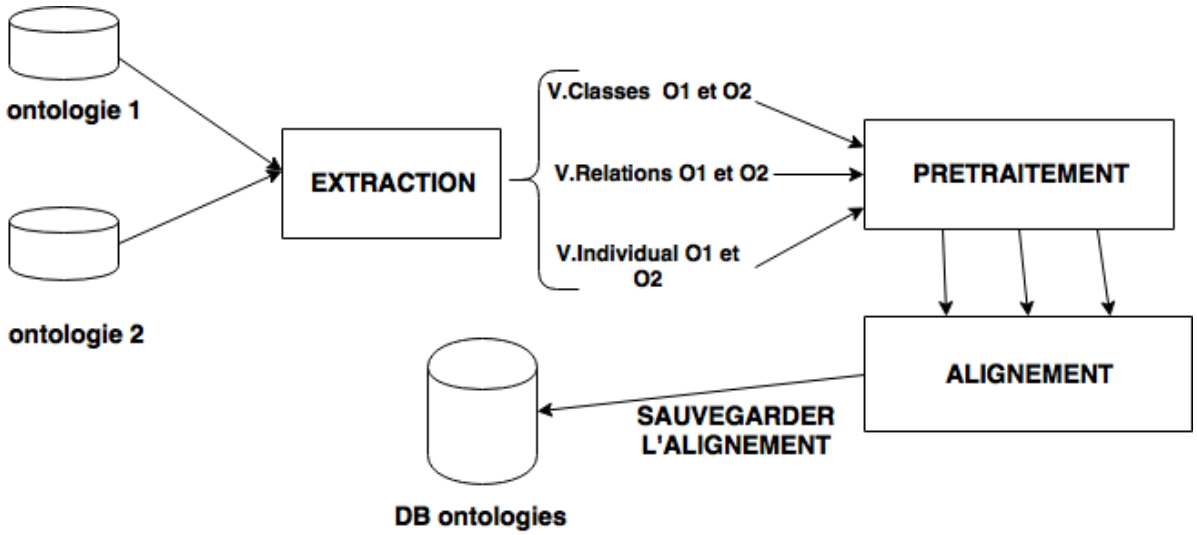

Figure 4 : Étapes d'alignement de deux ontologies O_1 et O_2

2.2.1 Etape d'extraction

L'exécution commence par l'importation d'ontologies. Ainsi, cela débute par les deux ontologies à aligner, à partir desquelles le nom des classes et des relations est obtenu et séparé en vecteurs, comme le montre la Figure 4.

Les caractéristiques des entités ontologiques (concepts C, relations R et instances) sont extraites des définitions d'ontologie extensionnelle et intentionnelle afin de comparer des entités provenant de deux ontologies différentes. Chacune de ces caractéristiques doit être utilisée pour calculer la similarité. Par conséquent, les étiquettes (label, comment, etc.) représentent la caractéristique la plus couramment utilisée dans l'alignement des ontologies.

En général, les étiquettes des entités, dans un environnement tel que le Web, peuvent être une seule lettre par exemple « D » pour date, une combinaison de lexèmes par exemple FirstName, DepNumber, ce qui les rend syntaxiquement différents. En conséquence, le prétraitement est nécessaire pour améliorer la correspondance entre les ontologies ; avant le début du processus d'alignement.

2.2.2 Etape de prétraitement

Bien qu'il existe de nombreuses tâches de traitement automatique de la langue qui peuvent être effectuées, nous nous concentrerons uniquement sur un sous-ensemble de ces tâches.

2.2.2.1 Détection de phrases.

La détection de phrases ou la segmentation de phrases est un processus de recherche du début et de la fin d'une phrase dans un texte. Elle est assez difficile pour plusieurs raisons dont l'une

d'entre elles est la suivante : le symbole « . » qui désigne habituellement la fin d'une phrase, aussi il peut également apparaître dans les adresses e-mail, abréviations, décimales, etc.

Ces scénarios appellent à construire nos propres modèles ; à partir de nos propres données d'entraînement, pour notre propre usage.

Le système[56] élaboré permet de détecter les phrases d'un texte. Il comporte un ensemble des étapes comme la montre la figure suivante :

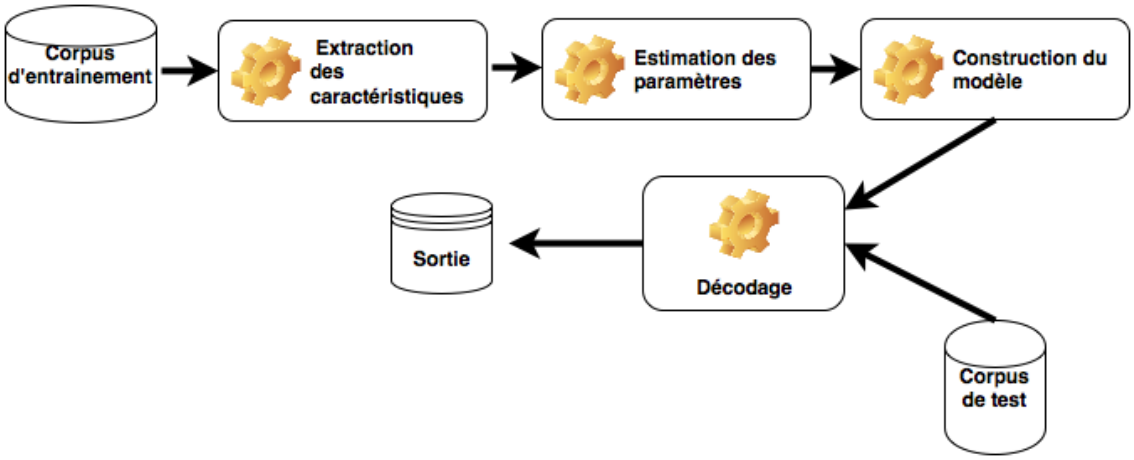

Figure 5 : Processus du système de détection de phrases

En premier lieu, la phase d'extraction des caractéristiques s'intéresse au dégagement des attributs à partir d'un corpus d'entraînement. Ensuite, les paramètres de Maximum de vraisemblance sont estimés dans la deuxième phase. Après, dans la troisième phase un algorithme d'entropie maximale est adopté pour l'élaboration du modèle. Finalement, la phase de décodage permet de détecter les phrases dans un corpus de test en se basant sur le modèle construit.

Le fonctionnement de ce système est fondé sur le travail de Reynar et Ratnaparkh [57] qui se base sur un ensemble de règles :

- L'utilisation des paramètres simples comme la lettre finale d'un mot ou sa longueur dont le coût de calcul est minime.
- L'adoption des règles de désambiguïsation pour les signes de ponctuation.

Ces règles sont extraites à partir des informations tirées des statistiques sur des corpus non annotés et sont applicables pour les textes écrits en anglais, français ou arabe.

2.2.2.2 Tokenisation

Pour déterminer les limites d'une phrase, il est nécessaire de la séparer et de déterminer quels éléments marquant la fin de la phrase.

Les éléments du texte qui déterminent où les éléments doivent être séparés sont appelés délimiteurs. Pour la plupart des textes anglais, français et arabe, les espaces blancs sont utilisés comme délimiteur. Ce type de délimiteur comprend généralement des espaces, des tabulations, des tirets, etc.

Puisque le nommage d'entités dans les ontologies joue un rôle essentiel dans le processus d'appariement, les noms des entités telles que les classes et les propriétés dans les ontologies OWL sont analysées en premier. Nous avons remarqué que les développeurs utilisent des styles différents pour nommer des mots composés, ce qui fait que les correspondants ont des difficultés à trouver les relations entre les noms composés. Par conséquent, dans la phase de tokenisation, les mots composés sont analysés plus que de simples mots.

Dans les ontologies d'évaluation, nous avons opéré qu'il existe deux types de tokenisation : le trait de soulignement (par exemple, In_book) dans l'ontologie 204 et le changement d'une lettre minuscule en majuscule (par exemple, BookPart) dans l'ontologie 304.

2.2.2.3 Détection de la partie grammaticale.

La détection de la partie grammaticale, ou POS-tagging[58], assigne des catégories de mots grammaticaux à des mots individuels ou à d'autres symboles tels que des nombres. Un ensemble de tags définit comment certaines entités sont représentées. Par exemple, dans la base de données Penn Treebank pour la langue anglaise, un nom commun singulier est marqué NN, tandis qu'un nom commun pluriel est marqué NNS. Les tagsets incluent souvent des catégories morphosyntaxiques telles que le genre, le nombre et le cas.

1. CC Coordinating conjunction	25.TO to
2. CD Cardinal number	26.UH Interjection
3. DT Determiner	27.VB Verb, base form
4. EX Existential there	28.VBD Verb, past tense
5. FW Foreign word	29.VBG Verb, gerund/present participle
6. IN Preposition/subord.	30.VBN Verb, past participle
7. JJ Adjective	31.VBP Verb, non-3rd ps. sing. present
8. JJR Adjective, comparative	32.VBZ Verb, 3rd ps. sing. present
9. JJS Adjective, superlative	33.WDT wh-determiner
10.LS List item marker	34.WP wh-pronoun
11.MD Modal	35.WP Possessive wh-pronoun
12.NN Noun, singular or mass	36.WRB wh-adverb
13.NNS Noun, plural	37. # Pound sign
14.NNP Proper noun, singular	38. $ Dollar sign
15.NNPS Proper noun, plural	39. . Sentence-final punctuation
16.PDT Predeterminer	40. , Comma
17.POS Possessive ending	41. : Colon, semi-colon
18.PRP Personal pronoun	42. (Left bracket character
19.PP Possessive pronoun	43.) Right bracket character
20.RB Adverb	44. " Straight double quote
21.RBR Adverb, comparative	45. ` Left open single quote
22.RBS Adverb, superlative	46. " Left open double quote
23.RP Particle	47. ' Right close single quote
24.SYM Symbol	48. " Right close double quote

Tableau 1 : Jeu de tags Penn Treebank POS

De nombreux étiqueteurs sont disponibles et peuvent souvent être configurés pour utiliser des modèles de langage personnalisés. De tels modèles de langage sont spécifiques au langage

étiqueté et sont généralement générés à partir d'un arbre - un corpus avec des annotations produites selon une précision suffisante pour être utilisée comme une base d'entrainement pour construire un modèle de langue. Des modèles de marquage de texte anglais, français et arabe sont intégrés dans le système proposé.

2.2.2.4 Élimination des Stop-word

À la suite de la tokenisation et la détection de la partie grammaticale, il est nécessaire de réduire la dimensionnalité des données résultantes. Ainsi, les filtres sont appliqués aux données. Le filtre de mots d'arrêt (Stop-word[58]) est un filtrage standard qui a été appliqué en tant qu'une étape primordiale de prétraitement.

Les mots d'arrêt sont les mots utiles pour former des phrases, mais ne contient pas des informations utiles. Néanmoins, de tels mots provoquent une fraction importante du texte dans les documents, de sorte que, lors du prétraitement, des filtres de mots vides sont utilisés pour éliminer ces mots du contenu. A cet effet, une liste des mots est construite, cette liste peut être fournie par l'utilisateur ou le système peut la construire automatiquement. Pour la langue anglaise la liste des mots vides[59] contient des articles (« the », « a » et « an »), des conjonctions (« and », « or », « but », et « yet »), des prépositions (« by », « from », « about », « below », « in » et « on »), etc. Cette liste contient aussi des mots non significatifs très fréquents qui surviennent très souvent, aussi elle comporte des mots qui n'ont pas d'importance statistique significative. La liste des mots d'arrêt est de nature contextuelle ou dépend du domaine, donc selon les exigences de l'application, cette liste peut être personnalisée.

2.2.2.5 Lemmatisation et radicalisation

Des mots arabes comme'يراقب ' peuvent être fléchis avec un suffixe ou préfixe morphologique pour produire'يراقبونهم , مراقبة. Ils partagent la même tige « راقب ». Souvent, mais pas toujours, il est utile de cartographier toutes les formes fléchies dans la tige. C'est un processus complexe, car il peut y avoir de nombreux cas exceptionnels (par exemple, département # départ, être # étaient dans la langue française). Le stemmer le plus couramment utilisé dans la langue anglaise est le Porter Stemmer et khoja pour la langue arabe.

La lemmatisation et la radicalisation[60] tentent toutes les deux de normaliser les mots de leurs formes inférées à une forme commune. La racine supprime les préfixes et les suffixes communs, laissant derrière la « racine ». La lemmatisation modifie les mots à leur lemme, ou la forme non infléchie.

La lemmatisation est préférée dans l'apprentissage de l'ontologie, car il est utile de distinguer différents sens d'un mot tout en fusionnant différentes inflexions du même sens. Elle est généralement basée sur des règles pouvant engendrer des parties importantes de mots qui ressemblent à des préfixes ou suffixes, mais qui font partie du lemme, voir exemple suivant.

يراقبونهم	ي + راقب + ون + هم	راقب
Ils les observent	il + (s) +les+ observe+(ent)	Observer

La lemmatisation a besoin d'un dictionnaire complet qui a tous les mots de base avec des formes fléchies ou des règles pour produire des formes fléchies alors que les stemmers n'appellent pas ce lexique.

2.2.3 Etape d'alignement

Pour chaque type de vecteur (exemple de classe de vecteur de O_1 et O_2), on calcule la similarité sémantique entre les deux éléments du vecteur en utilisant les algorithmes expliqués précédemment et BabelNet[61].

La méthode proposée utilise différents types d'information d'ontologie : les étiquettes, commentaire des concepts, les propriétés, et les instances. Avant d'introduire chaque stratégie spécifique, nous donnons d'abord la définition de la matrice de similarité.

Définition : Matrice de similarité

Pour trouver la relation de correspondance entre l'ontologie O_1 et O_2, nous utilisons une stratégie pour calculer leur similarité d'entités. La valeur de chaque élément de la matrice est entre 0 et 1. La représentation de la matrice de similarité M est définie par l'équation 13 :

$$M_{n,m} = \begin{pmatrix} S(e_{11}, e_{2,1}) & S(e_{11}, e_{2,2}) & \cdots & \cdots\cdots & \cdots\cdots\cdots & S(e_{11}, e_{2,m}) \\ S(e_{12}, e_{2,1}) & S(e_{11}, e_{2,2}) & \cdots & \cdots\cdots & \cdots\cdots & S(e_{12}, e_{2,m}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & S(e_{1i}, e_{2,j}) & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S(e_{1n}, e_{2,1}) & S(e_{11}, e_{2,2}) & \cdots & \cdots & \cdots & S(e_{1n}, e_{2,m}) \end{pmatrix} \quad (13)$$

Sachant que :

- $1 \leq i \leq n \; et \; 1 \leq j \leq m$ sont les entités dans l'ontologie O_1, O_2.
- $S(e_{1i}, e_{2,j})$ est la similarité entre l'entité i de l'ontologie O_1 et l'entité j de l'ontologie O_2.

2.2.3.1 Calcul de similarité d'étiquette (label) ou de commentaire

L'utilisation de la similarité de l'étiquette d'entité pour trouver la relation de correspondance est la méthode la plus élémentaire dans l'alignement des ontologies. En effet, il existe plusieurs recherches réalisées dans ce domaine, comme [62],[63], et [64]. Ces méthodes dépendent principalement du message texte d'entité ; elles utilisent les approches syntaxiques pour calculer la similarité de deux étiquettes d'entités. Pour prendre en compte l'échec de l'information sémantique de l'étiquette d'entité ; le système proposé utilise une méthode de calcul de similarité basée sur l'étiquette d'entité dans BabelNet, en adoptant les similarités linguistiques et la similarité de jaro-winkler. Les informations sémantiques et grammaticales de ces étiquettes ont été prises en considération. Les expériences montrent que cela est non seulement efficace dans la condition lorsque les éléments ont le même nom, mais il est efficace quand les noms ne sont pas les mêmes, mais avec quelques relations sémantiques. BabelNet est un système de vocabulaire web sémantique multilingue, il contient 271 langues. Il organise le vocabulaire comme un thesaurus de synset « liste des synonymes » qui indique les concepts lexicaux et crée différents indicateurs entre les concepts pour exprimer la sémantique des relations comme l'hyponymie, la synonymie, l'antonyme[65], etc.

