

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

UNIVERSITE SAAD DAHLEB DE BLIDA
Faculté des sciences
Département d'informatique



Mémoire de Fin d'Etudes
Pour l'obtention
Du Diplôme de Master en Informatique

Option : Ingénierie Des Logiciels

THÈME

Traitement des concepts composés pour
l'alignement des ontologies

Présenté par :

Hechchad Nesrine Belkhir Meroua

Soutenu le : 14/07/2021, devant le jury composé de :

- Présidente : Mme. MANCER
- Examinatrice : Mme. FERDI
- Promotrice : Mme. FAREH
- Co-promoteur : Mr. RIALI

Année Universitaire : 2020/2021

Remerciements

Nous remercions ALLAH de nous avoir donné la santé et le courage afin de pouvoir réussir ce travail. Ce travail est l'aboutissement d'un long cheminement au cours duquel nous avons bénéficié d'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement merci.

En premier lieu, nous adressons nos remerciements, les plus vifs, notre profonde gratitude et notre respects à notre promotrice Mme. FAREH d'avoir accepté de nous encadrer, pour les conseils et orientations tant précieux qu'elle nous a prodigué durant ce travail. Nous la remercions aussi vivement pour la démarche fructueuse qu'elle a adoptée pour nous introduire dans ce fabuleux domaine

Nous remercions aussi notre Co-promoteur M. RIALI Ishak qui nous a guidé dans notre travail et nous a aidé à trouver des solutions pour avancer. Nous présentons tous nos respects et nos remerciements aux membres du jury qui ont accepté d'évaluer ce travail.

Nos remerciements les plus sincères à toutes les personnes qui auront contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire

Dédicace

Je dédie ce modeste travail à :

*A mes parents, aucun hommage ne pourrait être à la hauteur de l'amour
dont ils ne cessent de me combler. Que dieu leur procure bonne santé et
longue vie*

*A celui qui m'a soutenue tout au long de ce projet : mon fiancé, et bien sur A
mes frères Amine , Youcef , et mon petit prince Hani*

*A la femme la plus chère à mes yeux : ma grande mère , A mon grand père
A toute ma famille, ainsi qu'à mes deux nièces Célia et Mellissa*

A mon binôme Belkhir Meroua

A mon chère amie Kheffi Zahra

*Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit
possible.*

je vous dis merci.

Nesrine

Dédicace

*Je dédie ce modeste travail à la femme qui m'a mise au monde.
Qui m'a conduit par la force avec patience et sans impuissance.
A Qui a été avec moi dans les difficultés des temps, le temps de ma joie et de
ma douleur.*

*Grâce à elle et à ses encouragements je suis arrivée à ce stade
A ma chère mère, la femme la plus adorable de ma vie.
A ce qui m'ont aidés et soutenu dans tous mes études mes chères sœurs :
Amel et Sabrina.*

*A ma chère sœur Nassima qui a durement travaillé avec moi et d'avoir
m'encouragé pour réaliser ce simple travail.*

A mon binôme : Hechchad Nesrine.

A mes chères amies : Limani Fatma et Achour Hafsa.

*A mes chers amis qui m'ont vraiment aidés durant mes années d'étude
universitaire : Mohamed Chakib Ibnaissa, Ismail Abed El hafid et Bellil
Mounir.*

Je les souhaite tout le bonheur, la prospérité et la réussite dans toute la vie.

Meroua

Résumé

De nombreuses méthodes d'alignement dédiées aux ontologies ont vu le jour cette dernière décennie. L'alignement est une approche intéressante parce qu'il consiste à découvrir la correspondance sémantique entre les éléments similaires de différentes ontologies.

Cependant, dans la littérature actuelle, il existe très peu de méthodes d'alignement qui visent à traiter les concepts composés pour l'alignement des ontologies. Dans les méthodes existantes, le traitement des concepts composés s'effectue de la même manière que celui des concepts simples, et cela diminue la performance du système.

Pour contribuer à résoudre ce problème, nous proposons une approche pour effectuer un alignement entre deux ontologies. Cette approche utilise un algorithme de décomposition « Treat-ConceptComposé ». Notre algorithme a l'avantage de ne garder aucun concept composé, mais traiter tous les termes de ce concept, en gardant la sémantique des concepts composés.

L'approche combine les mesures syntaxiques, lexicales et structurelles. Ceci afin de définir une mesure de similarité globale calculée en combinant ces mesures de similarité. Nos résultats préliminaires ont montré que cette approche est efficace et prometteuse.

Mots clés :

Ontologie, alignement d'ontologies, mesures de similarité, concepts composés, sémantique.

Abstract

Many methods alignment dedicated to the ontologies have seen the day in the last decade. Alignment is an interesting approach because it consists to discover semantic correspondence between similar elements of different ontologies.

However, in actual literature, it exists very little of alignment methods who aim to treat compound concepts of alignment of ontologies in existing methods, the treatment of compound concepts to perform in the same manner than that of simple concept, and that decrease system performance.

To contribute to solve this problem, we propose an approach to make an alignment between two ontologies. This approach uses a decomposition algorithm « TreatConceptComposé ». Our algorithm has the advantage not to keep any compound concept but treat all the terms of concept, keeping the semantic of compound concept.

The approach combines syntactic measures, lexical and structural. This is in order to define a global measure of similarities calculated by combining these similarity measures. Our preliminary results showed that this approach is effective and promising.

Keywords :

ontology, ontology alignment, similarity measure, compound concept, semantic.

Abstract

Table des matières

Table des figures	11
Liste des tableaux	12
Introduction générale	14
Contexte de travail	14
Problématique	14
Objectifs du travail	15
Organisation du mémoire	16
1 Alignement d'ontologie	17
1.1 Introduction	17
1.2 Ontologies	17
1.3 Historique des ontologies	18
1.4 Composants des ontologies	19
1.5 Langages d'ontologies	20
1.5.1 XML et XML schéma	20
1.5.2 RDF et RDF schéma	20
1.5.3 OWL (Ontology Web Language)	21
1.6 Domaines d'applications des ontologies	21
1.6.1 Système d'information	21
1.6.2 Web sémantique	22
1.7 Alignement des ontologies	22
1.7.1 Définition	22
1.7.2 Les objectifs de l'alignement des ontologies	22
1.7.3 Domaines d'application de l'alignement	22
1.7.4 Etapes du processus d'alignement	23
1.8 Les concepts composés	24
1.8.1 Définition	24

1.8.2	Exemples des concepts composés	24
1.8.3	Problèmes d'alignement dans les concepts composés	25
1.9	Conclusion	26
2	Etat de l'art sur les méthodes existantes d'alignement d'ontologies	27
2.1	Introduction	27
2.2	La similarité	27
2.2.1	Définition	28
2.2.2	Dissimilarité	28
2.2.3	Distance	28
2.3	Classification des mesures de similarité	29
2.3.1	Mesures simples	29
2.3.1.1	Mesures terminologiques	29
2.3.1.2	Mesures structurelles	30
2.3.1.3	Mesures sémantiques	31
2.3.1.4	Mesures extensionnelles	31
2.3.2	Mesures combinées	32
2.3.2.1	Mesures séquentielles (hybrides)	32
2.3.2.2	Mesures parallèles (composites)	32
2.4	Méthodes d'alignement des ontologies	32
2.4.1	Travail de (Gracia et Asooja,2013)	32
2.4.2	Travail de (Li et al., 2015)	33
2.4.3	Travail de (Hitti et al., 2016)	33
2.4.4	Travail de (Shvaiko et al., 2018)	33
2.4.5	Travail de (Hertling et al., 2018)	34
2.4.6	Analyse	34
2.4.6.1	Comparaison entre les différentes méthodes	34
2.4.6.2	Discussion	36
2.5	Conclusion	37
3	Conception du système d'alignement	38
3.1	Introduction	38
3.2	Schéma global du système	38
3.3	Description générale de notre approche	39
3.4	Description détaillée de l'approche proposée	40
3.4.1	Extraction des concepts	40
3.4.2	L'alignement d'ontologies	41
3.4.3	Agrégation des résultats	51
3.5	Conclusion	52

4	Implémentation et test du système	54
4.1	Introduction	54
4.2	Outils de développement de notre système	54
4.2.1	Python	54
4.2.2	Tkinter	55
4.2.3	NumPy	55
4.2.4	Cython	55
4.2.5	Owlready2	56
4.2.6	NLTK	56
4.2.7	WordNet	56
4.2.8	Protégé	56
4.2.9	WebVOWL	57
4.3	Présentation de notre système	57
4.4	Test du système	63
4.4.1	Résultats expérimentaux et discussion	64
4.5	Validation	69
4.6	Conclusion	73
	Conclusion et perspectives	74
	Bibliographie	76

Table des figures

1.1	Le triangle sémantique[1]	19
1.2	Les trois dimensions de l’alignement[2]	24
1.3	Extrait de l’ontologie «humain.owl»	25
2.1	Mesures de calcul de similarités [3]	29
3.1	Processus d’alignement des concepts composés	39
3.2	Schéma d’extraction d’un ensemble de concepts de l’ontologie « conference.owl » .	41
3.3	Une partie des résultats de la vérification des types de concepts	43
3.4	Résultats de la décomposition des concepts composés	44
3.5	Le fichier XML	52
4.1	Interface d’accueil	57
4.2	Interface choix d’emplacement d’ontologies	58
4.3	Interface de traitement des concepts composés	59
4.4	Interface des synonymes des concepts composés	60
4.5	Interface de calcul de la similarité des concepts simples	61
4.6	Interface de calcul de la similarité des concepts composés	62
4.7	Le fichier exporté « RésultatXML »	63

Liste des tableaux

2.1	Comparaison des méthodes d'alignement	36
4.1	Ontologies de test	64
4.2	Les résultats obtenus entre les concepts simples des deux ontologies	64
4.3	Les résultats obtenus entre les concepts composés des deux ontologies	65
4.4	Les résultats obtenus entre les concepts simples et les concepts composés	66
4.5	Les meilleures mesures choisies entre les concepts simples	68
4.6	Les meilleures mesures choisies entre les concepts composés	68
4.7	Les meilleures mesures choisies entre les concepts simples et les concepts composés	69
4.8	Les valeurs des poids	69
4.9	Extrait des correspondances manuelles et les correspondances de notre système	71
4.10	Les résultats des rappels et précisions	72

Introduction générale

Contexte de travail

Les ontologies sont au cœur du web sémantique, elles permettent de représenter les connaissances du web, elles sont proposées comme élément central du web sémantique. Ces dernières permettent de décrire la structure et la sémantique des données contenues dans un document web. Elles assurent aussi l'organisation des informations sous forme d'une taxonomie de concept et de relation entre ces derniers. Cependant, il n'existe pas d'ontologie universelle partagée, adoptée par tous les utilisateurs d'un domaine donné. Les problématiques et les tentatives d'amélioration de l'interopérabilité du système comptent donc sur la réconciliation des différentes ontologies utilisées dans un domaine par les différents systèmes. Cette réconciliation consiste à trouver les liens de correspondance entre ces ontologies, on parle donc de l'alignement des ontologie.

L'alignement des ontologies cherche des correspondances sémantiques entre les différents systèmes d'information et joue un rôle important afin de permettre l'interopérabilité des systèmes. Cet ensemble de correspondances ou encore alignement peut être par la suite utilisé pour fusionner les ontologies, créer une troisième ontologie à partir des entités des deux ontologies en entrée. . . etc.

Les ontologies se développent à toute vitesse dans différents domaines d'application réels, tel que le domaine médical, elles contiennent beaucoup de concepts représentés par plusieurs termes, ce qu'on appelle des concepts composés.

Le traitement des concepts composés, au niveau d'alignement des ontologies, est donc nécessaire, pour augmenter la précision du système, en choisissant une meilleure combinaison des mesures de similarité.

Problématique

L'alignement des ontologies se base sur l'utilisation des mesures de similarité, pour la détection des correspondances entre les différents concepts des deux ontologies. Le calcul de la

similarité sémantique entre les mots est important dans la recherche d'information, l'acquisition de connaissances et de nombreux autres domaines. Les études existantes traitent principalement des concepts simples composés d'un seul terme, et elles ignorent les concepts composés de plusieurs termes, et les traitent comme des concepts simples, en négligeant les caractéristiques de la composition, ce qui va réduire la précision du système.

Parmi les mesures utilisées dans l'alignement, il existe la mesure lexicale qui est importante dans le processus d'alignement, vu qu'elle détecte la correspondance selon la définition lexicale des concepts, ces mesures nécessitent l'accès et l'utilisation d'une ressource externe telles que les terminologies et les dictionnaires.

Le problème c'est que, les ressources lexicales reconnaissent que des concepts simples c.a.d. composés d'un seul mot, mais elles ne reconnaissent pas les concepts composés de plusieurs mots, ce qu'il influence fortement la précision du système et la qualité des correspondances extraits.

Pour cette raison, nous allons consacrer notre travail à répondre aux points suivants :

- Comment effectuer l'alignement ?
- Comment traiter les concepts composés, en gardant la sémantique ?
- Comment traiter les concepts composés, pour que WordNet les reconnaissent ?
- Comment utiliser et adapter les mesures de similarité pour l'alignement des ontologies selon nos besoins ?

Objectifs du travail

L'objectif de ce projet consiste à concevoir et réaliser un système d'alignement des ontologies qui traite des concepts composés, en effet, de plus en plus, des domaines représentent leur connaissances à travers des ontologies réelles, contenant beaucoup de concepts composés de deux ou trois mots, d'où, ce problème a une très grande importance.

Une solution possible pour résoudre ce problème est d'essayer de traiter les concepts composés avant de les utiliser dans le processus d'alignement, et pour cela nous proposons de décomposer ces concepts en plusieurs mots afin que WordNet les reconnaissent.

Dans les parties suivantes, nous allons expliquer en détail les étapes effectuées pour atteindre notre objectif et pour obtenir de bons résultats. Nous allons exprimer comment :

- Extraire les concepts d'une ontologie.
- Traiter les concepts extraits.
- Utiliser et adapter les mesures de similarité sur des concepts composés.

Organisation du mémoire

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

- **Chapitre 1 :Alignement d'ontologie**

Dans le premier chapitre nous avons présenté une étude sur les ontologies, nous avons abordé les définitions, les composants, les langages d'ontologies et le domaine d'application, nous avons mentionné aussi la notion des concepts composés et l'alignement d'ontologies.

- **Chapitre 2 :Etat de l'art sur les méthodes existantes d'alignement d'ontologies**

Dans ce chapitre, nous avons enchaîné avec un état de l'art sur les méthodes existantes d'alignement d'ontologies et de définir quelques travaux existants qui traitent les concepts composés afin d'effectuer une comparaison entre eux.

- **Chapitre 3 : Conception du système d'alignement**

Ce chapitre, consiste à concevoir une solution pour l'alignement des ontologies, en prenant en considération les concepts composés. Nous présentons dans ce chapitre les différentes mesures utilisées afin de calculer la similarité globale entre les concepts.

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

Dans ce dernier chapitre, nous avons présenté l'environnement de développement de notre solution, les outils utilisés, et une présentation visuelle de l'implémentation de notre système. Et nous concluons notre travail avec une évaluation.

Chapitre 1

Alignement d'ontologie

1.1 Introduction

L'exploitation de connaissance en informatique a pour objectif de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue entre le système et les utilisateurs. Alors, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leur est associée, afin qu'une communication efficace soit possible.

Actuellement, la connaissance visée par les ontologies est un sujet de recherche populaire dans diverses communautés tel que l'ingénierie des connaissances, la recherche d'information, le traitement du langage naturel, les systèmes d'information coopératifs, l'intégration intelligente d'information et la gestion des connaissances. Elles offrent une connaissance partagée sur un domaine qui peut être échangée entre des personnes et des systèmes hétérogènes. Elles ont été définies en intelligence artificielle afin de faciliter le partage des connaissances et leur réutilisation. Elles répondent aussi aux problèmes de représentation et de manipulation des connaissances

Dans ce chapitre, en premier lieu nous avons élaboré la notion des ontologies, en seconde lieu nous avons montré l'alignement des ontologies et en dernier nous avons parlé à propos de l'alignement dans les concepts composés.

1.2 Ontologies

Les ontologies sont un ensemble des objets reconnus comme existant dans le domaine, construire une ontologie c'est aussi décider de la manière d'être et d'exister des objets.

Neeches et ses collègues furent les premiers à proposer une définition dans le cadre de l'intelligence artificielle à savoir : « Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ». [4]

Aussi, Gruber a proposé une autre définition qui est : «une ontologie est une spécification explicite d'une conceptualisation».[5]

Selon Borst « Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée » avec la signification des termes suivants [6]

Formelle : réfère au fait qu'une ontologie doit être compréhensible par la machine, c'est-à-dire que cette dernière doit être capable d'interpréter la sémantique de l'information fournie.

Explicite : signifie que le type des concepts utilisés et les contraintes sur leur utilisation doivent être explicitement définis.

Conceptualisation : se réfère à un modèle abstrait de certains phénomènes dans le monde qui identifie les concepts appropriés de ce phénomène.

Partagée : indique que l'ontologie supporte la connaissance consensuelle, et elle n'est pas restreinte à certains individus mais est partagée entre un groupe de personnes.

Ainsi une ontologie représente un modèle conceptuel sur lequel il est possible de développer des systèmes de connaissances qui soient partageable et réutilisables permettant ainsi, l'interopérabilité entre plusieurs sources d'information et de connaissances.

