

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

UNIVERSITE SAAD DAHLEB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



Mémoire de Fin d'Etudes

Pour l'obtention

Du Diplôme de Master en Informatique

Option : Ingénierie de Logiciel

THEME

Passage à l'échelle pour l'alignement des ontologies

Réaliser par :

➤ **AISSANI ABDELGHEFFAR**

➤ **ATTOUT MOHAMED**

Soutenue le : 24/10/2019, devant le jury composé de :

- Présidente : Mme BOUSTIA
- Examinatrice : Mme GHEBGHOUB
- Promotrice : Mme FAREH

Année Universitaire : 2018/ 2019

Remerciements

Nous tenons en premier lieu à exprimer nos profondes gratitudee et nos vifs remerciements à Dieu qui nous a donné la force et la patience d'accomplir ce modeste travail.

Nous tenons à remercier notre promotrice Mme : FAREH M pour son aide précieuse.

Nous remercions sincèrement les membres de jury qui nous avoir fait l'honneur de juger ce travail.

Nos remerciements les plus chaleureux vont vers toute nos familles, nos parents pour leurs encouragements et leur soutenue depuis toujours.

Résumé

Dans la littérature actuelle, il existe très peu d'algorithmes d'alignement qui visent à traiter le problème du passage à l'échelle des méthodes d'alignement. L'une des solutions du passage à l'échelle suppose la possibilité de partitionner les ontologies en blocs avant de réaliser l'alignement. En effet, pour diminuer l'espace de recherche des correspondances, il faut limiter la taille des ensembles de concepts en entrée de l'outil d'alignement. Dans ce mémoire, nous proposons une solution qui traite l'alignement des ontologies volumineuses qui est basé sur le partitionnement.

Nous commençons par la phase de partitionnement d'ontologie en bloc, qui se base sur la structure d'ontologie, et les critères d'ingénierie ontologique, en effet, on parcourt la hiérarchie de l'ontologie, pour extraire tous les composants afin de construire les partitions, qui sont représentées par des représentants. Ces blocs sont les entrées du mapping. Pour la phase de mapping entre blocs, nous calculons les mesures de similarités sémantiques de chaque centre de blocs de la première ontologie avec tous les autres centres des blocs de la deuxième ontologie afin de trouver l'entité correspondante avec un sens équivalent ou le sens le plus proche pour produire comme sortie un ensemble de correspondances entre blocs. Ces résultats de mapping entre blocs sont utilisés pour découvrir le mapping entre concepts.

Summary

In the current literature, there are very few alignment algorithms to address the problem of scaling alignment methods. One of the scaling solutions assumes the ability to partition ontologies into blocks before performing alignment. Indeed, to reduce the search space of the matches, it is necessary to limit the size of the sets of concepts at the input of the alignment tool. In this thesis, we propose a solution that deals with the alignment of large ontologies that is based on partitioning.

We start with the bulk ontology partitioning phase, which is based on the ontology structure, and the ontological engineering criteria, in fact, we go through the ontology hierarchy, to extract all the components in order to build partitions, which are represented by representatives. These blocks are the inputs of the mapping. For the mapping phase between blocks, we calculate the semantic similarity measures of each block center of the first ontology with all the other centers of the blocks of the second ontology in order to find the corresponding entity with an equivalent meaning or the direction the closer to produce as output a set of matches between blocks. These mapping results between blocks are used to discover the mapping between concepts.

ملخص

في الأدبيات الحالية، هناك عدد قليل جداً من خوارزميات المحاذاة لمعالجة مشكلة توسيع نطاق أساليب المحاذاة. يتضمن أحد حلول القياس إمكانية تقسيم الأنطولوجيا إلى كتل قبل تنفيذ المحاذاة، ولتقليل مساحة البحث في التطابقات، من الضروري تحديد حجم مجموعات المفاهيم عند إدخال أداة المحاذاة. في هذه الرسالة، نقترح حلاً يتعامل مع محاذاة الأنطولوجيات الكبيرة والذي يعتمد على التقسيم..

نبدأ مع مرحلة تقسيم الأنطولوجيا إلى كتل، والتي تعتمد على هيكل الأنطولوجيا، ومعايير الهندسة الأنطولوجية، من أجل ذلك نسير من خلال التسلسل الهرمي للأنطولوجيا، لاستخراج جميع المكونات من أجل بناء أقسام، حيث كل قسم له عنصر (مركز) يمثله. هذه الكتل هي العناصر الأساسية لمرحلة المحاذاة. بالنسبة إلى مرحلة المحاذاة بين الكتل، نقوم بحساب مقاييس التشابه الدلالي لكل مركز كتلة في أول أنطولوجيا مع جميع المراكز الأخرى في الكتل الموجودة في الأنطولوجيا الثانية من أجل العثور على الكيان المقابل له في معنى أو الأقرب له من ناحية المعنى لإنتاج مجموعة من التطابقات بين الكتل. يتم استخدام نتائج المحاذاة بين الكتل لاكتشاف المحاذاة أو التعيين بين المفاهيم.

Sommaire

Introduction générale.....	10
----------------------------	----

Chapitre 1 : Etat de l'art sur les ontologies

1. Introduction.....	13
2. Ontologies.....	13
2.1. Définition.....	13
3. Composants des ontologies.....	14
3.1. Concepts.....	14
3.2. Les relations et fonctions.....	15
3.3. Les axiomes.....	15
3.4. Les instances.....	15
4. Classification des Ontologies.....	15
4.1. Selon l'objet de conceptualisation.....	15
4.2. Selon le niveau de détail.....	16
4.3. Selon le niveau de formalisme.....	16
4.4. Selon le niveau de complétude.....	17
5. Hétérogénéités des ontologies.....	18
5.1. Les hétérogénéités au niveau du langage.....	18
5.2. Les hétérogénéités au niveau ontologique.....	18
6. Ontologies et Systèmes d'information.....	19
7. Langages de Représentation des ontologies.....	20
7.1. XML et XML Schéma.....	20
7.2. RDF et RDF Schéma.....	20
7.3. Langage DAML+OIL.....	21
7.4. Langage OWL.....	21
8. Mapping d'ontologies.....	22
8.1. Définitions.....	23
8.2. Besoin d'alignement.....	24
8.3. Domaines d'application.....	25
9. Principe d'ingénierie ontologique.....	25
10. Conclusion.....	26

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

1. Introduction	28
2. Le mapping d'ontologie	28
2.1. Dimensions de l'alignement	28
2.2. Techniques de base pour l'alignement.....	30
3. La similarité.....	31
3.1. Définition de la similarité	31
3.2. Mesures Simples	32
3.3. Mesures combinées.....	34
4. Méthodes d'alignement d'ontologies	35
4.1. Comparaison de méthodes d'alignement	43
4.2. Analyse.....	47
5. Conclusion.....	48

Chapitre 3 : conception du système d'alignement d'ontologies

1. Introduction	49
2. Motivation	49
3. L'approche proposée pour l'alignement	50
3.1. Partitionnement	50
3.2. Détermination du représentant.....	56
3.3. Mapping entre blocs	56
3.4. Alignement entre concepts	57
3.5. Les mesures de similarité utilisées	58
4. Conclusion.....	62

Chapitre 4 : Implémentation du système

1. Introduction	63
2. Les outils utilisés.....	63
3. Présentation de l'application	65
4. Tests et validation.....	68
4.1. Evaluation.....	68
4.2. Résultats expérimentaux et discussion	70

5. Conclusion.....	70
Conclusion générale	71
Références	73

Table des figures

Figure 1: Le triangle sémantique (Ogden et al, 1923).....	15
Figure 2: Description d'une ressource selon le modèle RDF.....	20
Figure 3: Alignement comme solution pour l'interopérabilité sémantique.....	23
Figure 4: Mapping d'ontologies.....	23
Figure 5: Alignement d'ontologies.....	24
Figure 6: Processus d'alignement (Euzenat et al, 2007).....	29
Figure 7: Mesures de calcul de similarité (Euzenat et al, 2004).....	32
Figure 8: Schéma du Processus de partitionnement d'ontologie.....	51
Figure 9: Exemple d'un cas égal ($K = 2$).....	54
Figure 10: Exemple d'un cas inférieur ($K = 2$).....	55
Figure 11: Exemple d'un cas inférieur ($K = 3$).....	56
Figure 12: processus de mapping entre blocs.....	57
Figure 13: processus de mapping entre concepts.....	58
Figure 14: Extrait de l'ontologie UnivBench.....	61
Figure 15: Interface montre les boutons d'opération de partitionnement et de mapping.....	66
Figure 16 : Fenêtre de choix d'emplacement des deux ontologies.....	66
Figure 17: Interface montre les champs pour détermination de nombre de partitions.....	67
Figure 18: Onglet d'affichage de résultat.....	67
Figure 19: Affichage de résultats.....	68

Liste des tableaux

Tableau 1: Classification des techniques d'alignement.....	30
Tableau 2: Comparaison des méthodes d'alignement.....	46
Tableau 3: Les ontologies de tests.....	68
Tableau 4: Valeurs résultantes de rappel et de précision.....	70

Introduction générale

1. Contexte

Les ontologies ont été reconnues comme une composante essentielle pour le partage des connaissances et la réalisation de cette vision. Elles sont devenues très populaires, en occupant une place de choix dans le domaine de l'ingénierie des connaissances et ont prouvé leur efficacité pour la représentation des connaissances. Les ontologies se développent à toute vitesse dans différents domaines d'application réels, elles deviennent de plus en plus volumineuses en termes de nombre de concepts (milliers de concepts).

Au niveau du passage à l'échelle, le partitionnement d'ontologies en plusieurs partitions est nécessaire pour faciliter l'application de différentes opérations qui peuvent être effectuées sur les ontologies comme le mapping, la fusion et l'intégration.

Le mapping d'ontologies est un nouveau paradigme dans le web sémantique où les ontologies sont utilisées pour la représentation de l'information en utilisant un langage de description tel que OWL (Ontology Web Language). Le mapping tente de découvrir les correspondances entre les entités de deux ontologies afin de les combiner et de produire des nouvelles informations.

2. Problématique

L'alignement des ontologies est une solution pour résoudre l'hétérogénéité sémantique des données, en proposant des correspondances entre des entités sémantiques similaires de différentes ontologies. Donc le but de l'alignement consiste à déduire l'hétérogénéité, en effet, les ontologies ont été créées pour résoudre le problème de l'hétérogénéité des données, cependant elles sont devenues elles-mêmes source d'hétérogénéité qui se présente au niveau syntaxique et sémantique.

Dans les domaines d'applications réelles, les ontologies devenant de plus en plus volumineuses. Le partitionnement d'une ontologie en plusieurs partitions indépendantes les unes des autres sert à faciliter différentes opérations sur les ontologies tel que le mapping d'ontologies, ce dernier pose des difficultés qui peuvent se résumer par :

- L'hétérogénéité des ontologies vues la manière dont elles sont développées, à savoir l'hétérogénéité conceptuelle, terminologique, syntaxique et sémantique. En effet, l'hétérogénéité sémantique présente un défi majeur dans le processus d'intégration des ontologies, elle est due aux différentes interprétations des objets du monde réel.
- Leur nombre et leur taille qui ne cesse de croître vu l'augmentation continue des sources d'informations sur le web, d'où l'efficacité des méthodes de mapping diminue considérablement, vu la complexité de ce processus, avec la prise en compte de la dimension sémantique, sur un très grand nombre de concepts dans les ontologies initiales.

Objectifs

L'objectif de ce projet consiste à concevoir et de réaliser un système d'alignement d'ontologies, étant données deux ontologies représentées par le langage OWL (Ontology Web langage). En effet, le problème du passage à l'échelle a une très grande importance.

Une solution possible pour résoudre ce problème est d'essayer de limiter la taille des ensembles de concepts en entrée, et pour cela de partitionner les ontologies en plusieurs blocs, afin de n'avoir à traiter que des blocs de taille raisonnable.

Organisation du mémoire

Afin d'atteindre nos objectifs, ce mémoire est organisé en 4 chapitres :

Dans le premier chapitre, nous avons abordé une étude sur les ontologies, une vue d'ensemble sur les concepts de base des ontologies. Nous y aborderons les définitions, les composants, hétérogénéité des ontologies et les langages de représentation, une étude de la problématique "le mapping d'ontologies" qui est apparu dans le web sémantique, où nous avons abordé sa définition, le besoin d'alignement ainsi les domaines d'applications. Puis dans le Second Chapitre nous enchaînons avec un état de l'art sur les méthodes de mapping d'ontologies et une comparaison entre ces derniers, ce qui va nous aider à élaborer notre système avec les différentes étapes détaillées

Le Chapitre 3 détaille la conception de notre solution pour l'alignement, une description de méthode de partitionnement des ontologies, ainsi que les différentes mesures de similarité utilisées dans la phase d'alignement.

Le Chapitre4 est consacré pour détailler l'environnement de développement de notre approche, les outils utilisés, et une présentation visuelle de l'implémentation de notre système. Enfin, nous concluons ce travail par une évaluation de notre algorithme en termes de précision et rappel.

Chapitre 1 : Etat de l'art sur les ontologies

1. Introduction

L'apparition des ontologies entendues comme une nouvelle approche dans le domaine de la représentation des connaissances, elles ne sont considérées que relatives aux différents domaines de connaissances.

Les ontologies sont très utilisées dans la représentation des connaissances sur le Web, elles constituent le socle du Web sémantique. Grâce aux ontologies, les ressources ne sont plus décrites par des textes statiques mais plutôt, mobilisées et exprimées sous une forme permettant de refléter les concepts et leurs relations dans un vocabulaire commun facilitant ainsi le partage de la connaissance et son traitement par des non-humains.

Les ontologies répondent aux problèmes de représentation et de manipulation des connaissances, elles ont pour but de saisir la connaissance dans un domaine, d'une façon générale et de fournir une représentation communément acceptée qui pourra être réutilisée et partagée par diverses applications et groupes.

2. Ontologies

2.1. Définition

« Une ontologie est une spécification explicite et formelle d'une conceptualisation d'un domaine de connaissance » (Gruber, 1993).

La conceptualisation est le résultat d'une analyse ontologique du domaine étudié et donc est l'abstraction du monde de ce domaine. Cette conceptualisation est représentée dans une forme concrète, donc spécification, où les concepts, les relations entre eux et les contraintes pour les employer sont explicitement définis dans un format et langage formel.