Les mesures linguistiques utilisent la composition syntaxique de la phrase ou de l'information sémantique contenue dans la phrase pour déterminer la similarité sémantique.

Définition : similarité sémantique

Pour deux phrases p_1 et p_2 ; nous calculons le score de similarité de mot maximum pour chaque mot de p_1 avec des mots dans la même partie de discours grammaticale dans p_2 et ensuite nous répétons le processus pour la phrase p_2. Ensuite, un score de similarité est déterminé selon Mihalcea par l'équation 14 :

$$Sim_{sem}(P_1, P_2) = \frac{1}{2}\left(\frac{\sum_{w \in P_1}(maxS(w, P_2) * idf(w))}{\sum_{w \in P_1} idf(w)} + \frac{\sum_{w \in P_2}(maxS(w, P_1) * idf(w))}{\sum_{w \in P_2} idf(w)}\right) \quad (14)$$

Sachant que

- P_1 et P_2 sont les deux phrases.
- W est chaque mot de la phrase.
- $maxS(w, P)$ est le maximum de similarité des mots pour chaque mot dans une phrase P.

- IDF(w) (fréquence de document inverse) est donnée par le nombre total de documents divisé par le nombre de documents contenant le terme « w ».

Par exemple, soient P_1 et P_2 deux phrases telles que :

- P_1: Eventually, a huge cyclone hit the entrance of my house.
- P_2: Finally, a massive hurricane attacked my home.

	cyclone /NN	**hit /VBD**	**entrance /NN**	**house /NN**
hurricane/NN	0.9565	-	0.2857	0.3158
attacked/VBD	-	0.8571	-	-
home/NN	0.3529	-	0.6667	1.0000

Tableau 2 : Matrice de similarité de Wu & Palmer entre deux phrases

$$Sim_{sem}(P_1, P_2) = \frac{1}{2}\left(\frac{[0{,}9565 * \log\left(\frac{2}{1}\right) + 0{,}8571 * \log\left(\frac{2}{1}\right) + 1 * \log\left(\frac{2}{1}\right)]}{3 * \log\left(\frac{2}{1}\right)} + \right.$$

$$\left.\frac{[0{,}9565 * \log\left(\frac{2}{1}\right) + 0{,}8571 * \log\left(\frac{2}{1}\right) + 0{,}6667 * \log\left(\frac{2}{1}\right) + 1 * \log\left(\frac{2}{1}\right)]}{4 * \log\left(\frac{2}{1}\right)}\right)$$

$$= \frac{(0{,}93 + 0{,}865)}{2} = 0{,}89$$

D'après ce résultat, on trouve que les deux phrases P_1 et P_2 sont similaires.

2.2.3.2 Calcul de similarité d'identifiant

Le calcul de la similarité d'identifiant des couples d'entités est effectué par l'intermédiaire de l'equation 15 (similarité d'identifiant).

$$Sim_{iden}(e_1, e_2) = \begin{cases} 1 & si \quad sim_{jacc}(e_1, e_2) = 1 \\ \max\left(sim_{jacc}(e_1, e_2), sim_{wup}(e_1, e_2)\right) \end{cases} \quad (15)$$

2.2.3.3 Calcul de similarité global

La valeur de la similarité globale entre deux entités e_1 et e_2 est déterminée comme suit :

$$sim_g(e_1, e_2) = \begin{cases} 1 & si \quad 0{,}9 \leq sim_{iden}(e_1, e_2) \leq 1 \\ max(Sim_c(e_1, e_2); Sim_e(e_1, e_2); Sim_{iden}(e_1, e_2)) \end{cases} \quad (16)$$

Sachant que :

- $sim_{iden}(e_1, e_2)$: est la similarité d'identifiant entre deux entités.
- $Sim_c(e_1, e_2)$: est la similarité de commentaire.
- $Sim_e(e_1, e_2)$: est la similarité d'étiquette.

2.2.3.4 Calcul de similarité de sub et super

La similarité de super entité est la somme de similarité globale entre un ensemble des entités parentes. Elle est définie par l'équation 17 :

$$sim_{super}(e_1, e_2) = \frac{\sum_i^{sup_1} \sum_j^{sup_2} sim_g(i, j)}{sup_1 \times sup_2} \quad (17)$$

Où :

- sup_1 et sup_2 sont les éléments de super entité de l'entité 1 et 2.

La similarité de sub entité est la somme de similarité globale entre un ensemble des entités fils. Elle est définie par l'équation 18 :

$$sim_{sub}(e_1, e_2) = \frac{\sum_i^{sub_1} \sum_j^{sub_2} sim_g(i, j)}{sub_1 \times sub_2} \quad (18)$$

Où:

- sub_1 et sub_2 sont les éléments de sub entité de l'entité 1 et 2.

2.2.3.5 Calcul de similarité de propriété

La similarité de propriétés est calculée par l'équation 19:

$$Sim_P(P_1, P_2) = \frac{sim_{domaine}(P_1, P_2) + sim_{super}(P_1, P_2) + sim_{sub}(P_1, P_2) + sim_g(P_1, P_2)}{4} \quad (19)$$

Sachant que :

- $sim_{super}(P_1, P_2)$ et $sim_{sub}(P_1, P_2)$ sont les similarités de sub et super de propriétés.

- $sim_{domaine}(P_1, P_2)$ est la similarité entre l'ensemble des instances de la propriété P_1 et la propriété P_2 ; elle est définie par l'équation.

$$sim_{domain}(P_1, P_2) = \frac{\sum_i^{d_1} \sum_j^{d_2} sim_g(i,j)}{d_1 \times d_2} \quad (20)$$

Avec :

- d_1 et d_2 sont les éléments de domaine de la propriété 1 et 2.

2.2.3.6 Calcul de similarité de classe

La valeur de la correspondance entre deux classes c_1 et c_2 est déterminée par la formule suivante :

$$Sim_c(c_1, c_2) = \frac{sim_g(c_1, c_2) + sim_{super}(c_1, c_2) + sim_{sub}(c_1, c_2)}{3} \quad (21)$$

2.2.3.7 Filtration d'alignement

Après les calculs de similarités entre les éléments de deux entités en utilisant un seuil 0.8. Ce seuil a été sélectionné après avoir traversé plusieurs expériences sur les bases OAEI. Par conséquent, cette étape donne le degré de correspondance entre deux éléments, l'un des premiers vecteurs et l'autre du second.

2.3 Description de la méthode DROM

La méthode proposée comprend deux phases, comme la montre la figure 6. La première phase, appelée phase d'apprentissage, s'intéresse à l'extraction des caractéristiques d'entraînement. La deuxième phase, nommée phase d'évaluation, vise à trouver les paires correspondantes entre deux ontologies.

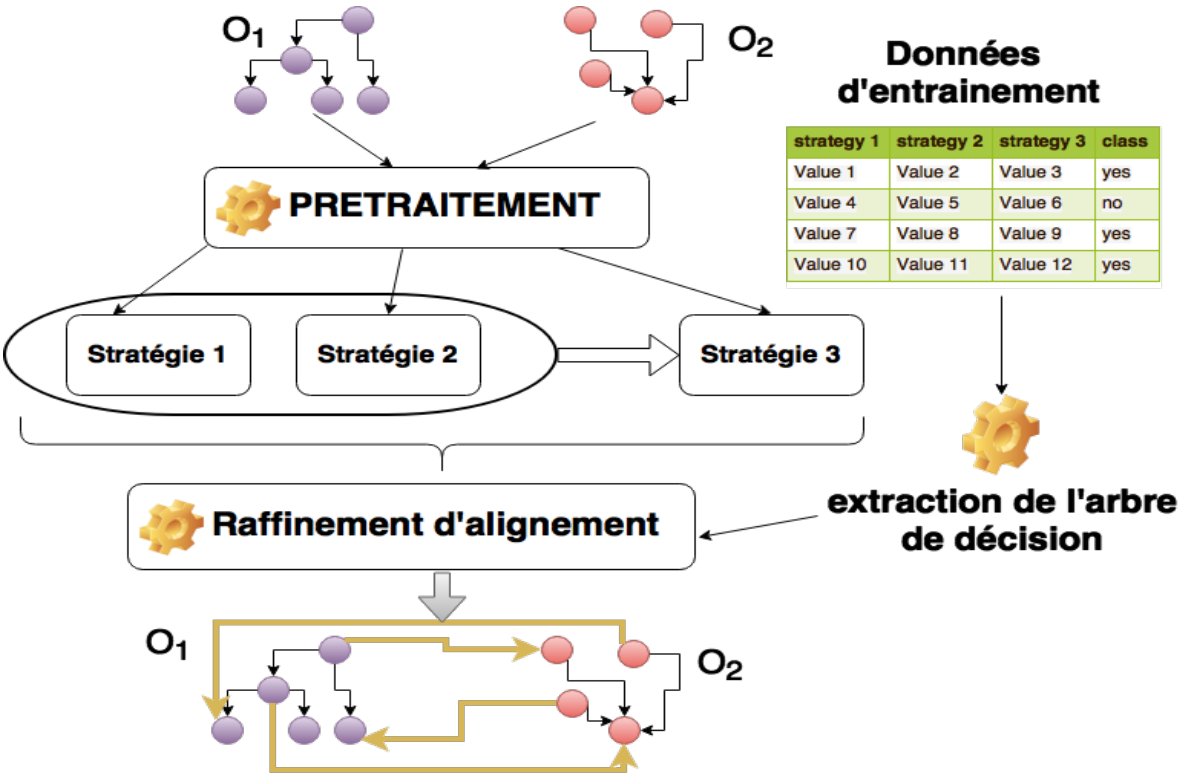

Figure 6 : Architecture de DROM

2.3.1 Phase d'entraînement

Après la classification des deux entités à aligner ou non à partir de la référence OAEI, toutes les paires d'alignements validées sont traitées par trois stratégies de calcul de la similarité. Chaque stratégie (expliquée ci-dessous) combine la fonction de similarité qui renvoie une valeur numérique enregistrée avec la paire de caractéristiques dans le test d'entraînement. Nous utilisons l'algorithme C4.5[66] comme classificateur qui distingue les entités qui s'alignent et celles qui sont disjointe

2.3.2 Phase d'évaluation

Cette phase comprend trois étapes essentielles : a) le prétraitement, b) la détermination des stratégies et c) le raffinement de l'alignement.

2.3.2.1 Prétraitement

Cette étape contient deux sous-étapes. La première consiste à extraire les classes avec ces annotations (nom, étiquettes, commentaires et instances) et les propriétés avec ces annotations (plage, domaine, nom, étiquettes et commentaires) pour chaque ontologie. La seconde

comprend des techniques de traitement du langage naturel telles que : tokenisation et suppression des mots d'arrêt.

2.3.2.2 Détermination des stratégies

Cette étape contient trois stratégies :

- Stratégie 1 consiste à calculer la similitude par la formule :

$$Sim_1(c1, c2) = max \begin{cases} jacc(c1_n, c2_n) \\ jacc(c1_l, c2_l) \\ jacc(c1_c, c2_c) \\ VSM(c1, c2) \\ WUP(c1_n, c2_n) \end{cases} \quad (22)$$

Sachant que :

- $\mathrm{jacc}(\mathrm{c1_n}, \mathrm{c2_n})$,$\mathrm{jacc}(\mathrm{c1_l}, \mathrm{c2_l})$ et $\mathrm{jacc}(\mathrm{c1_c}, \mathrm{c2_c})$ sont la similitude de Jaccard entre les noms, les étiquettes et les commentaires respectivement.
- $\mathrm{VSM}(\mathrm{c1}, \mathrm{c2})$ est la similitude basée sur l'instance entre deux classes.
- $\mathrm{WUP}(\mathrm{c1_n}, \mathrm{c2_n})$ est la mesure de similarité utilisant l'algorithme de Wu et Palmer entre deux étiquettes de deux classes.

- Stratégie 2 consiste à calculer la similitude par la formule 23 :

$$Sim_2(c1, c2) = max \begin{cases} Levein(c1_n, c2_n) \\ Levein(c1_l, c2_l) \\ Levein(c1_c, c2_c) \\ VSM(c1, c2) \\ AdLesk(c1_n, c2_n) \end{cases} \quad (23)$$

Tels que :

- $\mathrm{Levein}(\mathrm{c1_n}, \mathrm{c2_n})$, $\mathrm{Levein}(\mathrm{c1_l}, \mathrm{c2_l})$, $\mathrm{Levein}(\mathrm{c1_c}, \mathrm{c2_c})$ sont les similitudes de Levenshtein entre les noms, les étiquettes et les commentaires respectivement.
- $\mathrm{VSM}(\mathrm{c1}, \mathrm{c2})$ est la similarité basée sur l'instance entre deux classes.
- $\mathrm{AdLesk}(\mathrm{c1_n}, \mathrm{c2_n})$ est la mesure de similarité utilisant l'algorithme de Lesk adapté entre deux étiquettes de deux classes.

- Stratégie 3 utilise la similarité footing pour calculer la similarité entre deux classes en sachant que la matrice initiale des paires est calculée à partir des stratégies 1 et 2 par la formule 24 :

$$\sigma^0(c1, c2) = \max \begin{cases} Sim_1(c1, c2) \\ Sim_2(c1, c2) \end{cases} \quad (24)$$

2.3.2.3 Raffinement de l'alignement

Cette étape se focalise sur l'affinement de l'alignement généré à l'aide de règles de décision extraites de la phase d'apprentissage.

2.4 Description d'OM-NEURAL-NSGA-II

Pour résoudre le problème d'alignement de l'ontologie, nous construisons un réseau de neurones qui agrège les différentes similarités pour détecter la correspondance entre les entités. Comme l'apprentissage du réseau neuronal ne nous donne que l'optimum local, nous pouvons utiliser NSGA-II[67] pour calculer les poids de ce réseau. Commençant par générer un certain poids, en conséquence nous trouvons plusieurs adaptations des chromosomes, ce sont les poids générés. Deux poids sont tirés au hasard dans deux réseaux et croisés pour obtenir les nouveaux poids. Pour la mutation, un bruit est ajouté aléatoirement à un poids et en conséquence un nouveau réseau de neurones est obtenu. Enfin, les solutions sont à nouveau évaluées. La figure 7 schématise les étapes essentielles de notre approche, ces étapes sont décrites ci-dessous :

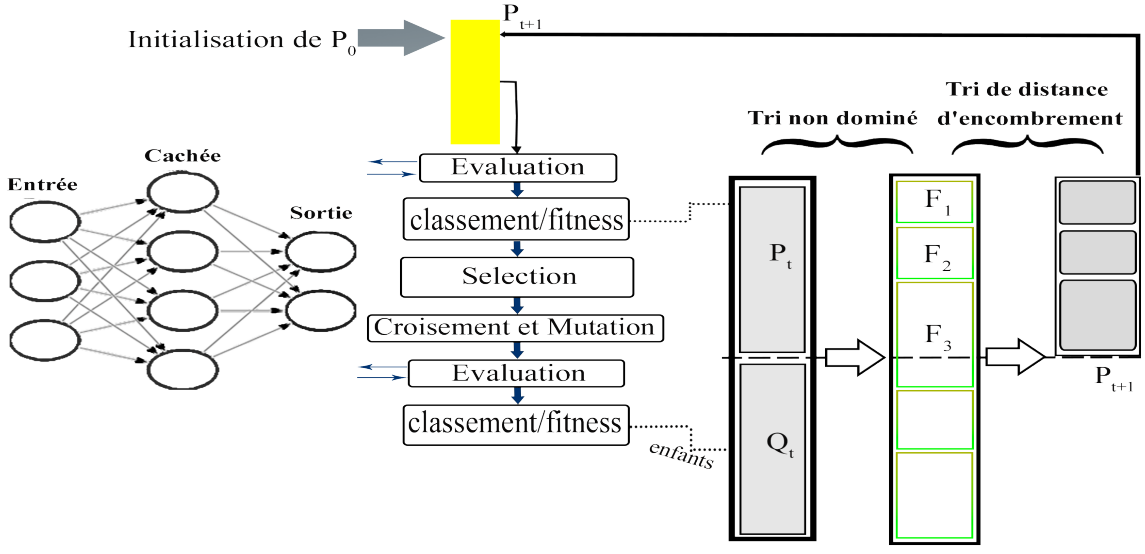

Figure 7 : Étapes de l'algorithme Neural NSGA-II

2.4.1 Encodage de chromosome

Tout d'abord, il est nécessaire de représenter les différents états possibles de la variable dont la valeur optimale est recherchée dans la forme utilisable. Où à chaque paramètre W_i (poids de nos réseaux neuronaux), il faut faire correspondre le gène. Un chromosome regroupe un ensemble de gènes ordonnés. Dans le cas d'un codage binaire, les gènes possibles sont 0 ou 1. Dans notre algorithme, les gènes ont des valeurs continues entre [0,1].