1.3 Historique des ontologies

Les ontologies sont apparues au début des années 90 dans la communauté Ingénierie des connaissances, dans le cadre des démarches d'acquisition des connaissances pour les systèmes à base de connaissances (SBC). Faisant suite aux systèmes experts qui séparaient une base de connaissance, et un moteur d'inférence, les SBC proposaient alors de spécifier, d'un côté, des connaissances du domaine modélisé et de l'autre des connaissances de raisonnement décrivant les règles heuristiques d'utilisation de ces connaissances du domaine.

L'idée de cette séparation modulaire était de construire mieux et plus rapidement des SBC en réutilisant le plus possible des composants génériques, que ce soit au niveau du raisonnement ou des connaissances du domaine.

L'ontologie est une notion philosophique, et on considérait que l'étude de l'ontologie était une partie de la métaphysique, qui s'intéresse à l'étude des propriétés de l'être, et par extension de l'existence. John McCarthy a introduit l'ontologie en intelligence artificielle en 1980, par le principe que les concepteurs des systèmes intelligents fondés sur la logique devraient d'abord établir une base de données complète de ce qui existe, et ensuite utiliser ces données. Les

ontologies sont apparues en intelligence artificielle, comme réponses aux problématiques de représentation et de manipulation des connaissances au sein des systèmes informatiques.

1.4 Composants des ontologies

Les ontologies rassemblent les connaissances propres à un domaine donné. En représentation des connaissances, ces ontologies existent sous la forme de concepts, relations, fonctions, axiomes et instances. Nous détaillons ci-dessous ces différents composants de l'ontologie :

- **Les concepts**

Appelés classes de l'ontologie, le concept représente un objet, une notion ou une idée. Il peut décrire une fonction, un processus de raisonnement, une stratégie, etc.

On désigne un concept par un label (terme), ce concept est défini par ses différentes relations avec les autres concepts de l'ontologie, ses attributs et les contraintes qui lui sont associées.

Un concept peut être divisé en trois parties : un terme (ou plusieurs), une notion et un ensemble d'objets.

- ▶ Le terme est un élément lexical qui permet d'exprimer le concept en langue naturelle, il peut admettre des synonymes.
- ▶ La notion également appelée intention du concept, contient la sémantique du concept, exprimée en terme de propriété, attributs et de contrainte.
- ▶ L'ensemble d'objets appelé extension du concept, regroupe les objets manipulés à travers le concept ; ces objets sont appelés instances du concept. Par exemple, le terme « table » renvoie à la fois à la notion de table comme objet de type « meuble » possédant un « plateau » et des « pieds », et à l'ensemble des objets de ce type.

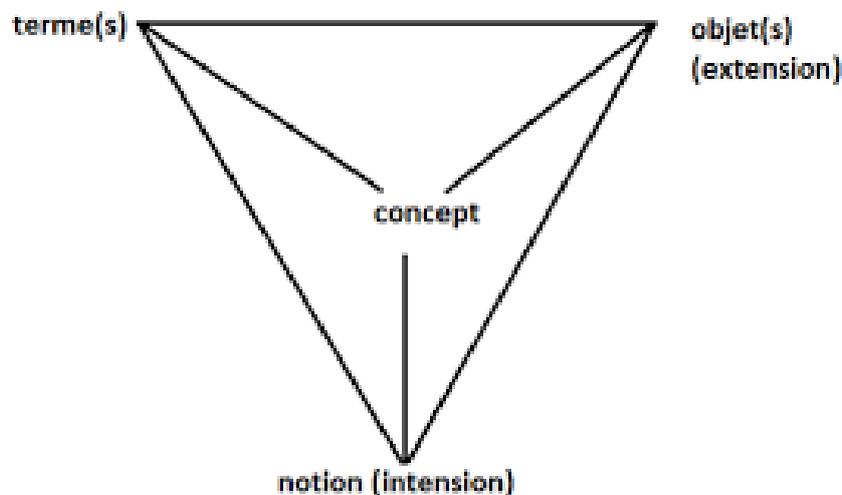


FIGURE 1.1 – Le triangle sémantique[1]

- **Les relations**

Traduisent les associations entre les concepts présents dans le segment analysé de la réalité, ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres

- **Les fonctions**

Constituent des cas particuliers de relations, dans laquelle un élément de la relation, (le nième) est défini en fonction des N-1 éléments précédents.

- **Les axiomes**

Constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie.

- **Les instances**

Les instances représentent les éléments des concepts et des relations dans un domaine donné

1.5 Langages d'ontologies

Une des principales décisions à prendre dans le processus de développement d'ontologie consiste à choisir le langage dans lequel l'ontologie sera exprimée et utilisée

1.5.1 XML et XML schéma

XML est un langage de structuration et de balisage développé par w3c comme un standard. Il a pour vocation d'être un format d'échange de documents entre diverses applications[7]. Il constitue le premier langage de description d'ontologies sur le web. Il est destiné pour l'échange des données sur le web, rend possible la description des concepts et leurs hiérarchisations sous forme de taxonomies à l'aide des balises.

Ces lacunes ont motivé le développement d'autres langages comme XML Schéma qui est un groupe de travail du W3C qui s'occupe des extensions de XML, il offre des grammaires qui permettent de spécifier des restrictions sur la structure des documents

1.5.2 RDF et RDF schéma

RDF (Resource Description Framework) c'est le langage de base du web sémantique, publié par le consortium W3C. C'est un modèle de graphe destiné à décrire de façon formelle les ressources web et leurs métadonnées grâce à sa syntaxe basée sur XML

Un document RDF est basé sur le triplet (ressource, propriété, valeur) ou encore (sujet, prédicat, objet) de sorte que la ressource (sujet) représente l'entité décrite, peut-être une page web identifiée par son URL ou une partie de page identifiée par une balise, la propriété (prédicat) exprime la relation utilisée pour décrire la source, et la valeur (objet) peut être des ressources ou

des littéraux.

RDF-Schéma définit les relations entre ces ressources.[8]Il permet de définir des propriétés sémantiques pour les ressources par un schéma. Dans un schéma, on peut définir de nouvelles ressources comme des spécialisations d'autres ressources. Les schémas contraignent aussi le contexte d'utilisation des ressources. Avec RDFS, de nouvelles notions sémantiques apparaissent, la principale est la distinction entre une classe (concept d'une ontologie) et une instance (individu d'une ontologie).[9]

1.5.3 OWL (Ontology Web Language)

OWL est une recommandation de W3C depuis février 2004. Il s'agit d'une extension du langage RDF et RDFS. OWL est un langage basé sur la syntaxe XML, très expressif et avec un vocabulaire très riche. Il permet de définir des ontologies de domaines complexes. OWL se compose de trois sous langages d'expressivité croissante :

- **OWL-Lite**

C'est le moins expressive des sous-langages d'OWL. Il est destiné aux cas de modélisation se limitant à une classification simple de concept et de contrainte. L'avantage de ce langage est d'avoir une complexité formelle faible par rapport aux deux autres sous-langages d'OWL.

- **OWL-DL**

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

- **OWL-FULL**

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

1.6 Domaines d'applications des ontologies

1.6.1 Système d'information

L'intégration d'une ontologie dans un système d'information vise à réduire, voire éliminer, la confusion conceptuelle et terminologique à des points clefs du système, et à tendre vers une compréhension partagée pour améliorer la communication, le partage, l'interopérabilité et le degré de réutilisation possible.

1.6.2 Web sémantique

Il s'agit d'une extension du Web actuel, dans laquelle l'information se voit associée à un sens bien défini, améliorant la capacité des logiciels à traiter l'information disponible sur le Web.

1.7 Alignement des ontologies

1.7.1 Définition

L'alignement des ontologies est un nouveau paradigme dans le web sémantique où les ontologies sont utilisées pour la représentation d'information en utilisant un langage de description. L'alignement des ontologies peut être vu comme un processus qui relie sémantiquement deux vocabulaires différents, il tente de découvrir les correspondances sémantiques et les relations implicites qui peuvent exister entre deux ou plusieurs ontologies en appliquant des mesures de similarité (équivalence, disjonction, généralisation, spécialisation etc.) pour les combiner afin de dériver d'autres informations [10]

1.7.2 Les objectifs de l'alignement des ontologies

Face au problème d'hétérogénéité d'ontologies, il devient impératif de penser à des moyens pour assurer leur interopérabilité pour pouvoir les faire collaborer, c'est pour cela que l'alignement des ontologies est considéré comme une solution pour que ces ontologies soient interopérables en essayant de réduire l'hétérogénéité et le partage des connaissances avec granularité.

L'alignement joue aussi un rôle très important pour garantir l'augmentation de la précision du système lors du traitement des concepts ainsi que l'augmentation de la qualité des correspondances extraites, et permet aussi de garantir un système d'alignement très pertinent.

1.7.3 Domaines d'application de l'alignement

L'alignement est considéré comme une étape primordiale et cruciale dans plusieurs domaines métiers. L'efficacité de ces systèmes dépend directement de l'efficacité d'alignement à savoir la pertinence des résultats et sa rapidité dans l'exécution [11] parmi ces systèmes, nous citons :

- **Ingénierie d'ontologie**

Le contexte dont les utilisateurs sont confrontés face aux ontologies hétérogènes est l'ingénierie d'ontologies. Cette activité nécessite un processus d'alignement dû à la diversité des ontologies, leurs natures distribuées ainsi que leurs perpétuelles évolutions.

- **Intégration d'information**

L'intégration d'information est une problématique majeure en informatique, elle est la plus

ancienne classe d'application qui considère l'alignement comme une solution probable. Il existe deux sortes d'intégration, l'intégration des schémas qui a pour but de fusionner des schémas pour avoir un schéma global, et l'intégration des données qui permet de combiner des données appartenant à différentes sources et de fournir une vue unique de ces données.

1.7.4 Etapes du processus d'alignement

Le processus d'alignement est une tâche pendant laquelle est déterminé un alignement A entre deux ontologies O et O' , cette tâche est réalisée en utilisant un certains nombres de techniques d'alignement.

Un alignement est d'écrit comme un ensemble de cinq éléments : $\langle id, e, e', r, n \rangle$ telle que :

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

En général, l'alignement regroupe trois dimensions :

- **L'input**

Est constitué essentiellement des ontologies (décrites en OWL, RDFS... etc.) qui sont destinées à être aligner ou des instances d'une base de données ou d'une ontologie.

- **Le processus d'alignement**

Comme le montre la figure en dessous, l'alignement peut être considéré comme une fonction f , tel que à partir de deux ontologies O et O' , d'un ensemble des paramètres p et d'un ensemble de ressources externes on aboutit un alignement A' .

- **L'output**

Est un ensemble d'alignement reliant les entités qui constituent les deux ontologies

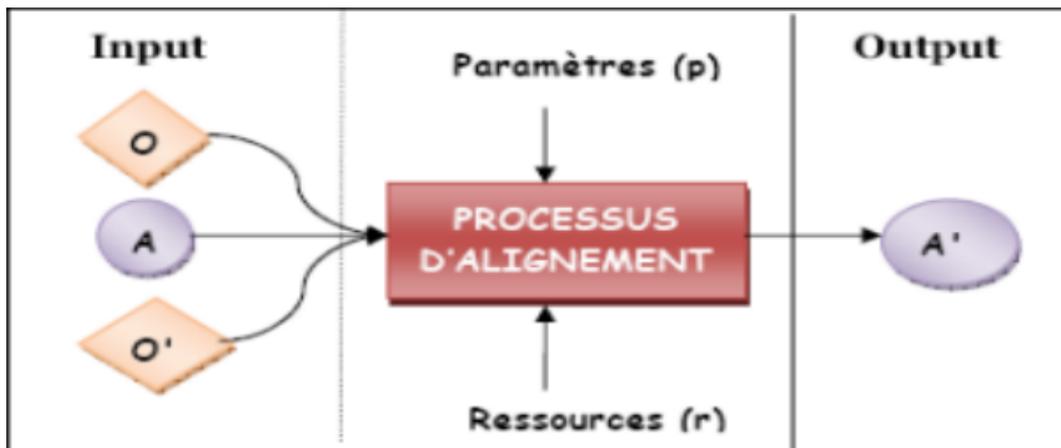


FIGURE 1.2 – Les trois dimensions de l’alignement[2]

1.8 Les concepts composés

1.8.1 Définition

Un concept composé est une combinaison des termes simples utilisés pour transmettre une seule unité de sens. Les concepts composés jouent un rôle important dans l’expression de la valeur nominale des concepts. Chaque terme de ce concept a un impact différent sur le calcul de la sémantique de ce dernier, ainsi que la précision et la performance du système.

Les concepts composés sont construits de plusieurs termes liés entre eux par des traits d’union et/ou des lettres en majuscule au milieu des concepts.

1.8.2 Exemples des concepts composés

Les concepts composés sont souvent utilisés dans le domaine médical, la biologie, l’agronomie etc.

Nous prenons l’exemple d’une ontologie médicale illustré dans la figure suivante :

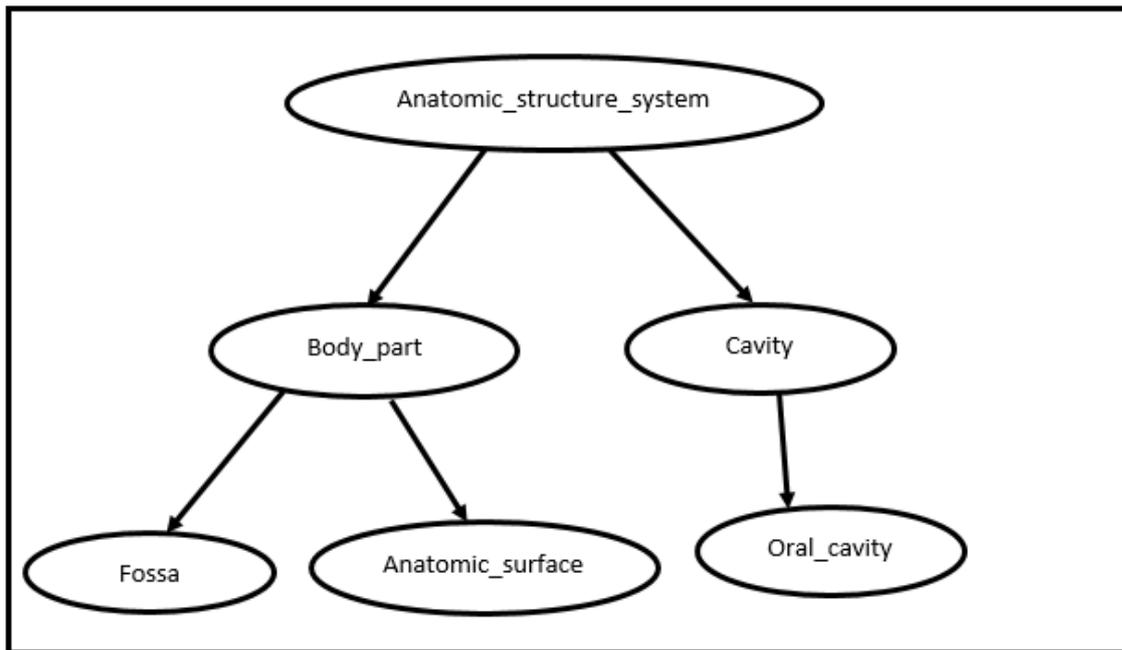


FIGURE 1.3 – Extrait de l’ontologie «humain.owl»

Cette figure représente un extrait d’une ontologie d’anatomie nommé « humain.owl », elle comporte des concepts composés ainsi que des concepts simples par exemple les concepts «Anatomic_structure_system», «Body_part», «Anatomic_surface» et «Oral_cavity» sont concepts construits de plusieurs termes donc sont des concepts dites composés.

Les concepts «Cavity» et «Fossa» sont des concepts simples.

En linguistique, aussi on peut trouver un mot composé de plusieurs éléments, chacun de ces constituants peut lui-même être un mot composé comme indique l’exemple suivant : le concept « ConferenceParticipant », est composé de deux mots : « Conference » et « Participant », mais ce concept composé n’existe pas dans la terminologie WordNet par contre les deux mots «Conference» et «Participant» existent.

1.8.3 Problèmes d’alignement dans les concepts composés

L’alignement des ontologies ayant pour objectif de permettre une utilisation conjointe de plusieurs ontologies, généralement, ces ontologies sont constituées de différents types de concepts, des concepts simples ainsi que des concepts composés, ces derniers nécessitent un traitement particulier pour le processus d’alignement, qui peut avoir quelques difficultés.

- ▶ Le processus d’alignement s’appuie sur le calcul de la similarité entre chaque couple de concept des deux ontologies, et pour cela, des mesures de similarité sont utilisées, qui ne prennent pas en considération les différents types de concepts, en considérant le concept composé comme étant un concept simple.
- ▶ Les méthodes basées sur un langage se fondent sur des techniques de traitement du langage naturel afin de trouver des associations entre les entités ou les classes. Ces méthodes exigent

l'utilisation des ressources externes tel que WordNet, ces ressources ne reconnaissent pas les concepts composés, qui va diminuer la précision des résultats.

- Les concepts composés nécessitent un traitement particulier afin de les rendre des concepts simples qui seront utilisés par la suite dans le processus d'alignement et cela augmente la durée d'exécution du processus.

1.9 Conclusion

D'après ce que nous avons présenté dans ce chapitre, il ressort que la notion d'ontologie est l'une des approches les plus efficaces dans la représentation des connaissances. Nous avons présenté en premier lieu la définition des ontologies et ses composants, ensuite nous avons exploré les principaux langages pour la représentation et la construction des ontologies. Finalement nous avons parlé à propos d'alignement des concepts composés et son objectif ainsi que ses étapes, son domaine d'application, et ses problèmes.