Une ontologie définit les termes utilisés pour décrire et représenter le domaine de connaissance. Les ontologies sont utilisées pour partager un domaine de connaissance, ou un domaine d'intérêt spécifique, comme la médecine, la manufacture d'objets, l'immobilier etc... Les ontologies comprennent des définitions, traitables par machine, de concepts de bases du

domaine et leurs relations. Elles encodent les connaissances afférentes à un domaine mais qui peuvent aussi s'étendre à d'autres domaines.

De cette façon, elles doivent rendre ce savoir ré-exploitable. Le mot ontologie a été utilisé pour décrire des artefacts de différents degrés de complexité. Ceci va de simples taxonomies, comme la hiérarchie Yahoo, aux schémas de métadonnées comme le Dublin Core, et les théories logiques.

La définition de(Gruber, 1993)Nous semblons, caractériser le mieux l'essence d'une ontologie : « Une ontologie est une **spécification formelle**, et **explicite** d'une **conceptualisation partagée** »

3. Composants des ontologies

Les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants (Gruber, 1993).

3.1. Concepts

Les concepts, aussi appelés termes ou classe de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème) retenus en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie

On désigne un concept par un label (terme, nom). 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 représenter plusieurs objets.

Un concept peut donc décrire une tâche, une fonction, une action, une stratégie, un processus de raisonnement, etc.

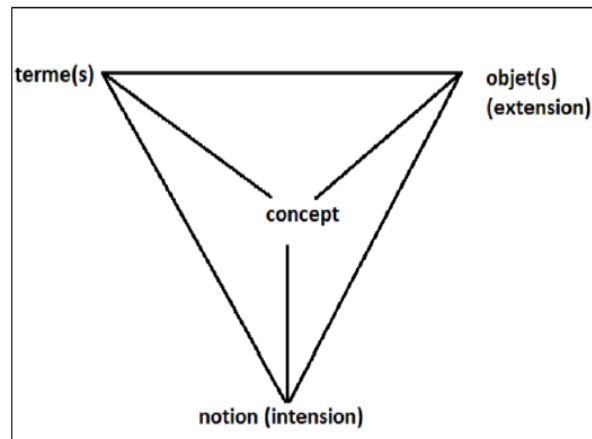


Figure 1:Le triangle sémantique (Ogden et al, 1923).

3.2. Les relations et fonctions

Les relations représentent un type d'interaction entre les notions d'un domaine. Une relation est définie comme une notion de lien entre des entités.

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.

3.3. Les axiomes

Les axiomes sont utiles à la structuration de phrases qui sont toujours vraies. Ils permettent de contraindre les valeurs de classes ou d'instances.

3.4. Les instances

Les instances sont utilisées pour représenter des éléments extensionnels spécifiques des concepts et des relations au domaine.

4. Classification des Ontologies

Il existe plusieurs critères de classification des ontologies :

4.1. Selon l'objet de conceptualisation

On distingue cinq types d'ontologie

- **Ontologie de représentation de connaissances**

Modélise les représentations primitives utilisées pour la formalisation des connaissances sous un paradigme donné.

- **Ontologie Générique**

Elle est appelée également noyau ontologique ou méta-ontologie, modélise des connaissances assez générales néanmoins pour être réutilisées à travers différents domaines.

- **Ontologie du domaine**

Cette ontologie exprime des conceptualisations spécifiques à un domaine, elle fournit les concepts et les relations permettant de couvrir les vocabulaires, activités et théories de domaines.

- **Ontologie de Tâches**

Fournit un vocabulaire systématisé des termes employés pour résoudre des problèmes liés aux tâches.

- **Ontologie d'application**

Elle contient des concepts dépendants d'un domaine et d'une tâche particulière, ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité.

4.2. Selon le niveau de détail

Cette classification est en fonction du degré de granularité, c'est-à-dire selon le niveau de détail des objets de la conceptualisation.

- **Granularité fine**

Correspondant à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche(FÜRST, 2004).

- **Granularité large**

Correspondant à un vocabulaire moins détaillé.

4.3. Selon le niveau de formalisme

(Uschold et al, 1996) ont identifié trois types d'ontologies

- **Les ontologies informelles**

Exprimées en langage naturel

- **Les ontologies semi informelles**

Elles sont exprimées sous une forme limitée, restreinte et structurée du langage naturel

- **Les ontologies formelles**

Exprimées dans un langage défini artificiellement et formellement, contenant une sémantique formelle.

4.4 Selon le niveau de complétude

Le niveau de complétude a été abordé par [Mizoguchi98] et [Bachimont00]. À titre d'exemple, nous décrivons la typologie de [Bachimont00]. Ce dernier propose la classification sur trois niveaux suivants :

- **Niveau sémantique**

Tous les concepts (caractérisés par un terme/libellé) doivent respecter les quatre principes différentiels :

1. Communauté avec l'ancêtre.
2. Différence (spécification) par rapport à l'ancêtre.
3. Communauté avec les concepts frères (situés au même niveau).
4. Différence par rapport aux concepts frères (sinon il n'aurait pas lieu de le définir).

Ces principes correspondent à l'engagement sémantique qui assure que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sémantiques sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

- **Niveau Référentiel**

Outre les caractéristiques énoncées au niveau précédent, les concepts référentiels (ou formels) se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension.

- **Niveau Opérationnel**

Outre les caractéristiques énoncées au niveau précédent, les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences (engagement computationnel). Deux concepts opérationnels sont identiques s'ils possèdent le même potentiel d'inférence.

5. Hétérogénéités des ontologies

Les ontologies visent à constituer une représentation réelle qui puisse être acceptée par une communauté, et que les langages de représentation des ontologies et les méthodes de conceptualisation sont différents, ce qui engendre des hétérogénéités entre les ontologies ce qui limitent les possibilités d'interopérabilité entre eux, et gênent l'intégration aussi.

L'hétérogénéité se manifeste sur deux grands niveaux, au niveau langage ou au niveau ontologique.

5.1. Les hétérogénéités au niveau du langage

Se manifestent quand le langage de spécification diffère d'une ontologie à l'autre : les classes, les relations ne sont pas définies de la même manière. Les hétérogénéités de ce niveau sont situées sur le plan de la syntaxe, de la représentation logique et de l'expressivité.

- **Syntaxe**

La syntaxe définit la structure des représentations, les différences dans ce plan concernent le formalisme plutôt que le contenu. La réécriture des ontologies dans un même langage suffit pour résoudre ce problème.

- **Représentation logique**

Les notions logiques sont exprimées de manière différente, par exemple dans un langage le fait que deux concepts A et B sont disjoints s'exprime par $A \cap B = \text{false}$, et que dans un autre langage cette même relation s'exprime par $A \cap B = 0$. Ce problème est facilement résolu par des règles de translation d'une représentation logique à l'autre.

- **Expressivité**

Implique que certains langages sont capables d'exprimer des notions alors que les autres ne peuvent pas, par exemple les listes, la négation, etc.

5.2. Les hétérogénéités au niveau ontologique

(Visser et al, 1999) Ont distingué dans ce niveau, les hétérogénéités conceptuelles et celles terminologiques.

- **Niveau terminologique**

Ce type d'hétérogénéité concerne toutes les différences liées au processus de nomination des entités (classes, propriétés etc.). Il est représenté très souvent par la présence des

synonymes, des homonymes, multilinguismes, les abréviations, etc. La difficulté major de ce niveau est la présence des homonymes qui n'est pas facile à résoudre.

- **Niveau Conceptuel**

Est appelé aussi hétérogénéité sémantique, c'est la différence dans l'interprétation du domaine qui a comme conséquence différents concepts ou différentes relations entre concepts.

6. Ontologies et Systèmes d'information

L'intégration d'une ontologie dans un système d'information vise à réduire, voire à éliminer, la confusion conceptuelle et trilogique à des points clefs du système, et a tendre vers une compréhension partagée pour améliorer la communication, le partage, l'interopérabilité et le degré de réutilisation possible, ce qui permet de déclarer formellement un certain nombre de connaissances utilisées pour caractériser les informations gérées par le système, et de se baser sur ces caractérisations et la formalisation de leur signification pour automatiser des tâches de traitement de l'information.

L'ontologie se retrouve maintenant dans une large famille de système d'information.

Elle est utilisée pour :

- Décrire et traiter des ressources multimédia.
- Assurer l'interopérabilité d'application en réseau.
- Permettre l'intégration des ressources hétérogènes d'information.
- Vérifier la cohérence de modèle.
- Faire des approximations logiques ; etc.

Ces utilisations des ontologies se retrouvent dans de nombreux domaines d'application tel que

- Intégration d'information géographique.
- Gestion de ressources humaines.
- Aide à l'analyse en biologie.
- Commerce électronique.
- Enseignement assisté par ordinateur.
- Bibliothèque numérique.
- Recherche d'information. Etc...

7. Langages de Représentation des ontologies

Le langage de description d'ontologies OWL (Smith et al, 2004) est le successeur d'un ensemble de technologies qui ont commencé avec le langage XML, passant par RDF, RDF Schema et arrivant au Langage (Ontology Web Language) OWL.

7.1. XML et XML Schéma

L'émergence d'XML, développé par le W3C comme un standard, constitue une première étape vers un langage de description d'ontologies sur le Web. XML est destiné plus particulièrement à l'échange de données (des documents) sur le Web, rend possible la description des concepts et leurs attributs ainsi que leur hiérarchisation sous forme de taxonomies à l'aide des balises. Cependant, l'XML reste très léger et limité quand il d'exprimer des contraintes et des relations entre des concepts ou des axiomes.

Ces lacunes ont motivé le développement d'autres langages comme DTD (Document Type Définition) et XMLSchema. En effet, DTD et XMLSchema offrent des grammaires qui permettent de spécifier des restrictions sur la structure des documents (taxonomies) et préciser les noms et le typage des attributs. Cependant, ils n'offrent aucun mécanisme pour exprimer des contraintes ou des relations sémantiques entre les concepts.

7.2. RDF et RDF Schéma

Le RDF (Ressources Description FrameWork) est un formalisme standard développé par le W3C pour décrire les ressources sur le Web (Decker & al, 2000). Le RDF ne représente pas un langage mais un méta-modèle pour décrire les ressources sur le Web en utilisant la syntaxe d'XML. Il est équivalent au formalisme des réseaux sémantiques. Ainsi, et suivant ce modèle, chaque ressource sur le Web peut être décrite sous forme d'un triplet composé d'une ressource, d'une propriété et d'une valeur (voir l'exemple donné par la *Figure 2*).

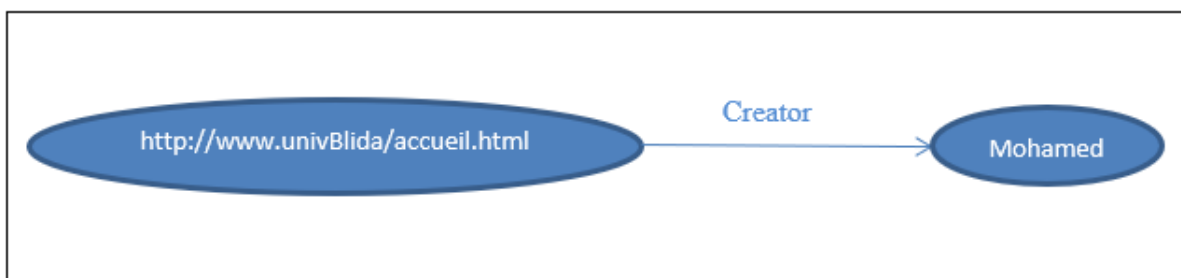


Figure 2: Description d'une ressource selon le modèle RDF.

- **Ressource**

Décrite par des expressions de RDF et nommée souvent par des URIs. Elle peut être une page Web ou une partie d'une page définie par une balise XML (la page Web identifiée par l'URI (<http://www.univBlida/accueil.html>) dans la **Figure 2**).

- **Propriété**

Représente un aspect particulier, une caractéristique, un attribut ou une relation qui décrit la ressource (la propriété Creator dans la **Figure 2**).

- **Valeur**

Peut-être à son tour une ressource, assignant une valeur à une propriété d'une ressource spécifique (la valeur Mohamed dans la **Figure 2**).

7.3. Langage DAML+OIL

DAML+OIL (Connolly et al, 2001) est le fruit d'un projet commun de la fusion de deux langages l'OIL (Ontology Interchange Language) développé par l'union européenne et DAML (DARPA Agent Modeling Language) développé par les États Unis dans le cadre du projet DARPA (The Defense Advanced Research Projects Agency). DAML+OIL est construit sur le modèle RDFs et basé sur le modèle théorique des logiques de descriptions. Il est considéré le premier langage à fournir des mécanismes d'inférence sur les concepts d'ontologies.

7.4. Langage OWL

Le langage OWL est le descendant de l'ensemble des langages ci-haut mentionnés, développé par le W3C comme une révision du langage DAML+OIL. OWL (Smith et al, 2004) permet de décrire les structures d'un domaine en termes de classes et de propriétés dans une approche similaire à celle de l'approche orienté objet. L'objectif visé par l'OWL est de permettre de décrire les ontologies dans un langage commun permettant ainsi le partage, l'import et l'export d'ontologies ainsi que de pallier au problème d'hétérogénéité des autres langages.

A l'instar de DAML+OIL, OWL s'appuie sur les fondements théoriques des logiques de descriptions : les classes et les propriétés dans OWL sont équivalentes aux concepts et aux rôles respectivement. OWL fournit une multitude d'opérateurs pour définir des concepts (classes, objets...) Plus complexes à partir des concepts élémentaires ou combinés. Cet aspect s'ajoutant à la possibilité de supporter toute sorte d'axiomes, dote le langage d'une grande expressivité. OWL offre aussi des mécanismes d'inférence et de raisonnement sur les

ontologies à savoir la vérification de la consistance de classes, de sous-assomptions et d'autres types d'assertions pour supporter l'intégration et le déploiement d'ontologies.

OWL vient en trois versions dont le degré d'expressivité et de décidabilité sont mutuellement inverses. Le but est de permettre un choix de langage offrant un compromis entre le degré d'expressivité et la qualité d'inférence permise. Les trois versions du langage sont :

7.5. OWL-Lite

Il est destiné aux cas de modélisation se limitant à une classification simple de concepts et de contraintes. Il supporte un sous ensemble de constructeurs comparativement à OWL-DL et OWL-Full.

7.6. OWL-DL

Permet une plus grande expressivité faisant usage un l'ensemble de constructeurs du langage OWL. En revanche la complétude et la décidabilité des algorithmes d'inférences ne sont pas garanties.