2.4.2 Evaluation

L'évaluation s'assure que les chromosomes performants seront préservés, alors que les chromosomes inadaptés seront progressivement éliminés de la population. Puisque l'évaluation d'un chromosome ne dépend pas de celle des autres chromosomes. Pour calculer le coût d'un point de l'espace de recherche, nous utilisons une fonction d'évaluation multi objective définie par l'équation (25) :

$$f(X,Y) = \{\text{Précision}(X), Rappel(X), -Erreur(Y)\}, \begin{matrix} avec\ Y = (y_1, \ldots . y_n)^T\ et\ y_i \epsilon [0,1] \\ avec\ X = (x_1, \ldots . x_n)^T\ et\ x_i \epsilon \{0,1\} \end{matrix} \qquad (25)$$

Sachant que :

- La fonction $Erreur(Y) = \frac{1}{2}\sum_{i=1}^{m}(\boldsymbol{y_i} - \boldsymbol{c_i})^2$ est la moitié du carré de la distance euclidienne entre la sortie $\boldsymbol{c_i}$ du réseau de neurones et la cible $\boldsymbol{y_i}$.
- Précision : mesure la capacité du système à refuser des solutions non pertinentes.
- Rappel : Mesure la capacité du système à fournir toutes les solutions pertinentes.

Le résultat serait de permettre de sélectionner un chromosome, ou de le refuser pour nous garder que les chromosomes qui ont le meilleur coût par la population actuelle.

2.4.3 Opérateurs génétiques

2.4.3.1 Sélection

Pour guider le processus de sélection, l'opérateur de comparaison d'encombrement $<_n$ est utilisé comme suit : chaque solution (i) de la population est identifiée par son rang (i_{rang}) et la distance d'encombrement ($i_{distance}$) calculée par le périmètre formé par les points (i) les plus proches de chaque objectif. Le calcul de la distance d'encombrement nécessite avant tout le tri des solutions par objectif dans l'ordre croissant. Ensuite, pour chaque objectif, les chromosomes

possédant les valeurs limites (la plus petite et la plus haute valeur de la fonction objective) sont associés à une distance infinie. Pour les autres solutions intermédiaires, la distance d'encombrement[67] est calculée par l'algorithme suivant :

Algorithme 1 : Calcul de la distance d'encombrement

```
l←|I| ;          // Nombre de solutions dans l'ensemble I
Pour chaque i : l
I[i]distance← 0 ; // initialiser les distances
Pour chaque objective m
        I←Trier(I,m); // trier en fonction de la valeur de l'objectif m
        I[1]distance←infini ;
        I[l]distance←infinie ;
        Pour i←2 jusqu'à (l-1) ;
                I[i]distance←I[i]distance + ( f_m^{i+1} - f_m^{i-1})/( f_m^{Max} - f_m^{Min}) ;
               Fin Pour ;
        FinPour ;
FinPour ;
```

Sachant que :

- $\boldsymbol{f_m^{i+1}}$ et $\boldsymbol{f_m^{i-1}}$ représentent respectivement la valeur de la fonction objective de la solution i+1 et i-1.
- $\boldsymbol{f_m^{Max}}$ et $\boldsymbol{f_m^{Min}}$ représentent les valeurs maximale et minimale de la fonction objective.

L'opérateur ($<_n$) défini ci-dessous dans l'équation (26) identifie un ordre de préférence entre deux solutions :

$$i <_n j \; si[\,(i_{rang} < j_{rang}) \; Ou \; ((i_{rang} = j_{rang}) \; et \; (i_{distance} > j_{distance}))] \tag{26}$$

Entre deux solutions de rangs différents, la solution est préférée avec le plus petit rang (ou le plus petit front). Pour deux solutions qui appartiennent au même front, on préfère la solution qui se situe dans la région où la densité des solutions est la plus faible (l'individu possédant la plus grande valeur de distance d'encombrement).

2.4.3.2 Croisement

Dans cette partie, nous présentons les différentes procédures de croisement [68] que nous allons tester pour implémenter Neural NSGA-II.

2.4.3.2.1 Croisement génétique Breeder (BGX)

À partir de deux parents P_1 et P_2, ce croisement génère un seul enfant en utilisant l'équation suivante :

$$enfant(i) = p_1(i) \pm \frac{(p_2(i) - p_1(i))}{\|p_2(i) - p_1(i)\|} \Delta_i \gamma \tag{27}$$

Sachant que :

- Δ_i est la moitié du domaine de définition du paramètre i.
- $\|...\|$ est la distance euclidienne.
- $\gamma = 2^{-\alpha\beta}$. Ou α et β sont des variables aléatoires réparties uniformément dans l'intervalle [0,1].

2.4.3.2.2 Croisement binaire simulé (SBX)

Le croisement binaire simulé reproduit les mécanismes du croisement standard à un point utilisé lorsque les variables d'objet sont représentées sous la forme de chaînes binaires. À partir de deux parents P_1 et P_2, ce croisement génère deux enfants $enfant_1$ et $enfant_2$, en utilisant l'équation suivante :

$$\begin{cases} \text{enfant}_1(i) = \frac{1}{2}[(1+\phi)]p_1(i) + (1-\phi)p_2(i) \\ \text{enfant}_2(i) = \frac{1}{2}[(1-\phi)]p_1(i) + (1+\phi)p_2(i) \end{cases} \tag{28}$$

Sachant que :

- ϕ est un facteur de dispersion défini par :

$$\varphi = \begin{cases} (2\beta)^{\frac{1}{\alpha+1}} \ si\ \beta < 0.5 \\ (\frac{1}{2(1-\beta)})^{\frac{1}{\alpha+1}} \ sinon \end{cases} \quad (29)$$

Où β est une variable aléatoire répartie uniformément dans l'intervalle [0,1] et $\alpha \in \mathbb{R}^+$ est un paramètre qui caractérise la forme de la distribution des enfants par rapport aux parents (pour trouver des enfants près de leurs parents, on augmente α).

2.4.3.2.3 Mutation

L'opérateur de mutation classique prend comme entrée un individu sélectionné pour la mutation et renvoie un individu mutant obtenu par transformation locale d'un gène du P. Dans le cas présent, un gène est codé par un sous-domaine des valeurs possibles de la variable correspondante ; Par analogie, la mutation d'un individu consiste à remplacer un de ses gènes / sous-domaines par un autre sous-domaine sélectionné aléatoirement. Dans notre approche, nous avons adopté l'opérateur Michalewicz[69] qui est défini par l'équation suivante.

$$x'_k = \begin{cases} x_k + \Delta(t, UB - x_k) \ , si\ rand(r) = 0 \\ x_k - \Delta(t, x_k - LB) \ , si\ rand(r) = 1 \end{cases} \quad (30)$$

Sachant que :

- ***LB*** et ***UB*** sont les valeurs maximales et minimales de la variablex_k.
- $\boldsymbol{\Delta(t, y)}$ est une fonction qui renvoie une valeur entre [0, y]. Elle est définie par l'équation (31) :

$$\Delta(t, y) = y(1 - r^{(1-\frac{t}{T})^b}) \quad (31)$$

Avec :

- ***r*** est une variable uniforme entre [0,1].
- ***T*** est le nombre de générations maximal.
- ***b*** est un paramètre système qui détermine le degré de dépendance entre le nombre d'itérations.

2.4.3.2.4 Production de la population de la prochaine génération

Une nouvelle population parente (P_{t+1}) est formée en additionnant les fronts entiers (premier front 1, deuxième front 2, etc.) s'ils ne dépassent pas N. Si le nombre d'individus présents dans (P_{t+1}) est inférieur à (N), une procédure d'encombrement est appliquée sur le front avant le suivant (F_i) non inclus dans (P_{t+1}). L'objectif de cet opérateur est d'insérer les meilleurs individus manquants (N- | P_{t+1} |) dans la population (P_{t+1}).

Répétition de toutes les étapes en utilisant des individus de la population (P_{t+1}) au lieu de la population (P_0), jusqu'à un critère d'arrêt.

2.5 Conclusion

Un nouveau système pour l'alignement d'ontologies a été construit par l'intégration de certaines caractéristiques importantes de l'alignement afin d'atteindre des résultats de haute qualité lors de la recherche et échange d'information entre les ontologies. Le système contient trois méthodes (deux automatiques et une semi-automatique). Il permet l'interopérabilité sémantique, syntaxique et structurelle entre les ontologies. Notre objectif était d'atteindre le plus grand nombre de correspondances exactes. Notre système est de type multi stratégies qui peut traiter et résoudre plus d'un problème critique. Par conséquent, il est susceptible d'être plus commodément applicable dans différents domaines.

CHAPITRE 3 : RESULTATS EXPERIMENTEAUX DES APPROCHES PROPOSEES

3.1 Introduction

Dans ce chapitre, l'accent est mis sur l'évaluation et l'expérimentation des méthodes d'alignement introduites dans ce manuscrit. Cette évaluation permet d'expertiser les résultats obtenus. Cette appréciation est réalisée en exploitant un ensemble de métriques d'évaluation. Ces métriques permettent, d'une part, d'estimer la qualité de l'alignement obtenu ; et d'autre part, de proposer un mécanisme de comparaison avec d'autres méthodes. L'expérimentation est réalisée sur des bases de test Benchmark, conférence, anatomie, et multiforme. Ces bases sont mises à la disposition de la communauté, qui travaille dans le cadre de l'alignement d'ontologies pour développer un processus de comparaison des méthodes d'alignement.

Le reste du chapitre est organisé comme suit : la section 4.2 décrit les différentes métriques d'évaluation permettant d'estimer la qualité des résultats obtenus pour chaque méthode. Les sections 4.3, 4.4, 4.5 et 4.6 décrivent les différents tests effectués sur les bases de test et proposent une étude comparative avec d'autres méthodes existantes dans la littérature.

3.2 Initiative d'évaluation de l'alignement d'ontologies

L'OAEI est une initiative internationale coordonnée pour établir un accord pour l'évaluation et l'amélioration des techniques disponibles de l'alignement d'ontologies. La campagne OAEI organise un concours chaque année depuis 2004, comprenant plusieurs types de tests, des mesures et des processus d'évaluation des résultats.

Les objectifs de l'Initiative d'évaluation de l'alignement d'ontologies sont :

- L'évaluation des forces et faiblesses des systèmes d'alignement.
- La comparaison des performances de techniques.
- L'augmentation de la communication entre les développeurs de l'algorithme.
- La comparaison des systèmes d'alignement et l'aider à améliorer le niveau de l'alignement d'ontologies.

L'OAEI traite ses objectifs par l'organisation d'une évaluation annuelle et de l'événement en publiant les tests et résultats de l'événement pour plus d'analyse.

3.3 Les mesures d'évaluation

L'une des tâches les plus difficiles est la sélection des mesures visant à évaluer une approche. C'est pour cela, les mesures standard qui ont été utilisées dans le domaine de la recherche d'information sont appliquées.

Afin d'évaluer les différentes approches ou le degré de conformité des résultats de l'appariement, les mesures de recherche d'information standard sont utilisées, à savoir : la précision, le rappel et le $F_\beta - mesure$.

3.3.1 Précision

Précision[11] mesure le nombre d'alignements corrects trouvés par rapport au nombre total de l'ensemble des alignements obtenus par une certaine méthodes.

Définition : Étant donné un alignement de référence R, la précision d'un certain alignement A est une fonction Précision : $A \times A \rightarrow [0,1]$; où :

$$\text{Précision} = \frac{\text{Nombre des alignements corrects trouvés}}{\text{Nombre des alignements extrait par une certaine méthode}} \quad (32)$$

Elle peut être définie comme suit :

$$\text{Précision} = \frac{|R \cap A|}{|A|} \quad (33)$$

Tels que :

- R est l'ensemble des alignements de référence.
- A est l'ensemble des alignements obtenus par une certaine méthode.

Une précision de la valeur 1 indique que tous les alignements trouvés sont corrects, mais cela ne signifie pas que tous les alignements sont trouvés. Alors, la précision doit être équilibrée par rapport à la mesure de rappel.

3.3.2 Rappel

Le rappel[11] mesure le nombre des alignements corrects trouvés par rapport au nombre total des alignements existants.

Définition : Étant donné un alignement de référence R, le rappel d'un certain alignement A est une fonction $Rappel : A \times A \rightarrow [0,1]$ telles que :

$$\text{Rappel} = \frac{\text{Nombre d'alignements corrects trouvés}}{\text{Nombre d'alignements existants}} \quad (34)$$

Il peut être défini comme suit :

$$Rappel = \frac{|R \cap A|}{|R|} \quad (35)$$

Une haute valeur de rappel indique qu'un grand nombre de ces alignements sont effectivement trouvés, mais il ne donne aucune information sur le nombre de faux alignements.

3.3.3 F_β-mesure

Le $F_\beta - mesure$[11] combine les deux mesures : la précision et le rappel en une seule mesure, elle est définie de la façon suivante :

$$F_\beta - mesure = (1 + \beta^2) \times \frac{\text{Précision} \times \text{Rappel}}{\beta^2 \times \text{Précision} + \text{Rappel}} \quad (36)$$

Où :

- β est une valeur de pondération réelle et positive.

3.4 Base Benchmarks

La base benchmarks fait partie de l'Initiative d'Évaluation de l'Alignement des Ontologies (OAEI) depuis sa création en 2005. Cette initiative est organisée par l'INRIA. Les ensembles de données sont restés presque constants en dehors des corrections mineures.

Depuis 2010, les bases de tests OAEI ont été menées dans le contexte du projet SEALS.

3.4.1 Description

Cette base de test vise à identifier les forces et les faiblesses des systèmes d'alignement en utilisant un ensemble de données généré systématiquement. À cette fin, une ontologie du

domaine de la bibliographie est systématiquement modifiée en omettant des caractéristiques (ou des combinaisons de caractéristiques) qui peuvent être exploitées par les systèmes d'alignement afin de trouver des correspondances. Chaque version modifiée doit être alignée avec l'ontologie d'origine. La modification systématique de l'ontologie comprend la suppression ou la modification d'étiquettes, de commentaires, d'individus revendiqués, de propriétés ou de la hiérarchie de subsomption.

Elle se compose de cinq groupes de test.

- Famille de tests 101 à 104 : les ontologies sources contiennent des classes et des propriétés portant le même nom que celles des ontologies de référence. Nous pouvons facilement obtenir toutes les paires d'entités appariées.
- Famille de tests 201 à 210 : l'ontologie 201 contient des étiquettes et des commentaires, mais il a des noms aléatoires qui n'ont pas de sens ; et 202 ne contient pas de noms ni de commentaires ; l'ontologie 205 comporte des synonymes. Enfin les ontologies de 206 à 210 représentent les traductions françaises de l'ontologie originale.
- Famille de tests 221 à 247 : la structure est modifiée (ne contient pas de spécialisation, hiérarchie aplatie), les instances sont supprimées et le type de données n'est pas spécifié.
- Famille de tests 248 à 266 : ces tests sont les plus difficiles. Encore une fois, les étiquettes et les commentaires ont été supprimés ainsi que les divers éléments structuraux. La seule information restante est les liens entre les instances et les classes.
- Famille de tests 301 à 304 : représentent des ontologies du monde réel modélisé par d'autres organisations, mais couvrant le même champ de métadonnées bibliographiques. Ces ontologies combinent les difficultés des tests précédents. Les ontologies réelles et l'ontologie de référence ne sont pas sémantiquement différentes en termes des concepts, de hiérarchies de concepts, de hiérarchies de propriétés et des propriétés.