Le chapitre suivant se focalise sur les méthodes existantes de traitement des concepts composés dans le processus d'alignement d'ontologies.

Chapitre 2

Etat de l'art sur les méthodes existantes d'alignement d'ontologies

2.1 Introduction

Les ontologies ont été reconnues comme une composante essentielle pour le partage des connaissances. En définissant les concepts associés à des domaines particuliers, elles permettent à la fois de décrire le contenu des sources à intégrer et d'explicitier le vocabulaire utilisable dans des requêtes par des utilisateurs.

L'alignement d'ontologies cherche des correspondances sémantiques entre les différents systèmes d'information et joue un rôle important pour permettre l'interopérabilité sémantique.

Les techniques d'alignement jouent un rôle crucial dans la construction d'un lien sémantique entre les ontologies d'un même domaine. Les techniques actuelles d'alignement s'appuient en général sur des mesures calculant la similarité de couples de concepts issus des deux ontologies. Ces mesures sont pour la plupart fondées sur les caractéristiques lexicales, des labels, des concepts ou sur les caractéristiques structurelles des ontologies.

Dans ce chapitre nous allons présenter les types des mesures de similarités ainsi que les travaux existants pour le traitement des concepts composés dans l'alignement d'ontologies.

2.2 La similarité

L'identification de la similarité dans les ontologies est un concept fondamental, cette notion a été très tôt perçue comme un concept clé en intelligence artificielle, comme le souligne [12], elle est au cœur du paradigme qui énonce qu'un transfert de connaissances d'un cas connu vers un cas inconnu est possible dans la mesure où ils sont suffisamment similaires.

Dans notre contexte, on se base sur la notion de similarité sémantique, qui est également appelée la proximité sémantique. Elle est déterminée grâce à l'association à des documents, des termes ou des entités, d'une métrique basée sur la similarité de leurs significations ou de leurs contenus sémantiques. La similarité est évaluée par une mesure de similarité ou une distance.

2.2.1 Définition

La similarité est une fonction d'une paire d'entités à un nombre réel exprimant la similarité entre ces deux entités. La similarité est définie comme suit : [12]

- $\forall a, b \in O, S(a, b) \geq 0$ *(positivité)*
- $\forall a, b, c \in O, S(a, a) \geq S(b, c)$ et $S(a, a) = S(a, b) \leftrightarrow a = b$ *(autosimilarité ou maximalité)*
- $\forall a, b \in O, S(a, b) = S(b, a)$ *(symétrie)*
- $\forall a, b, c \in O, S(a, b) = S(b, c) \rightarrow S(a, b) = S(a, c)$ *(transitivité)*
- $\forall a, b \in O, S(a, b) \leq \infty$ *(finitude)*

2.2.2 Dissimilarité

La dissimilarité est parfois utilisée au lieu de la similarité. Elle est définie de manière analogue à la similarité, sauf qu'elle n'est pas transitive [11]

La dissimilarité $DS : O \times O \rightarrow \mathbb{R}$ est une fonction d'une paire d'entités à un nombre réel exprimant la dissimilarité entre ces deux entités comme suit :

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

2.2.3 Distance

La distance est une mesure utilisée aussi souvent que les mesures de similarité. Elle mesure la dissimilarité de deux entités, elle est l'inverse de la similarité, si la valeur de la fonction de similarité de deux entités est élevée, la distance entre ces entités est petite et vice-versa. [11] Elle est donc définie comme suit :

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

2.3 Classification des mesures de similarité

Une mesure de similarité est un concept de base, (Rada & al.,1989) ont suggéré que la similarité dans un réseau sémantique peut être calculée en se basant sur les liens taxonomiques « is-a ».

La figure suivante résume les différentes mesures de similarité, catégorisées selon les techniques utilisées. Qui est une synthèse des travaux présentés

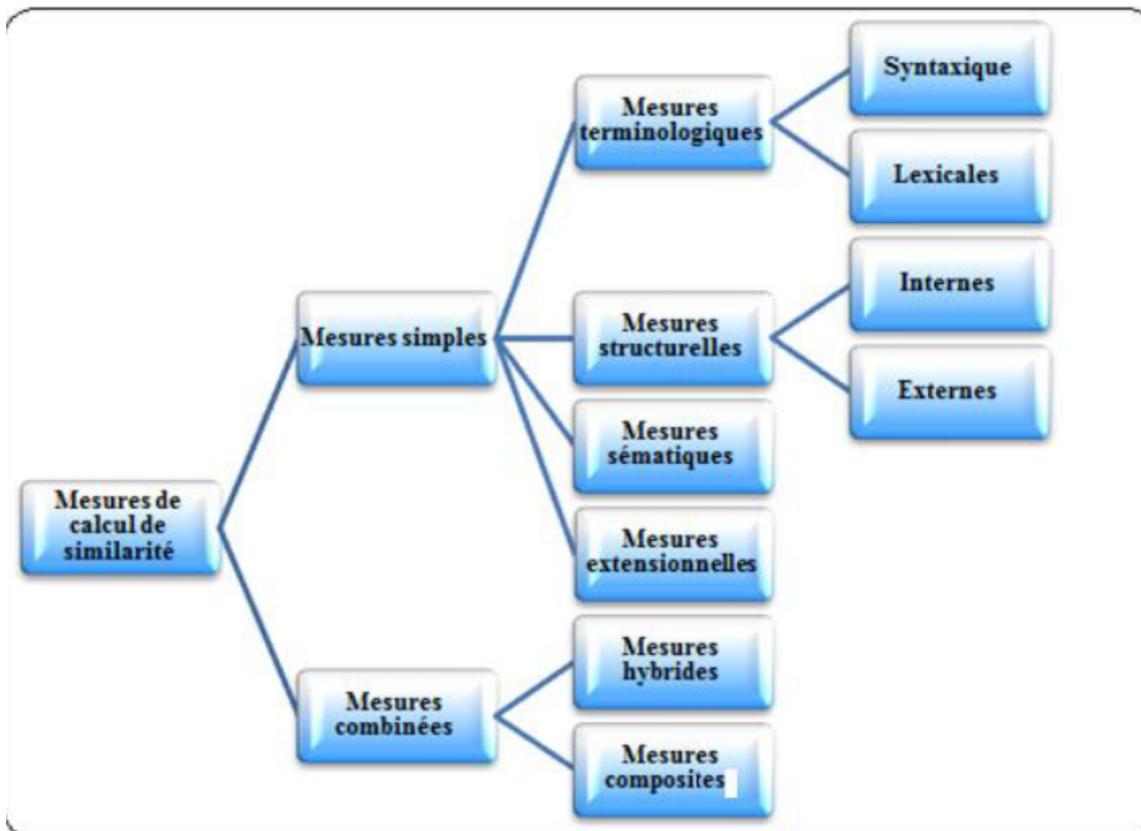


FIGURE 2.1 – Mesures de calcul de similarités [3]

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

2.3.1 Mesures simples

Les mesures simples sont identifiées par quatre types de mesures

2.3.1.1 Mesures terminologiques

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

Nous trouvons dans cette méthode deux approches : l'approche syntaxique et l'approche lexicale, appelée aussi linguistique [13]

- **L'approche lexicale ou linguistique**

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

La similarité est calculée à partir des liens sémantiques déjà existants dans les ressources externes. Nous pouvons déduire cette dernière entre ces termes en s'appuyant sur des connaissances de la langue naturelle et / ou sur des vocabulaires et des dictionnaires

- **L'approche syntaxique**

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

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

- **Distance de Jaccard**

Soit s et t deux chaînes de caractères et S et T deux ensembles des caractères de s et t respectivement, soit $P(x)$ la probabilité d'une instance aléatoire être dans l'ensemble X , la distance de Jaccard est une fonction de la dissimilarité telle que : [14]

$$DS_{Jaccard}(s, t) = 1 - \frac{P(S \cap T)}{P(S \cup T)}$$

2.3.1.2 Mesures structurelles

Ce sont les méthodes qui déduisent la similarité entre deux entités en exploitant leurs positions dans une hiérarchie et en fonction des informations structurelles. En effet, les entités sont reliées entre elles par des liens sémantiques ou syntaxiques. On peut distinguer entre deux méthodes structurelles. L'une qui n'exploite que des informations concernant des attributs d'entités (les méthodes structurelles interne) et l'autre qui considère des relations entre des entités (les méthodes structurelles externes)

- **Méthodes structurelles internes**

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

etc.).

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

- **Méthodes structurelles externes**

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

2.3.1.3 Mesures sémantiques

Les méthodes d'alignement sémantiques exploitent l'interprétation sémantique formelle dans le traitement de l'information pour calculer la similarité entre les concepts de deux ontologies. la représentation de la mesure sémantique englobe les notions, les concepts et les objets du domaine étudié

Cependant l'aspect sémantique fait le lien entre les trois dimensions du concept le terme l'intention et l'extension.

2.3.1.4 Mesures extensionnelles

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

Dans le cas où les ensembles d'instances partagent une partie commune, on peut avoir des mesures extensionnelles qui emploient des opérations de l'ensemble, telles que la distance de Hamming ou la mesure de Jaccard. Fondamentalement, la mesure de Hamming compte un nombre d'éléments différents entre deux ensembles à comparer et la mesure de Jaccard est le rapport entre l'intersection des ensembles et leur union. Ces mesures peuvent être adaptées pour construire des mesures extensionnelles. [15]

- **Distance de Hamming**

Soit S et T deux ensembles, la distance de Hamming (appelée aussi la différence symétrique) entre S et T est une fonction de la dissimilarité [14]

$$\overline{DS}_{Hamming}(S, T) = \frac{|S \cup T - S \cap T|}{|S \cup T|}$$

Ces mesures produisent la similarité de deux entités qui est en fait la similarité entre les deux ensembles de leurs instances en se basant sur la comparaison exacte des éléments dans deux ensembles.

2.3.2 Mesures combinées

Les différentes techniques citées auparavant peuvent ensuite être utilisées ensemble dans un algorithme « hybride » (deux ou plusieurs techniques dans un même algorithme) ou en un paramétrage d'algorithmes exécutés en parallèle « composite ». [16]

2.3.2.1 Mesures séquentielles (hybrides)

La méthode la plus simple pour combiner les mesures est l'utilisation séquentielle de ces dernières en choisissant un ordre d'exécution.[16]

2.3.2.2 Mesures parallèles (composites)

Une autre manière de combiner les résultats des différentes mesures (c.-à-d. les valeurs de similarité) consiste tout d'abord à lancer parallèlement plusieurs mesures, puis par la suite à combiner leurs résultats. [16]

2.4 Méthodes d'alignement des ontologies

L'alignement des ontologies est une tâche cruciale dans plusieurs domaines d'application. Le problème actuel est qu'étant donné un même domaine ou des domaines connexes, il est possible que plusieurs ontologies soient disponibles donc l'alignement des ontologies devient nécessaire.

Plusieurs techniques d'alignement basées sur des critères différents, sont actuellement proposées dans la littérature des travaux récents. Le choix d'une technique ou d'une autre ou la composition de plusieurs d'entre elles n'est pas une tâche aisée.

Ces techniques exploitent les ontologies décrites dans différents langages tel que RDF(S), OWL, XML(S) et sont souvent testées sur des ontologies qui traitent les concepts composés en considérant ces derniers comme un seul terme et cela influence sur la pertinence du système. Cependant il existe très peu d'algorithmes d'alignement qui traitent les concepts composés terme par terme, et pour cela nous allons analyser les travaux déjà existants.

2.4.1 Travail de (Gracia et Asooja,2013)

CIDER-CL (Context and Inference based aligner-Cross-Lingual) est une extension de CIDER (Trillo et al.,2007). La première étape dans CIDER-CL, consiste à extraire le contexte ontologique, jusqu'à une certaine profondeur. Cette extraction est faite en fonction du type de termes, leur synonyme, des descriptions textuelles, propriétés, domaines, les rôles, les concepts associés, etc. Afin d'ajouter des informations plus sémantiques, ce processus est enrichi par l'application d'un mécanisme d'inférence.

Dans la deuxième étape, les similarités sont calculées entre les différentes parties de contexte

ontologiques, et dans la troisième étape les différentes similarités sont composées par un ANN (Artificial Neural Networks) pour avoir le degré de similarité finale. Finalement, toutes les similarités sont mises dans une matrice.

CIDER-CL peut fonctionner en prenant deux ontologies OWL comme entrée et la sortie sera un fichier RDF. L'inconvénient de ce système est qu'il ne peut pas être utilisé avec des ontologies larges. [17].

2.4.2 Travail de (Li et al., 2015)

Ils ont proposé une nouvelle approche de calcul de similarité sémantique de concept composé appelé CCSS qui exploite les fonctionnalités de constitution de concept et qui permet d'atteindre le degré maximal de la précision

Dans CCSS, le concept composé est décomposé en en-têtes de sujet et en mots auxiliaires, et les relations entre ces deux ensembles sont utilisées pour mesurer la similarité, de plus, plusieurs sources d'information d'ontologie telles que les caractéristiques taxonomiques, la densité locale et la profondeur sont prises en compte.[18]

2.4.3 Travail de (Hitti et al., 2016)

Ils ont développé un « système Badea » conçu pour l'enrichissement semi-automatisé de lexiques ontologiques.

Ils ont utilisé une approche basée sur des modèles utilisant une ontologie composée d'un petit ensemble de paires d'antonymes, pour extraire les paires de mots d'un corpus donné avec l'antonyme sémantique, ensuite, les paires découvertes sont utilisées pour améliorer l'ontologie. Afin d'éviter ce problème et d'améliorer la précision score, (Hitti et al.,2016) ont étendu leurs travaux susmentionnés en utilisant le score LogDice et en calculant un score pour chaque motif basé sur sa co-occurrence. [19]

2.4.4 Travail de (Shvaiko et al., 2018)

Ils ont développé un système d'alignement appelé EVOCROS qui est un outil d'alignement d'ontologies basé sur une mesure de similarité composée reposant sur des techniques de similarité syntaxique et sémantique, la similarité syntaxique peut être comprise comme un score calculé sur la base d'une chaîne d'analyse (extraite d'un label de concepts), alors que la similarité sémantique est calculée en tenant compte des connaissances de base. Cette approche fonctionne en comparant la similarité calculée entre un concept d'une ontologie à un autre concept d'une ontologie différente en utilisant la distance d'édition Levenshtein comme mesure de similarité syntaxique. Les termes du concept sont traduits en un langage naturel pivot visant à utiliser les ressources externes disponibles telles que thésaurus, corpus, dictionnaires. . . etc., pour surmonter les barrières linguistiques et alphabétiques. [20]

2.4.5 Travail de (Hertling et al., 2018)

Dome (Deep Ontology MatchEr) est un matcher évolutif qui s'appuie sur de grands textes décrivant des concepts ontologiques, Il utilise l'approche doc2vec pour former une représentation vectorielle à longueur fixe des concepts, l'idée est de représenter un texte de longueur variable comme des phrases, des paragraphes et des documents, en tant que vecteur de caractéristiques de longueur fixe, ce vecteur est formé pour prédire les mots apparaissant dans le document.

Le modèle doc2vec est formé sur tous les textes disponibles dans les deux ontologies

Pour chaque concept des deux ontologies, le vecteur est formé, et les concepts qui ont les vecteurs les plus similaires par rapport à la seconde ontologie est récupérée

Chaque concept est comparé en utilisant une similarité basée sur les chaînes de caractères. [21]

2.4.6 Analyse

2.4.6.1 Comparaison entre les différentes méthodes

La comparaison des méthodes d'alignement se fait selon les critères suivants :

- **La sortie du système**

Consiste à définir le résultat de l'alignement qui est un ensemble de correspondances entre les ontologies alignées. Il peut être employé pour effectuer plusieurs autres tâches émergentes telles que la fusion, la gestion des versions d'une ontologie, le partage de la connaissance, l'intégration sémantique etc. La sortie doit être un fichier XML, OWL, RDF ... etc. pour pouvoir l'utiliser avec d'autres logiciels.

- **Concept Composé**

Un concept composé est une combinaison de deux termes ou plus qui sont reliés entre eux par un trait d'union et/ou une lettre majuscule au milieu de la chaîne.

Ce critère permet de mentionner si le travail proposé prend en considération ou néglige les concepts composés.

- **Traitement**

Les méthodes d'alignement permettent de traiter les concepts d'une ontologie et de calculer leur mesure de similarité, avec ce critère on peut préciser si ces méthodes ont traité ces concepts en analysant chaque terme de ce dernier.

- **Mesure de similarité**

La mesure de similarité est un critère permettant de faire la comparaison entre les travaux mentionnés en se basant sur les mesures terminologique (lexicale et/ou syntaxique), structurelle, sémantique et extensionnelle.

- **Aspect sémantique**

La sémantique joue aujourd'hui un rôle important dans l'alignement des ontologies, notamment il est nécessaire de disposer d'un aspect sémantique pour pouvoir effectuer des comparaisons entre concepts et, par conséquent, exploiter toutes les informations de

ces concepts. C'est pourquoi nous nous sommes intéressés à prendre en considération cet aspect comme un critère de comparaison qui permet d'identifier si le travail est basé sur cet aspect ou pas, cependant, l'aspect sémantique englobe les trois dimensions du concept : le terme, l'intention et l'extension.