7.7. OWL-Full

Il est adressé à des utilisateurs qui cherchent un maximum d'expressivité et la liberté syntaxique de RDF. Ainsi, une classe par exemple, peut être considérée comme une collection d'individus ou comme un individu à part entière. OWL-Full permet par conséquent, d'augmenter le sens du vocabulaire déjà défini de RDF et celui d'OWL. Cependant, pareil au OWL-DL, la décidabilité et la complétude des inférences ne sont pas assurées.

8. Mapping d'ontologies

La réutilisation des ressources ontologiques est limitée à la diversité des ressources ainsi que l'hétérogénéité de leur contenu ce qui nécessite une réflexion approfondie sur la manière de les réutiliser. A ce stade l'alignement des ontologies intervient comme solution d'interopérabilité des ressources sémantiques indépendantes.

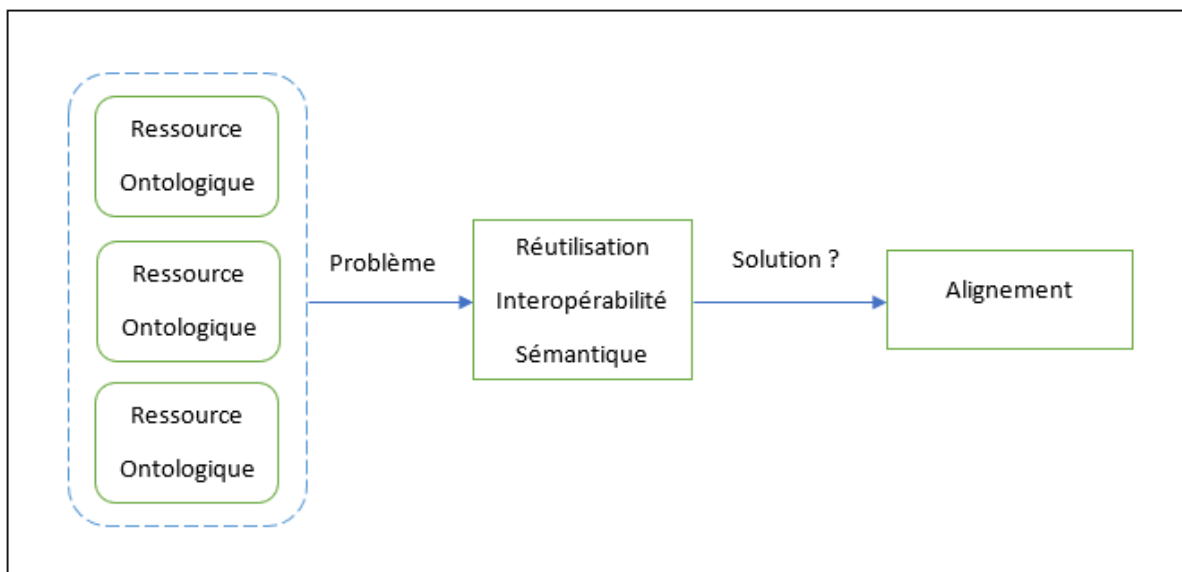


Figure 3:Alignement comme solution pour l'interopérabilité sémantique.

Cependant, trouver les concepts en commun entre les ressources ontologiques manuellement est une tâche fastidieuse, pratiquement infaisable dans le cas des ontologies volumineuses. Par conséquent, trouver des moyens pour automatiser ce processus aura un apport important dans le domaine d'ingénierie d'ontologies plus particulièrement celui du Web sémantique.

8.1. Définitions

- **Le mapping**

C'est l'expression formelle d'une relation sémantique entre deux entités appartenant à deux ontologies différentes. Lorsque la relation est orientée, elle devient une restriction de la signification mathématique du mapping, qui est la fonction.

Le résultat du processus de mapping représente l'ensemble des fonctions qui relient les concepts de différentes ontologies.

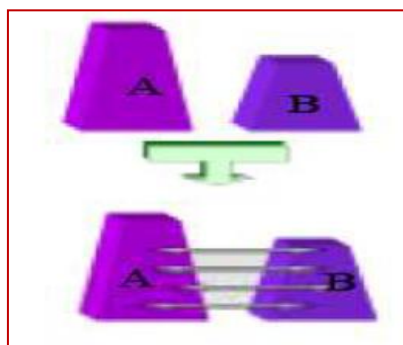


Figure 4:Mapping d'ontologies

- **L’alignement**

L’alignement d’ontologies consiste à déterminer, par un processus global, les correspondances (similarité) entre deux ontologies, en utilisant ou en mettant en œuvre des solutions aux différents problèmes d’hétérogénéité, et des méthodes qui permet de calculer la similarité (Bach et al, 2006).

Le mapping d’ontologies peut être vu comme un processus qui relie sémantiquement deux vocabulaires différents (Euzenat et al, 2007).

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 concevoir des ponts entre ces ontologies et de dériver d’autres informations, ou de procéder à d’autres opérations de manipulation comme la fusion ou l’intégration partielle. Pour cette raison, l’alignement est considéré comme une technique à base de toute autre opération de réconciliation entre les ontologies.

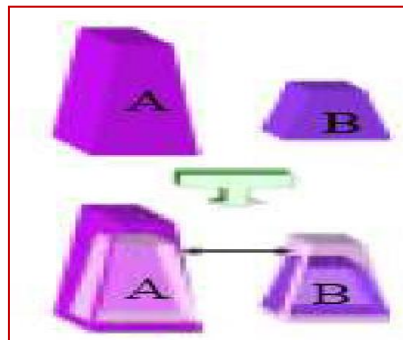


Figure 5:Alignement d’ontologies

8.2. Besoin d’alignement

L’interopérabilité sémantique entre les sources d’information est une problématique importante en raison du nombre croissant de sources d’information disponibles sur le web et leurs hétérogénéités.

L’alignement des ontologies est vu comme une solution prometteuse pour que les systèmes hétérogènes et les applications du web sémantique (où les informations sont présentées par des ontologies) soient interopérables. Il essaie de réduire l’hétérogénéité et permettre ainsi l’échange d’information d’une manière sémantique en créant une couche commune.

L'alignement est donc nécessaire pour pouvoir déterminer les concepts similaires entre les ontologies, il est parmi les techniques de réconciliation entre les ontologies et qui s'intéresse plutôt à trouver les entités similaires des ontologies en question.

8.3. Domaines d'application

Le mapping est considéré comme une étape primordiale et cruciale dans plusieurs domaines métiers. L'efficacité de ces systèmes dépend directement de l'efficacité du mapping à savoir la pertinence des résultats et sa rapidité dans l'exécution. Parmi ces systèmes, nous citons :

- **Ingénierie d'ontologie**

Le contexte dont les utilisateurs sont confrontés face aux ontologies hétérogènes est l'ingénierie d'ontologies, plus précisément, la tâche de concevoir, d'implémenter et de maintenir des applications basées sur ces dernières.

Cette activité nécessite un processus de mapping dû à la diversité des ontologies, leurs natures distribuées ainsi que leurs perpétuelles évolutions.

- **Intégration d'information**

L'intégration d'information est la plus ancienne classe d'application qui considère le mapping comme une solution probable (souhaitable). Il existe deux sortes d'intégration : L'intégration des schémas et l'intégration de données.

L'intégration des schémas a pour but de fusionner (d'intégrer) des schémas pour avoir un schéma global où la première étape consiste à trouver les correspondances sémantiques entre les entités des deux schémas avant qu'elles soient émergées. Cette étape est connue sous le nom de mapping.

L'intégration de données est un processus de combinaison des données appartenant à différentes sources et fournissant à l'utilisateur une vue unique de ces données appelée schéma globale. Pour intégrer les données de diverses sources d'information, les systèmes d'intégration de données doivent faire face à plusieurs problèmes majeurs comme l'hétérogénéité, la scalabilité et l'évolution. Comme étape nécessaire pour ce processus (le processus d'intégration de données) le mapping.

9. Principe d'ingénierie ontologique

Il existe un ensemble de critères et de principes qui ont fait leurs preuves dans le développement des ontologies et qui peuvent être résumés comme suit :

9.1. Clarté et Objectivité

L'ontologie doit fournir la signification des termes définis en fournissant des définitions objectives ainsi qu'une documentation en langage naturel (Gruber., 1993).

9.2. La Complétude

Une définition exprimée par des conditions nécessaires et suffisantes est préférée à une définition partielle (définie seulement par une condition nécessaire et suffisante) (Gruber., 1993).

9.3. Extensibilité maximale

De nouveaux termes généraux et spécialisés devraient être inclus dans l'ontologie d'une façon qui n'exige pas la révision des définitions existantes (Gruber., 1993).

9.4. Diversification des hiérarchies

Ce principe est adopté pour augmenter la puissance fournie par les mécanismes d'héritage multiple. Si suffisamment de connaissances sont représentées dans l'ontologie et que suffisamment de différentes classifications de critères sont utilisées, il est plus facile d'ajouter de nouveaux concepts (puisqu'ils peuvent être facilement spécifiés à partir des concepts et des classifications de critères pré-existants) et de les faire hériter de propriétés de différents points de vue (Arpirez et al., 1998).

9.5. Distance sémantique minimale

Il s'agit de la distance minimale entre les concepts enfants de mêmes parents. Les concepts similaires sont groupés et représentés comme des sous-classes d'une classe, et devraient être définis en utilisant les mêmes primitives, considérant que les concepts qui sont moins similaires sont représentés plus loin dans la hiérarchie (Arpirez et al., 1998).

10. Conclusion

Dans ce chapitre, nous avons rappelé la notion des ontologies, leurs différents types et composants, ainsi l'hétérogénéité des ontologies.

Les ontologies sont représentées au moyen de langages formels dédiés, offrant des structures de données adaptées à la représentation de concepts. Dans ce chapitre nous avons présenté les langages les plus utilisés pour la représentation des ontologies.

Dans le noyau du Web sémantique, les ontologies permettent de capitaliser, de représenter, d'exploiter et de partager sémantiquement des connaissances et des informations. En tant qu'entités du Web, un environnement très large, très complexe et divers, les entités du Web

sémantique possèdent aussi des caractéristiques de cet environnement si hétérogène. Les ontologies du Web sémantique peuvent donc être variées et hétérogènes même si elles sont créées dans un même domaine. Une des clés importantes pour le succès du Web sémantique est de maîtriser cette hétérogénéité et de cohabiter avec elle.

L'alignement des ontologies représente un grand intérêt pour plusieurs domaines d'applications qui manipulent des connaissances hétérogènes. A cet effet, nous allons présenter dans le chapitre suivant les différentes méthodes d'alignement.

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

1. Introduction

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

Ce chapitre a pour premier objectif d'aborder les techniques d'alignement dans le domaine du web sémantique en présentant quelques critères sur lesquels une comparaison doit s'appuyer pour les classifier.

2. Le mapping d'ontologie

Le mapping entre deux ontologies « o1 ; o2 » signifie que pour chaque entité dans une ontologie « o1 », il faut trouver l'entité correspondante dans l'ontologie « o2 » avec un sens équivalent ou le sens le plus proche. La caractéristique importante du mapping est qu'il produit en sortie un ensemble de correspondances sans modifier les deux ontologies impliquées.

Nous présentons les différents éléments du processus de mapping entre ontologies :

2.1. Dimensions de l'alignement

L'alignement regroupe trois dimensions : *l'input, le processus d'alignement et l'output.*

- **L'input**

Est constitué essentiellement des structures destinées à être alignées et qui peuvent être, comme énoncé précédemment, des schémas XML, des schémas relationnels, des ontologies décrites en OWL, RDFS...etc, ou des instances d'une ontologie.

L'input peut être enrichi par un alignement en entrée (qui aurait besoin d'être complété par une nouvelle itération d'alignement).

- **Le processus d'alignement**

Comme le montre **Figure 6** le processus d'alignement peut être considéré comme une fonction f , qui donne un alignement A' entre une paire d'ontologie O et O' à partir de ces deux derniers, un alignement en entrée A (optionnel), un ensemble de paramètre p (ex : paramètres de pondération, seuils ...) et un ensemble de ressources externes r (ex : thésaurus, lexicque, etc.), $A' = f(O, O', A, p, r)$. Ceci peut être représenté schématiquement de la manière suivant :

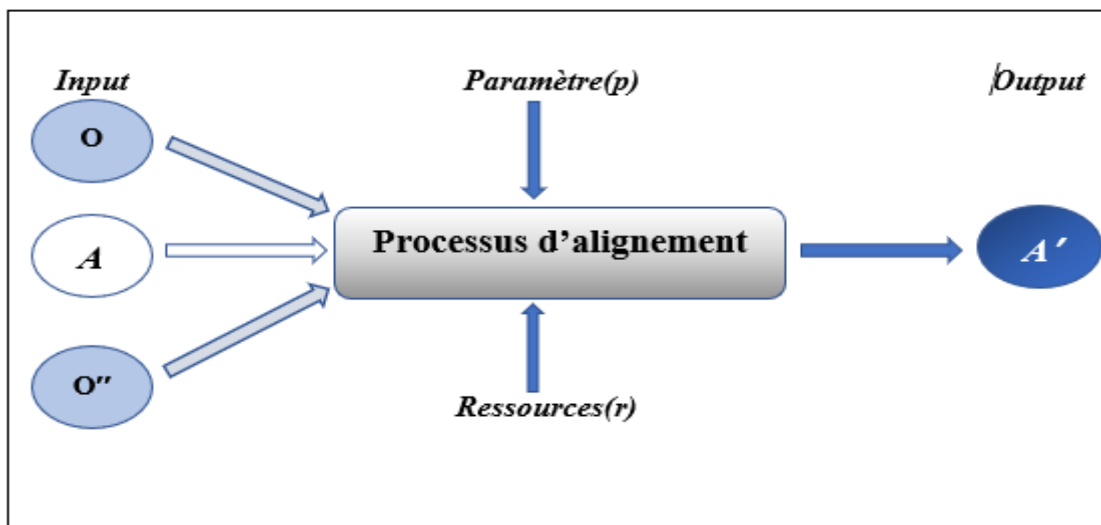


Figure 6: Processus d'alignement (Euzenat et al, 2007)

- **L'output**

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

Un alignement est décrit par un ensemble de cinq éléments :

$\langle id, e, e', r, n \rangle$ telle que :

- id : identifiant unique d'un mapping ;
- e : une entité, à aligner, appartenant à O (class, propriété, contraint, instance) ;
- e' : une entité, à aligner, appartenant à O' ;

- r : la relation qui relie e à e' ($=, \supseteq, \subseteq, \prod, \perp$) ;
- n : la mesure de confiance de la relation r , généralement une valeur réelle compris entre 0 et 1, $n \in [0, 1]$. Plus le n est proche du 1, plus la relation est considérée comme étant fort.