3.4.2 Résultats

Les expériences ont été exécutées à l'aide d'un JDK 1.8 sur une machine à 8 processeurs (Intel i7 5th, 2,9 GHz, 16 Go de RAM, disque dur 512SSD) et MacOS high sierra.

Par conséquent, les résultats sont présentés ci-dessous dans les groupes correspondants.

3.4.2.1 Famille de tests 101-104 :

La famille de tests 101-104 est un simple test d'alignement de l'ontologie sur laquelle notre système a fonctionné très bien, parce que les ontologies n'avaient pas des caractéristiques ou des difficultés. La source d'ontologies continue les concepts et les propriétés avec les mêmes noms que ceux de l'ontologie de référence ; par suite, les similarités terminologiques, linguistiques, et structurelles ont été utilisées pour l'alignement. Cependant, comme les résultats sur les deux premières stratégies appliquées à des termes dans chaque document sont tout à fait différents, cette combinaison peut trouver facilement la plupart des alignements.

	OMBWSD		DROM		OM-NEURAL-NSGA-II	
Teste ID	P	R	P	R	P	R
101	1,00	1,00	1,00	1,00	1,00	1,00
102	1,00	1,00	1,00	1,00	1,00	1,00
103	1,00	1,00	1,00	1,00	1,00	1,00
104	1,00	1,00	1,00	1,00	1,00	1,00

Tableau 3 : Résultats de la comparaison entre OMBWSD, DROM et OM-NEURAL-NSGA-II ; Famille 101-104

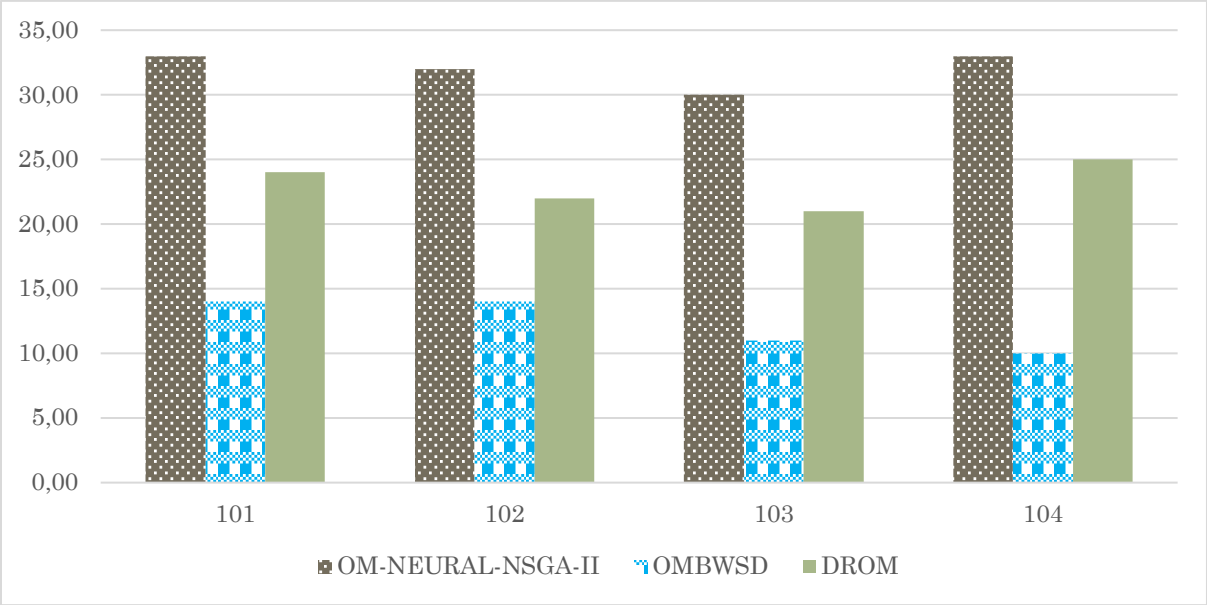

Figure 8 : Comparaison de temps d'exécution du système proposé pour la famille 101-104

Les résultats obtenus dans la figure 8 montrent que les méthodes DROM et OM-Neural-NSGA-II deviennent lentes, pour cette famille de test. En revanche, l'OMBWSD

est plus rapide. Les performances de ces méthodes sont détaillées dans le Tableau 3 sachant que P et R sont respectivement la Précision et le rappel.

Ces résultats sont dus au fait que DROM et OM-Neural-NSGA-II combinent les similarités terminologiques, linguistiques, structurelles et une couche de raffinement de l'alignement pour identifier l'alignement des relations, ce qui nécessite beaucoup d'appels récursifs et plusieurs calculs. En revanche, l'OMBWSD est plus rapide, car il utilise des similarités terminologiques, linguistiques, structurelles et un seuil qui permet de déterminer aisément les entités correspondantes.

3.4.2.2 Famille de tests 201-210 :

Pour la deuxième famille, elle n'y avait pas de nom pour certaines entités, d'autres ont changé de nom, où il n'y a pas de commentaire. Ces difficultés ne peuvent pas être résolues par les méthodes terminologiques. Néanmoins, notre système a obtenu de bons résultats pour l'ensemble de tests, de sorte que ce système est capable d'appliquer différentes stratégies dans les différents tests. En effet, notre système a trouvé la plupart des alignements corrects en utilisant la plupart des fonctions d'ontologies, telles que les commentaires, la synonymie, la structure et les instances, comme le montre le tableau 4.

	OMBWSD		DROM		OM-NEURAL-NSGA-II	
Teste ID	P	R	P	R	P	R
201	0,96	0,47	0,95	0,80	1,00	1,00
202	0,80	0,46	0,90	0,78	1,00	1,00
203	0,70	0,56	0,63	0,81	1,00	1,00
204	0,87	0,51	0,88	0,83	1,00	1,00
205	0,80	0,46	0,89	0,87	1,00	1,00
206	0,82	0,80	0,90	0,80	1,00	1,00
207	1,00	0,78	1,00	0,78	1,00	1,00
208	1,00	0,81	1,00	0,81	1,00	1,00
209	1,00	0,83	1,00	0,83	1,00	1,00
210	1,00	0,87	1,00	0,87	1,00	1,00

Tableau 4 : Résultats de la comparaison entre OMBWSD, DROM et OM-NEURAL-NSGA-II ; Famille 201-210

Dans les ontologies de test 201, 203, 204 et 205, lorsque des étiquettes de concepts et de propriétés ont été remplacées par des chaînes aléatoires ou par synonymes, il n'y a pas de similarité entre ontologies de tests et ontologie de référence dans les stratégies basées sur des chaînes de caractères. En revanche, les stratégies basées sur la structure ont réussi à produire des paires de concepts et de propriétés. Dans le test 202, les noms et les commentaires n'apparaissent pas ; notre système utilise les caractéristiques structurelles pour identifier l'alignement.

Pour les ontologies de tests 206-210, ils avaient des étiquettes et le nom en langue française. Ce qui explique l'utilisation d'un module de traduction dans notre système pour calculer la similarité entre les entités. L'utilisation du module de traduction, des étiquettes et des noms ont donné une précision et un rappel maximal 1.00 pour la méthode OM-NEURAL-NSGA-II. En revanche, le temps d'exécution est affecté (voir la figure 9), puisqu'on ajoute une couche supplémentaire qui normalise la langue des ontologies utilisées.

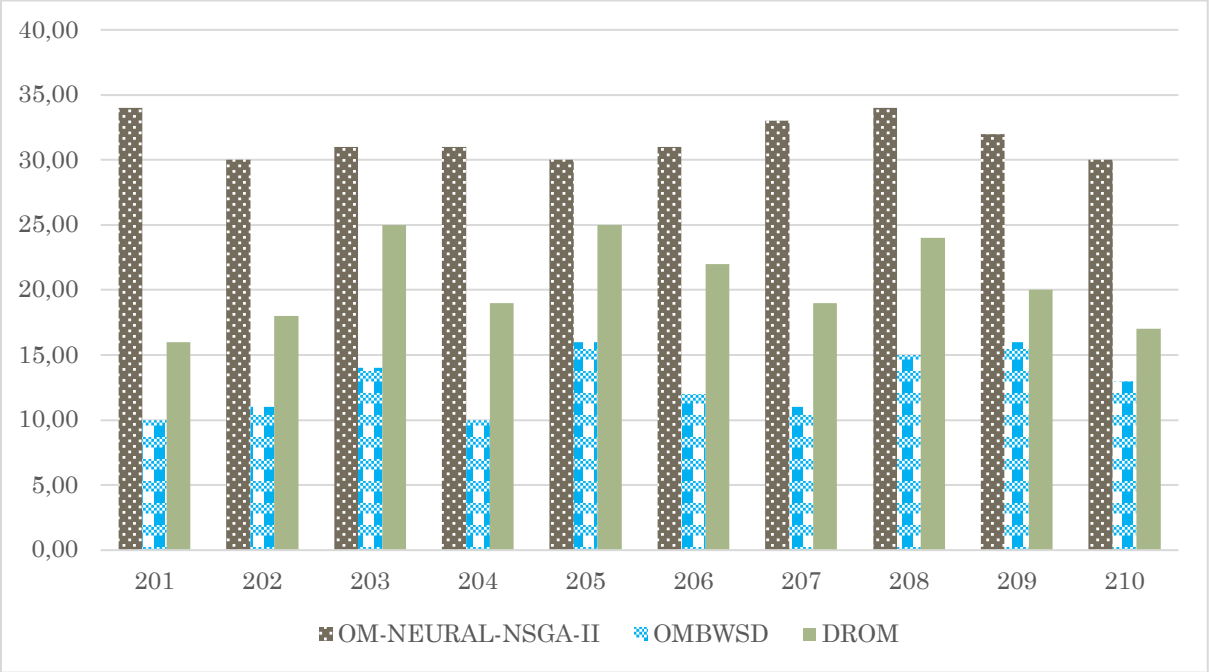

Figure 9 : Comparaison de temps d'exécution du système proposé pour la famille de tests 201-210

En fait, tout au long de ces dix tests ; la précision variait de 0,7 à 1,0, le rappel de 0.50 à 1,00 et la F_1-mesure de 0,37 à 1,00 (voir le tableau 4 et la figure 10).

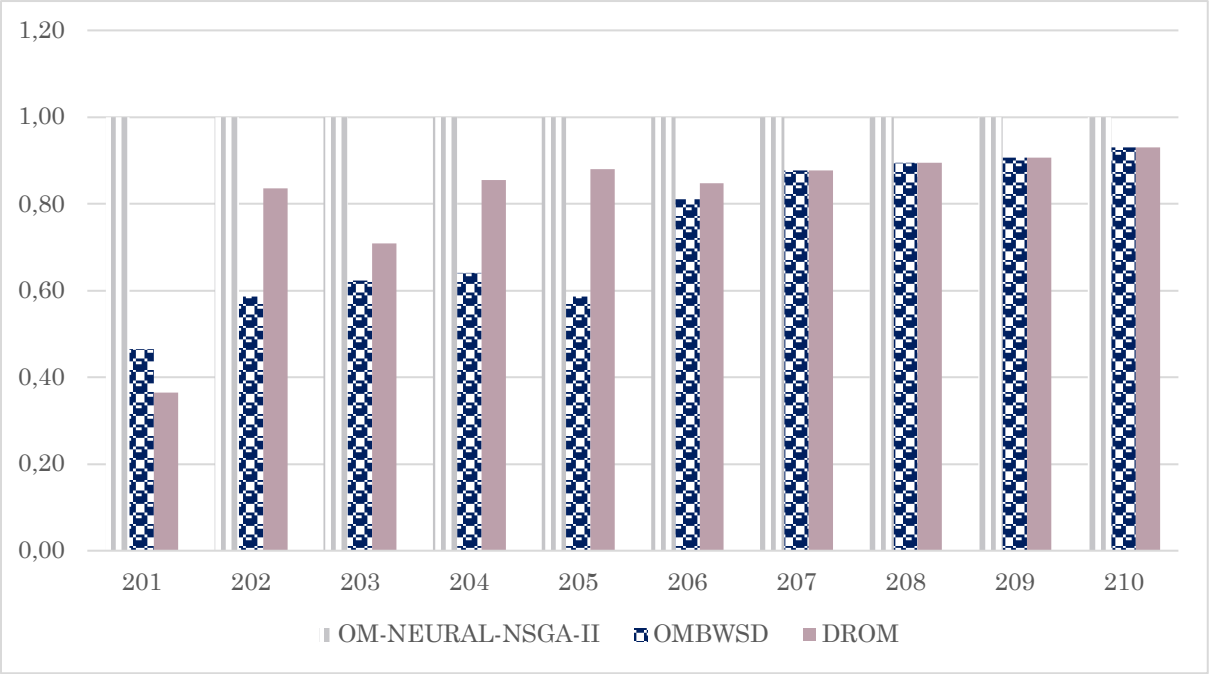

Figure 10 Résultats de F_1-mesure pour le système proposé sur OAEI-2016 pour la famille de tests 201-210

3.4.2.3 Famille de tests 221 à 247 :

Dans la troisième famille de tests, les noms, les étiquettes et les commentaires ne subissent aucune modification, mais les structures de ces ontologies ont été manipulées et dans certains cas des propriétés ont été ajoutées. Par conséquent, dans ces ontologies notre système s'exécute très bien en utilisant les méthodes terminologiques, linguistiques, et des stratégies fondées sur des heuristiques dans le calcul de la similarité.

Les questions les plus importantes auxquelles chacun de ces tests a été exposé ci-dessus et est brièvement repris ici. Pas de spécialisation dans le test 221, une hiérarchie plate pour le test 222, l'expansion de hiérarchie pour le test 223, le test 224 ne contient aucune instance, pas de restriction pour le 225, il n'y'a aucun type de données pour le 226 ; dans 227 les valeurs sont exprimées dans différents types de données ; pas de propriétés dans le 228, certaines classes sont devenues des instances pour le 229 et le 230 contient des classes aplaties ; tous ces tests ont été combinés avec un très haut taux de rappel et de précision. En conclusion, sur cette famille de tests notre système fonctionne bien.

Bien que les structures d'ontologies de tests aient été changées, notre système a constaté que la plupart des alignements sont corrects à l'aide des méthodes terminologiques (étiquette, commentaire), des méthodes linguistiques et heuristiques. Par suite, à la fois la précision, le rappel et F_1-mesure ont été excellents, le tableau 5 et la figure 11 montrent les résultats.