Travaux	Sortie	Concept composé	Traitement des concepts	Mesure similarité	Sémantique
Travail de (Gracia et Asooja,2013)	Fichier RDF	Oui	la première étape est d'extraire le contexte ontologique jusqu'à une certaine profondeur ensuite ils calculent la similarité de chaque concept, pour les mettre dans une matrice	Terminologique et sémantique (ils ont travaillé sur tout le contexte ontologique)	Oui car ils ont pris en considération tous les composant d'un concept.
Travail de (Li et al.,2015)	-	Oui	Li & al ont proposé une approche pour calculer la similarité des concepts composé en décomposant ce concept en mots	Sémantique (selon eux ils ont le réflexe d'avoir traité l'aspect sémantique en traitant le terme seulement	Non (ils ont traité seulement les mots)
Travail de (Hitti et al.,2016)	-	Non	Dans ce travail, ils ont basé sur des paires de mots d'un corpus donné	Terminologique (lexicale)	Non (ils ont basé sur l'aspect lexicale)
Travail de (Shvaiko et al., 2018)	-	Non	Ils ont calculé la similarité en utilisant des techniques de similarité syntaxique et sémantique ensuite ils comparent la similarité calculée entre un concept d'une ontologie et un autre concept d'une autre ontologie et non pas le terme de cette ontologie	Syntaxique	Non
Travail de (Hertling & al.,2019)	Vecteur	-	Ils ont traité les concepts en utilisant l'approche doc2vec pour former un vecteur	-	Oui

TABLE 2.1 – Comparaison des méthodes d'alignement

2.4.6.2 Discussion

La table ci-dessus (Table 2.1) représente une comparaison entre chaque méthode d'alignement mentionnée auparavant. Chaque méthode est différente d'une autre selon quelques critères et y en a d'autres qui partagent les mêmes critères.

- Les sorties se diffèrent d'un travail à un autre, il ya des travaux qui possèdent en sortie un fichier RDF (Gracia et Asooja,2013), un fichier XML (Shvaiko et al.,2018) ou un vecteur (Hertling et al.,2018).

- ▶ Parmi les travaux mentionnés, Les travaux de (Gracia et Asooja,2013), et de (Li et al.,2015) s'adaptent aux concepts composés vu qu'ils ont traité les concepts terme par terme contrairement aux autres qui ne prennent pas en considération chaque terme.
- ▶ Pour la découverte de la similarité entre les concepts d'une ontologies, la technique de mesure de similarité se change d'un travail à un autre, telle que dans le travail de (Gracia et Asooja,2013) ont utilisé la mesure terminologique et sémantique en traitant tous les composants du concept,(Li et al.,2015)) sont basés sur l'aspect sémantique selon leur façon de voir l'aspect sémantique, (Shvaiko et al., 2018) utilisent la mesure syntaxique et sémantique en traitant les termes et leur signification, contrairement aux (Hertling et al.,2018) et (Hitti et al.,2016) ont focalisé sur la mesure terminologique.
- ▶ La plupart des travaux mentionnés auparavant ont le réflexe d'avoir traité l'aspect sémantique selon leur propre définition de cet aspect, mais en réalité Ils ont visé seulement sur les termes en négligeant les autres composant du concept (l'intention et l'extension) seulement (Gracia et Asooja,2013) ont pris en considération tous les composant d'un concept.

2.5 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur les méthodes existantes d'alignement d'ontologies, nous avons analysé ces méthodes en comparant les différents travaux selon des critères de comparaison bien précis telle que les mesures de similarité, le traitement des concepts composés, l'aspect sémantique etc.

L'alignement des ontologies représente un grand intérêt pour plusieurs domaines d'applications qui manipulent des connaissances hétérogènes.

Dans le chapitre suivant, nous présentons la conception de notre système, nous allons présenter en premier lieu une architecture générale pour le fonctionnement de notre application ainsi que la méthode utilisée pour calculer la similarité entre les entités d'ontologies.

Chapitre 3

Conception du système d'alignement

3.1 Introduction

La conception d'un système est une étape cruciale pour l'évaluation des besoins d'un utilisateur. L'objectif de ce chapitre est de présenter une solution à notre problématique et de proposer un système d'alignement d'ontologies.

L'alignement d'ontologies est une tâche complexe reposant sur la définition des mesures de similarités. Le système que nous avons proposé est un système d'alignement qui permet de traiter les concepts composés dans l'alignement des deux ontologies, et calculer la similarité en combinant plusieurs méthodes pour avoir les meilleurs résultats.

Dans ce chapitre, nous commençons d'abord par présenter le schéma global de notre système d'alignement et son fonctionnement, ensuite nous expliquons chaque étape de ce schéma.

3.2 Schéma global du système

Notre système consiste à effectuer un alignement entre deux ontologies A et B écrites dans le format OWL qui est un langage ontologique, capable de décrire formellement la signification de la terminologie employée dans les documents web, constitue le premier niveau nécessaire du web sémantique après RDF.

Nous présentons le schéma global de notre approche, ensuite nous détaillons les différentes étapes et les différents algorithmes utilisés. Le schéma global proposé est illustré dans la figure suivante :

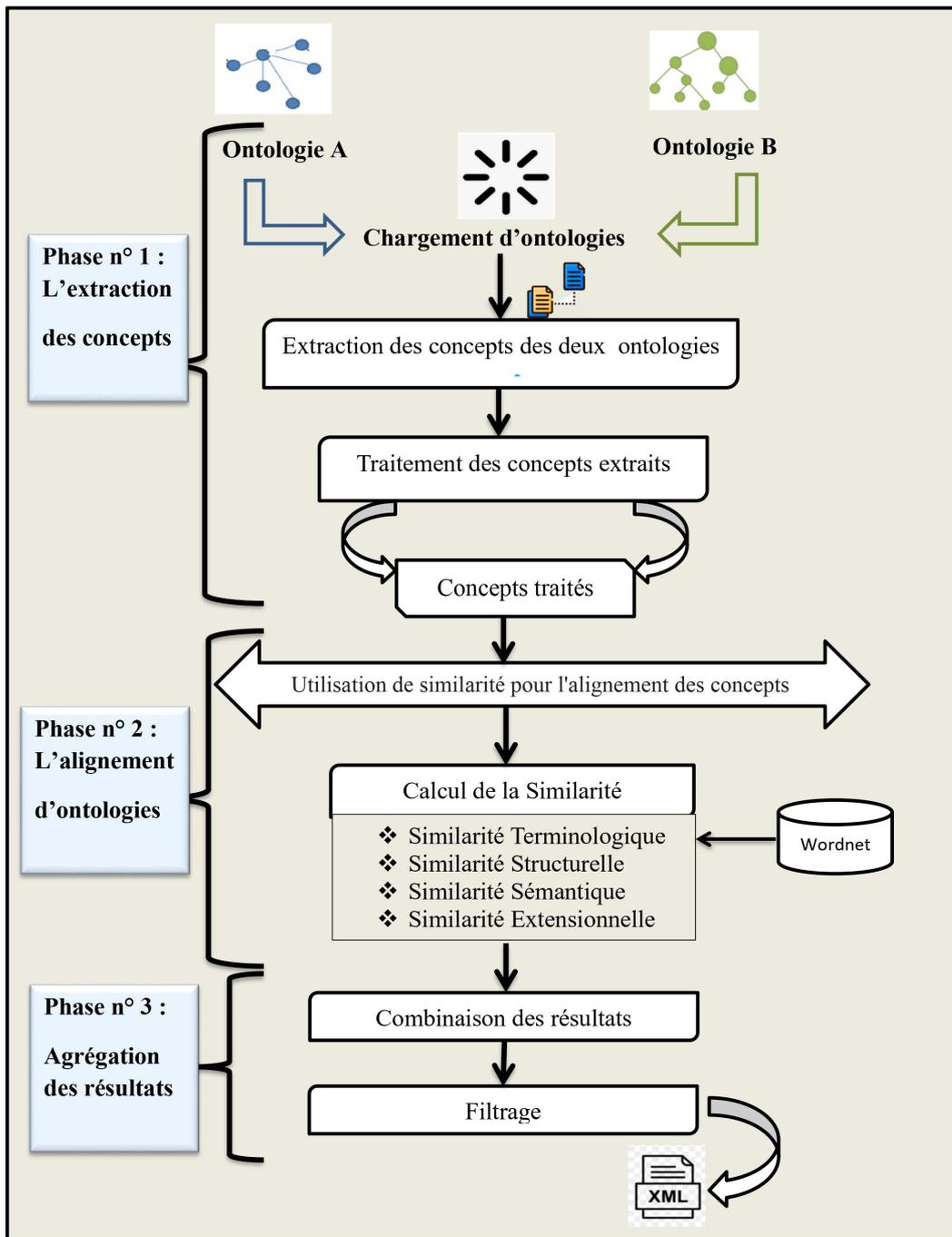


FIGURE 3.1 – Processus d'alignement des concepts composés

3.3 Description générale de notre approche

Notre approche est composée de trois phases, chaque phase présente une ou plusieurs étapes que nous avons suivies pour faire l'alignement d'ontologies.

Phase 1 : Extraction des concepts

Dans la première phase nous avons en entrée deux ontologies A et B décrites en format OWL, On doit tout d'abord parcourir les deux ontologies pour refléter toutes les entités, les relations et les individus de l'ontologie.

Phase 2 : l'alignement d'ontologies

Cette phase va répondre à notre besoin qui est le traitement des concepts composés pour l'alignement d'ontologies. Elle comporte deux étapes principales :

Étape 1 : traitement des concepts extraits.

Étape 2 : Le calcul de la similarité.

Phase 3 : l'agrégation des résultats

L'agrégation des résultats est la dernière phase dans notre approche. Elle permet de faire une combinaison des mesures de similarités utilisés pour en avoir une mesure globale (sémantique). Ainsi que de faire un filtrage sur les résultats obtenus.

3.4 Description détaillée de l'approche proposée

L'approche que nous avons proposée consiste à concevoir un système d'alignement entre deux ontologies. Pour expliquer notre solution, nous avons choisi comme exemple d'ontologies, appartenant au même domaine "Organisation de conférence" qui sont téléchargées depuis l'OAEI (Ontology Alignment Evaluation Initiative) de Benchmark, notre choix s'est porté sur l'ontologie « conference.Owl » et l'ontologie « cmt.Owl ».

Dans cette section, nous allons expliquer en détail l'approche proposée.

3.4.1 Extraction des concepts

Cette phase consiste à extraire les concepts de chaque ontologie d'une manière automatique en utilisant les étapes suivantes :

On doit tout d'abord charger les deux ontologies A et B, et parcourir toute la hiérarchie de ces ontologies afin d'extraire tous les concepts. Les concepts extraits constituent essentiellement les structures destinées à être alignées qui seront stockées dans deux listes différentes, les concepts extraits de l'ontologie A seront stockés dans la liste « listeConceptOnto1 » et les concepts de l'ontologie B dans la liste « listeConceptOnto2 ».

Pour chaque concept on doit spécifier les relations qui sont présentés par des liens (hiérarchiques ou sémantiques). On doit extraire tous les sous classes qui existent entre chaque classe comme le montre la figure suivante :

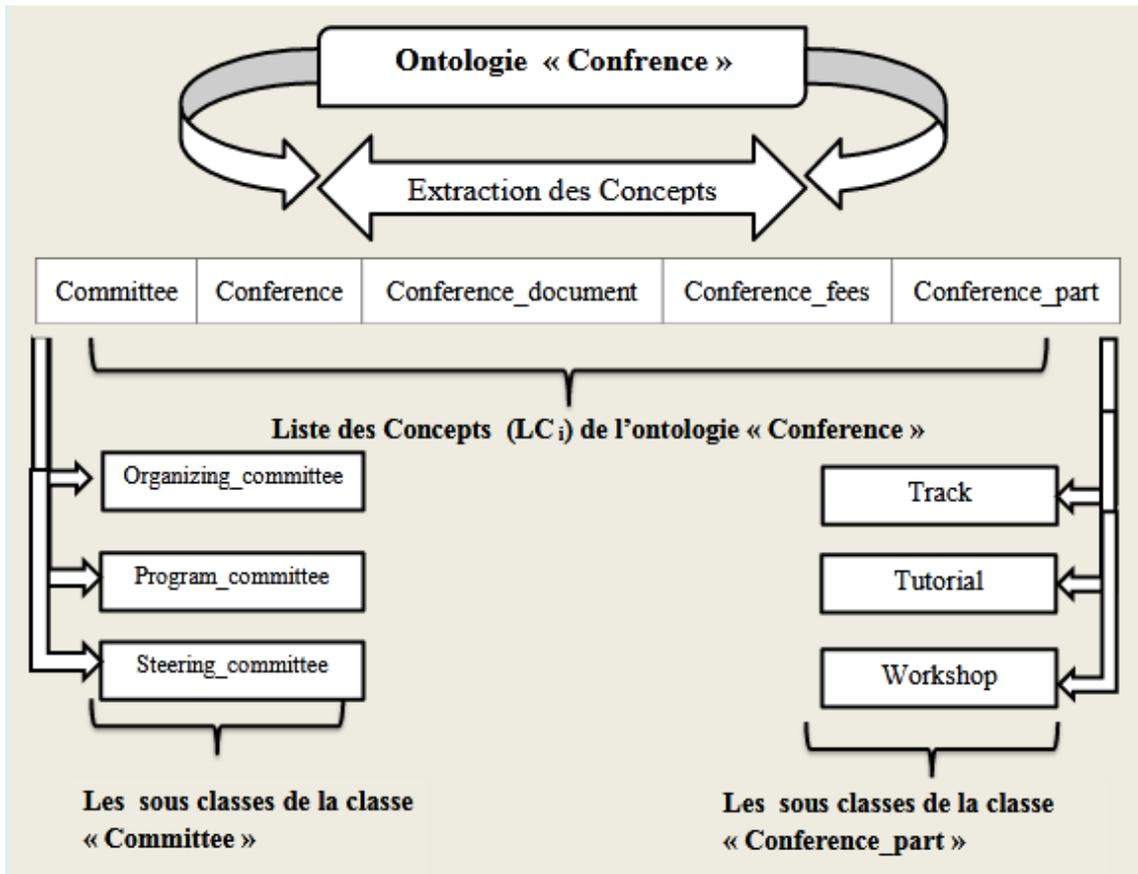


FIGURE 3.2 – Schéma d'extraction d'un ensemble de concepts de l'ontologie « conference.owl »

Cette figure représente un schéma qui explique la phase d'extraction des différents types de concepts de l'ontologie « conference.owl » ainsi que la hiérarchie entre les différents concepts. Les classes et les sous classes de l'ontologie « conference.Owl » sont décrites par des concepts simples et composés.

- La classe « Committee » présente un concept simple qui contient un seul terme et ses sous classes sont des concepts composés, ils sont construits à partir de plusieurs termes reliés entre eux par un trait d'union (_) « Organizing_committee », « Program_committee », « Steering_committee »
- La classe « Conference_part » présente un concept composé et ses sous classes sont des concepts simples « Track », « Tutoriel », « Workshop »

3.4.2 L'alignement d'ontologies

Cette phase joue un rôle crucial dans notre système

Étape 1 : Traitement des concepts extraits

Pour traiter les concepts extraits, nous avons proposé un algorithme « TreatConceptComposé » pour effectuer ce traitement. L'objectif de cet algorithme est de vérifier d'abord le type des concepts (concepts simples ou concepts composés) ensuite décomposer les concepts composés

retrouvés afin d'obtenir plusieurs termes simples.

Algorithme TreatConceptComposé

Inputs : « listeConceptOnto1 »

Outputs : « LCCT1 »

Début

1. /*importation de l'ontologie*/

Lire (Ontologie)

2. /*parcourir la liste des concepts de l'ontologie */

For each elementOnto1 ∈ listeConceptOnto1 **do**

3. /*rechercher des caractères (trait d'union et/ou lettres majuscules) dans les éléments de l'ontologie */

if (rechercher_TypeCaractères (elementOnto1)) **then**

4. /*ajouter les concepts trouvés dans la liste des concepts composés de l'ontologie*/

LCC1.**ajouter** (elementOnto1)

else

5. /*ajouter les concepts trouvés dans la liste des concepts simples de l'ontologie */

LCS1.**ajouter** (elementOnto1)

End if

End for each

6. /*parcourir la liste des concepts composés de l'ontologie */

For each elementOnto1 ∈ LCC1 **do**

7. /*rechercher l'emplacement des caractères (trait d'union ou lettres majuscules dans les éléments de liste des concepts composés */

if(rechercher_TypeCaractères(elementOnto1))

then

8. /*décomposer le concept composé en plusieurs termes à cet emplacement*/

Décomposer (elementOnto1)

9. /*ajouter le concept décomposé dans la liste des concepts composés traités de l'ontologie */

LCCT1.**ajouter** (elementOnto1)

End if End for each

END

Description détaillé de l'algorithme « TreatConceptComposé »

Le traitement des concepts extraits de l'ontologie se fait d'abord par la vérification du type des concepts (concepts simples ou concepts composés ensuite effectuer une décomposition sur les concepts composés trouvés par l'algorithme.