2.2. Techniques de base pour l'alignement

Les techniques représentées en deux principales catégories comme le représente le tableau ci-dessus :

Techniques basées sur les éléments isolés	Techniques basées sur la structure
<ul style="list-style-type: none"> • Techniques syntaxiques. • Techniques fondées sur les chaînes de caractères. • Techniques fondées sur le langage. • Techniques à base de contraintes. • Techniques utilisant des sources externes. • Des ressources linguistiques : lexiques, thésaurus. • Réutilisation. 	<ul style="list-style-type: none"> • Techniques syntaxiques. • Techniques fondées sur l'analyse de données et les statistiques : fréquence de distribution. • Techniques fondées sur les graphes : recherche d'homomorphismes, de chemins. • Techniques fondées sur des taxonomies. • Techniques utilisant des sources externes. • Techniques fondées sur des bibliothèques de structures. • Techniques sémantiques. • Techniques fondées sur le modèle des données : mécanismes de raisonnement en LD.

Tableau 1: Classification des techniques d'alignement.

3. La similarité

Il s'agit de la 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 similitude de leurs significations ou de leurs contenus sémantiques. La similarité est la quantité qui reflète la force du rapport entre deux objets ou deux caractéristiques.

3.1. Définition de la similarité

Définition 1 (similarité) : (Bach, 2006) $S : O \times O \rightarrow R$ est une fonction d'une paire d'entités à un nombre réel exprimant la similarité entre ces deux entités telle que :

- ❖ $\forall a, b \in O, S(a, b) \geq 0$ (positivity).
- ❖ $\forall a, b, c \in O, S(a, a) \geq S(b, c)$ et $S(a, a) = S(a, b) \Leftrightarrow a = b$ (auto similarité 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).

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

Définition 2 (distance) $D : O \times O \rightarrow R$ est une fonction de la dissimilarité satisfaisant la définitivité et l'inégalité triangulaire :

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

Les valeurs de similarité sont souvent normalisées pour pouvoir être combinées dans des formules plus complexes. Si la valeur de similarité et la valeur de dissimilarité entre deux entités sont normalisées, notées S et DS , alors on a $S + DS = 1$.

Définition 3 (Normalisation) N : Une mesure est dite normalisée si les valeurs calculées par cette mesure ne peuvent varier que dans un intervalle de 0 à 1. Ces valeurs calculées sont appelées valeurs normalisées. Les fonctions du calcul sont appelées fonctions normalisées et notées f .

Les mesures de la similarité, de la dissimilarité, de la distance peuvent être classées selon la nature des entités que l'on veut comparer : des termes, des chaînes de caractères, des structures, des instances (des individus des classes).

La Figure ci-dessus résume différentes mesures de similarité, catégorisées selon les techniques utilisées. Ce résumé est une synthèse des travaux présentés dans (Rahm et al, 2001), (Euzenat et al, 2004) et (Shvaiko et al, 2005).

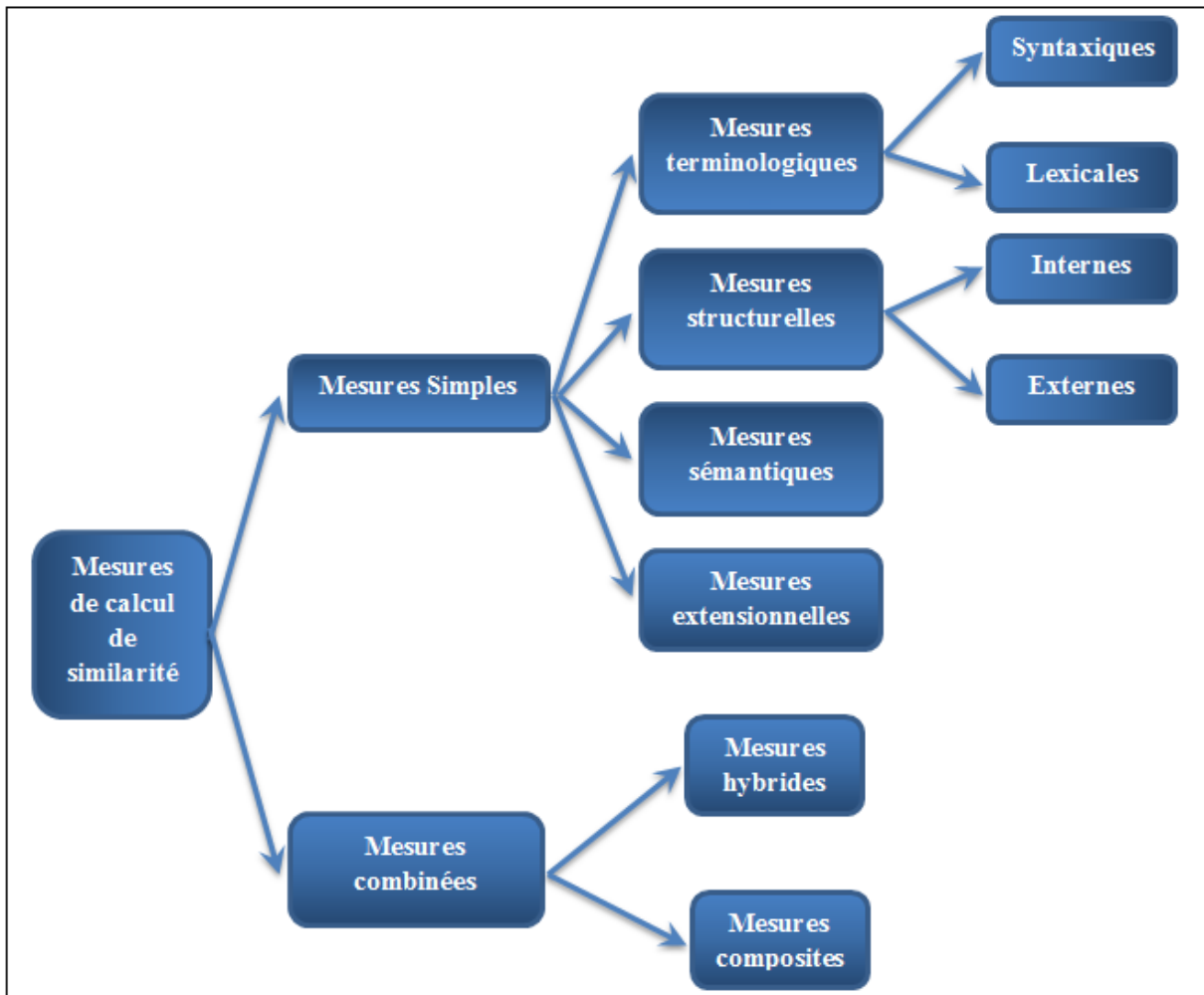


Figure 7: Mesures de calcul de similarité (Euzenat et al, 2004)

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

3.2. Mesures Simples

Elle regroupe les mesures suivantes :

- **Mesures terminologiques**

Ces mesures reposent sur la comparaison des termes ou des chaînes de caractères ou bien les textes. Elles sont exploitées pour calculer la valeur de la similarité des entités textuelles, telles que des 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 (Monge et al, 1996) :

- **Approche syntaxique** : approche basée sur les chaînes de caractères (texte) : comme leur nom l'indique, ces méthodes considèrent l'entité comme une séquence de lettres. Les résultats obtenus par ces méthodes sont utiles si les concepteurs utilisent des chaînes de caractères similaires pour définir la même entité, en revanche, s'il y a des synonymes avec des structures différentes ces méthodes donnent une mauvaise estimation de la similarité.
- **Approche lexicale ou linguistique** : les informations exploitées peuvent être celles intrinsèques (des propriétés linguistiques internes des termes telles que des propriétés morphologiques ou syntaxiques) ou celles extrinsèques en effectuant la correspondance à travers les relations lexicales (par exemple, synonymie, hyponymie, etc.) en faisant appel à des ressources auxiliaires telles que les dictionnaires de synonymes et hyponymes, mais aussi à des thésaurus et des ressources sémantiques ainsi qu'à des dictionnaires spécifiques aux domaines étudiés.

- **Mesures structurelles**

Elles se focalisent sur la comparaison des structures des entités à mapper, en fonction de la nature interne ou externe des structures à comparer, on peut distinguer :

- **Méthodes structurelles internes** : qui utilise la comparaison de structure interne d'une entité (ex : la comparaison des attributs, des noms, des types des attributs...) indépendamment des autres entités, ces méthodes sont généralement combinées avec celles des techniques terminologiques. Ce type de méthodes est très important pour le mapping du fait qu'il peut éliminer les correspondances incompatibles, cependant, il n'offre pas vraiment beaucoup d'information sur les entités à comparer, plusieurs entités très hétérogènes peuvent être décrites par la même structure interne et des entités très proches peuvent avoir des structures différentes.

- **Méthodes structurelles externes** : qui utilisent la comparaison externe en mettant par exemple en jeux la disposition des entités dans leur hiérarchie, le voisinage, ...

Le mapping qui utilise ces méthodes est très performant parce qu'il prend en compte toutes les relations entre les entités ce qui nécessite l'utilisation d'autres méthodes terminologiques.

- **Mesures sémantiques**

La similarité sémantique est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens. La définition donnée par Lin de la similarité sémantique repose sur trois suppositions. La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun (plus ils ont de caractéristiques communes, plus les concepts sont similaires) et à leurs différences (plus deux concepts sont différents, moins ils sont similaires). La similarité maximale est obtenue lorsque deux concepts sont identiques (Mellal, 2007).

- **Mesures extensionnelles**

Elles résultent de la similarité entre deux entités qui sont notamment des concepts ou des classes tout en analysant ainsi leurs extensions (leurs ensembles d'instances). 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 (Ziani et al, 2010).

On distingue deux approches pour comparer les ontologies à partir des instances associées aux concepts d'ontologies :

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

3.3. Mesures combinées

Ces mesures combinent plusieurs mesures lorsqu'une seule est insuffisante (Leacock et al, 1998). Il existe deux types de combinaison :

- **La combinaison séquentielle (hybride)**

La méthode la plus simple pour combiner les mesures est l'utilisation séquentielle de ces dernières en choisissant un ordre d'exécution. Par exemple, nous choisissons de lancer une mesure terminologique avant de lancer une autre mesure structurelle ou sémantique (Elbyed, 2009).

- **La combinaison parallèle (composite)**

Une autre manière de combiner les résultats des différentes mesures (c.-à-d. les valeurs de similarité) consiste tout d'abord à lancer parallèlement plusieurs mesures, puis par la suite à combiner leurs résultats (Elbyed, 2009).

4. Méthodes d'alignement d'ontologies

Les techniques d'alignement jouent un rôle capital dans la construction d'un lien sémantique entre les ontologies d'un même domaine. Donc la diversité d'approches vient du fait que chaque approche utilise les méthodes adéquates selon les types d'ontologies à aligner. Dans la littérature, plusieurs méthodes d'alignement d'ontologies ont été proposées, ces méthodes exploitent des ontologies décrites dans différents langages (RDF(S), DAML+OIL, OWL, etc.), elles sont souvent testées sur des ontologies de petite taille (quelques centaines de concepts). Quand les ontologies sont de très grande taille, par exemple en Agronomie ou en Médecine, des ontologies comportent plusieurs dizaines de milliers de concepts (AGROVOC : 28 439, NALT : 42 326, NCI : 27 652), l'efficacité des méthodes d'alignement automatique diminue considérablement la précision des mappings obtenus du fait de l'augmentation du bruit. Cependant, il existe très peu d'algorithmes d'alignement qui visent à traiter le problème du passage à l'échelle des méthodes d'alignement.

Le partitionnement est un processus au cours duquel une ontologie O est fractionnée en un ensemble de partitions P tel que l'union de l'interprétation de toutes les partitions ainsi créées soit équivalente à l'interprétation de l'ontologie initiale.

Dans cette partie, nous allons analyser des approches déjà existantes dans la littérature qui concernent le problème d'alignement d'ontologies. Nous allons voir un résumé sur quelques approches :

- **Anchor-PROMPT** (Noy et al, 2001)

Est un algorithme qui permet de trouver des concepts qui sont sémantiquement similaires dans des ontologies différentes, cela en se basant sur : une liste des paires d'ancres comme entrée de l'algorithme, ces paires d'ancres sont les paires de concepts similaires définis par les utilisateurs ou trouvés en utilisant des méthodes de correspondance lexicologique. Aussi sur le graphe qui représente l'ontologie, où les nœuds sont des concepts (classes) et les arcs représentent les relations (slots) entre les concepts, cet algorithme fait une différence entre les relations de type « is-a » et les autres relations car les classes liées par « is-a » constituent des groupes d'équivalences qui sont considérés comme un seul nœud. L'algorithme analyse les chemins dans les sous-graphes limités par les ancres pour déterminer quels concepts apparaissent fréquemment en positions similaires sur les chemins similaires. Ces classes sont susceptibles de représenter les concepts qui sont sémantiquement similaires. Les résultats de cet algorithme sont liés à la taille de la liste des paires d'ancre initiale et la longueur maximale du chemin. Cet algorithme trouve des problèmes si les structures des ontologies sont différentes car il ne trouvera pas les chemins similaires avec la même longueur et si les noms de relations sont différents.

- **GLUE** (Doan et al, 2002)

Est la version évoluée de LSD (Doan et al, 2000) dont le but est de trouver semi automatiquement des correspondances entre des schémas pour l'intégration de données. Comme LSD, GLUE utilise la technique d'apprentissage (telle que Naïve Bayes) pour trouver des correspondances entre deux ontologies. GLUE comprend plusieurs modules d'apprentissage (learners), qui sont entraînés par des instances des ontologies. Ces modules emploient la technique d'apprentissage Bayes naïf (Domingos et al, 1997) en exploitant différentes caractéristiques des instances telles que les valeurs textuelles des instances, les noms des instances, les formats des valeurs.... Les prévisions de ces modules de mise en correspondance sont combinées par un méta module de mise en correspondance en employant la somme pondérée. Le résultat final des correspondances sera déduit à partir des valeurs de similarité agrégées en employant la technique d'optimisation de contraintes « relaxation labeling » (la technique permettant de résoudre le problème d'assignement des étiquettes aux nœuds d'un graphe en donnant un ensemble de contraintes). Un inconvénient de cette approche est qu'elle se fonde principalement sur les instances des ontologies, qui ne sont pas toujours abondamment disponibles pour plusieurs ontologies. Un autre inconvénient est que

l'ontologie est modélisée comme une taxonomie des concepts et que chaque concept a quelques attributs. Avec cette organisation, GLUE n'emploie pas des informations contenues dans la taxonomie (hiérarchie) des relations. GLUE fait également l'utilisation modeste des informations sur la taxonomie des concepts.