	OMBWSD		DROM		OM-NEURAL-NSGA-II	
Teste ID	P	R	P	R	P	R
221	0,94	0,95	0,94	1,00	1,00	1,00
222	0,80	0,50	0,94	0,67	1,00	1,00
223	0,94	0,95	0,94	1,00	1,00	1,00
224	0,94	0,95	0,94	1,00	1,00	1,00
225	0,94	0,96	0,94	1,00	1,00	1,00
228	0,85	1,00	1,00	1,00	1,00	1,00
230	0,94	0,95	0,94	1,00	1,00	1,00
231	0,85	1,00	1,00	1,00	1,00	1,00
232	0,85	1,00	1,00	1,00	1,00	1,00
233	0,60	0,50	0,94	0,67	1,00	1,00
236	0,94	0,96	0,94	1,00	1,00	1,00
237	0,85	1,00	1,00	1,00	1,00	1,00
238	0,85	1,00	1,00	1,00	1,00	1,00
239	0,85	1,00	1,00	1,00	1,00	1,00
240	0,85	1,00	1,00	1,00	1,00	1,00
241	0,85	1,00	1,00	1,00	1,00	1,00
246	1,00	0,30	1,00	0,43	0,98	0,81
247	0,93	0,70	0,94	0,65	0,98	0,97

Tableau 5 : Résultats de la comparaison entre OMBWSD, DROM et OM-NEURAL-NSGA-II ; Famille 221-247

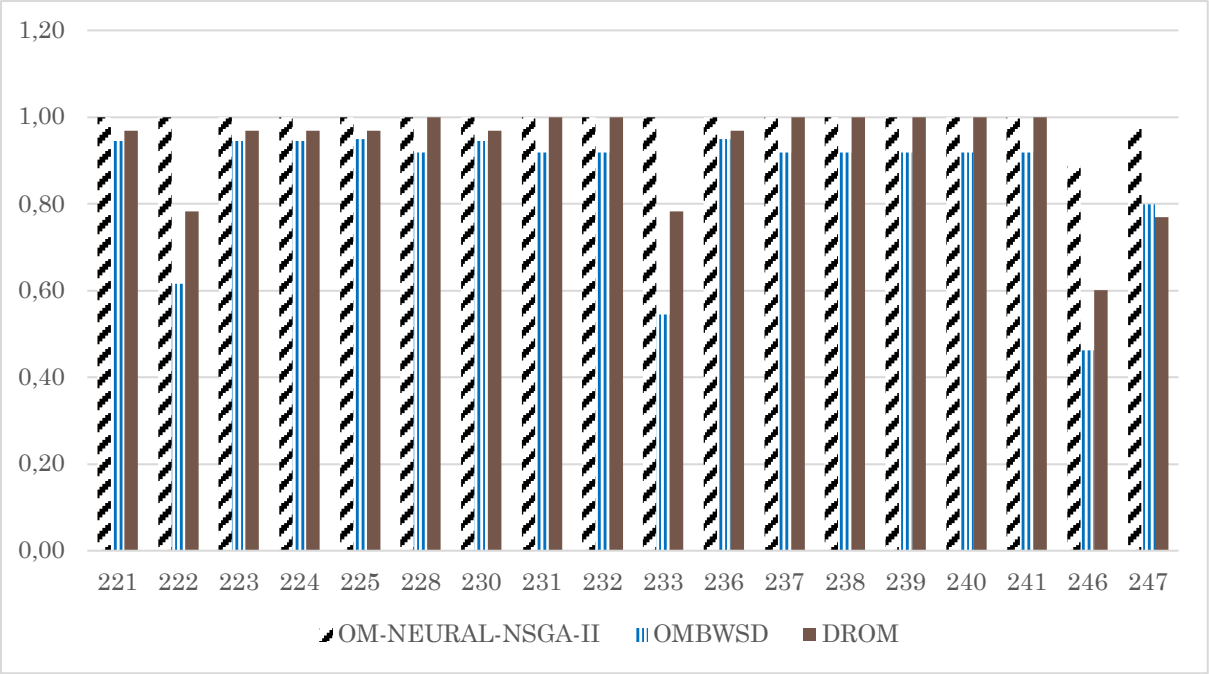

Figure 11 : Résultats de F_1-mesure pour le système proposé sur OAEI-2016 pour la famille de tests 221-247

Le tableau 5 présente les résultats des tests 221-247. Ces résultats sont très élevés et égaux à 1, qui est le résultat des méthodes proposées fortement en utilisant des algorithmes de correspondance basée sur la chaîne de caractère et les méthodes linguistiques.

Pour le temps d'exécution, la figure 12 illustre que la méthode OMBWSD est très rapide par rapport aux autres méthodes.

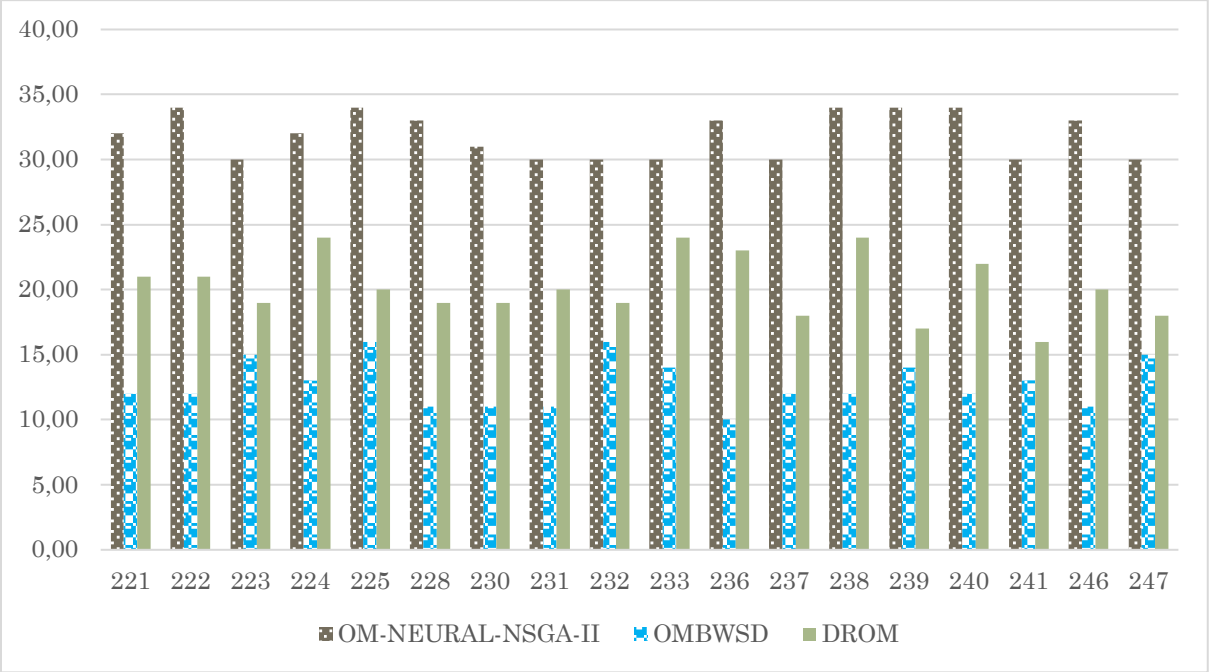

Figure 12 : Comparaison de temps d'exécution de système proposé pour la famille de tests 221-247

3.4.2.4 Famille de tests 248 à 266

Cette série de tests d'évaluation a été la plus difficile. Les étiquettes, les commentaires et les caractéristiques linguistiques avaient été fortement modifiés ou remplacés par des chaînes de caractères aléatoires, ou bien par modification de la structure de l'ontologie. Dans ce cas, les similarités terminologiques, linguistiques et structurelles entre la source et les ontologies de référence ont été faibles. De sorte, notre système a obtenu des résultats médiocres dans certains cas, lorsqu'il était très difficile d'identifier correctement les alignements.

Pour les tests 249, 250 et 257, des stratégies basées sur la structure ont commencé à être actives pour aider à améliorer l'alignement final. Les résultats de ces tests ont été donc raisonnables : le rappel variait de 0,23 à 0,89, tandis que la précision était beaucoup plus satisfaisante, allant de 0,73 à 1,00. Le tableau 6 présente ces résultats.

	OMBWSD		DROM		OM-NEURAL-NSGA-II	
Teste ID	P	R	P	R	P	R
248	0,97	0,42	0,98	0,39	0,99	0,89
249	0,97	0,39	0,95	0,33	0,94	0,82
250	0,97	0,43	1,00	0,49	1,00	0,72
251	0,80	0,44	0,92	0,27	0,99	0,87
252	0,98	0,44	0,97	0,36	0,98	0,88
253	0,98	0,47	0,94	0,32	0,93	0,81
254	0,93	0,46	1,00	0,46	1,00	0,67
257	0,96	0,46	1,00	0,44	0,89	0,49
258	0,88	0,33	0,96	0,31	0,90	0,78
259	0,95	0,44	0,95	0,42	0,93	0,81
260	0,88	0,35	1,00	0,53	0,97	0,71
261	0,91	0,45	1,00	0,46	0,92	0,53
262	0,92	0,44	1,00	0,50	0,94	0,53

265	1,00	0,23	1,00	0,40	1,00	0,03
266	0,73	0,32	0,82	0,40	0,84	0,76

Tableau 6 : Résultats de la comparaison entre OMBWSD, DROM et OM-NEURAL-NSGA-II ; Famille 247-266

D'après les figures 13 et 14, on remarque que le temps d'exécution et F_1-mesure en corrélation ; c'est à dire, lorsque la qualité de F_1-mesure augmente, le temps d'exécution augmente aussi.

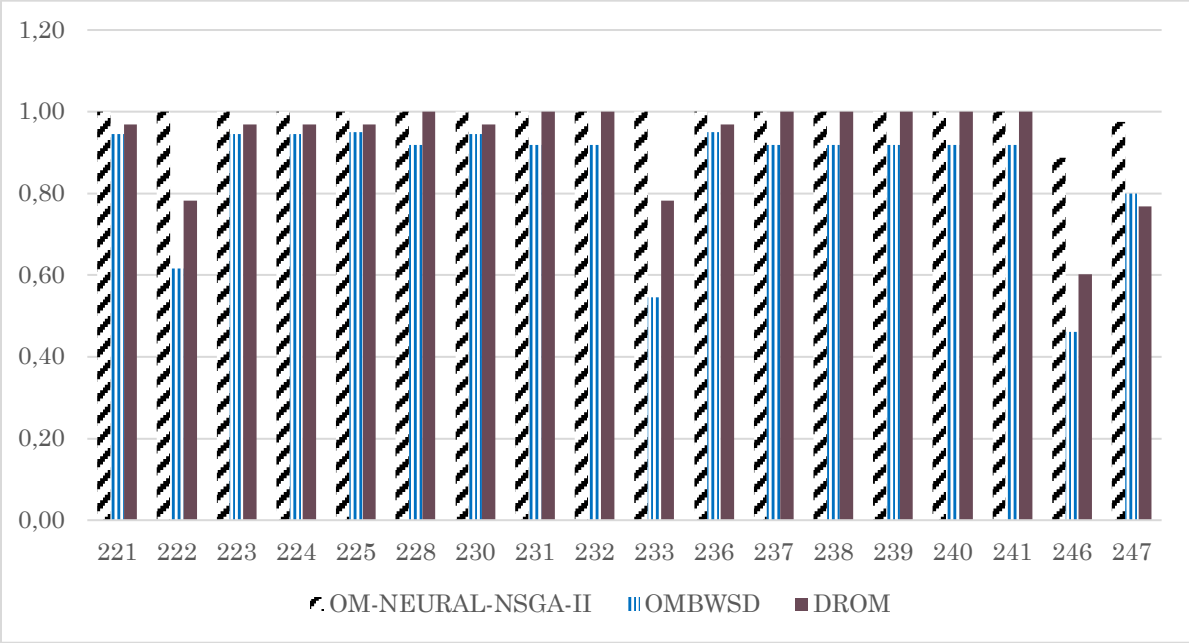

Figure 13 : Résultats de F_1-mesure pour le système proposé sur OAEI-2016 pour la famille de tests 221-247

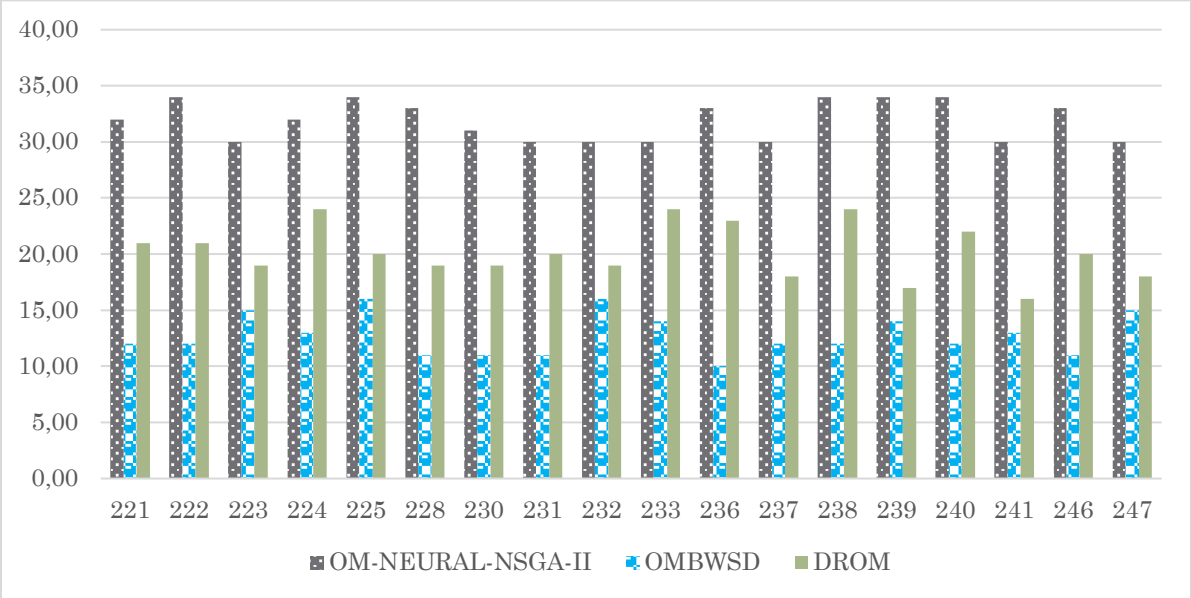

Figure 14 : Comparaison de temps d'exécution du système proposé pour la famille de tests 221-247

3.4.2.5 Famille de tests 301-304 :

Ces tests avaient été modelés par des institutions différentes, mais pour le même domaine de la bibliographie. Cette famille combine les difficultés de tous les essais précédents, l'ensemble du système a été utilisé afin d'obtenir de bons résultats. Dans l'ensemble, la précision varie de 0,85 à 1.00 et le rappel varie 0.39 à 0,89.

	OMBWSD		DROM		OM-NEURAL-NSGA-II	
Teste ID	P	R	P	R	P	R
301	0,85	0,42	0,92	0,80	1,00	0,89
302	0,85	0,39	0,97	0,56	1,00	0,86
303	0,85	0,43	0,94	0,51	1,00	0,88
304	0,85	0,44	0,91	0,55	1,00	0,85

Tableau 7 : Résultats de la comparaison entre OMBWSD, DROM et OM-NEURAL-NSGA-II ; Famille 301-304

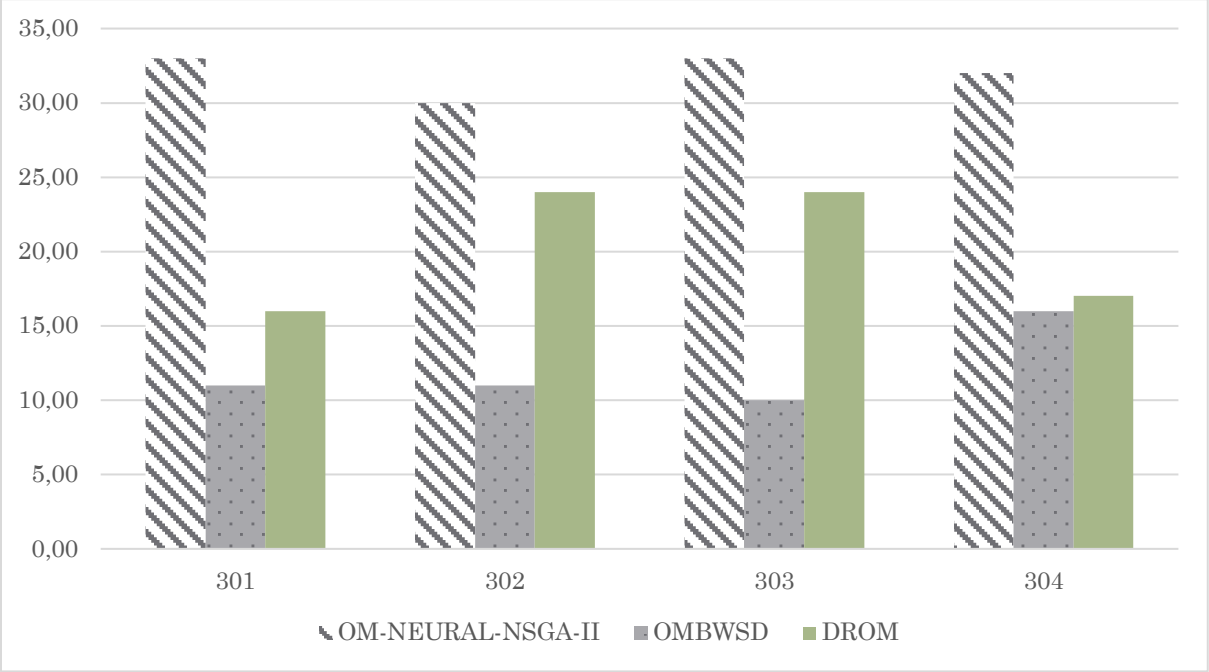

Figure 15 : Comparaison de temps d'exécution du système proposé pour la famille de tests 301-304

Dans les tests 302 et 303, les étiquettes sont souvent différentes. Donc, notre système est utilisé afin de trouver l'alignement. Le test 304 est proche de l'ontologie de référence, la hiérarchie de ce test contient des classes qui sont des sous-classes de plusieurs autres classes ; cette modification de la structure et le manque d'instances ont été une des principales causes de la

faible mesure de rappel, mais c'était encore raisonnable. La capacité de la base linguistique pour trouver des similitudes entre les deux ontologies candidates dans chaque test a été élevée, tandis que le rendement de la stratégie structurelle était raisonnable. Enfin, les stratégies basées sur l'heuristique ont joué un rôle essentiel dans l'amélioration des résultats correspondants, illustré dans le tableau 7 et la figure 15.