Après l'extraction des concepts de l'ontologies, on va les stocker dans une liste, ensuite nous commençons par la lecture de cette liste « listeConceptOnto1 » (la liste des concepts de l'ontolo-

gie)

Ensuite nous parcourons la liste des concepts afin de vérifier si l'élément trouvé dans la liste est de type simple ou composé et pour cela l'algorithme traite chaque élément trouvé comme une chaîne de caractère et vérifie si cette chaîne contient des caractères spéciaux (traits d'union (_)) et/ou des lettres majuscules (A...Z) au milieu de la chaîne alors c'est un concept composé et sera stocké dans la liste « LCCi », s'il ne trouve pas alors c'est un concept simple et sera stocké dans la liste « LCSi »

la figure suivante montre le résultat obtenu lors de la vérification des types des concepts sur l'ontologie « conference.Owl »

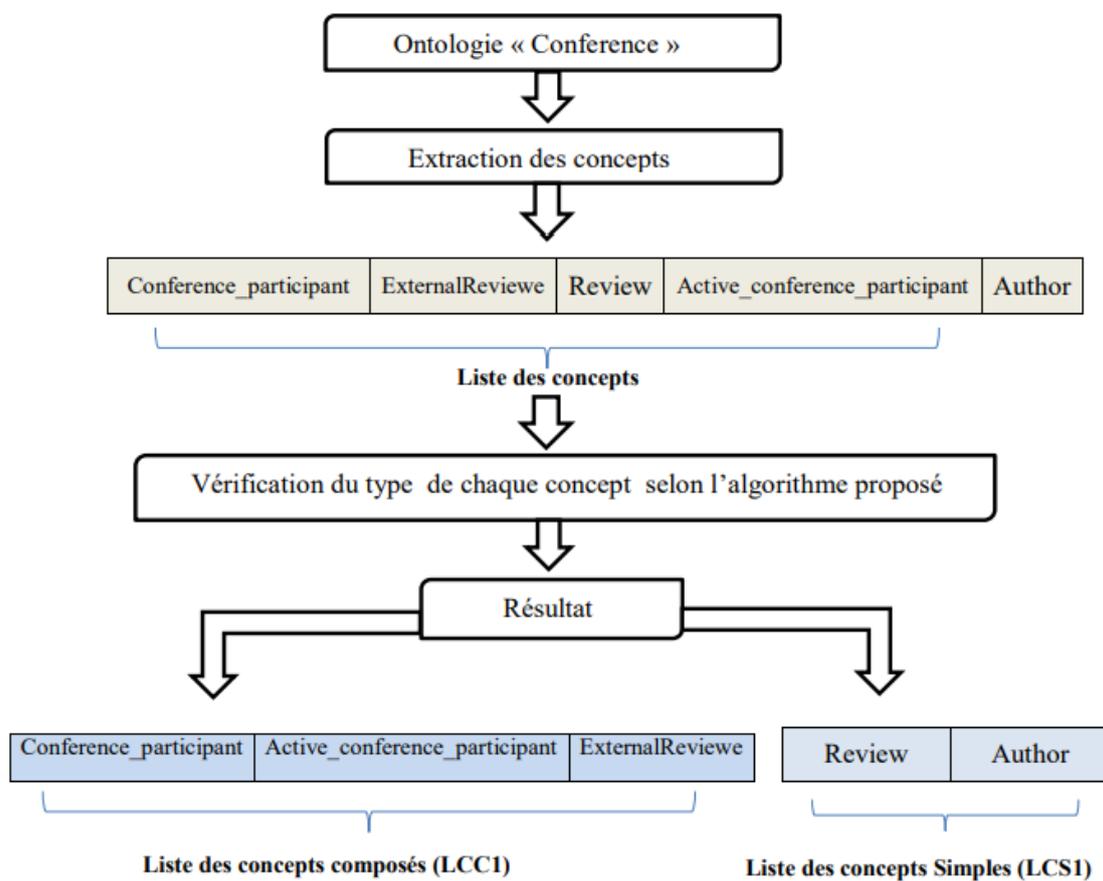


FIGURE 3.3 – Une partie des résultats de la vérification des types de concepts

Après la vérification, l'algorithme va parcourir la liste contenant les concepts composés de

l'ontologie « LCC1 » dans le but de décomposer le concept en plusieurs termes tout en gardant le sens du concept composé.

La décomposition du concept composé consiste en premier lieu à chercher l'emplacement des caractères spéciaux ou des lettres majuscules qui relient les termes puis insérer à cet emplacement un espace afin d'avoir plusieurs termes simples. Cette décomposition est nécessaire pour la reconnaissance des termes par WordNet.

Les résultats obtenus seront stockés dans une liste « LCCT 1 » qui est destinée aux concepts composés traités de cette ontologie.

La figure suivante montre les résultats de la décomposition appliqué sur la liste des concepts composés de l'ontologie « Conference.owl ».

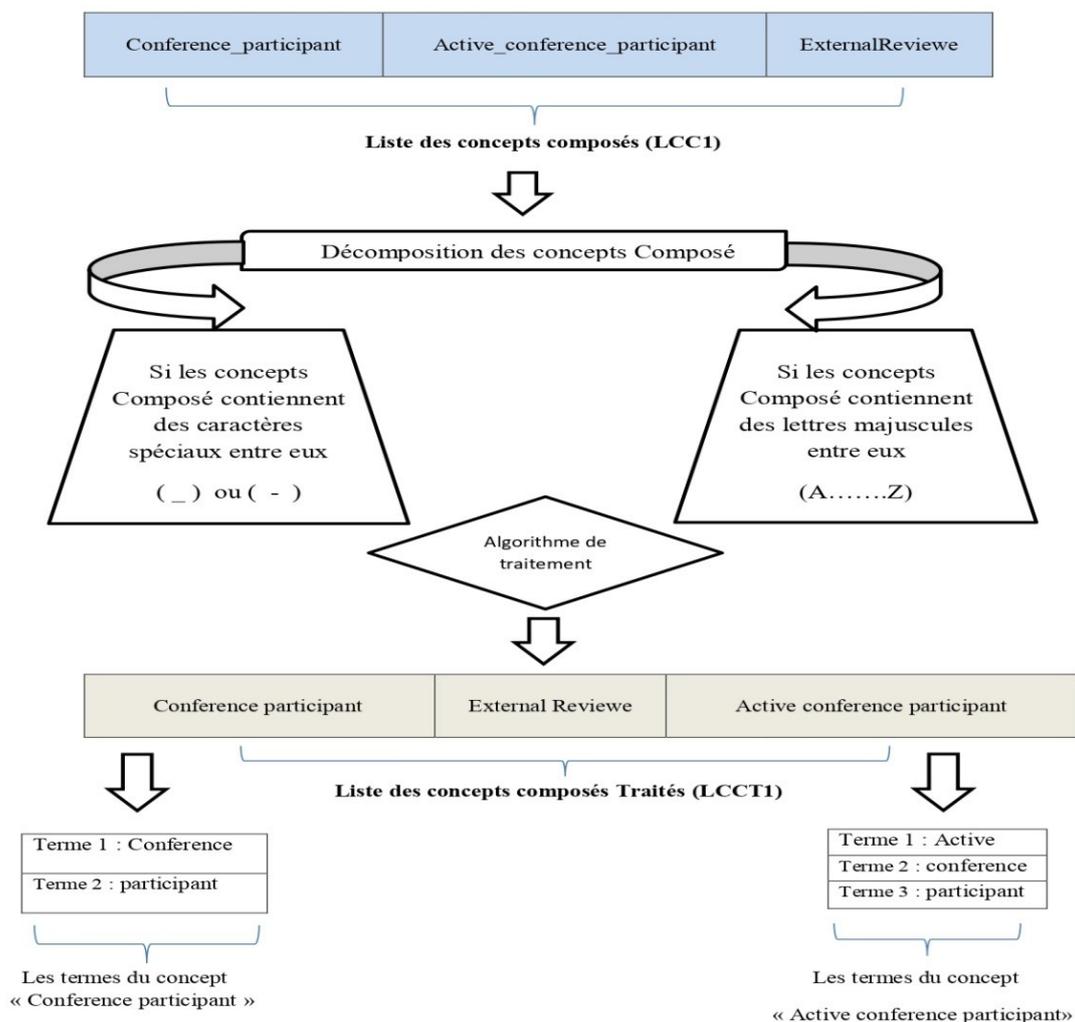


FIGURE 3.4 – Résultats de la décomposition des concepts composés

Etape 2 : Calcul de la similarité

Le calcul de la similarité entre les concepts est un problème de longue durée dans le domaine du traitement du langage naturel. Il existe plusieurs moyens pour évaluer la similarité entre deux entités. Dans notre travail, notre choix s'est basé sur l'utilisation des mesures terminologiques (lexicales et syntaxiques) et la mesure structurelle.

Avant d'enchaîner avec le calcul de similarité entre les différents types de concepts des deux ontologies, nous avons fait un pré-traitement qui consiste à normaliser les concepts composés. Dans notre système, on se basant sur les différentes mesures de similarités, mesures terminologiques (lexicales et syntaxiques) et mesures structurelles .

Les résultats obtenus seront agrégées avec une stratégie de combinaison pour atteindre une seule valeur de similarité finale entre deux concepts de deux ontologies différentes. En examinant cette valeur de similarité finale par rapport à un seuil prédéfini, deux concepts sont considérées comme similaires (équivalentes) ou différentes.

Pour chaque mesure utilisée, on calcule la similarité entre :

1. deux concepts simples.
2. deux concepts composés.
3. un concept simple et un concept composé.

Nous détaillons par la suite les différentes mesures de similarité utilisées :

1. Similarité terminologique

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

a) Mesures syntaxiques

Pour évaluer la correspondance entre les concepts de deux ontologies nous avons appliqué la distance de « dice » qui calcule la similarité entre deux paires de chaînes de caractères.

La similarité de « dice » est définie par le nombre des objets communs multipliés par 2 sur le nombre total d'objets. Elle est donc définie par la formule suivante :

$$Sim_{dc} = \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (3.1)$$

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

On considère X comme étant un concept de l'ontologie 1 et Y un concept de l'ontologie 2

La similarité entre X et Y est calculée en traitant chaque caractère qui compose les deux concepts :

La mesure syntaxique est calculée selon le principe du « dice » entre les différents concepts en traitant chaque paires de caractères qui compose ces concepts

- Si x et y sont deux concepts composés , la similarité entre eux est calculée en décomposant chaque terme de chaque concepts en paires de caractères
- Si x et y deux concepts simples :le calcul se fait en traitons deux paires de chaînes de caractères.
- Si x est un concept composé et y un concept simple : le calcul se fait en traitons tous les termes qui compose le concept composé avec le concept simple

Exemple

Prenons l'exemple d'une partie de deux listes qui contiennent les différents concepts des ontologies utilisés

- ListeOnto1 = [Conference part, Person]
- ListeOnto2 = [Conference, Conference Chair, Preference]

Etant donné les deux concepts composés « Conference_part » et « Conference_Chair»)la similarité syntaxique est calculée selon le principe ci-dessus :

Les paires de caractères des deux concepts composés respectifs sont :

« **Conference_part** » :co, on, nf, fe, er, re, en, nc, ce, pa, ar, rt = 12 paires

« **Conference_Chair** » : co, on, nf, fe, er, re, en, nc, ce, ch, ha, ai, ir = 13 paires

Le résultat de l'intersection entre les paires de caractères est : co, on, nf, fe, er, re, en, nc, ce.

Ainsi, l'équation devient :

$$Simdc(Conference_part, Conference_Chair) = \frac{2 * 9}{12 + 13} = 0.72 \quad (3.2)$$

Grâce à cette valeur de 0,72 il est possible de déterminer que ces deux mots sont à peu près semblables.

b) Mesures lexicales

Cette mesure consiste à utiliser des ressources externes (dictionnaires, taxonomies,...). Elle permettent de déterminer la similarité entre deux entités. Ces entités sont représentées par des termes. Plusieurs types de ressources peuvent être utilisées, notre choix s'est porté sur WordNet.

WordNet est le produit d'un projet de recherche à l'Université de Princeton[22]. C'est

une grande base de données lexicales de l'anglais. Dans WordNet, les verbes, les adverbes et les adjectifs sont organisés par une variété de relations sémantiques en ensembles de synonymes (synsets), qui représentent un concept. Un synset regroupe tous les termes dénotant un concept donné. Les synsets sont reliés entre eux par les relations sémantique : relation de généralisation / spécialisation, relation composant/-composé.

Pour le calcul de la similarité lexicale entre les concepts de deux ontologies. Nous avons utilisé la mesure de « lin ». l'objectif de cette mesure est de renvoyer un score indiquant à quel point les synsets de deux concepts sont similaires, en fonction du contenu d'information (CI) du sous-sommet le moins commun et de celui des deux synsets d'entrée.

$$SimLin(syn1, syn2) = \frac{2 * CI(lso(syn1, syn2))}{CI(syn1) + CI(syn2)} \quad (3.3)$$

Où :

CI : est le contenu informationnel, le contenu informationnel d'un concept traduit la pertinence d'un concept dans le corpus en tenant compte de sa spécificité ou généralité.

Le CI est calculé comme suit :

$$CI(synset) = -\log(P(synset)) \quad (3.4)$$

Où :

P(Synset) est la probabilité d'un synset donné, est calculée par la formule suivante :

$$P(Synset) = \text{frequence}(Synset)/N \quad (3.5)$$

Où :

N est le nombre total des Synsets et frequence(Synset) c'est le nombre d'apparition du synset.

Où :

Iso (lowest super-ordinate) est le sous-sommet commun le plus spécifique des deux synsets.

Le calcul de la similarité lexicale entre les concepts des deux ontologies comporte deux étapes que nous avons proposé afin d'utiliser le résultat pour la similarité globale.

i. **Mesure lexicale entre les concepts simples**

Pour le calcul de la mesure lexical entre les concepts simples de deux ontologies

différentes, on a utilisé la formule « SimLinS » qui calcule la similarité entre deux synsets de deux concepts simples en se basant sur les synsets de chaque concept en utilisant WordNet.

La formule est défini dans l'équation (3.3)

Exemple

Concept simple de l'ontologie 1 : [Person]

Concept simple de l'ontologie 2 : [Reviewer]

Synset [Person] : Synset('person.n.01')

Synset [Reviewer] : Synset('commentator.n.02')

SimLinS (Synset('person.n.01'), Synset('commentator.n.02'))= -0.0

ii. **Mesure lexicale entre les concepts composés**

Pour le calcul de la mesure lexicale entre deux concepts composés nous avons proposés la formule « SimLinC », qui permet de déterminer la similarité entre deux concepts composés en se basant sur les synsets de chaque concept en utilisant WordNet.

Et comme WordNet ne reconnaît pas les concepts composés, nous avons proposé l'algorithme « IdentifyTermsOfCC » suivant :

Algorithme IdentifyTermsOfCC

Inputs : « listeCCOnto1 ».

Outputs : « ESynsetCC1 ».

Début

1. /*Parcourir la liste des concepts composés */

For each CC1 ∈ listeCCOnto1 **do**

2. /*rechercher l'espace entre les termes du concept composé*/

if (rechercher_espace(CC1)) **then**

TermsCC1.**ajouter** (TCC1)

End if

3. /*parcourir la liste des termes du CC1 */

For each TCC1 ∈ TermsCC1 **do**

rechercher_Synset_TCC1 ()

4. /*ajouter les synonymes de chaque terme du CC dans un ensemble*/

ESynsetCC1.**ajouter**(SynsetCC1)

End for each

END

L'algorithme sert à identifier pour chaque concept composé les termes qui le compose, puis de chercher pour chaque terme leur synonymes sur WordNet et par la suite nous avons fait l'union entre l'ensemble des synsets du premier

terme avec l'ensemble des synsets du deuxième terme.

Etant donné un concept composé CC1 avec EnSynsetCC1, on doit parcourir chaque synset de l'ensemble EnSynsetCC1 puis utiliser la formule « SimLinC » comme elle est défini dans (formule 3.3), pour calculer la similarité entre chaque synset du « EnSynsetCC1 » avec les synsets d'autres concepts composés

A la fin nous avons fait une adaptation pour maximiser les résultats obtenus en utilisant le maximum (Max) entre les valeurs obtenus de chaque synset.

Exemple

Concept composé de l'ontologie 1 : [Conference participant]

Concept composé de l'ontologie 2 : [Meta Reviewer]

La liste des synsets du concept [Conference participant] : Synset('player.n.01'), Synset('conference.n.03'), Synset('conference.n.01'), Synset('conference.n.03'), Synset('participant.n.01'), Synset('league.n.01')

La liste des synsets du concept [Meta Reviewer] : Synset('reviewer.n.01'), Synset('commentator.n.02')

Synset('player.n.01'), Synset('reviewer.n.01')=0.28983066155024034

Synset('player.n.01'), Synset('commentator.n.02')=0.25341961470444835

Synset('conference.n.03'), Synset('reviewer.n.01')=-0.0

Synset('conference.n.03'), Synset('commentator.n.02')=-0.0

Synset('conference.n.01'), Synset('reviewer.n.01')=-0.0

Synset('conference.n.01'), Synset('commentator.n.02')=-0.0

Synset('participant.n.01'), Synset('reviewer.n.01')=0.24186357092068844

Synset('participant.n.01'), Synset('commentator.n.02')=0.2159689057674491

Synset('league.n.01'), Synset('reviewer.n.01')=-0.0

Synset('league.n.01'), Synset('commentator.n.02')=-0.0

listScore [0.28983066155024034, 0.25341961470444835, -0.0, -0.0, -0.0, -0.0, 0.24186357092068844, 0.2159689057674491, -0.0, -0.0]

La liste des scores consiste à enregistrer les résultats obtenu entre les synsets du concept composé 1 et 2 puis on doit parcourir la liste pour obtenir le max qu'on va l'utiliser par la suite dans le calcul de la mesure globale.

maxOfList 0.28983066155024034

2. Similarité structurelle

Le principe de cette mesure est de déduire la similarité entre deux concepts en exploitant leurs positions dans une hiérarchie et en fonction des informations structurelles.