- **QOM** (Ehrig et al, 2004)

L'approche QOM (*Quick Ontology Matching*) est basée sur l'approche NOM (Naïve Ontology Matching) cette dernière a montré qu'elle est effective, mais inefficace, donc l'auteur a pensé à l'optimiser, la chose qui a donné naissance à l'approche QOM. Cette approche comme son prédécesseur exploite RDF-triples, et pour mesurer la similarité entre les ontologies à aligner elle utilise la similarité terminologique et structurelle. Et, après la phase de l'analyse de la similarité et ses interprétations, de nouvelles décisions doivent être prises, comme : quels alignements candidats doit-on ajouter à l'agenda pour l'itération suivante.

Cette approche a montré de très bons résultats en termes de temps d'exécution ($n \cdot \log(n)$ au lieu de n^2 , n étant le nombre d'entités dans les ontologies) en comparaison avec d'autres approches de la même classe de complexité.

- **S-Match** (Giunchiglia et al, 2004)

Est un algorithme et un système pour chercher sémantiquement des correspondances, basé sur l'idée d'employer le moteur de la satisfiabilité propositionnelle (SAT) (Giunchiglia et al, 2003) pour le problème de mise en correspondance des schémas. Il prend comme entrée deux graphes des concepts (schémas), et produit en sortie des rapports entre les concepts tels que l'équivalence, overlapping, différence (mismatch), plus général ou plus spécifique. L'idée principale de cette approche est d'utiliser la logique pour coder le concept d'un nœud dans le graphe et d'appliquer SAT pour trouver des rapports. Le concept à un nœud, qui est alors transformé en formule propositionnelle, est la conjonction de tous les concepts des étiquettes des nœuds sur le chemin de la racine du graphe jusqu'au nœud en question. Le concept d'étiquette d'un nœud est construit en deux étapes :

- i. la normalisation de l'étiquette : telle que la tokenization, la lemmatisation.

- ii. L'extraction du sens de l'étiquette normalisée (des lemmes) à partir de WordNet (Miller, 1995). Ensuite, les relations sémantiques (l'équivalence, plus général, plus spécifique) entre deux étiquettes de deux schémas sont :

- Calculées grâce aux « matchers », les modules qui calculent la similarité entre deux étiquettes en employant des mesures de similarité de base telles que la similarité des préfixes, des suffixes, la distance d'édition, la similarité de n-gram.
- Ou bien déduites en employant des matchers qui exploitent la sémantique dans WordNet, la similarité entre des hiérarchies, la similarité entre des commentaires (Giunchiglia et al, 2004). Ces relations sémantiques est aussi encodée en logique. Enfin, le rapport entre deux concepts qui doit être prouvé est également converti en formule propositionnelle. Le moteur SAT calcule sur l'ensemble de formules propositionnelles pour vérifier si le rapport supposé est vrai ou faux. Cela permet donc de déduire des correspondances entre deux ontologies.
- **OLA** (Euzenat et al, 2004)

(*OWL Lite Alignment*) Est un algorithme pour aligner des ontologies représentées en OWL. Il transforme les ontologies sous forme d'un graphe OL-graphs qui représente toutes les caractéristiques possibles des ontologies représentées en OWL-Lite. Les nœuds du graphe sont : classe (C), objet (O), relation (R), propriété (P), instance de propriété (A), type de donnée (D), valeur de type de donnée (V), les labels de restriction de propriété (L). Les arcs entre les nœuds sont :

- *rdfs:subClassOf* entre deux classes ou deux propriétés (S);
- *rdf:type* (I) entre les objets et les classes, instance de propriété et propriété, valeurs et types de données ;
- *A* entre les classes et les propriétés, objets et les instances de propriétés ;
- *owl:Restriction* (R) exprimant la restriction sur une propriété dans une classe;
- *Valuation* (U) d'une propriété d'un individu.

OLA utilise les différentes méthodes de calcul de similarité pour trouver les correspondances entre deux entités dans deux ontologies en se basant leurs caractéristiques (leurs types : classe, relation ou instance, leurs rapports avec d'autres entités : sous-classe, domaine, Co-domaine...) et de combiner les valeurs de similarités calculées pour chaque paire d'entités de manière homogène. La combinaison est la somme pondérée des valeurs de similarité de chaque caractéristique.

À partir des valeurs de similarité calculées par des mesures de base, l'algorithme applique un calcul du point fixe, avec des itérations pour améliorer la similarité de deux entités. Quand il n'y a plus d'améliorations, des alignements entre deux ontologies sont générés.

- **AROMA** (David, 2007).

(Association Rule Ontology Matching Approach) Est une approche d'alignement pour des ontologies représentées en OWL. Elle permet de découvrir des liens sémantiques de type « subsomption » ou « équivalence » entre deux entités (classes ou propriétés). Le processus d'alignement se déroule en trois étapes : la première procède à l'acquisition des termes contenus dans les descriptions et instances des entités à partir d'outils de Traitement Automatique du Langage (TAL). Ensuite, pour chaque entité, ainsi qu'à ses ancêtres est associé un ensemble de termes dits représentatifs. La deuxième étape permet de créer des relations de subsomptions entre les entités à partir de règles d'association construites sur le principe qu'une entité X est plus spécifique ou équivalente à une entité Y si le vocabulaire utilisé dans les descriptions et les instances de X a tendance à être inclus dans celui de Y. Des 'matchers' terminologiques sont utilisés pour comparer les différentes descriptions. Enfin, la dernière étape vise à analyser les règles d'associations trouvées afin de : (1) déduire des relations d'équivalence ; (2) trouver des incohérences (cycles) et de les éliminer (3) ; supprimer les relations redondantes ; (4) sélectionner le meilleur alignement pour chaque entité.

- **ASMOV** (Jean-Mary et al, 2007)

(Automated Semantic Mapping of Ontologies with Validation) Est un système d'alignement d'ontologies conçu pour l'intégration de sources de données hétérogènes représentées dans des ontologies. ASMOV permet de produire des mappings entre des concepts et/ou des propriétés et/ou des instances de deux ontologies. L'algorithme implémenté est automatique, il calcule de façon itérative, la similarité entre deux entités appartenant à deux ontologies différentes suivant quatre caractéristiques : (1) les éléments lexicaux (labels, commentaires) ; (2) les relations structurelles (ancêtres/descendants dans la hiérarchie) ; (3) la structure interne (restrictions sur les propriétés pour les concepts ; types, domaines et intervalles pour les propriétés ; valeurs pour les instances) ; (4) et les extensions (instances de classes et valeurs des propriétés). La similarité finale est calculée à partir de la somme pondérée des quatre mesures et permet d'obtenir un alignement. Le système vérifie ensuite cet alignement afin de s'assurer qu'il ne contienne pas d'incohérence sémantique.

- **ROMIE** (Elbyed, 2009)

(Resource based Ontology Mapping within and Interactive and Extensible environment) est une approche qui regroupe les méthodes d'alignement syntaxiques, linguistiques, structurelles et sémantiques pour offrir un alignement semi-automatique.

Cette approche se divise en deux phases : la phase d'enrichissement sémantique et la phase d'alignement. La première est basée sur l'analyse des informations développées par les ontologies à comparer (ressources web, données, documents, ...) et qui sont associées aux concepts de l'ontologie. A la fin de la phase d'enrichissement, une ontologie contient plus de relations sémantiques entre les concepts qui seront exploitées dans la deuxième phase. La phase d'alignement, prend deux ontologies et calcule la similarité entre les couples de chaque concept d'ontologies en employant plusieurs méthodes syntaxiques qui calculent par exemple la distance d'édition entre deux concepts, et/ou encore des méthodes linguistiques basées sur le dictionnaire WordNet, par exemple. Puis selon les valeurs de similarité retournées, l'auteur génère un ensemble d'alignements candidats et en utilisant des méthodes de filtrage, et élimine les alignements les moins pertinents. Ces méthodes sont basées sur des relations structurelles et sémantiques entre les concepts. En fin l'ensemble des hypothèses d'alignements fournis par l'étape précédente est ordonné selon les valeurs de similarité.

- **TaxoMap** (Kasri, 2010)

TaxoMap est un outil d'alignement qui a pour objectif de permettre un accès unifié via le Web aux documents d'un même domaine d'application. Il est adapté au traitement de taxonomies dont les structures sont hétérogènes et dissymétriques. L'objectif de TaxoMap est de mettre en correspondance les concepts de la taxonomie la moins structurée, la taxonomie source.

- **COMA++**(Massmann et al, 2011)

Un système dédié à l'alignement des graphes de schémas larges. La méthode proposée se base sur l'idée 'diviser pour régner'. Une ontologie est décomposée en des clusters (appelés fragments) qui seront, par la suite, mis en correspondance. Le clustering de l'ontologie est basé sur un ensemble de règles heuristiques prédéfinies pour la production d'un ensemble de clusters de taille réduite. Chaque cluster est identifié par un concept racine qui sera utilisé pour la comparaison des clusters dans la phase d'alignement de clusters. En effet, deux clusters dont les deux concepts racine sont sémantiquement similaires, ceci implique la

similarité des deux clusters. L'alignement sera ensuite mené sur les entités appartenant aux clusters jugés similaires en utilisant la technique structurelle et terminologique.

- **Travail de** (Kachroudi et al, 2013)

Les auteurs proposent également une méthode initiée par les concepts terminologiquement équivalents dans les deux ontologies à aligner. Le concept initié dans l'ontologie cible est considéré comme étant le centre d'un nouveau block. Ainsi, les concepts sémantiquement et structurellement similaires au centre seront rajoutés dans le cluster. A cette fin, les auteurs proposent d'utiliser la ressource externe WordNet (Miller, 1995) pour examiner le voisinage de chaque concept. Pour chaque concept, on identifie la liste des termes qui lui sont sémantiquement équivalents à partir de WordNet. De ce fait, chaque concept similaire sera assigné au cluster d'intérêt. Ceci permet d'assurer la cohésion des clusters produits. Le même traitement sera répété pour chaque concept rajouté au cluster. Le processus s'arrête lorsqu'il n'y aura aucun lien sémantique entre les concepts de cluster et les concepts restants ou lorsque la taille du cluster atteint une valeur maximale. Une fois que le partitionnement de l'ontologie cible est achevé, l'auteur crée le même nombre de clusters dans l'ontologie source. La phase d'alignement finale consiste à aligner les entités des clusters correspondants aux ancres (les concepts des ontologies source et cible).

- **Travail de** (Algergawy et al, 2014)

Le système d'alignement proposé est initié par la phase de clustering d'ontologies où chaque ontologie est clustérisée indépendamment en un ensemble des clusters disjoints. La tâche de clustering se base sur une mesure de similarité structurelle afin d'identifier les éléments structurellement similaires. Chaque concept sera assigné à un cluster, ensuite un algorithme itératif sera déclenché. Ce dernier consiste à fusionner les clusters structurellement similaires. Le processus s'arrête lorsque la taille du cluster atteint une valeur maximale. La deuxième phase, consiste à identifier les clusters source et cible similaires.

A cette fin, chaque cluster est converti en un vecteur de termes. Une fois les clusters similaires sont identifiés, on procède à l'alignement des éléments correspondants.

- **Travail de** (Setti et al, 2015)

Dans le but de partitionner une ontologie volumineuse donnée en modules de haute qualité, Setti et al, ont proposé une approche de partitionnement qui introduit une amélioration de

l'algorithme de classification k-means, en se basant sur une nouvelle mesure de similarité sémantique « Dennai ». C'est une amélioration de la mesure de similarité structurelle : Wu et Palmer.

L'approche consiste à extraire d'abord tous les constructeurs de l'ontologie (classes, axiomes, relations et propriétés), puis à convertir l'ontologie représentée en OWL sous forme d'un graphe. Après, différentes mesures de similarité sont calculées d'un nœud donné vers tous les nœuds du graphe. Il s'agit de déterminer le chemin le plus court entre un nœud donné et un autre nœud du graph. Cette étape génère différentes matrices de mesures de similarité. Ensuite, deux algorithmes de clustering sont utilisés. Le premier est introduit pour adapter l'algorithme traditionnel au clustering d'ontologies. Et il est modifié après pour intégrer les différentes mesures de similarité. Le deuxième algorithme a pour but d'éliminer les clusters inutiles générés auparavant. Enfin, un processus de validation du partitionnement est programmé comme une perspective.

Deux séries d'expérimentations ont montré que cette approche réduit le temps et l'espace requis pour traiter une ontologie, et produit des partitions de haute qualité. L'amélioration de l'algorithme k-means proposée et l'utilisation de la nouvelle mesure de similarité Dennai donnent de meilleures partitions et des clusters significatifs que l'algorithme k-means traditionnel.

4.1. Comparaison de méthodes d'alignement

Méthode d'alignement	Entré du système	Cardinalité	Mesure de similarité	Technique de similarité	Automatisation	Passage a l'échelle	Technique de passage à l'échelle
Anchor-PROMPT [Noy& Musen, 2001]	Ontologies : RDF(S), graph, RDF	(0, n)	-terminologique -Structurelle	-String-comparaison des classes/reliations -Comparaison des positions	Semi-automatique		
GLUE [Doan & al, 2002]	Schémas, bases De données relationnelles, Taxonomie XML	(1,1)	-Terminologique -Extensionnelle -Structurelle	-Tokenisation -String-comparaison pour des tokens	Semi-automatique		
QOM [Ehrig& Staab, 2004]	OWL, RDF(S)	(1,1)	Terminologique Structurelle	String-based, Application vocabulary,	Semi-automatique	Non	-

				Taxonomic structure			
S-Match [Giunchiglia & al, 2004]	SchémasXML, OWL	-	Terminologique Structurelle	-Tokenisation, lemmatisation, WordNet -logique propositionnelle	Automatique	-	-
OLA [Euzenat & al, 2004]	OWL, RDF(S)	(1, n)	Sémantique, structurelle, terminologique	Agrégation des similarités	Automatique	Non	-
AROMA [David, 2007]	OWL, RDF	-	Terminologiques Structurelle	String-based Outils de TAL pour l'extraction de termes et règles d'associations.	Automatique	Non	-
ASMOV [Jean-Mary & al, 2007]	OWL	(0, n)	Terminologique, structurelle interne, conceptuelle et extensionnelle	UMLS Meta thesaurus WordNet	Automatique	Non	-

ROMIE [Elbyed A., 2009]	Ensemble de ressources locales et d'ontologies	-	Terminologique Structurelle Extensionnelle	SubString Wordnet Hamming Structural matchers	Automatique	-	-
TaxoMap [Kasri, 2010]	Ontologie OWL RDF (taxonomies)	(1, n)	-terminologique -structurelle	Technique basée sur la mesure de similarité de Lin (SimLinLike)	Automatique	Oui	La méthode de clustering proposé dans Falcon.
COMA++ [Massmann& al, 2011]	XSD Schema graphe	(0, n)	-syntaxique -structurelle (descendants, ascendants)	Fragment-based Matching. Taxonomy Matcher	Automatique	Oui	-Clustering basé sur la structure.
[Kachroudi, et al., 2013]	Reseau sémantique	-	-Terminologique -sémantique	Wordnet	Automatique	Oui	Clustering basé sur WordNet
[Algergawy, et al., 2014]	-Schéma XML, Ontologies	-	-syntaxique -structurelle	Levenstein distance, N-gram distance,	-	Oui	-Clustering basé sur la

				and Jaro similarity Similarity datatype element			structure.
Setti et al., 2015] [Ontologie volumineuse OWL	-	-Terminologique -Structurelle	- Tbk, - Jaccard -Cosine - Dice - Euclidienne - Wu & Palmer - Mesure d'édition - Dennai	-	Oui	Clustering : k-means amélioré

Tableau 2: Comparaison des méthodes d'alignement.