3.4.3 Comparaison entre les systèmes participants de l'OAEI

Une comparaison entre les systèmes participants de l'OAEI 2016 dans la base de benchmark est présentée au tableau 8.

Système d'alignement	**P**	**R**	**F_1**
OMBWSD	0,90	0,67	0,77
DROM	0,96	0,73	0,83
OM-NEURAL-NSGA-II	0,98	0,89	0,94
AML	1,0	0,24	0,39
CroMatcher	0,96	0,83	0,89
Lily	0,97	0,83	0,89
LogMap	0,93	0,39	0,55
LogMapLt	0,43	0,50	0,46
PhenoMF	0,03	0,01	0,02
PhenoMM	0,03	0,01	0,02
PhenoMP	0,02	0,01	0,01
XMap	0,95	0,40	0,56
LogMapBio	0,48	0,24	0,32
ALin	NaN	0,00	NaN
CroLOM	NaN	0,00	NaN
DKP-AOM	NaN	0,00	NaN

DKP-AOM-Lite	NaN	0,00	NaN
DiSMatch	NaN	0,00	NaN
FCA-Map	NaN	0,00	NaN
LPHOM	NaN	0,00	NaN
LYAM++	NaN	0,00	NaN
RiMOM	0,00	0,00	NaN

Tableau 8 : Résultats de la comparaison entre le système proposé et les systèmes participants au OAIE-2016 ;

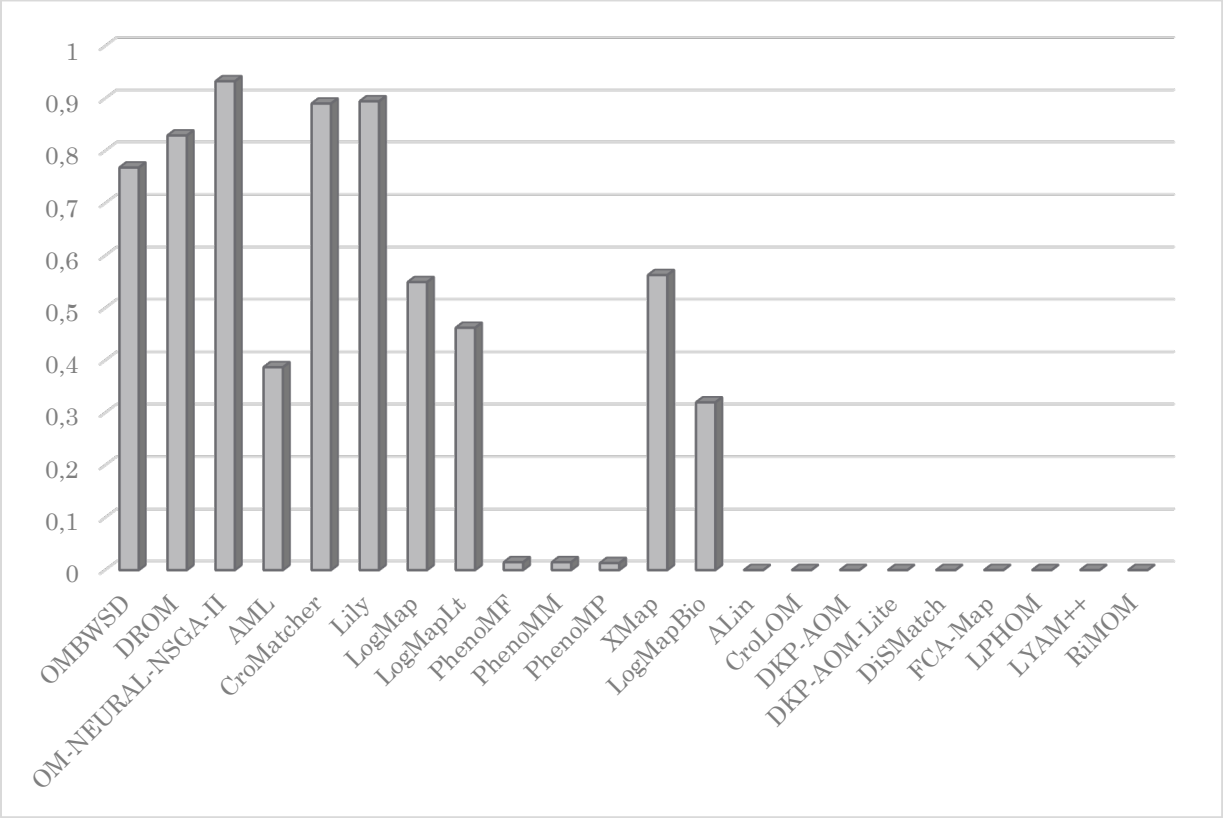

Figure 16 : Résultats de F_1-mesure pour le système proposé et les systèmes participants à l'OAEI-2016 pour la base benchmarks

Pour évaluer les résultats fournis par notre système, le Tableau 8 récapitule les valeurs des métriques de précision, de rappel et de F_1-mesure. Il présente aussi les valeurs obtenues par les méthodes AML, CroMatch, Lily, LogMap, LogMapLt, PhenoMF, PhenoMM, PhenoMP, LogMapBio, ALIN, CroLOM, DKP-AOM, DKP-AOM-Lite, DiSMatch, FCA-Map, LPHOM, LYAM++ et RIMOM. La méthode Neural NSGA-II de notre système donne de meilleurs résultats par rapport aux autres systèmes, notamment sur la famille de tests 30X (celle des

ontologies réelles). Notre système fournit de bons résultats sur la famille de tests 26X. Cette famille de tests se caractérise par l'absence des noms, des commentaires et des propriétés des entités ontologiques. Ces composants ontologiques sont des facteurs capitaux pour la détermination de l'alignement par tous les systèmes d'alignement. Les résultats expérimentaux montrent que les performances de notre système sont liées aux caractéristiques des différents composants ontologiques (noms, commentaires, étiquettes, et structure). En effet, l'absence de ces descripteurs dégrade considérablement la qualité de l'alignement fourni. Par exemple, les tests 257 et 260 marquent l'absence des noms et des commentaires ainsi que les relations et les propriétés. L'absence des descripteurs noms et commentaires diminue la valeur de la similarité linguistique composée. De même, l'absence des relations et des propriétés détériore la valeur de la similarité structurelle. La valeur de la similarité linguistique et structurelle affecte la valeur de la similarité agrégée. En conséquence, les valeurs des métriques de précision et de rappel s'affaiblissent. Les résultats de la figure 16 s'expliquent par le fait que notre système s'appuie sur l'agrégation des similarités. D'une part, ceci explique le choix adéquat de la mesure de similarité qui s'adapte le mieux à un descripteur donné. D'autre part, ils se justifient par une bonne exploitation des mesures de similarités terminologiques, linguistiques, extensionnelles et structurelles de notre système.

3.5 Base d'anatomie

La base d'anatomie, organisée par l'Université de Mannheim, fait partie de l'Initiative d'évaluation de l'alignement ontologique (OAEI) depuis 2005. Ces ontologies utilisées dans l'ensemble de données sont restées les mêmes, mais plusieurs améliorations ont été apportées depuis. Les ontologies et l'alignement de référence sont créés manuellement.

3.5.1 Description

Ce test vise à refléter le cas d'utilisation du monde réel consistant à aligner de grandes ontologies du domaine biomédical. L'ensemble de données utilisé dans la base d'anatomie comprend les deux ontologies « l'anatomie d'humain et l'anatomie de souris », ainsi qu'un alignement de référence créé manuellement. L'anatomie de la souris contient 2744 classes, l'anatomie de l'humain comprend 3304 classes et l'alignement de référence 1517 correspondances. À cet égard, la base d'anatomie est le plus grand ensemble de données de l'OAEI avec un alignement de référence de haute qualité disponible. Selon les organisateurs de la piste, l'ensemble de données contient environ 60 % de correspondances triviales dans le sens où elles peuvent être identifiées par des techniques de comparaison de chaînes de base.

Dans l'OAEI 2011, la base d'anatomie a été menée dans le contexte du projet SEALS. De plus, ce test a été réalisé dans le cadre de la deuxième campagne d'évaluation de SEALS au 2012.

3.5.2 Résultats et Discussion

Les résultats pour chaque participant à la base d'anatomie OAEI 2017 sont listés dans le Tableau 9. En raison de la taille des ontologies, 6 Systèmes sur 14 n'ont pas pu calculer un résultat en moins de 100 s. Les résultats de la Figure 17 et du tableau 9 rapportent que notre système et le système AML peuvent faire face à des ontologies qui contiennent plus de 1000 concepts.

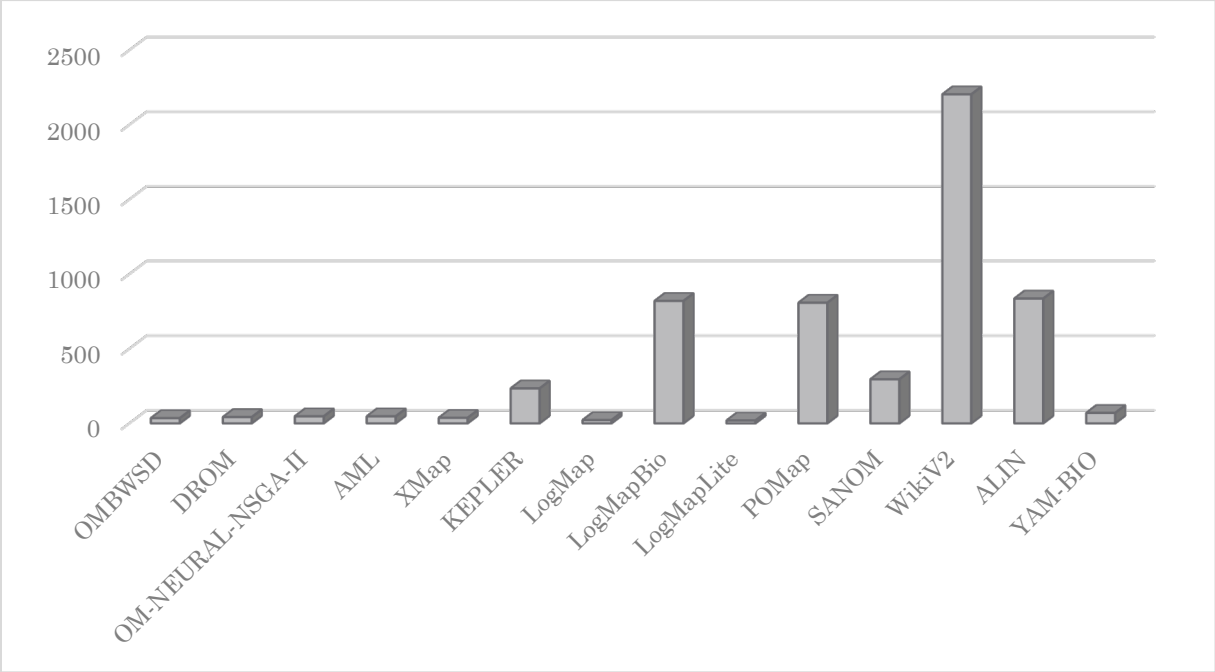

Figure 17 : Comparaison de temps d'exécution du système proposé et de tous les participants à la piste d'anatomie OAEI 2017

La figure 17 montre que LogMapLite a la durée d'exécution la plus courte. Il a besoin de 19 secondes pour aligner les ontologies de test.

Système	**Taille**	**P**	**R**	**F_1**
OMBWSD	1400	0,921	0,852	0,885
DROM	1401	0,954	0,851	0,885
OM-NEURAL-NSGA-II	1493	0.955	0.931	0.943
AML	1493	0,95	0,936	0,943

XMap	1412	0,926	0,863	0,893
KEPLER	1173	0,958	0,741	0,836
LogMap	1397	0,918	0,846	0,88
LogMapBio	1534	0,889	0,899	0,894
LogMapLite	1148	0,962	0,728	0,829
POMap	1492	0,94	0,925	0,933
SANOM	1304	0,895	0,77	0,828
WikiV2	1260	0,883	0,734	0,802
ALIN	516	0,996	0,339	0,506
YAM-BIO	1474	0,948	0,922	0,935

Tableau 9 : Taille de l'alignement, précision, rappel et résultat F_1-mesure pour le système proposé et tous les participants à la piste d'anatomie OAEI 2017

Le tableau 9 montre également les résultats pour F-mesure et la taille des alignements. En ce qui concerne F-mesure, les 5 premiers systèmes classés sont OM-NEURAL-NSGA-II, AML, YAM-BIO, POMap, LogMapBio et XMap. Parmi ceux-ci, AML et OM-NEURAL-NSGA-II ont obtenu la mesure F la plus élevée (0,943). En termes de nombre de correspondances, AML et OM-NEURAL-NSGA-II ont généré le même nombre de correspondances 1493, LogMapBio a généré 41 correspondances de plus, LogMapLite a généré 345 de moins, et XMap a généré 81 de moins.

3.6 Base de Conférence

La base de conférence, organisée par l'Université des Économies de Prague, en République tchèque, faisait partie d'OAEI depuis 2006. Elle contient 16 ontologies du même domaine et 11 expressions logiques diverses (tableau 10). Elle comprend aussi un sous-ensemble de 21 alignements de référence impliquant 7 ontologies.

Nom	Nombre de classe	Nombre de Datatype Propriétés	Nombre d'Object Propriétés	DL expressivité
Ekaw	74	0	33	SHIN
Sofsem	60	18	46	ALCHIF(D)
Sigkdd	49	11	17	ALEI(D)
Iasted	140	3	38	ALCIN(D)
Micro	32	9	17	ALCOIN(D)
Confious	57	5	52	SHIN(D)
Pcs	23	14	24	ALCIF(D)
OpenConf	62	21	24	ALCOI(D)
ConfTool	38	23	13	SIN(D)
Crs	14	2	15	ALCIF(D)
Cmt	36	10	49	ALCIN(D)
Cocus	55	0	35	ALCIF
Paperdyne	47	21	61	ALCHIN(D)
Edas	104	20	30	ALCOIN(D)
MyReview	39	17	49	ALCOIN(D)
Linklings	37	16	31	SROIQ(D)

Tableau 10 : Caractéristiques des ontologies de la base de conférence

Le but de cette base est de trouver des alignements dans un ensemble d'ontologies décrivant le domaine de l'organisation de la conférence. Par rapport aux autres bases d'OAEI, les ontologies dans la base de conférence sont plus expressives en termes de la logique de description qu'elles couvrent.