Notre choix s'est porté sur la similarité de la distance du chemin « pathsim » qui permet d'indiquer la similarité entre deux synsets en fonction du chemin le plus court qui les relie dans la taxonomie is-a.

La distance entre deux synsets est le plus court chemin allant de synset1 à synset2.

Si il existe un plus courts chemin entre synset 1 et synset2 la valeur de « pathsim » devient :

$$pathsim(syn1, syn2) = \frac{1}{distance + 1} \quad (3.6)$$

Sinon

$$pathsim(syn1, syn2) = 0 \quad (3.7)$$

La similarité structurelle entre les concepts des deux ontologies est calculé entre tous les types des concepts comme suit :

a) **Mesure structurelle entre les concepts simples**

Étant donné deux concepts simples de deux ontologies différentes, on doit extraire les synsets pour chaque concept ensuite on utilise la mesure « pathsim » afin d'obtenir une valeur pour cette mesure.

Exemple

Concept simple de l'ontologie 1 : [Person]

Concept simple de l'ontologie 2 : [Reviewer]

Synset [Person] : Synset('person.n.01')

Synset [Reviewer] : Synset('commentator.n.02')

SimStrS = pathsim(Synset('person.n.01'), Synset('commentator.n.02'))=0.06

b) **Mesure structurelle entre les concepts composés**

La mesure « pathsim » est calculée entre tous les synsets du concept composé 1 avec les synsets du concept composé 2, et a la fin nous avons fait une adaptation pour combiner les valeurs obtenus en utilisant le minimum entre ces valeurs.

Cette mesure peut être appliquer entre un concept composé et un concept Simple.

Exemple

Concept composé de l'ontologie 1 : [Conference participant]

Concept composé de l'ontologie 2 : [Meta Reviewer]

listScore [0.125, 0.16666666666666666, 0.06666666666666667, 0.07692307692307693, 0.06666666666666667, 0.07692307692307693, 0.11111111111111111, 0.14285714285714285, 0.06666666666666667, 0.07692307692307693]

La liste des scores consiste à enregistrer les résultats obtenus entre les synsets du concept composé 1 et 2, puis on doit parcourir la liste pour obtenir le min entre les valeurs qu'on va l'utiliser par la suite dans le calcul de la mesure globale.

Synset('player.n.01'),Synset('reviewer.n.01')=0.125
 Synset('player.n.01'), Synset('commentator.n.02')=0.16666666666666666
 Synset('conference.n.03'),Synset('reviewer.n.01')= 0.06666666666666667
 Synset('conference.n.03'),Synset('commentator.n.02')= 0.07692307692307693
 Synset('conference.n.01'), Synset('reviewer.n.01')=0.06666666666666667
 Synset('conference.n.01'), Synset('commentator.n.02')=0.07692307692307693
 Synset('participant.n.01'), Synset('reviewer.n.01')=0.11111111111111111
 Synset('league.n.01'), Synset('reviewer.n.01')=0.06666666666666667
 Synset('league.n.01'), Synset('commentator.n.02')=0.07692307692307693
 minOfList 0.06666666666666667

3.4.3 Agrégation des résultats

Après le calcul de la similarité entre les différents concepts provenant des deux ontologies d'entrée, vient la dernière phase de cette approche.

- **Combinaison des mesures de similarités**

La similarité sémantique est calculée par la combinaison des similarités terminologiques, (syntaxique et lexicale) et la mesure structurelle en utilisant la moyenne pondérée, cette similarité est calculée entre les concepts simples ainsi qu'entre les concepts composés selon les formules suivantes :

$$(3.8) \quad SimGlb(CS1, CS2) = \frac{\alpha_1 Simdc(CS1, CS2) + \alpha_2 SimLinS(syn1, syn2) + \alpha_3 SimStrS(syn1, syn2)}{\alpha_1 + \alpha_2 + \alpha_3}$$

$$SimGlb(CC1, CC2) = \frac{\alpha_1 Simdc(CC1, CC2) + \alpha_2 MaxSimLinC(syn1, syn2) + \alpha_3 MinSimStrC(syn1, syn2)}{\alpha_1 + \alpha_2 + \alpha_3} \quad (3.9)$$

D'ou α_1 , α_2 et α_3 représentent des valeurs de pondération qui seront détaillé dans le chapitre de l'implémentation et test du système.

- **Filtrage**

Le filtrage consiste à déterminer une valeur de seuil bien précise de sorte qu'on garde que les couples de concepts qui ont une valeur de similarité supérieure à la valeur du seuil et on élimine les couples qu'ils ont une valeur inférieure. Nous obtenons donc une matrice des couples des concepts les plus similaires. Le résultat sera par la suite décrits sous format XML qui présente l'alignement des ontologies. Comme le montre la figure suivantes :

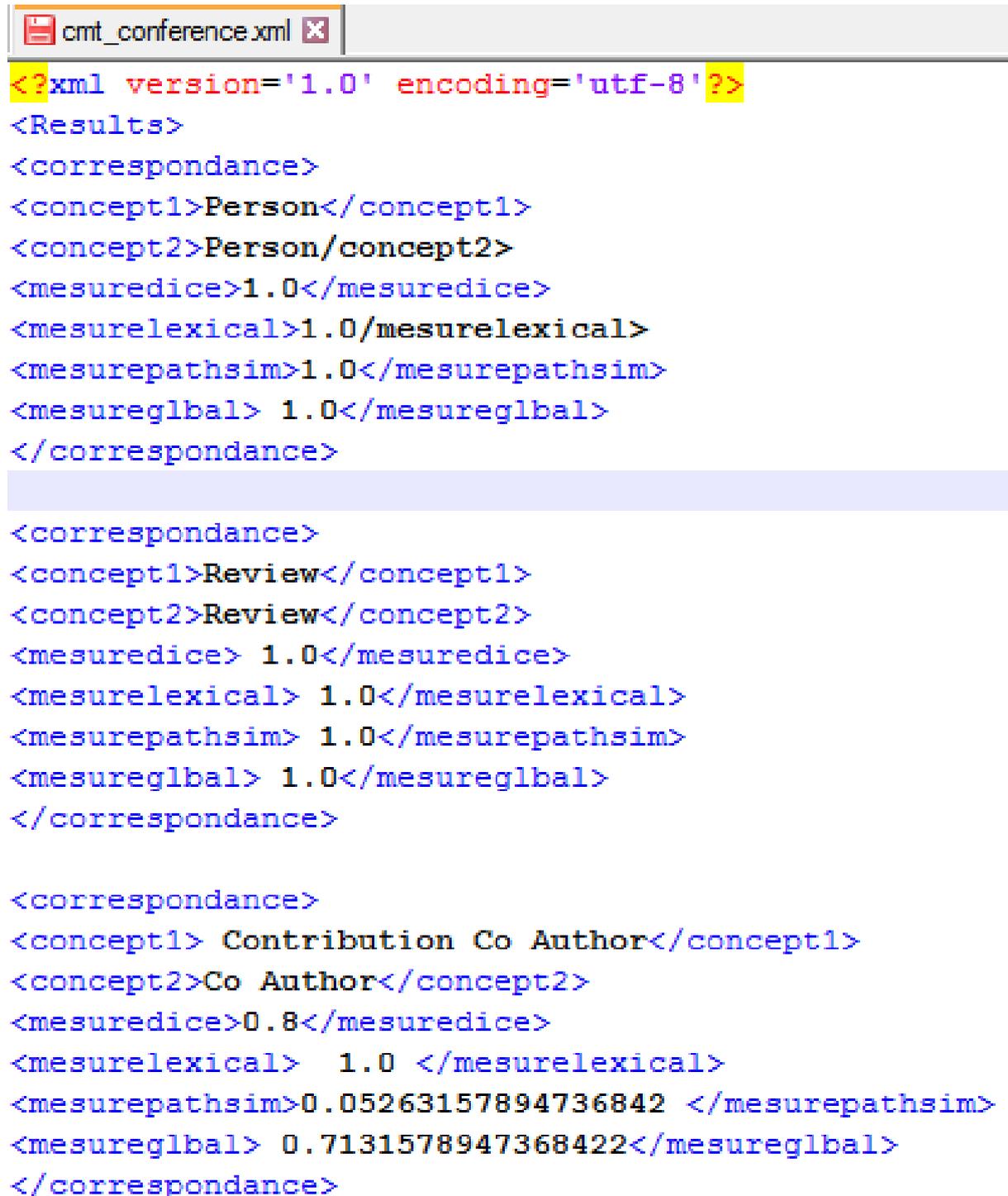
The image shows a screenshot of a text editor window titled 'cmt_conference.xml'. The window contains XML code for a conference document. The code is structured as follows: It starts with an XML declaration: `<?xml version='1.0' encoding='utf-8' ?>`. This is followed by a root element `<Results>`. Inside `<Results>`, there are three `<correspondance>` elements. Each `<correspondance>` element contains: `<concept1>` and `<concept2>` tags, and four measurement tags: `<measuredice>`, `<measurelexical>`, `<measurepathsim>`, and `<measureglbal>`. The first `<correspondance>` has `<concept1>Person</concept1>` and `<concept2>Person</concept2>`, with all measurements set to 1.0. The second `<correspondance>` has `<concept1>Review</concept1>` and `<concept2>Review</concept2>`, with `<measuredice>` set to 1.0 and the other three to 1.0. The third `<correspondance>` has `<concept1>Contribution Co Author</concept1>` and `<concept2>Co Author</concept2>`, with `<measuredice>` set to 0.8, `<measurelexical>` set to 1.0, `<measurepathsim>` set to 0.05263157894736842, and `<measureglbal>` set to 0.7131578947368422. The code ends with `</Results>`.

FIGURE 3.5 – Le fichier XML

3.5 Conclusion

Dans ce chapitre, nous avons présenté les étapes suivies pour la construction de notre système d'alignement ainsi que les étapes de traitement des concepts composés.

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

Chapitre 4

Implémentation et test du système

4.1 Introduction

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

4.2 Outils de développement de notre système

La réalisation de notre système nécessite les outils suivants :

4.2.1 Python

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

Python propose un système de typage dynamique et une gestion automatique de la mémoire. Il prend en charge plusieurs paradigmes de programmation, notamment orienté objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque standard étendue et complète.[23]

L'interpréteur Python est facilement étendu avec de nouvelles fonctions et de nouveaux types de données implémentés en C ou C ++ (ou d'autres langages pouvant être appelés à partir de C). Il convient également comme langage d'extension pour les applications personnalisables. Le

langage étant très complet et ne nécessitant aucune phase de compilation. Le choix de ce langage présente les avantages suivants :

- Python est entièrement gratuit.
- C'est un langage complet et puissant dans de nombreux domaines.
- Il est orienté objet mais n'impose pas ce type de programmation.
- Sa syntaxe reste très simple et le code peut être très lisible.
- Raccourcit le cycle de développement par rapport aux langages compilés et permet un prototypage rapide des projets.

4.2.2 Tkinter

Tkinter (Tool kit interface) est la bibliothèque d'interface graphique standard pour Python. Python, lorsqu'il est combiné à Tkinter, fournit un moyen rapide et facile pour créer des applications graphiques. Tkinter fournit une puissante interface orientée objet simple et conviviale.[23]

Tkinter est disponible sur Windows et la plupart des systèmes Unix, les interfaces créées avec Tkinter sont donc portables.

4.2.3 NumPy

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

4.2.4 Cython

Cython est un compilateur qui simplifie l'écriture d'extensions compilées pour Python. La syntaxe du langage est très similaire à Python mais il supporte en plus un sous-ensemble du langage C/C++ (déclarations des variables, appel des fonctions, ...).

Le premier intérêt de Cython est de produire du code nettement plus performant dans des programmes qui nécessitent par exemple la manipulation des grands tableaux, Cython permet d'écrire des interfaces Python à des bibliothèques externes écrites en C ou C++.

Le compilateur Cython convertit ses fichiers sources en langage Cython (extension .pyx) vers le langage C ou C++. Dans un deuxième temps, ces fichiers C/C++ sont compilés sous forme de bibliothèques prêtes à être utilisées avec Python.

Cython automatise entre autres :

- La conversion automatique entre type Python et type C.
- La compatibilité Python 2 / Python 3 .
- La gestion des erreurs.

4.2.5 Owlready2

Owlready2 est un module de programmation orientée ontologie en Python pour un accès de haut niveau aux ontologies OWL[24], incluant un quadre RDF optimisé.

Owlready peut :

- Importer les ontologies OWL au format NTriples, RDF / XML ou OWL / XML.
- Exporter les ontologies OWL vers NTriples ou RDF / XML.
- Manipuler de manière transparente les classes, les instances et les propriétés d'ontologie, comme s'il s'agissait d'objets Python normaux.
- Ajouter les méthodes Python aux classes d'ontologie.
- Effectuer une classification automatique des classes et des instances, en utilisant le raisonneur Hermit

4.2.6 NLTK

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

4.2.7 WordNet

Wordnet est une base de données lexicale, il a vu le jour en 1986 à l'Université de Princeton où il continue d'être développé et maintenu [22]. Il contient les noms, les verbes, les adjectifs et les adverbes en anglais qui sont organisés en des ensembles de synonymes[25]. Ces ensembles de synonymes, appelés synsets, sont reliés par différentes relations sémantiques. En utilisant Wordnet, nous pouvons trouver les synonymes d'un mot ou d'un terme. Par exemple, l'un peut choisir le terme « voiture » pour le nom de la classe dénotant un individu, mais un autre peut décider d'employer le terme « automobile » comme nom alternatif.

4.2.8 Protégé

Protégé est un environnement de développement d'ontologies open source avec des fonctionnalités pour éditer des classes, des slots (propriétés) et des instances [26]. Il est considéré aussi comme système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en Java. Il peut lire et sauvegarder des ontologies dans plusieurs formats RDF, RDFS, OWL, etc.

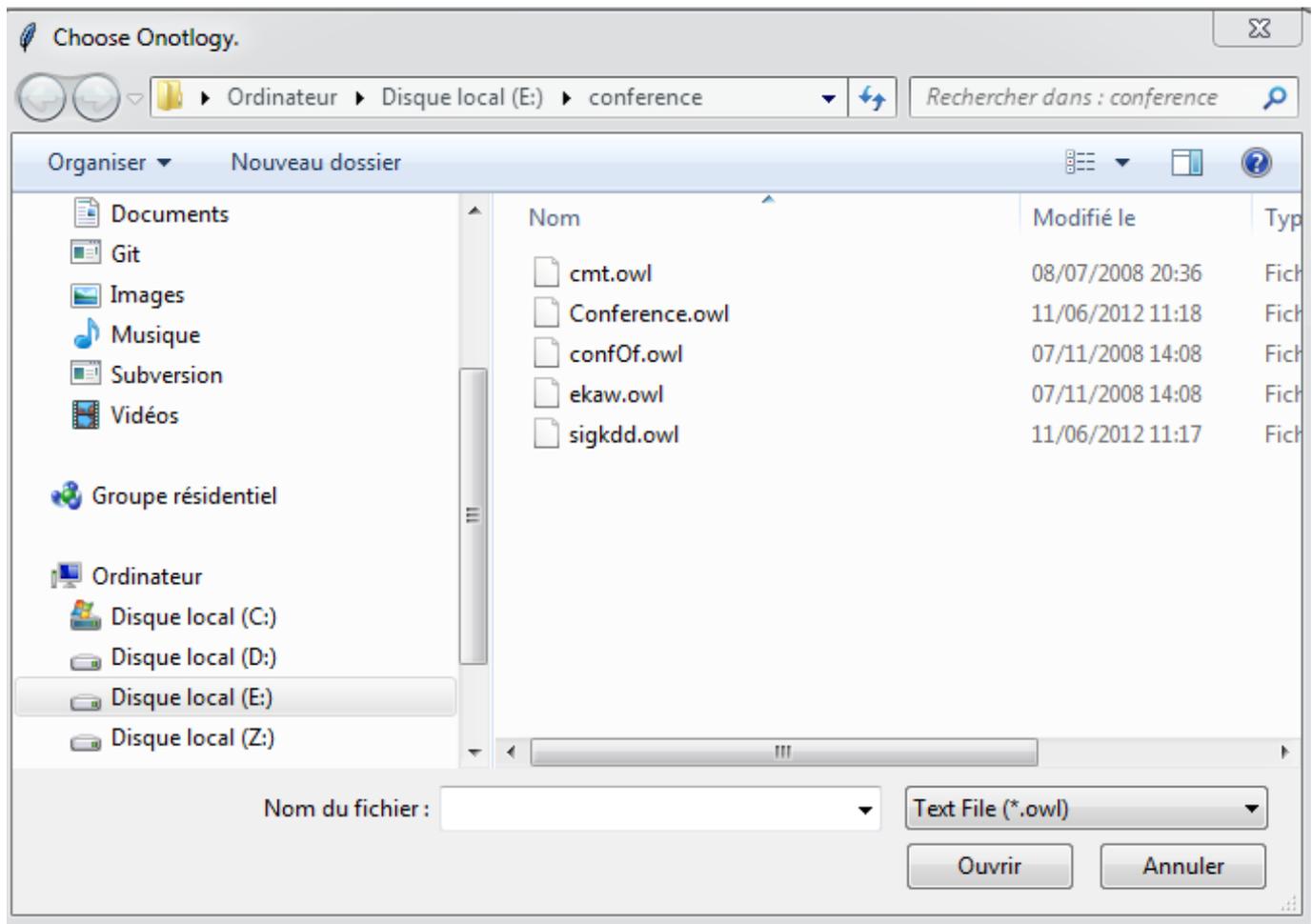


FIGURE 4.2 – Interface choix d’emplacement d’ontologies

Cette interface représente l’interface du choix de l’emplacement des ontologies, l’utilisateur doit appuyer sur le bouton « Traitement Concepts » afin d’obtenir cette interface.