4.2. Analyse

La table (*Tableau2*) présente une étude comparative qui montre les principales techniques de base utilisées par chaque méthode d'alignement. Ces techniques regroupent les méthodes terminologiques, les méthodes structurelles, les méthodes extensionnelles et les méthodes sémantiques.

Les algorithmes d'alignement cités se diffèrent dans quelques critères et partagent d'autres.

- La plupart des algorithmes comparés admettent en entrée deux ontologies OWL et pour certains (**GLUE** et **Taxomap**) un type particulier des ontologies OWL qui sont des taxonomies, (**Algergawy** et **S-Match**) ont des schémas XML comme entrée.
- Parmi les différentes approches d'alignement cités auparavant, il y a des systèmes de mapping qui sont complètement automatisés, dans le sens où les similarités entre les concepts des deux ontologies à faire correspondre sont identifiées sans aucune intervention de l'utilisateur. Cependant, les approches (**Anchor-PROMPT**, **GLUE** et **QOM**) nécessitent une intervention humaine pour accomplir la tâche de mapping.
- Pour la découverte de la similarité entre les concepts des ontologies, la technique de calcul de similarité n'est pas la même, la plupart des algorithmes utilise la mesure de similarité structurelle et terminologique, celle de **kachroudi** utilise une mesure sémantique. La mesure structurelle est commune dans (**Algergawy**, **COMA++** et **OLA**), d'autres mesures dans (**Algergawy** et **COMA++**) qui est la mesure syntaxique, (**kachroudi** et **OLA**) utilise la mesure sémantique, et la mesure extensionnelle pour (**ROMIE**, **ASMOV**, et **GLUE**).
- Les approches d'alignement d'ontologies volumineuse utilisent le clustering comme une technique pour le passage à l'échelle.
- La phase de clustering est basée sur le calcul des mesures de distance (qui est utilisé pour l'alignement) pour construire des parties, ces parties facilitent le passage à échelle pour l'alignement des ontologies volumineuse. Le travail d'alignement est fait une deuxième fois pendant l'étape d'alignement des ontologies.

5. Conclusion

Le mapping d'ontologies est une tâche difficile en raison des problèmes posés par l'hétérogénéité des ontologies disponibles dans le web, des ontologies qui exposent des hétérogénéités terminologiques, conceptuelles et syntaxiques etc. Le mapping joue un rôle dans l'interopérabilité et la communication entre les applications basées sur les ontologies.

La littérature du domaine propose plusieurs méthodes d'alignement d'ontologies, ces méthodes exploitent différents formats d'ontologies et repose sur le calcul des mesures de similarité, Cependant, il existe très peu d'algorithmes d'alignement qui visent à traiter le problème du passage à l'échelle des méthodes d'alignement, pour atteindre l'objectif du passage à l'échelle des méthodes d'alignement qui permette la construction et l'exploitation d'un Web sémantique dans une organisation hétérogène, nous envisageons la problématique de l'hétérogénéité de l'ontologie, qui est une composante de base importante.

Chapitre 3 : conception du système d'alignement d'ontologies

1. Introduction

L'objectif de ce chapitre est de présenter la conception du système d'alignement d'ontologies, nous proposerons une approche qui traite l'alignement des ontologies volumineuses en combinant les mesures de similarité et en prenant en considération les points forts et les points faibles de chaque méthode déjà analysée pour obtenir un bon résultat.

Nous commencerons d'abord par citer les critères de notre solution, ensuite nous présentons ses différentes étapes, le partitionnement et le mapping.

2. Motivation

La majorité des travaux existants que nous avons déjà analysé dans le chapitre précédent ne traitent pas le passage à l'échelle pour réduire l'espace de recherche, nous avons distingué aussi que les approches d'alignement d'ontologies volumineuses utilisent le clustering comme une technique pour le passage à l'échelle, cette phase de clustering est basée sur le calcul des mesures de distance qui est représenté par les mesures de similarité, d'où sur des ontologies volumineuses le calcul de similarité entre tous les concepts des deux ontologies est utilisé pour le partitionnement, à base de clustering, et d'une manière itérative, une fois le clustering est réalisé, les mesures de similarités sont utilisées à nouveau pour trouver l'alignement.

Nous avons mis au point un système de mapping d'ontologies volumineuse qui est basé sur le partitionnement, il se résume comme suit :

Le partitionnement :

- ❖ Fournir une solution basée sur les critères d'ingénierie ontologique pour élaborer le processus de partitionnement.
- ❖ Utilisation de la distance sémantique minimale entre les concepts.
- ❖ Partitionnement d'ontologies sans passer par les mesures de similarité

- ❖ Partitionnement base sur le critère de distance semantique minimale d'ingénierie ontologique.
- ❖ Utilisation de la structure d'ontologie pour réaliser le partitionnement.
- ❖ L'automatisation du processus de partitionnement et d'alignement.

Le mapping :

- ❖ Mettre un système qui sert à réaliser le mapping sémantique des ontologies volumineuses.
- ❖ Le calcul d'une similarité sémantique entre les concepts de l'ontologie, il s'agit de combiner deux mesures de calcul de similarité (terminologique, structurelle).
- ❖ L'utilisation des ressources externe pour avoir un mapping sémantique avec l'utilisation de Wordnet.
- ❖ Utilisation du standard OWL comme un langage de représentation d'ontologie.
- ❖ Réaliser des tests du système sur des ontologies volumineuses.

3. L'approche proposée pour l'alignement

Notre approche consiste à faire l'alignement entre deux ontologies volumineuses, d'abord nous abordons le partitionnement ensuite nous réalisons le mapping en se basant sur les résultats du partitionnement.

3.1. Partitionnement

Dans le but de relever le challenge du passage à l'échelle des méthodes d'alignement, nous avons proposé une méthode de partitionnement des ontologies.

Le schéma de partitionnement d'ontologie est illustré dans la *Figure 8* suivante :

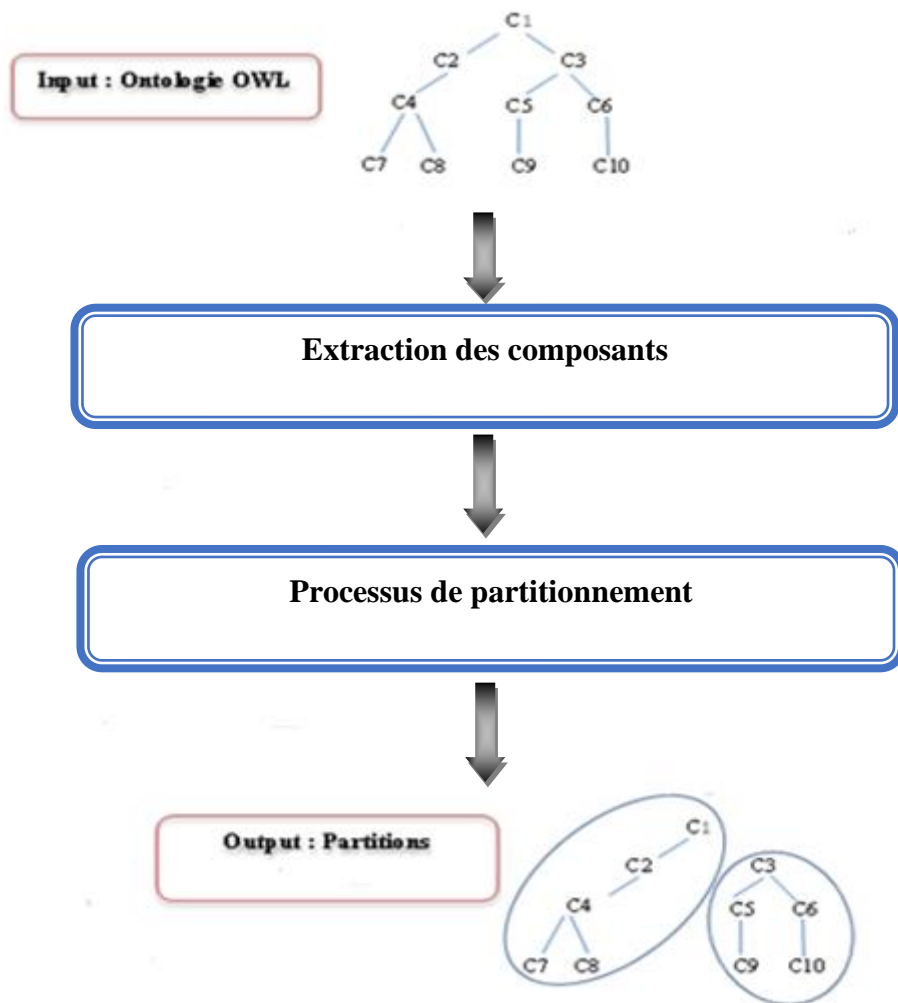


Figure 8: Schéma du Processus de partitionnement d'ontologie.

Le partitionnement d'ontologie est composé de deux étapes suivantes :

- **Extraction des composants de l'ontologie OWL**

Dans la première phase nous avons en entrée une ontologie OWL, ensuite on doit parcourir toute la hiérarchie de l'ontologie, pour extraire tous les composants qui se trouvent dans cette dernière.

- **Construction des partitions d'ontologies**

Le parcours d'ontologie se fait par niveau de la hiérarchie, dans chaque niveau on compare le nombre de concepts avec le nombre de partitions voulu, si le nombre des concepts ne suffit pas pour construire les (K) partitions voulus on descend dans le niveau de hiérarchie suivant.

L'algorithme de partitionnement prend en entrée l'ontologie à partitionner, ainsi que le nombre de partitions K , l'algorithme extrait la liste des concepts à travers un parcours de l'ontologie. A la fin il retourne une liste qui contient les partitions construites. La section suivante décrit en détail le processus de partitionnement proposé :

Algorithme : Partitionner

Données : 1. O : ontologie à partitionner

2. K : nombre de partitions voulu

Résultat : LP : liste des partitions

Début

3. /* parcours des nœuds de la racine de l'ontologie */

4.*ExtraireNoeuds*(O)

5. Compare (*nbNoeud*, K)

6. **Cas 1** : si (*nbNoeud*= K) **alors**

7. /*Chaque nœud représente la racine d'une partition*/

8. **pour chaque** (*Noeud*) **faire**

9. /* Ajouter *Noeud* à LP */

10. **AJOUTER** (*Nœud*, LP)

11. **Cas 2** : si (*nbNoeud*> K)

12. /*Regrouper les nœuds afin de construire les K parties*/

13. $LP = NbConceptSup()$

14. **Cas 3** : si (*nbNoeud*< K)

15. /*Descend dans le niveau de hiérarchie */

16. $LP = NbConceptInf()$

17. **Retourner** (LP)

Fin

Pour le cas où le nombre de concepts est supérieur au nombre de partition K , nous allons appliquer la fonction *NbConceptSup()* qui prend comme paramètre la liste des concepts et qui retourne la liste des partitions construites :

Fonction : *NbConceptSup()*

Données : 1. *LNœud* : liste des Nœuds

2. k : nombre de partitions voulu

Résultat : *LP*: liste des partitions

Début

3. /* Division de nombre des Nœuds sur K et stocke le résultat dans la variable */

4. $n = (nbNoeud / K)$

5. /* initialisé le compteur des partitions np à 0 */

6. $np = 0$

7. /* récupérer la taille de la liste des Nœuds dans la variable t */

8. $t = \text{taille}(LNœud)$

9. /* Construire des blocs de n concepts */

10. Pour i allant de 1 à t pas n faire

11. Si $(np < k)$ alors

12. Si $np < (k-1)$ alors

13. /* limiter l'indice j de la boucle au nombre des Nœuds de chaque partition */

14. $m = i + (n-1)$

15. Sinon /* construction de k -ième partition */

16. /* laisser l'indice j de la boucle parcourt le reste de la liste des Nœuds */

17. $m = t$

18. Pour j allant de i à m faire

19. /* ajouter les n Nœuds dans une liste Ln */

20. AJOUTER (*Nœud*, *Ln*)

21. /* Ajouter la liste des Nœuds dans *LP* */

22. AJOUTER (*Ln*, *LP*)

23. Vider (*Ln*)

24. $np++$

25. Retourner (*LP*)

Fin

La fonction traite le cas où le nombre de concepts est inférieur au nombre de partition K :

Fonction : NbConceptInf()

Données :

1. *LNœud* : liste des Nœuds

Résultat : *LP*: liste des partitions

Début

2. /* Descend dans le niveau de hiérarchie suivant */

3. Pour chaque *Nœud* dans *LNœud* faire

4. **AJOUTER** (*Liste*, **RecupererFils**(*Nœud*))

5. *LNœud* = *Liste*

6. /* Appeler de fonction partitionner avec la nouvelle liste *LNœud* comme paramètre */

7. *LP* = **Partitionner**(*LNœud*)

8. /* Construire la k-ième partition à partir des Nœuds de niveau de hiérarchie supérieur */

9. /* récupérer les Nœuds de niveau de hiérarchie supérieur dans la liste *kp* */

10. *kp* = liste_tout_Nœuds - *LP*

11. **AJOUTER** (*LP*, *kp*)

12. **Retourner** (*LP*)

Fin

Par la suite, on va donner un exemple pour chaque cas traité :

- **Nombre de partition voulu égal aux nombres de fils de la racine**

C'est le cas le plus simple, où chaque fils devient la racine d'une partition :

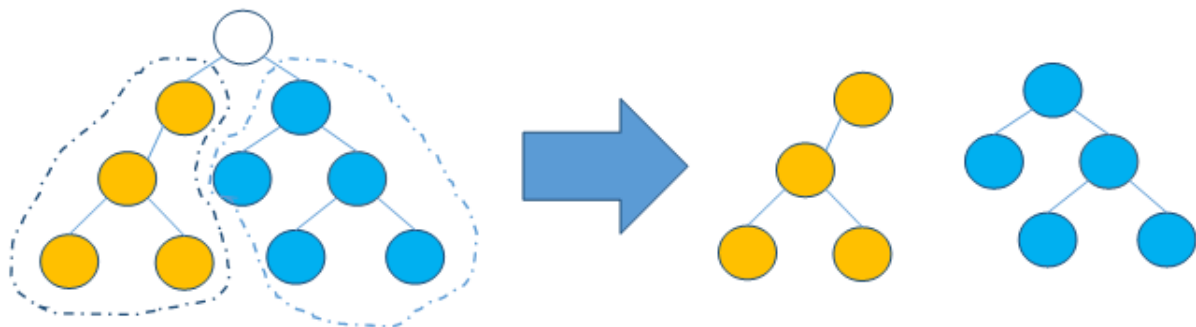


Figure 9: Exemple d'un cas égal ($K = 2$)

- **Nombre de partitions voulu est inférieur aux nombres de fils de la racine**

Dans ce cas, on divise le nombre de fils de la racine sur le nombre de partitions voulu (K), le quotient représente le nombre de fils des premières ($k-1$) parties, quand le nombre de partitions obtenu est $k-1$, on regroupe le reste des concepts de niveau de la hiérarchie parcouru dans un seul bloc qui représente le K -ième partition.