Dans l'OAEI 2010 et 2011, la base de la conférence a été menée dans le contexte du projet SEALS. De plus, cette base a été réalisée dans le cadre de la deuxième campagne d'évaluation de SEALS au printemps 2012.

3.6.1 Résultats

L'évaluation de cette base est basée sur des alignements de référence (RA1) et deux autres alignements impliqués (RA2 et RAR2) déduits de RA1. Notre évaluation est limitée aux alignements RA1, car ils sont le seul ensemble disponible publiquement. RA1 est divisé en trois sous-évaluations, comme suit :

- RA1-M1, seuls les alignements entre les classes sont évalués ;
- RA1-M2, seuls les alignements entre les propriétés (objet et données) sont évalués ;
- RA1-M3, les deux alignements entre les classes et les propriétés sont évalués.

Nous comparons les résultats de notre approche avec les résultats des 12 participants à la campagne de l'OAEI 2017. Ces résultats ont été obtenus à partir de la page Web décrivant les résultats de la campagne.

L'évaluation est basée sur la précision (P), le rappel (R), la F_1-mesure (F1), F_2-mesure (F2) et $F_{0.5}$-mesurer (F0.5).

❖ **RA1-M1**

Pour cette évaluation, nous avons évalué notre système. Les résultats moyens pour les systèmes sont résumés dans le tableau 11.

	Seuil	**P**	**R**
OMBWSD	0,80	0,72	0,63
DROM	0,00	0,78	0,64
OM-NEURAL-NSGA-II	0.00	0,87	0,74
AML	0,00	0,83	0,7
LogMap	0,00	0,84	0,64
Xmap	0,00	0,84	0,64

KEPLER	0,73	0,76	0,61
LogMapLt	0,00	0,84	0,54
WikiV3	0,00	0,83	0,55
POMap	0,00	0,88	0,47
ALIN	0,00	0,89	0,32
SANOM	0,00	0,81	0,29
ONTMAT	0,00	0,11	0,52

Tableau 11 : Résultats de précision et rappel pour le système proposé et tous les participants à la base de conférence RA1-M1 OAEI 2017

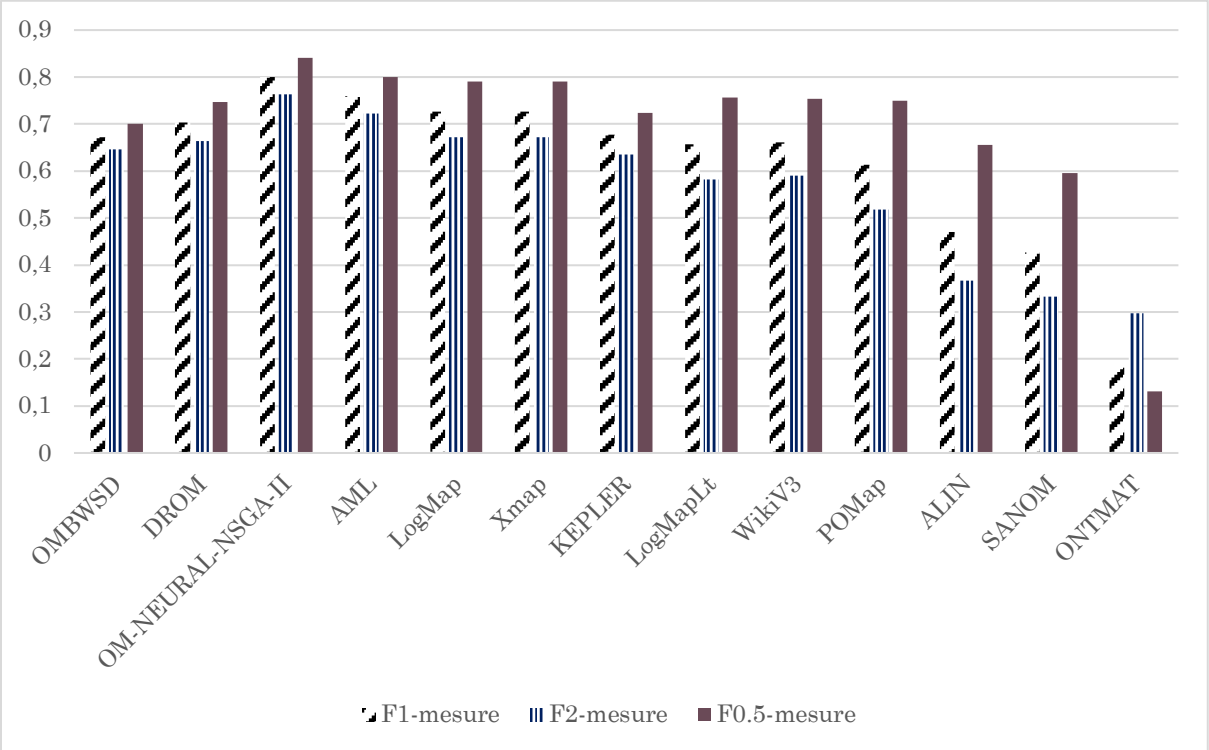

Figure 18 : Résultats de F_1-mesure, F_2-mesure et de $F_{0.5}$-mesure pour le système proposé et les systèmes participants à l'OAEI-2017 pour la base conférence RA1-M1

Nous observons que nos résultats sont situés au top. Les meilleures approches bénéficient de stratégies plus élaborées et de ressources externes pour calculer les similitudes. Ces résultats sont légèrement plus proches, voire mieux que le rappel des participants AML et LogMap. En plus ; la figure 18 indique que notre système et le système AML plus stable au changement de la valeur de pondération de précision et de rappel.

❖ **RA1-M2**

Comme le montrent le tableau 12 et la figure 19, nous observons que notre système et AML fonctionnent bien dans cette tâche, et toutes les autres approches ont des difficultés à aligner les propriétés. Les résultats de notre approche sont encore une fois bons. Nous avons remarqué que 9 systèmes ayant réussi à donner des résultats, et les autres participants ont échoué.

	Seuil	**P**	**R**
OMBWSD	0,80	0,87	0,37
DROM	0,00	0,94	0,43
OM-NEURAL-NSGA-II	0,00	1,00	0,62
AML	0,00	1,00	0,44
LogMap	0,79	0,62	0,28
Xmap	0,00	0,75	0,2
LogMapLt	0,00	0,24	0,22
KEPLER	0,00	0,17	0,28
WikiV3	0,67	0,29	0,11
ALIN	-	-	-
POMap	-	-	-
ONTMAT	-	-	-
SANOM	-	-	-

Tableau 12 : Resultats de précision et rappel pour le système proposé et tous les systèmes participants à la base conférence RA1-M2 OAEI 2017

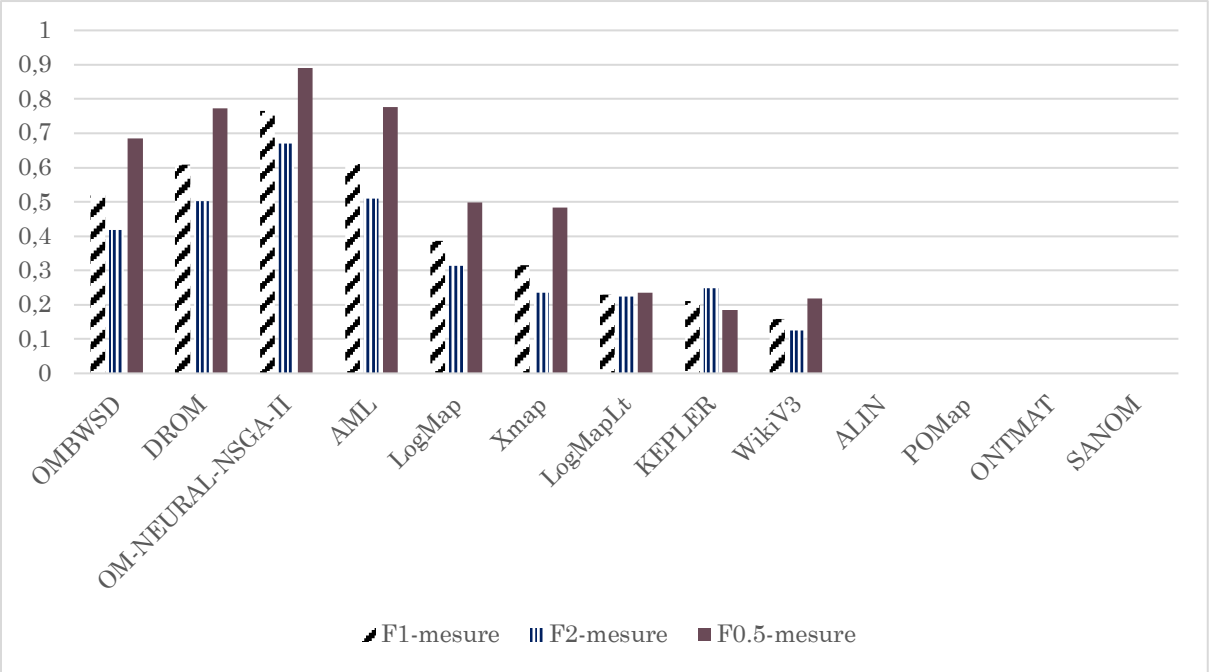

Figure 19 : Résultats de F2-mesure, F1-mesure et de F0.5-mesure pour le système proposé et les systèmes participants à l'OAEI-2017 pour la base conférence RA1-M2

❖ **RA1-M3**

Le tableau 13 et la figure 20 résument nos résultats par rapport aux résultats des autres participants pour l'évaluation des classes et des propriétés. Notre approche garde un rang stable par rapport aux autres approches. Nous remarquons qu'ALIN et SANOM ont des positions non stables à travers les évaluations. L'utilisation de toutes les contraintes semble avantageuse pour le rappel plus que pour la précision en raison du bruit provoqué par les propriétés faussement alignées positives.

	Seuil	**P**	**R**
OMBWSD	0,80	0,68	0,31
DROM	0,00	0,83	0,53
OM-NEURAL-NSGA-II	0.00	0,89	0,69
AML	0,00	0,84	0,66
LogMap	0,00	0,82	0,59
Xmap	0,00	0,84	0,57

LogMapLt	0,00	0,73	0,5
KEPLER	0,92	0,76	0,48
WikiV3	0,00	0,67	0,49
POMap	0,94	0,73	0,4
ALIN	0,00	0,89	0,27
SANOM	0,00	0,81	0,25
ONTMAT	0,00	0,06	0,43

Tableau 13 : Resultats de précision et rappel pour le système proposé et les systèmes participants à la base de conférence RA1-M3 OAEI 2017

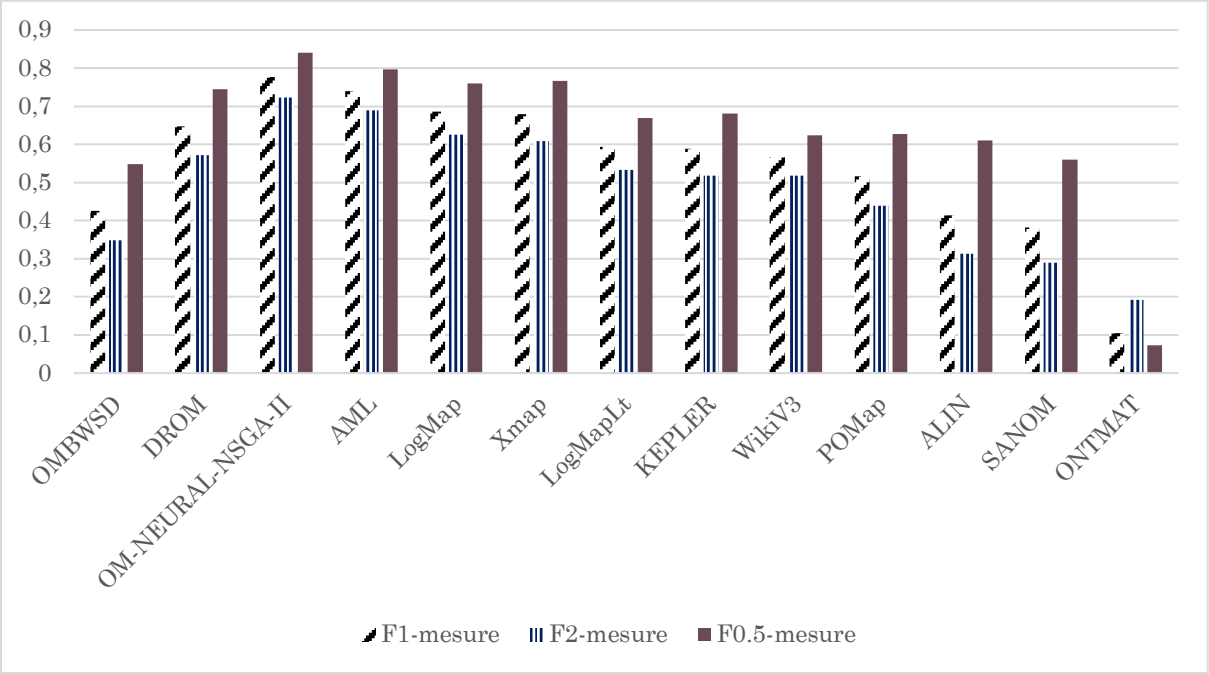

Figure 20: Resultats de F_2-mesure, F_1-mesure et de $F_{0.5}$-mesure pour le système proposé et les systèmes participants à l'OAEI-2017 pour la base conférence RA1-M3

En résumé, notre approche atteint des résultats prometteurs pour sa première comparaison en ce qui concerne le problème d'appariement d'ontologies par paires. Notre modèle est plus efficace lorsque nous utilisons toutes les contraintes proposées (RA1-M3). L'interaction entre les contraintes conduit à des résultats sémantiquement significatifs plus proches des références qui sont illustrés par un bon rappel. Enfin, le temps d'exécution moyen de notre système (prétraitement, génération et résolution de programme linéaire) sur 21 paires de la piste de

conférence était de 20 s en utilisant la méthode OM-Neural-NSGA-II, de 13 s en utilisant la méthode DROM et de 10 s en utilisant la méthode OMBWSD.

3.7 Conclusion

La base des benchmarks OAEI est devenue une norme pour l'évaluation des systèmes d'alignement ontologique et la comparaison de nouveaux systèmes avec l'état de l'art. Cependant, les ensembles de données doivent être utilisés avec soin lors de la création de rapports sur les performances des systèmes d'alignement. Cela vaut en particulier pour les alignements référencés fournis, car ils sont systématiquement générés avec les ontologies de test individuelles. Ceci signifie que pour chaque modification de l'ontologie d'origine, l'alignement référencé est modifié en conséquence. Celui-ci demande un soin particulier quand il est fait automatiquement, puisque la suppression des caractéristiques de l'ontologie peut conduire à des situations où il n'y a pas de preuve que certaines entités participent à une correspondance particulière. De telles correspondances ne doivent pas être incluses dans un alignement de référence. Dans la version 2016 d'OAEI, les correspondances (#PersonList, # dsqdbz) et (#MastersThesis, #xsqlknk) font partie de l'alignement de référence (avec une valeur de confiance de 100 %). Cette assignation précise ne peut même pas être faite par un être humain, puisque toutes les caractéristiques qui supportent l'une des deux correspondances ont été supprimées dans le cas de test n ° 262.

Puisque, dans cet exemple, les deux correspondances ne peuvent pas être des informations théoriquement justifiées, elles ne doivent pas faire partie de l'alignement de référence. Un autre remède consisterait à réduire la valeur de confiance dans l'alignement référencé pour les deux correspondances.

En conclusion, la base benchmarks OAEI ne doit pas être utilisée qu'en tenant compte du fait que dans de nombreux cas de test, la précision et le rappel de 100 % ne peuvent pas être atteints de manière fiable. De plus, il n'est pas documenté qu'elles sont les scores de précision et de rappel les plus élevés possible pour chaque cas de test, de sorte que les ensembles de données ne peuvent pas être utilisés que pour comparer des systèmes de précision et de rappel.