Concept1	Concept_Traité
Conference_participant	Conference participant
Active_conference_participant	Active conference participant
Passive_conference_participant	Passive conference participant
Review_expertise	Review expertise
Submitted_contribution	Submitted contribution
Regular_author	Regular author
Conference_contributor	Conference contributor
Contribution_1th-author	Contribution 1th author
Contribution_co-author	Contribution co author
Conference_contribution	Conference contribution
Co-chair	Co chair
Committee_member	Committee member
Extended_abstract	Extended abstract
Important_dates	Important dates
Invited_speaker	Invited speaker
Conference_proceedings	Conference proceedings
Information_for_participants	Information for participants
Conference_document	Conference document
Regular_contribution	Regular contribution
Written_contribution	Written contribution
Conference_applicant	Conference applicant
Camera_ready_contribution	Camera ready contribution

FIGURE 4.3 – Interface de traitement des concepts composés

Après avoir choisir les ontologies souhaités, il peut afficher les résultats de traitement des concepts composés après avoir appuyer sur le bouton « Traitement Concepts ».

Traitement des Concepts Composés

parcourir ontologie 1	Synonymes of Concepts 1	parcourir ontologie 2	Synonymes of Concepts 2
	Concept1	Concept_Traité	
	Conference participant	{Synset('conference.n.01'), Synset('con	
	Active conference participant	{Synset('league.n.01'), Synset('active.a.:	
	Passive conference participant	{Synset('passive.s.02'), Synset('passive_	
	Review expertise	{Synset('expertness.n.01'), Synset('revie	
	Submitted contribution	{Synset('submit.v.06'), Synset('relegate.	
	Regular author	{Synset('regular.s.03'), Synset('even.s.04	
	Conference contributor	{Synset('conference.n.01'), Synset('con	
	Contribution 1th author	{Synset('contribution.n.01'), Synset('co	
	Contribution co author	{Synset('contribution.n.01'), Synset('co	
	Conference contribution	{Synset('contribution.n.01'), Synset('co	
	Co chair	{Synset('moderate.v.01'), Synset('profe:	
	Committee member	{Synset('member.n.04'), Synset('memb	
	Extended abstract	{Synset('unfold.v.03'), Synset('run.v.03'	
	Important dates	{Synset('go_steady.v.01'), Synset('date.1	
	Invited speaker	{Synset('receive.v.05'), Synset('invite.v.(
	Conference proceedings	{Synset('proceed.v.04'), Synset('confere	
	Information for participants	{Synset('information.n.01'), Synset('infr	
	Conference document	{Synset('document.v.01'), Synset('docu	
	Regular contribution	{Synset('regular.s.03'), Synset('regular.s	
	Written contribution	{Synset('spell.v.03'), Synset('written.a.0:	
	Conference applicant	{Synset('conference.n.01'), Synset('con	
	Camera ready contribution	{Synset('contribution.n.01'), Synset('co	

FIGURE 4.4 – Interface des synonymes des concepts composés

Après avoir traité les concepts composés, il peut afficher les synonymes de chaque concept composé après avoir appuyé sur le bouton « Synonymes of Concepts ».

Traitement des Concepts Composés pour l'alignement des Ontologies						
Mesure Concept Simple						
Concept1	Concept2	dice	Lin_sim	Path_sim	SimGlobal	
Person	Reviewer	0.3333333333333333	0.37979996724478	0.1666666666666666	0.2852533202312477	
Person	Decision	0.61538461538461	-0.0	0.0833333333333333	0.1564102564102564	
Person	Person	1.0	1.0	1.0	1.0	
Person	Document	0.42857142857142	-0.0	0.0909090909090909	0.12207792207792209	
Person	Preference	0.6666666666666666	-0.0	0.1	0.17333333333333334	
Person	Bid	0.0	-0.0	0.1	0.04000000000000001	
Person	Conference	0.61538461538461	-0.0	0.1	0.1630769230769231	
Person	Chairman	0.30769230769230	0.44101802628847	0.0909090909090909	0.27430930841748946	
Person	Acceptance	0.30769230769230	-0.0	0.1111111111111111	0.105982905982906	
Person	Rejection	0.42857142857142	-0.0	0.1111111111111111	0.13015873015873017	
Person	Review	0.18181818181818	-0.0	0.125	0.08636363636363636	
Person	Paper	0.54545454545454	0.28933790840553	0.1	0.2648260724531234	
Person	Author	0.3333333333333333	0.60113841004871	0.3333333333333333	0.4404553640194864	
Person	User	0.6	0.32407537714186	0.2	0.32963015085674435	
Person	Administrator	0.5	0.38908230076395	0.25	0.35563292030558236	
Reviewer	Reviewer	1.0	1.0	1.0	1.0	
Reviewer	Decision	0.30769230769230	-0.0	0.05882352941176470!	0.08506787330316742	
Reviewer	Person	0.3333333333333333	0.37979996724478	0.1666666666666666	0.2852533202312477	
Reviewer	Document	0.14285714285714	-0.0	0.0625	0.053571428571428575	
Reviewer	Preference	0.3333333333333333	-0.0	0.06666666666666667	0.09333333333333334	
Reviewer	Bid	0.22222222222222	-0.0	0.06666666666666667	0.07111111111111111	
Reviewer	Conference	0.30769230769230	-0.0	0.06666666666666667	0.08820512820512821	

FIGURE 4.5 – Interface de calcul de la similarité des concepts simples

La figure ci-dessus représente les résultats obtenus des mesures de similarité entre les concepts simples, ainsi que la similarité globale.

Traitement des Concepts Composés pour l'alignement des Ontologies						
Mesure Concept Compose						
Concept1	Concept2	Dice	Lin_Sim	Path_Sim	SimGlobal	
Conference participant	Meta Reviewer	0.5454545454545454	0.28983066155024	0.0666666666666666	0.25168984037767	
Conference participant	Program Committee	0.6956521739130435	0.48774914009921	0.0555555555555555	0.35645231304451	
Conference participant	Conference Chair	0.8695652173913043	1.0	0.0625	0.59891304347826	
Conference participant	Conference Member	0.6956521739130435	1.0	0.07142857142857	0.56770186335403	
Conference participant	Associated Chair	0.72	0.32416942663073	0.0625	0.29866777065229	
Conference participant	Program Committee Mem	0.64	0.82531105864861	0.0555555555555555	0.48034664568166	
Conference participant	Author Not Reviewer	0.48	0.40308992043698	0.0666666666666666	0.28390263484145	
Conference participant	Subject Area	0.5217391304347826	0.35341805785078	0.0666666666666666	0.27238171589393	
Conference participant	External Reviewer	0.56	0.28983066155024	0.0625	0.25293226462009	
Conference participant	Program Committee Chair	0.6666666666666666	0.48774914009921	0.0555555555555555	0.35065521159523	
Conference participant	Co author	0.6	0.40308992043698	0.0625	0.30623596817479	
Conference participant	Paper Abstract	0.6086956521739131	0.46055551860811	0.0625	0.33096133787802	
Conference participant	Paper Full Version	0.6153846153846154	0.46055551860811	0.05882352941176	0.33082854228487	
Conference participant	Meta Review	0.4761904761904761	0.29680266036102	0.05882352941176	0.23748857114721	
Active conference participant	Meta Reviewer	0.6086956521739131	0.28983066155024	0.0666666666666666	0.26433806172154	
Active conference participant	Program Committee	0.5833333333333334	0.48774914009921	0.0555555555555555	0.33398854492857	
Active conference participant	Conference Chair	0.75	1.0	0.0625	0.57500000000000	
Active conference participant	Conference Member	0.5833333333333334	1.0	0.07142857142857	0.54523809523809	
Active conference participant	Associated Chair	0.6923076923076923	0.32416942663073	0.0625	0.29312930911383	
Active conference participant	Program Committee Mem	0.5384615384615384	0.82531105864861	0.0555555555555555	0.46003895337397	
Active conference participant	Author Not Reviewer	0.6153846153846154	0.40308992043698	0.0666666666666666	0.31097955791838	
Active conference participant	Subject Area	0.5833333333333334	0.35341805785078	0.0666666666666666	0.28470055647364	

FIGURE 4.6 – Interface de calcul de la similarité des concepts composés

Cette figure représente les résultats obtenus des mesures de similarité entre les concepts composés, ainsi que la similarité globale.

```
interfaceOntoSimple.py x filenameXML.xml x
1 <root><doc>
2   <concept1 name="res1">Person</concept1>
3   <concept2 name="res2">Person</concept2>
4   <measureGlobal name="measureglobal">1.0</measureGlobal>
5 </doc>
6
7 <doc>
8   <concept1 name="res1">Review</concept1>
9   <concept2 name="res2">Review</concept2>
10  <measureGlobal name="measureglobal">1.0</measureGlobal>
11 </doc>
12 <doc>
13  <concept1 name="res1">Contribution Co Author</concept1>
14  <concept2 name="res2">Co Author</concept2>
15  <measureGlobal name="measureglobal">0.7131578947368422</measureGlobal>
16 </doc>
17
18
19 <doc>
20  <concept1 name="res1">Conference</concept1>
21  <concept2 name="res2">Conference</concept2>
22  <measureGlobal name="measureglobal">1.0</measureGlobal>
23 </doc>
24
25
26 <doc>
27  <concept1 name="res1">Program Committee</concept1>
28  <concept2 name="res2">Program Committee</concept2>
29  <measureGlobal name="measureglobal">1.0</measureGlobal>
30 </doc>
31
32 <doc>
```

FIGURE 4.7 – Le fichier exporté « RésultatXML »

Cette figure représente le fichier XML contenant les résultats de la similarité globale qui dépasse le seuil proposé.

4.4 Test du système

Afin de tester la fiabilité et de montrer la faisabilité de notre approche, nous avons développé le système d'alignement en combinant plusieurs mesures de similarité. Ce système a besoin

d’être testé sur des ontologies afin de vérifier leur fonctionnement.

Nous avons effectué des expériences sur des ontologies contenant des concepts composés, selon le tableau suivant :

Ontologie (OWL)	Nombre de classe	Lien
conference	60	http://conference
cmt	30	http://cmt
confOf	39	http://confOf
ekaw	74	http://ekaw

TABLE 4.1 – Ontologies de test

4.4.1 Résultats expérimentaux et discussion

Dans cette section nous présentons les résultats obtenus des différentes mesures qu’on a testé et utilisé pour faire l’alignement entre les concepts de différentes ontologies.

Concernant la mesure syntaxique, nous avons utilisé deux mesures différentes la distance de « jaro » et la distance de « dice », et pour la mesure lexicales nous avons utilisé la mesure de « lin_similarité », et pour la mesure structurelle nous avons utilisé la mesure de « path_similarity » et « wu_palmer ».

La mesure wu_palmer consiste à effectuer une comparaison entre les synsets de la première ontologie et les synsets de la deuxième ontologie selon leur position dans la taxonomie wordNet.

Nous avons appliqué toutes ces mesures entre les concepts simples, entre les concepts composés, et ainsi qu’entre les concepts simples et composés, nous avons pris l’exemple des deux ontologies « conference » et « cmt », les résultats obtenus sont mentionnés dans les tableaux ci-dessous.

1. Entre les concepts simples des deux ontologies « conference » et « cmt »

Concept 1	Concept 2	Jaro	dice	lin_sim	wu_palmer	path_similarity
Person	Reviewer	0.43	0.33	0.16	0.44	0.16
Person	Decision	0.72	0.61	-0.0	0.12	0.09
Person	Person	1.0	1.0	1.0	1.0	1.0
Person	Preference	0.60	0.66	-0.0	0.12	0.08
Reviewer	Reviewer	1.0	1.0	1.0	1.0	1.0
Reviewer	Decision	0.5	0.30	-0.0	0.13	0.06
Reviewer	Document	0.41	0.14	-0.0	0.12	0.07
Committee	Decision	0.41	0.46	0.06	0.25	0.08
Committee	Person	0.42	0.33	-0.0	0.12	0.1
Committee	Document	0.56	0.57	-0.0	0.23	0.1
Committee	Preference	0.47	0.16	0.06	0.25	0.07
Committee	Bid	0.48	0.22	0.06	0.11	0.08

TABLE 4.2 – Les résultats obtenus entre les concepts simples des deux ontologies

Le tableau ci-dessus représente les différentes mesures appliquées sur les concepts simples de deux ontologies

- Concernant la mesure syntaxique, nous avons utilisé la mesure de « jaro » et la mesure de « dice », d'après les résultats mentionnés dans le tableau nous pouvons remarquer que la mesure de « jaro » montre des résultats plus élevés par rapport à la mesure de « dice » dans certains concepts, nous prenons l'exemple de Person et Reviewer, la mesure de jaro donne 0.43 et la mesure de dice qui donne 0.3, contrairement aux concepts Person et Preference la mesure de jaro donne 0.60 et la mesure de dice qui donne 0.66. Après le test de ces deux mesures, notre choix s'est porté sur la mesure de « dice » car elle nous aide à améliorer nos résultats de la similarité globale.
- Par rapport à la mesure structurelle, les résultats obtenus en appliquant la mesure de « wu_palmer » sont bien plus élevés par rapport à ceux de « path_similarity », nous prenons l'exemple de Reviewer et Decision, la mesure de « wu_palmer » donne 0.13 et la mesure de « path_similarity » donne 0.06. Après le test de ces deux mesures, notre choix s'est porté sur la mesure de « path_similarity » car elle nous donne des meilleurs résultats lors de l'application de la similarité globale.

2. Entre les concepts composés des deux ontologies « conference » et « cmt »

Le tableau suivant montre les résultats obtenus en les appliquant sur les concepts composés des deux ontologies.

Concept 1	Concept 2	jaro	dice	lin_sim	wu_palmer	path_similarity
Conference participant	Meta Reviewer	0.48	0.54	-0.0	0.60	0.08
Conference participant	Program Committee	0.59	0.69	-0.0	0.24	0.09
Conference participant	Conference Chair	0.81	0.86	0.07	0.26	0.10
Conference participant	Conference Member	0.60	0.69	0.11	0.24	0.11
Active conference participant	Meta Reviewer	0.75	0.60	-0.0	0.24	0.06
Active conference participant	Program Committee	0.55	0.58	-0.0	0.23	0.11
Active conference participant	Conference Chair	0.59	0.75	0.11	0.27	0.07
Active conference participant	Conference Member	0.58	0.58	0.06	0.28	0.07

TABLE 4.3 – Les résultats obtenus entre les concepts composés des deux ontologies

Le tableau ci-dessus représente les différentes mesures appliqués sur les concepts composés de deux ontologies.

- Nous constatons que concernant les concepts composés, la mesure de « jaro » donne des résultats moins élevé contrairement à la mesure de « dice », par exemple Conference Participant et Meta Reviewer la mesure de « Jaro » donne 0.48 et la mesure de « dice » donne 0.54. Après le test de ces deux mesures, notre choix s’est porté sur la mesure de « dice » car elle nous aide à améliorer nos résultats de la similarité globale.
- Cependant, pour la mesure structurelle, la mesure de « wu_palmer » montre toujours des résultats élevé par rapport à la mesure « path_similarity », nous prenons l’exemple de Active conference participant et Conference chair « wu_palmer » donne 0.27 et « path_similarity » donne 0.07. Après le test de ces deux mesures, notre choix s’est porté sur la mesure de « path_similarity » car elle nous donne des meilleures résultats lors de l’application de la similarité globale.

3. Entre les concepts composés et les concepts simples des deux ontologies « conference » et « cmt »

Concept 1	Concept 2	jaro	dice	lin_sim	wu_palmer	path_similarity
Person	Meta Reviewer	0.41	0.25	1.0	0.12	0.05
Person	Program committe	0.55	0.47	1.0	0.1	0.03
Person	Conference chair	0.47	0.47	1.0	0.11	0.04
Reviewer	Meta Reviewer	0.71	0.75	1.0	0.11	0.05
Reviewer	Program committe	0.39	0.35	1.0	0.26	0.03
committe	Meta Reviewer	0.49	0.37	1.0	0.22	0.05
committe	Program committe	0.61	0.70	1.0	0.2	0.03

TABLE 4.4 – Les résultats obtenus entre les concepts simples et les concepts composés

Le tableau ci-dessus représente les différentes mesures appliquées entre les concepts composés et les concepts simples des deux ontologies.

- Les résultats obtenus lors de l’application de la mesure « jaro » sont plus élevé par rapport à « dice » dans certain concept tel que Person et Meta Reviewer, la mesure « jaro » donne 0.41 et la mesure de « dice » donne 0.25, par contre par rapport aux concepts Reviewer et Meta Reviewer la mesure « jaro » donne 0.71 et la mesure de « dice » donne 0.75. Après le test de ces deux mesures, notre choix s’est porté sur la mesure de « dice » car elle nous aide à améliorer nos résultats de la similarité globale.