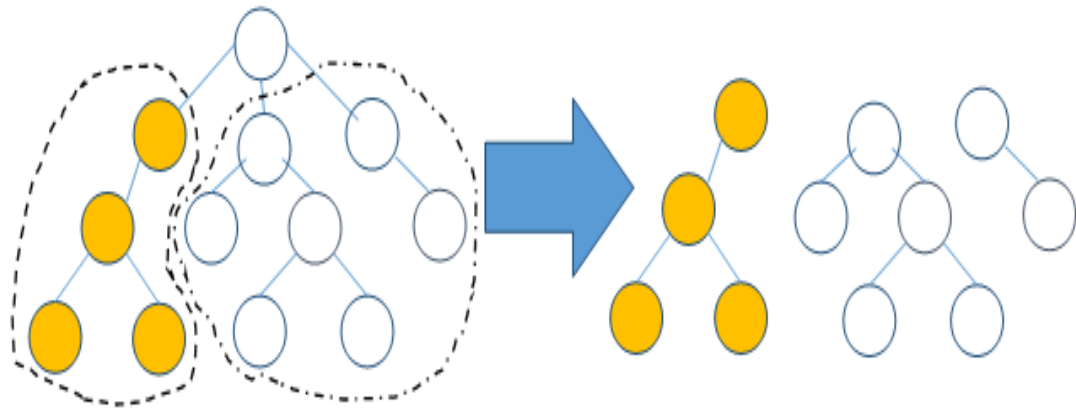


Figure 10: Exemple d'un cas inférieur ($K = 2$)

- **Nombre de partition voulu est supérieur aux nombres de fils de la racine**

On parcourt l'ontologie par niveau de profondeur, dans la hauteur (H) tel que le nombre de concept est inférieur ou égale au nombre de partition voulu -1 ($K-1$) on applique le même algorithme que les cas précédents pour ($K-1$) parties, dans le but de ne pas avoir des concepts ignorés, Ces concepts de niveau supérieur de la hauteur précédente seront regroupés dans une partition qui représente la K -ième partition :

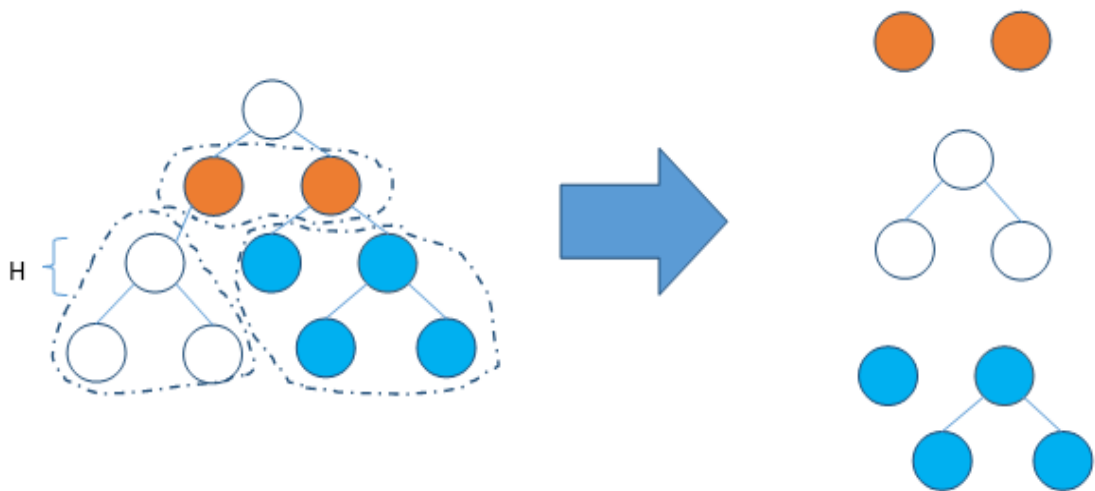


Figure 11: Exemple d'un cas inférieur ($K = 3$)

L'objectif principal du partitionnement est de réduire l'espace de travail, en utilisant les représentants de chaque partition afin de calculer les correspondances entre ces représentants.

3.2. Détermination du représentant

Le représentant d'un bloc est un concept qui existe réellement dans bloc, qui ressemble à la notion du médoïde dans l'algorithme K-médoïde. Nous avons utilisé la formule du médoïde pour trouver le médoïde dans chaque cluster qui est le point le plus central d'un cluster et chaque cluster est représenté par un de ses objets.

Pour déterminer le représentant du nouveau cluster, on calcule la somme des similarités de chaque concept par rapport aux autres concepts du cluster. Le concept dont la valeur des similarités avec les autres éléments est la plus élevée (valeur max de similarités) représente le nouveau médoïde.

3.3. Mapping entre blocs

En partant d'un partitionnement d'ontologies et la désignation de centres de chaque cluster de l'ontologie partitionnée ces derniers se sont les entrées du mapping nous calculons les mesures de similarités de chaque centre des blocs de la première ontologie avec tous les autres centres des blocs de la deuxième ontologie afin de trouver l'entité correspondante avec un sens équivalent ou le sens le plus proche pour produire comme sortie un ensemble de correspondances entre blocs.

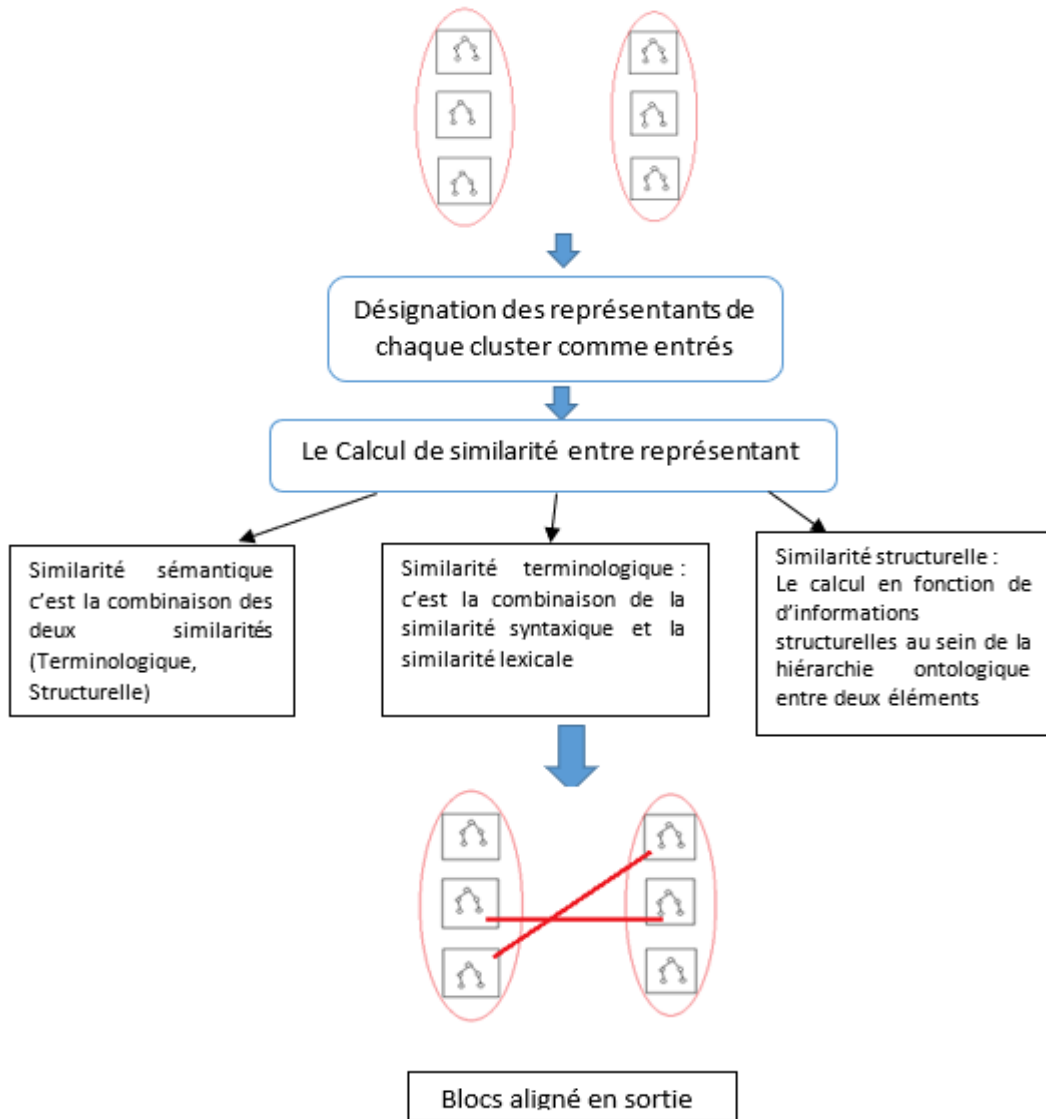


Figure 12: processus de mapping entre blocs.

3.4. Alignement entre concepts

Après avoir trouver les correspondances avec les centres des clusters (blocs) de chaque ontologie, nous lançons l'alignement entre concepts, cela veut dire à calculer la similarité entre tous les concepts qui se trouvent dans des blocs similaires c.à.d. faire un calcul de similarité sémantique entre les concepts du bloc A et les concepts du bloc B qui sont similaires, chaque concept du bloc A va avoir toutes les valeurs de similarité avec les autres concepts du bloc B pour ensuite déterminer le concept le plus proche.

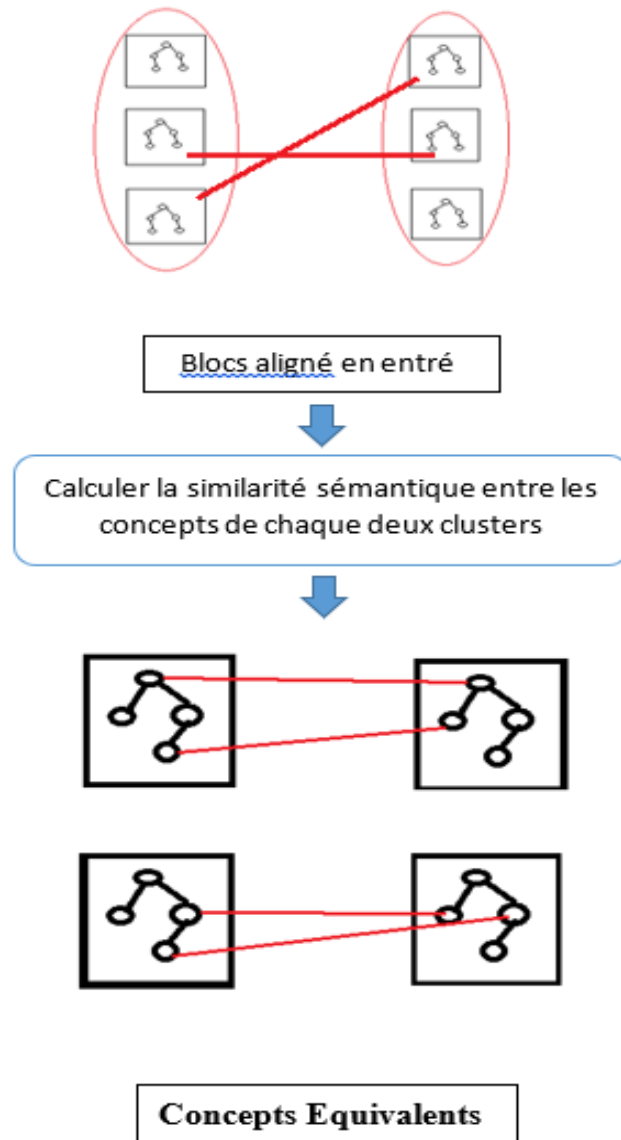


Figure 13: processus de mapping entre concepts

3.5. Les mesures de similarité utilisées

Nous présentons par la suite les mesures de similarité utilisées dans notre travail :

- **Similarité terminologique**

Afin de mesurer la similarité terminologique entre les concepts, nous allons utiliser les deux mesures suivantes :

❖ Similarité lexicale

Les méthodes basées sur un langage consistent à trouver des associations entre les entités ou les classes, en utilisant des ressources externes. Plusieurs types de ressources peuvent être employés, notre choix s'est porté sur WordNet.

WordNet est une ressource lexicale de langue anglaise, disponible sur internet, qui regroupe des termes (noms, verbes, adjectifs et adverbes) en ensembles de synonymes appelés *synsets*. Un synset regroupe tous les termes dénotant un concept donné. WordNet contient des liens entre les synsets qui représentent plusieurs relations : is-a(est-un(e)), part-of (fait-partie-de), synonymie, antonymie, hyponymie, homonymie etc.