Les résultats de la base benchmark ainsi que les résultats des autres bases présentées par les expériences dans ce chapitre démontrent que l'objectif de la proposition a été atteint avec succès. Pour la combinaison des différentes métriques de similarité, les résultats obtenus sont meilleurs dans tous les domaines, par rapport aux approches de l'état de l'art dans l'alignement

de l’ontologie. Dans tous les scénarios testés, le système proposé occupe le meilleur emplacement par rapport aux systèmes participant à OAEI 2016 ou OAEI 2017.

CONCLUSION GÉNÉRALE

Le nombre croissant d'ontologies hétérogènes qui décrivent souvent le même domaine d'intérêt a été présenté comme un défi pour l'interopérabilité des données. Ce défi a été confronté à la zone d'étude de l'alignement des ontologies, qui s'intéresse à l'identification de la correspondance entre les entités. Les études dans le domaine de l'alignement des ontologies ont progressé au fil des ans et de nouveaux défis sont apparus face à de nouvelles demandes.

Parmi ces nouveaux défis, celui qui est traité dans ce mémoire concerne la recherche de solutions pour l'alignement des ontologies dont le but d'améliorer la qualité des alignements générés, de s'adapter à différents domaines et aussi d'agréger de différentes similarités. Le système proposé met en œuvre une variété de techniques qui sont basées sur les similarités terminologiques, linguistiques et structurelles. Plus précisément, une nouvelle approche générique a été développée pour l'alignement automatique et semi-automatique d'ontologies avec un minimum d'effort humain. Par conséquent, une contribution principale de cette thèse est de présenter un nouveau processus d'appariement multi stratégies pouvant d'améliorer considérablement le processus d'alignement de l'ontologie elle-même et sa production.

Le système élaboré compare plusieurs caractéristiques de l'Ontologie pour trouver des entités similaires : il compare les étiquettes des entités, les relations entre eux et leurs extensions connues (instances et classes), ainsi que leur structure. L'objectif principal de ces mesures est d'automatiser le processus de comparaison des méthodes d'alignement et de l'évaluation de la qualité de leurs produits.

D'après les résultats de l'étude expérimentale dans cette thèse ; certaines approches ou outils sont basés sur une seule stratégie d'alignement, tandis que d'autres utilisent plus d'une stratégie ; c'est ce qu'on appelle multi stratégie. Il semble qu'un système multi stratégie est en fait mieux qu'un système utilise une seule stratégie, parce qu'il aborde et résout plus d'un niveau d'hétérogénéité.

Pour les techniques basées sur des classifications linéaires et des règles ont généralement l'avantage d'une vitesse de fonctionnement relativement rapide. Pour les méthodes basées sur des règles, elles partagent l'inconvénient d'ignorer les informations supplémentaires. Pour les travaux basés sur des classifications linéaires (algorithme génétique[62], PSO[70], perceptron), ils recherchent un hyperplan séparant des classes de discrimination linéairement séparables.

C’est une limitation sévère qui ne peut pas résoudre les problèmes non linéaires. Pour remédier au problème de l’absence du séparateur linéaire, l’idée est d’utiliser des réseaux neuronaux en considérant le problème dans un espace de dimensions supérieures, éventuellement de dimensions infinies. Dans ce nouvel espace, il est probable qu’il y ait un séparateur linéaire. Plus formellement, la transformation non linéaire est appliquée aux vecteurs d’entrée. En outre, il existe deux problèmes avec ce type d’algorithme :

- Au lieu d’obtenir un optimum global, nous aurons un optimum local grâce à la phase d’entraînement (utilisation de la descente en pente).
- La complexité augmente pour mettre à jour les poids.

En fait, les résultats obtenus par l’approche proposée dans les expériences ont été comparés avec les systèmes évalués par OAEI 2016 et 2017. Ces résultats démontrent que l’approche présentée est prometteuse, car les alignements générés présentent la qualité compatible avec les solutions de référence d’OAEI pour chaque base, compte tenu les mesures de précision, de rappel et de F_{β}-mesure.

1. Limitations

Les limites de la recherche sont les suivants :

- Dans le monde réel, la possibilité d’interaction entre deux systèmes d’information de différents domaines est très fréquente. Cette réalité limite notre étude puisqu’on n’a pas de base de tests de différents domaines pour valider le système développé.
- Une autre limitation de l’approche proposée est liée à la configuration correcte des variables de l’approche OM-NEURAL-NSGA-II pour l’adapte aux autres domaines.

2. Travaux futurs

Les résultats réalisés avec le système proposé acceptent plusieurs perspectives, à savoir :

- L’exploration d’autres domaines, y compris le domaine médical.
- L’intégration des outils Big-Data pour traiter les ontologies de grande taille.
- L’inclusion et l’exploration des contraintes (les axiomes simples, complexes ...)

REFERENCES

[1] J. Domingue, D. Fensel, and J. A. Hendler, Handbook of Semantic Web Technologies. Springer Science & Business Media, 2011.

[2] L. Yu, A Developer's Guide to the Semantic Web. Springer, 2014.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, 'The Semantic Web', Scientific American, pp. 29–37, May 2001.

[4] T. R. Gruber, 'A translation approach to portable ontology specifications', Knowledge acquisition, vol. 5, no. 2, pp. 199–220, 1993.

[5] J. Hendler, T. Berners-Lee, and E. Miller, 'Integrating Applications on the Semantic Web', Journal of the Institute of Electrical Engineers of Japan, vol. 122, no. 10, pp. 676–680, Ocotbre 2002.

[6] A. Sheth, Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications. Idea Group Inc (IGI), 2013.

[7] S. Lecomte and T. Boulanger, XML par la pratique : bases indispensables, concepts et cas pratiques. Editions ENI, 2008.

[8] J. Hjelm, Creating the Semantic Web with RDF: Professional Developer's Guide. Wiley, 2001.

[9] G. Antoniou, F. van Harmelen, and R. Hoekstra, A Semantic Web Primer. MIT Press, 2012.

[10] J. Subercaze and P. Maret, 'Programming Semantic Agent for Distributed Knowledge Management', in Semantic Agent Systems, Springer, pp. 47–65, 2011.

[11] J. Euzenat and P. Shvaiko, Ontology Matching, Second Edition. Springer Science & Business Media, 2013.

[12] Y. Sun, 'A Comparative Evaluation of String Similarity Metrics for Ontology Alignment', Journal of Information and Computational Science, vol. 12, no. 3, pp. 957–964, Feb. 2015.

[13] S. Melnik, H. Garcia-Molina, and E. Rahm, 'Similarity flooding: A versatile graph matching algorithm and its application to schema matching', in Data Engineering, 2002. Proceedings. 18th International Conference on, pp. 117–128, 2002.

[14] M. Lesk, 'Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone', in Proceedings of the 5th Annual International Conference on Systems Documentation, New York, NY, USA, pp. 24–26, 1986.

[15] T. Joachims, Learning to Classify Text Using Support Vector Machines. Springer Science & Business Media, 2002.

[16] S. Banerjee, 'Adapting the Lesk algorithm for word sense disambiguation to WordNet', University of Minnesota Duluth, 2002.

[17] Z. Wu and M. Palmer, 'Verbs semantics and lexical selection', in Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp. 133–138, 1994.

[18] C. Fellbaum, WordNet: An Electronic Lexical Database. MIT Press, 1998.

[19] A. Budanitsky and G. Hirst, 'Evaluating wordnet-based measures of lexical semantic relatedness', Computational Linguistics, vol. 32, no. 1, pp. 13–47, 2006.

[20] P. Resnik and others, 'Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language', J. Artif. Intell. Res.(JAIR), vol. 11, pp. 95–130, 1999.

[21] R. Tous and J. Delgado, 'A vector space model for semantic similarity calculation and OWL ontology alignment', in Database and Expert Systems Applications, pp. 307–316, 2006.

[22] J. Ramos, 'Using tf-idf to determine word relevance in document queries', in Proceedings of the first instructional conference on machine learning, 2003.

[23] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, 'The agreementmakerlight ontology matching system', in On the Move to Meaningful Internet Systems: OTM 2013 Conferences, pp. 527–541, 2013.

[24] D. Faria et al., 'OAEI 2016 results of AML.', in ISWC-2016, pp. 138–145, 2016.

[25] V. Geroimenko, Dictionary of XML Technologies and the Semantic Web. Springer Science & Business Media, 2012.

[26] W. Sunna and I. F. Cruz, 'Using the AgreementMaker to Align Ontologies for the OAEI Campaign 2007.', in OM, 2007.

[27] Y. Zhang, H. Jin, L. Pan, and J.-Z. Li, 'RiMOM results for OAEI 2016.', in ISWC-2016, Japan, pp. 210–216, 2016.

[28] J. da Silva, F. A. Baião, and K. Revoredo, 'ALIN results for OAEI 2017.', in ISWC-2017, pp. 130–137, 2017.

[29] E. Jiménez-Ruiz, B. C. Grau, and V. V. Cross, 'LogMap family participation in the OAEI 2017.', in ISWC-2017, pp. 185–189, 2017.

[30] E. Jiménez-Ruiz and B. C. Grau, 'Logmap: Logic-based and scalable ontology matching', in The Semantic Web–ISWC 2011, Springer, pp. 273–288, 2011.

[31] W. E. DJEDDI, M. T. KHADIR, and S. B. YAHIA, 'XMap: Results for OAEI 2016', in ISWC-2016, Japan, 2016.

[32] W. E. DJEDDI, M. T. KHADIR, and S. B. YAHIA, 'XMap: Results for OAEI 2017', in ISWC-2017, 2017.

[33] N. L. of Medicine (U.S.), UMLS Knowledge Sources: Metathesaurus, Semantic Network, [and] SPECIALIST Lexicon. U.S. Department of Health and Human Services, National Institutes of Health, National Library of Medicine, 2003.

[34] M. Bennett, J. Fried, M. Kehoe, and N. Voskresenskaya, Professional Microsoft Search: FAST Search, SharePoint Search, and Search Server. John Wiley & Sons, 2010.

[35] S. Hertling, 'WikiV3 results for OAEI 2017', in ISWC-2017, Austria, 2017.

[36] D. J. Barrett, MediaWiki: Wikipedia and Beyond. O'Reilly Media, Inc., 2008.

[37] P. Ayers, C. Matthews, and B. Yates, How Wikipedia Works: And how You Can be a Part of it. No Starch Press, 2008.

[38] A. Laadhar, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste, and F. Gargouri, 'POMap results for OAEI 2017', in ISWC-2017, Austria, 2017.

[39] M. Majid, 'Simulated annealing based ontology matching', in ISWC-2017, Austria, 2017.

[40] S. G. M. Khadir, 'ONTMAT: Results for OAEI 2017', in ISWC-2017, Austria, 2017.

[41] M. KACHROUDI, G. DIALLO, and S. B. YAHIA, 'OAEI 2017 results of KEPLER', in ISWC-2017, Austria, 2017.

[42] A. N. Tigrine, Z. Bellahsene, and K. Todorov, 'Light-Weight Cross-Lingual Ontology Matching with LYAM++', in On the Move to Meaningful Internet Systems: OTM 2015 Conferences, Springer International Publishing, pp. 527–544, 2015.

[43] A. Annane, Z. Bellahsene, F. Azouaou, and C. Jonquet, 'YAM-BIO: Results for OAEI 2017', in ISWC-2017, Austria, 2017.

[44] D. Ngo and Z. Bellahsene, 'YAM++: A multi-strategy based approach for ontology matching task', in International Conference on Knowledge Engineering and Knowledge Management, pp. 421–425, 2012.

[45] M. Gulić, B. Vrdoljak, and M. Banek, 'CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment', Web Semantics: Science, Services and Agents on the World Wide Web, vol. 41, pp. 50–71, Dec. 2016.

[46] P. Wang and W. Wang, ‘Lily results for OAEI 2016.’, in ISWC-2016, Japan, pp. 178–184, 2016.

[47] A. Khiat, ‘CroLOM results for OAEI 2017’, in ISWC-2017, Austria, 2017.

[48] A. Khiat, ‘CroLOM results for OAEI 2016’, in ISWC-2016, Japan, 2016.

[49] F. Muhammad, ‘DKP-AOM: results for OAEI 2016’, in ISWC-2016, Japan, pp. 161–165, 2016.

[50] M. Rybinski, M. del M. R. García, J. García-Nieto, and J. F. A. Montes, ‘DisMatch results for OAEI 2016.’, in ISWC-2016, Japan, pp. 161–165, 2016.

[51] M. Zhao and S. Zhang, ‘FCA-Map results for OAEI 2016.’, in ISWC-2016, Japan, pp. 172–177, 2016.

[52] A. Berro, I. Megdiche, and O. Teste, ‘A Linear Program for Holistic Matching: Assessment on Schema Matching Benchmark’, in Database and Expert Systems Applications, pp. 383–398, 2015.

[53] A. Billionnet, Optimisation discrète: de la modélisation à la résolution par des logiciels de programmation mathématique. Dunod, 2007.

[54] M. A. R. Garcıa, G. V. Gkoutos, P. N. Schofield, and R. Hoehndorf, ‘Integrating phenotype ontologies with PhenomeNET’, Ontology Matching, p. 201, 2016.

[55] A. Barton, A. Rosier, A. Burgun, and J.-F. Ethier, ‘The Cardiovascular Disease Ontology.’, in Formal Ontology in Information Systems, 2014.

[56] B. Mohamed, E. A. Rachid, and F. Mohamed, ‘Word Boundary Detection in Tifinagh using MaxEnt and n-gram algorithms’, in ACIT 2016, Beni-Mellal, 2016.

[57] J. C. Reynar and A. Ratnaparkhi, ‘A maximum entropy approach to identifying sentence boundaries’, in Proceedings of the fifth conference on Applied natural language processing, 1997.

[58] R. M. Reese, Natural Language Processing with Java. Packt Publishing Ltd, 2015.

[59] E. Kwiatek, Contrastive Analysis of English and Polish Surveying Terminology. Cambridge Scholars Publishing, 2014.

[60] S. Boukil, F. E. Adnani, A. E. E. Moutaouakkil, L. Cherrat, and M. Ezziyyani, ‘Arabic Stemming Techniques as Feature Extraction Applied in Arabic Text Classification’, in Advanced Information Technology, Services and Systems, 2017.

[61] R. Navigli, ‘A quick tour of babelnet 1.1’, in Computational Linguistics and Intelligent Text Processing, Springer, 2013.

[62] X. Xue, Y. Wang, and W. Hao, ‘Optimizing Ontology Alignments by using NSGA-II’, International Arab Journal of Information Technology (IAJIT), vol. 12, no. 2, 2015.

[63] F. Jauro, S. B. Junaidu, and S. E. Abdullahi, ‘Falcon-AO++: An Improved Ontology Alignment System’, International Journal of Computer Applications, vol. 94, no. 2, pp. 1–7, 2014.

[64] lti Mathur, N. Joshi, H. Darbari, and A. Kumar, ‘Shiva++: An Enhanced Graph based Ontology Matcher’, International Journal of Computer Applications, vol. 92, no. 16, Apr. 2014.

[65] L. Depecker, Entre signe et concept: éléments de terminologie générale. Presses Sorbonne Nouvelle, 2002.

[66] G. L. Agrawal and H. Gupta, ‘Optimization of C4. 5 Decision Tree Algorithm for Data Mining Application’, International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 3, pp. 341–345, 2013.

[67] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ‘A fast and elitist multiobjective genetic algorithm: NSGA-II’, IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[68] J. Stender, E. Hillebrand, and J. Kingdon, Genetic Algorithms in Optimisation, Simulation and Modelling. IOS Press, 1994.

[69] L. D. Chambers, Practical Handbook of Genetic Algorithms: Complex Coding Systems. CRC Press, 1998.

[70] U. Marjit, ‘Aggregated Similarity Optimization in Ontology Alignment through Multiobjective Particle Swarm Optimization’, IJARCCE, pp. 258–263, Feb. 2015.