- Concernant la mesure structurelle, la mesure de « path_similarity » donne toujours des résultats moins élevés par rapport à « wu_palmer ». Après le test de ces deux mesures, notre choix s'est porté sur la mesure de « path_similarity » car elle nous donne des meilleurs résultats lors de l'application de la similarité globale.

Discussions

Après une étude comparative entre les résultats obtenus des différentes mesures de similarité, nous avons constaté que :

- Pour la mesure syntaxique, la distance de « jaro » permet de mesurer le nombre et l'ordre des caractères communs entre deux chaînes de caractères selon la formule suivante :

$$SymSynJaro = \frac{1}{3} \left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right) \quad (4.1)$$

Où :

m est le nombre de caractères correspondants.

t est le nombre de transpositions.

Deux caractères identiques de s1 et s2 sont considérés comme correspondant si leur éloignement (i.e la différence entre leur position dans leur chaîne respective) ne dépasse pas :

$$\left(\frac{\max(|s1|, |s2|)}{2} \right) - 1 \quad (4.2)$$

Le nombre de transposition est obtenu en comparant le i^{ème} caractère correspondant de s1 avec le i^{ème} caractère correspondant de s2.

Le nombre de fois où ces caractères sont différents, divisé par 2 donne le nombre de transposition.

- Pour la mesure lexicale, nous avons utilisé la mesure de lin_similarity qui donne des bons résultats et compare chaque contenu informationnel de chaque synset pour chaque concept simple. Les valeurs sont comprises entre 0 et 1.
- Pour la mesure structurelle nous avons utilisé la mesure de « wu_palmer » [28] selon la taxonomie du wordNet puisque nous tenons à comparer entre les synsets des concepts de la première ontologie avec les synsets des concepts de la deuxième ontologie, nous avons remarqué que la mesure de « wu_palmer » donne un résultat plus élevé par rapport à « path_similarité » ce qui nous amène à utiliser « path_similarity » dans nos calculs de la similarité globale.

La mesure de « wu_palmer » est calculée comme suit :

$$wu_palmer = \frac{2 * Prof(C)}{dist(ci, c) + dist(cj, c)} \quad (4.3)$$

Où C est le concept le plus spécifique qui sépare c1 et c2, Prof(c) est le nombre d'arcs qui sépare C de la racine et dist(ci, c) le nombre d'arcs qui séparent ci de c.

- Pour conclure, dans le calcul de la similarité globale entre les différents concepts ; nous avons choisi d'utiliser la mesure de « dice », « lin_similarity » et « path_similarity », le résultat de la similarité globale est mentionné dans les tableaux ci-dessous

Les tableaux suivants comporte les mesures finales choisi pour notre système ainsi que la similarité globale.

Concept 1	Concept 2	dice	lin_sim	path_similarity	Sim_global
Person	Reviewer	0.33	0.16	0.16	0.29
Person	Decision	0.61	-0.0	0.09	0.15
Person	Person	1.0	1.0	1.0	1.0
Person	Preference	0.66	-0.0	0.08	0.16
Reviewer	Reviewer	1.0	1.0	1.0	1.0
Reviewer	Decision	0.30	-0.0	0.06	0.08
Reviewer	Document	0.14	-0.0	0.07	0.05
Committee	Decision	0.46	0.06	0.08	0.14
Committee	Person	0.33	-0.0	0.1	0.10
Committee	Document	0.57	-0.0	0.1	0.17
Committee	Preference	0.16	0.06	0.07	0.08
Committee	Bid	0.22	0.06	0.08	0.10

TABLE 4.5 – Les meilleures mesures choisies entre les concepts simples

Concept 1	Concept 2	dice	lin_sim	path_similarity	Sim_glb
Conference participant	Meta Reviewer	0.54	0.28	0.06	0.29
Conference participant	Program Committee	0.69	0.04	0.05	0.43
Conference participant	Conference Chair	0.86	1.0	0.06	0.73
Conference participant	Conference Member	0.69	1.0	0.07	0.69
Active conference participant	Meta Reviewer	0.60	0.02	0.06	0.31
Active conference participant	Program Committee	0.58	0.04	0.05	0.40
Active conference participant	Conference Chair	0.75	1.0	0.06	0.70
Active conference participant	Conference Member	0.58	1.0	0.07	0.66

TABLE 4.6 – Les meilleures mesures choisies entre les concepts composés

Concept 1	Concept 2	dice	lin_sim	path_similarity	Sim_glb
Person	Meta Reviewer	0.25	1.0	0.05	0.47
Person	Program committe	0.47	1.0	0.03	0.50
Person	Conference chair	0.47	1.0	0.04	0.51
Reviewer	Meta Reviewer	0.75	1.0	0.05	0.57
Reviewer	Program committe	0.35	1.0	0.03	0.48
committe	Meta Reviewer	0.37	1.0	0.05	0.49
committe	Program committe	0.70	1.0	0.03	0.55

TABLE 4.7 – Les meilleures mesures choisies entre les concepts simples et les concepts composés

Nous avons choisi d'utiliser aussi des poids pour chaque mesure afin d'améliorer la précision de notre système, en comparant avec les résultats de l'OAEI.

Les poids se changent d'un type de concept à un autre comme le montre le tableau suivant :

Simple _ Simple	$\alpha_1 = 0.1, \alpha_2 = 0.2, \alpha_3 = 0.2$
Composé _ Composé	$\alpha_1 = 0.2, \alpha_2 = 0.4, \alpha_3 = 0.2$
Simple _ Composé	$\alpha_1 = 0.1, \alpha_2 = 0.6, \alpha_3 = 0.2$

TABLE 4.8 – Les valeurs des poids

Après avoir testé plusieurs valeurs de poids différentes pour le calcul de la similarité globale entre les concepts simples, les concepts composés et entre les concepts simples et les concepts composés, les valeurs représentées dans le tableau ci-dessus sont les meilleures valeurs qui nous a permis d'avoir des meilleurs résultats.

4.5 Validation

Le calcul des mesures de similarité est basé sur la comparaison entre les correspondances produites par notre système d'alignement d'ontologies, noté : C, et un ensemble de correspondance de référence produit par un humain qu'on notera H

- Les correspondances correctes trouvées par un système sont appelées « The true positive (TP) » et elles sont calculées ainsi :

$$TP = C \cap H \quad (4.4)$$

- Les correspondances incorrectes trouvés par un système sont appelés « The false positive (FP) » et sont calculées ainsi :

$$FP = C - C \cap H \quad (4.5)$$

- Les correspondances correctes omises par un système sont appelés « The false négative (FN) » et sont calculées ainsi :

$$FN = H - C \cap H \quad (4.6)$$

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

$$Precision = \frac{|TP|}{|TP + FP|} = \frac{C \cap H}{C} \quad (4.7)$$

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

$$Rappel = \frac{|TP|}{|TP + FN|} = \frac{C \cap H}{H} \quad (4.8)$$

Le tableau suivant montre un exemple des correspondances manuelles et des correspondances de notre système entre les différents concepts des ontologies initiales.

Concept de l'ontologie1	Concept de l'ontologie2	correspondances manuelles	Mesures de similarité	correspondances de notre système
Person	Reviewer	-	0.29	-
Person	Decision	-	0.15	-
Person	Person	√	1.0	√
Person	Document	-	0.25	-
Person	Preference	-	0.17	-
Person	Bid	-	0.036	-
Person	Conference	-	0.16	-
Person	Chairman	-	0.33	-
Person	Paper	-	0.24	-
Person	Administrator	-	0.36	-
Regular Author	Author	√	0.74	√
Contribution Co Author	Co Author	√	0.71	√
Abstract	Paper Abstract	√	0.76	√
Review	Review	√	1.0	√
Review	Paper	-	0.09	-
Review	User	-	0.07	-
Review Preference	Preference	√	0.74	√
Conference Document	document	√	0.75	√
Conference	Conference	√	1.0	√
Conference Volume	Conference	√	0.76	√
Organizer	Conference	-	0.10	-
Chair	Chairman	-	0.45	-
Program Committee	Program Committee	√	1.0	√
Topic	Subject Area	√	0.68	√
Chairman	Chair	√	0.19	-

TABLE 4.9 – Extrait des correspondances manuelles et les correspondances de notre système

Le tableau ci-dessus représente les résultats de la mesure de similarité globale obtenus par notre système, ainsi qu'une comparaison entre les correspondances manuelles de l'OAEI et les correspondances de notre système.

On parcourt les résultats obtenus à la recherche des concepts qui ont un degré de similarité élevé fixé à un seuil > 0.67

Alors :

produites par le système : $|C| = 11$

Correspondances produites manuellement : $|H| = 12$

$$TP = C \cap H \quad (4.9)$$

$$FP = C - C \cap H \quad (4.10)$$

$$FN = H - C \cap H \quad (4.11)$$

$$Precision = \frac{|TP|}{|TP + FP|} = \frac{C \cap H}{C} \quad (4.12)$$

$$Rappel = \frac{|TP|}{|TP + FN|} = \frac{C \cap H}{H} \quad (4.13)$$

Ontologies	C	H	TP	FP	FN	Précision	Rappel
Cmt et conference	11	12	11	0	1	0.91	1
Cmt et confOf	10	11	10	0	1	0.90	1
conference et ekaw	22	24	22	0	2	0.91	1

TABLE 4.10 – Les résultats des rappels et précisions

Selon les résultats obtenus dans le tableau 4.9, l'expert du domain a validé les correspondances entre :

- **Les concepts simples :**
 - Person _ Person
 - Review _ Review
 - Conference _ Conference
- **Les concepts composés :**
 - Contribution Co Author _ Author
 - Programm committe _ Programm committe
- **Les concepts composés et les concepts simples :**
 - Review preference _ preference
 - Regular author _ author
 - Conference document _ document
- **Les concepts simples et les concepts composés :**
 - Abstract _ Paper abstract
 - Topic _ Subject Area

Notre système a obtenu des valeurs de similarités supérieur à 0.67 donc il a considéré que ces concepts sont similaires. Cependant, ce n'est pas le cas pour les autres concepts qui ont des valeurs de similarités inférieur à 0.67.

4.6 Conclusion

Dans ce chapitre, nous avons vu les outils qu'on a utilisé pour l'implémentation de notre système, qui sont, le vocabulaire Wordnet qui nous a permis de trouver tous les synonymes d'un mot ou d'un terme en anglais, langage Python qu'on a utilisé comme un langage de scriptage, ensuite l'environnement Tkinter qu'on le considère comme une plate-forme de programmation, et protégé pour l'édition d'ontologie. Et on a fait un petit aperçu sur notre interface.

Nous avons aussi présenté les résultats d'évaluation et l'expérimentation de notre système qui ont permis de répondre au but de notre système qui est la combinaison des mesures de similarités pour avoir un bon résultat d'alignement.

Conclusion et perspectives

Ces dernière années, les ontologies ont suscité de nombreux travaux dans le domaine du web sémantique. Toutefois, en raison de la nature décentralisé du web sémantique, les ontologies sont très hétérogènes.

L'alignement des ontologies est une solution pour résoudre l'hétérogénéité sémantique des données, en proposant des correspondances entre des entités sémantiques similaires des différentes ontologies.

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans le domaine de l'ingénierie des connaissances et plus particulièrement l'ingénierie ontologique. Notre objectif a été de chercher et proposer une solution d'alignement d'ontologies des concepts composés et de concevoir un système d'alignement pour deux ontologies OWL en utilisant les différentes mesures de similarité .

Dans le premier chapitre, nous nous sommes intéressés à ce qu'était une ontologie. Pour cela, nous sommes partis des origines philosophiques du terme pour ensuite définir son sens en ingénierie des connaissances. Par la suite, nous avons mentionnés les composants d'une ontologie, les langages d'ontologies et leur domaine d'application.

Nous avons aussi parlé à propos de l'alignement des ontologies et les concepts composés qui est l'objectif principale de ce travail.

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

Dans le troisième chapitre, afin de répondre à notre objectif, nous avons proposé une nouvelle approche d'alignement d'ontologies qui permet de traiter les concepts composés. Cette approche se base sur une combinaisons des mesures de similarité syntaxique, lexicale et structurelle pour obtenir des bons résultats d'alignement.

Sur la base de cette approche, nous avons implémenté notre système d'alignement. Des expérimentations ont été menées sur plusieurs ontologies contenant des concepts composés, afin de qualifier le travail effectué et de démontrer l'efficacité de notre approche dans la production

de bonne qualité, des tests sur les différentes ontologies générées ont été effectués en utilisant les mesures d'évaluation « Rappel » et « Précision ».

Enfin, on a eu quelques perspectives tel que l'utilisation d'autres mesures de similarité terminologiques et structurelles, afin d'améliorer la précision du système, tester le système sur des ontologies réelles et aussi calculer le temps d'exécution pour les différents types de mesures de similarité

Enfin, ce travail a été une expérience très enrichissante qui nous a permis d'apporter des tentatives de solution à un problème d'actualité.

Bibliographie

- [1] Charles Kay Ogden and Ivor Armstrong Richards. The meaning of meaning : A study of the influence of thought and of the science of symbolism. 1923.
- [2] Jérôme Euzenat and Pavel Shvaiko. Classifications of ontology matching techniques. *Ontology matching*, pages 61–72, 2007.
- [3] Jérôme Euzenat. Quelques pistes pour une distance entre ontologies. In *Actes 1er atelier EGC 2008 sur similarité sémantique*, pages 51–66. No commercial editor., 2008.
- [4] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R Swartout. Enabling technology for knowledge sharing. *AI magazine*, 12(3) :36–36, 1991.
- [5] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220, 1993.
- [6] WN Borst. Construction of engineering ontologies for knowledge sharing and reuse, ctit phd. *DTheis series*, (97-14), 1997.
- [7] Jérôme Euzenat, Amedeo Napoli, and Jean-François Baget. Xml et les objets (objectif xml). *Obj. Logiciel Base données Réseaux*, 9(3) :11–37, 2003.
- [8] Nathalie Hernandez. *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. PhD thesis, Université Paul Sabatier-Toulouse III, 2005.
- [9] Hanène Ghorbel, Afef Bahri, and Rafik Bouaziz. Les langages de description des ontologies : Rdf & owl. *Acte des huitièmes journées scientifique des jeunes chercheurs en Génie Electrique et Informatique (GEI), Sousse-Tunisia*, 2008.
- [10] Jean Charlet, Sylvie Szulman, Nathalie Aussenac-Gilles, Adeline Nazarenko, Nathalie Hernandez, Nadia Nadah, Éric Sardet, Jean Delahousse, Henry Valéry Téguiak, Audrey Baneyx, et al. Dafoe : une plateforme pour construire des ontologies à partir de textes et de thésaurus. In *EGC*, pages 631–632, 2010.

- [11] Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.
- [12] Edwina L Rissland. Ai and similarity. *IEEE Intelligent Systems*, 21(3) :39–49, 2006.
- [13] Mina Ziani, Danielle Boulanger, and Guilaine Talens. Système d’aide à l’alignement d’ontologies métier. *Ingénierie des systèmes d’information*, 16(1) :89–112, 2011.
- [14] Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-owl : Contextualizing ontologies. In *International Semantic Web Conference*, pages 164–179. Springer, 2003.
- [15] Chawki Djeddi, Imran Siddiqi, Labiba Souici-Meslati, and Abdellatif Ennaji. Text-independent writer recognition using multi-script handwritten texts. *Pattern Recognition Letters*, 34(10) :1196–1202, 2013.
- [16] Abdeltif Elbyed. *ROMIE, une approche d’alignement d’ontologies à base d’instances*. PhD thesis, Evry, Institut national des télécommunications, 2009.
- [17] Jorge Gracia and Kartik Asooja. Monolingual and cross-lingual ontology matching with cider-cl : evaluation report for oaei 2013. *OM*, 1111 :109–116, 2013.
- [18] Ming Li, Bo Lang, and Jinmiao Wang. Compound concept semantic similarity calculation based on ontology and concept constitution features. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 226–233. IEEE, 2015.
- [19] Edward Hitti, Tala Bakheet, Norah Al-Souhibani, Walid Moghrabi, Suhad Al-Yahya, Maha Al-Ghamdi, Maher Al-Saif, Mohamed M Shoukri, András Lánckzy, Renaud Grépin, et al. Systematic analysis of au-rich element expression in cancer reveals common functional clusters regulated by key rna-binding proteins. *Cancer research*, 76(14) :4068–4080, 2016.
- [20] Pavel Shvaiko, Jérôme Euzenat, E Jimenez-Ruiz, M Cheatham, and O Hassanzadeh. Ontology matching om-2018. In *CEUR Workshop Proceedings*, volume 2288, pages i–ii, 2018.
- [21] Sven Hertling and Heiko Paulheim. Dbkwik : A consolidated knowledge graph from thousands of wikis. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 17–24. IEEE, 2018.
- [22] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology : computer applications*, pages 231–243. Springer, 2010.
- [23] Younes Derfoufi. Programmation en langage python. 2019.
- [24] Jean-Baptiste Lamy. Owlready : Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*, 80 :11–28, 2017.

- [25] George A Miller. Wordnet : a lexical database for english. *Communications of the ACM*, 38(11) :39–41, 1995.
- [26] Holger Knublauch, Mark A Musen, and Alan L Rector. Editing description logic ontologies with the protégé owl plugin. In *International workshop on description logics*, volume 49. Citeseer, 2004.
- [27] Sylvie Despres and Jérôme Nobécourt. Quelles fonctionnalités pour un outil de visualisation d'ontologie. *Atelier VIF Visualisation d'informations, Interaction, et Fouille de données EGC*, 2018.
- [28] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*, 1994.