Pour le calcul de la similarité linguistique, la fonction $Syn(c)$ calcule l'ensemble des Synsets de WordNet du concept c retenus après l'enrichissement de l'ontologie soit

$S = Syn(c1) \cap Syn(c2)$ l'ensemble des sens communs entre $c1$ et $c2$ à comparer, la cardinalité de S est :

$$\lambda(S) = |Syn(c1) \cap Syn(c2)| ;$$

Soit $\min(|Syn(c1)|, |Syn(c2)|)$ le minimum entre la cardinalité des deux ensembles $Syn(c1)$ et $Syn(c2)$ alors la mesure de similarité lexicale entre deux concepts $c1$ et $c2$ est définie comme suit :

$$Simlex(c1, c2) = \frac{\lambda(s)}{\min(Syn(c1), Syn(c2))}$$

❖ Similarité syntaxique :

Dans notre système nous allons utiliser la distance de Jaro, cette dernière mesure la similarité entre deux chaînes de caractères. Plus la distance de Jaro entre deux chaînes est élevée, plus elles sont similaires.

Cette mesure est particulièrement adaptée au traitement de chaînes courtes comme des noms ou des mots de passe. Le résultat est normalisé de façon à avoir une mesure entre 0 et 1, le zéro représente l'absence de similarité.

La distance de Jaro entre les deux chaînes $s1$ et $s2$ est définie par :

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

Où :

- m est le nombre de caractères correspondants.

- t est le nombre de transpositions.

Deux caractères identiques de s_1 et de s_2 sont considérés comme correspondants si leur éloignement (i.e. la différence entre leurs positions dans leurs chaînes respectives) ne dépasse pas :

$$\left(\frac{\max(|s_1|, |s_2|)}{2} \right) - 1$$

Le nombre de transpositions est obtenu en comparant le i ème caractère correspondant de s_1 avec le i ème caractère correspondant de s_2 . Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de transpositions.

DSJaro : $S \times S \in [0, 1]$ telle que :

$$\mathit{SymSin}(\mathit{Marie}, \mathit{Maires}) = \frac{1}{3} \left(\frac{5}{|5|} + \frac{5}{|6|} + \frac{5-1}{5} \right) = 0.87$$

Nous avons ensuite, combiné ces deux mesures, le résultat est la formule suivante :

$$\mathit{Sim}_{\text{ter}}(c_1, c_2) = \frac{\mathit{SimLex}(c_1, c_2) + \mathit{SimSyn}(c_1, c_2)}{2}$$

- **Similarité structurelle**

Cette dernière sera calculée par la mesure de Wu et Palmer :

$$\mathit{Simwp}(c_1, c_2) = \frac{2 \times \mathit{Profondeur}(c)}{\mathit{Profondeur}(c_1) + \mathit{Profondeur}(c_2)}$$

Exemple

Soit l'ontologie de la Figure 13, on dénote par C1, C2 et C3 les concepts « Person », « PostDoc » et « AdministrativeStaff ». En appliquant la mesure de Wu et Palmer, la valeur de similarité est calculée comme suit :

$$\text{Simwp}(C1, C2) = 2 \cdot 1 / (1 + 4) = 0.4$$

$$\text{Simwp}(C2, C3) = 2 \cdot 2 / (4 + 3) = 4/7 = 0.57$$

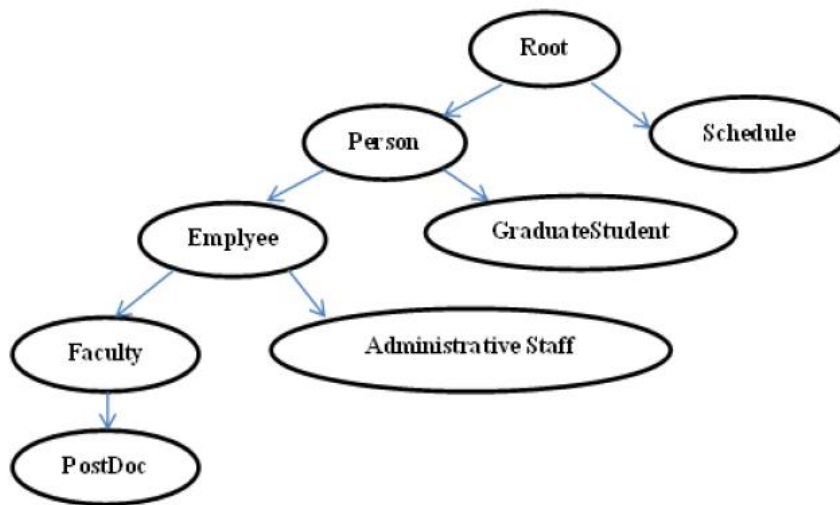


Figure 14: Extrait de l'ontologie UnivBench

- **Similarité sémantique**

La similarité sémantique est une évaluation du lien sémantique afin d'estimer à quel point deux concepts sont proches dans leurs sens, elle est calculée par la combinaison hybride des similarités terminologique et structurelle. La similarité sémantique définis selon la formule suivante :

$$\text{SimSem}(c1, c2) = \frac{\text{SimTer}(c1, c2) + \text{SimStr}(c1, c2)}{2}$$

4. Conclusion

Dans ce chapitre, nous avons vu les différentes étapes de l'algorithme proposée, qu'est utilisé pour aligner les ontologies représentées en OWL, en donnant comme résultats la liste des correspondances entre les entités des deux ontologies à aligner. Cet algorithme fait l'extraction de tous les concepts pour chaque ontologie, finalement seuls les éléments qui ont une valeur de similarité supérieure ou égale à ce seuil vont être affichés.

Chapitre 4 : Implémentation du système

1. Introduction

Dans ce chapitre on va expliquer l'implémentation de cette approche qui sert à effectuer l'ensemble des opérations qui permettent de définir un projet et de le réaliser. Le chapitre s'organise comme suit : tout d'abord on va présenter l'environnement de développement ainsi que les différents outils utilisés dans ce travail, ensuite nous allons décrire de façon visuelle l'implémentation de notre système, en effectuant des captures d'écran des différentes interfaces du système. Nous passons par la suite à la validation et les tests de notre système, nous montrons l'évaluation du mapping avec quelques résultats et discussions.

2. Les outils utilisés

Avant de commencer l'implémentation de notre système d'alignement, nous allons spécifier les différents outils utilisés dans notre travail :

2.1. Python

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par **Guido van Rossum**, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au *Centrum voor Wiskunde en Informatica* aux Pays-Bas, jusqu'à se voir associer une organisation à but non lucratif particulièrement dévouée, la **Python Software Foundation**, créée en 2001. Ce langage a été baptisé ainsi en hommage à la troupe de comiques les « Monty Python ».

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau efficaces et d'une approche simple mais efficace de la programmation orientée objet. La syntaxe élégante et le typage dynamique de Python, associés à sa nature interprétée, en font un langage idéal pour la création de scripts et le développement rapide d'applications dans de nombreux domaines sur la plupart des plates-formes.

L'interpréteur Python et la bibliothèque standard étendue sont librement disponibles sous forme de source ou de binaire pour toutes les principales plates-formes à partir du site Web Python, et peuvent être distribués librement. Le même site contient également des distributions et des pointeurs vers de nombreux modules, programmes et outils Python gratuits, ainsi que de la documentation supplémentaire.

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). Python convient également comme langage d'extension pour les applications personnalisables.

- **Paquets utilisés dans Python**

Dans l'implémentation de notre système, on a utilisé les paquets suivants dans python :

- ❖ **Numpy**

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).

- ❖ **Owlready2**

Owlready est un module de programmation orientée ontologie en Python, incluant un quadre RDF optimisé.

Owlready peut :

- Importer les ontologies OWL 2.0 au format NTriples, RDF/XML ou OWL/XML.
- Exporter les ontologies OWL 2.0 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.

- ❖ **Wordnet**

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

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

2.2. Le logiciel Protégé pour la visualisation des ontologies

Protégé est un 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 est gratuit et son code source est publié sous une licence libre (la Mozilla Public Licence).

Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc.

3. Présentation de l'application

Notre système contient une interface globale afin de faciliter l'interaction et l'utilisation de système, nous avons implémenté une interface graphique permettant de faire le partitionnement et le mapping des deux ontologies.

L'interface de notre système contient deux boutons, un bouton pour le partitionnement, et le deuxième bouton permet de faire l'alignement entre les deux ontologies :

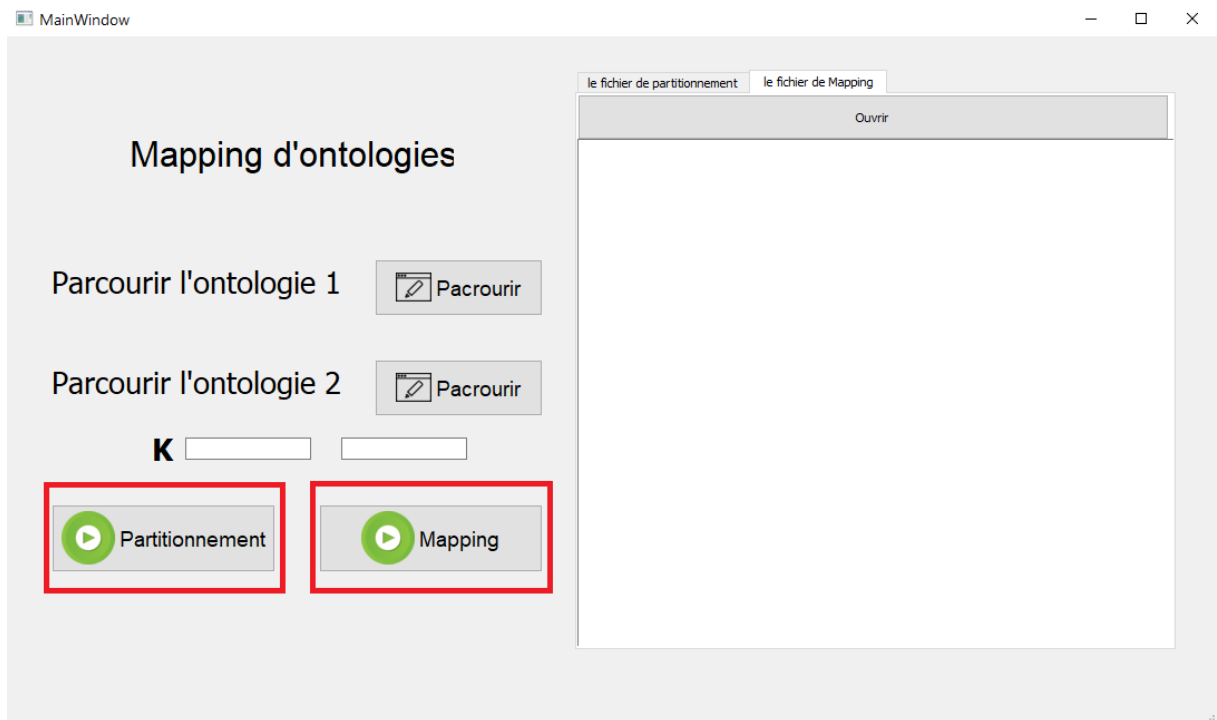


Figure 15:Interface montre les boutons d'opération de partitionnement et de mapping.

Afin qu'un utilisateur puisse aligner deux ontologies, il doit cliquer sur le bouton Parcourir pour choisir l'ontologie 1 puis la sélectionner, de même pour l'ontologie 2 :

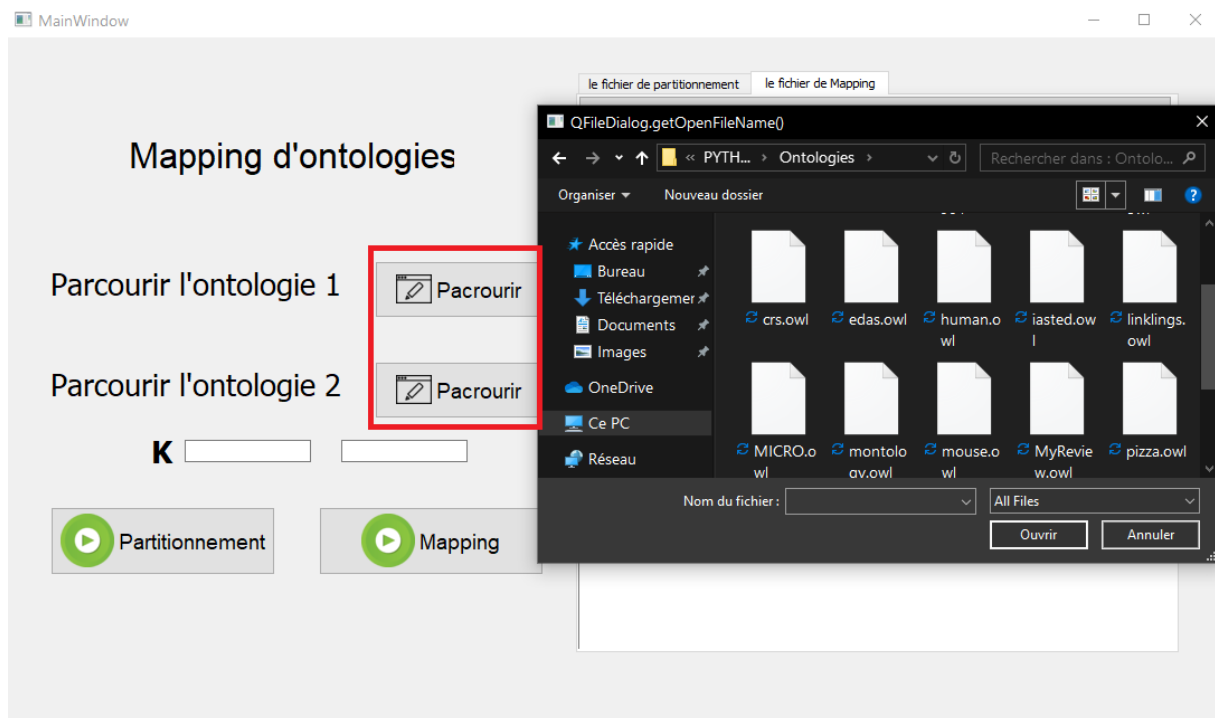


Figure 16 : Fenêtre de choix d'emplacement des deux ontologies.

Après avoir choisi les deux ontologies à aligner, l'utilisateur doit spécifier le nombre de partitions voulu pour chaque ontologie ($O1$ et $O2$) en introduisant la valeur de K dans les deux champs respectivement :

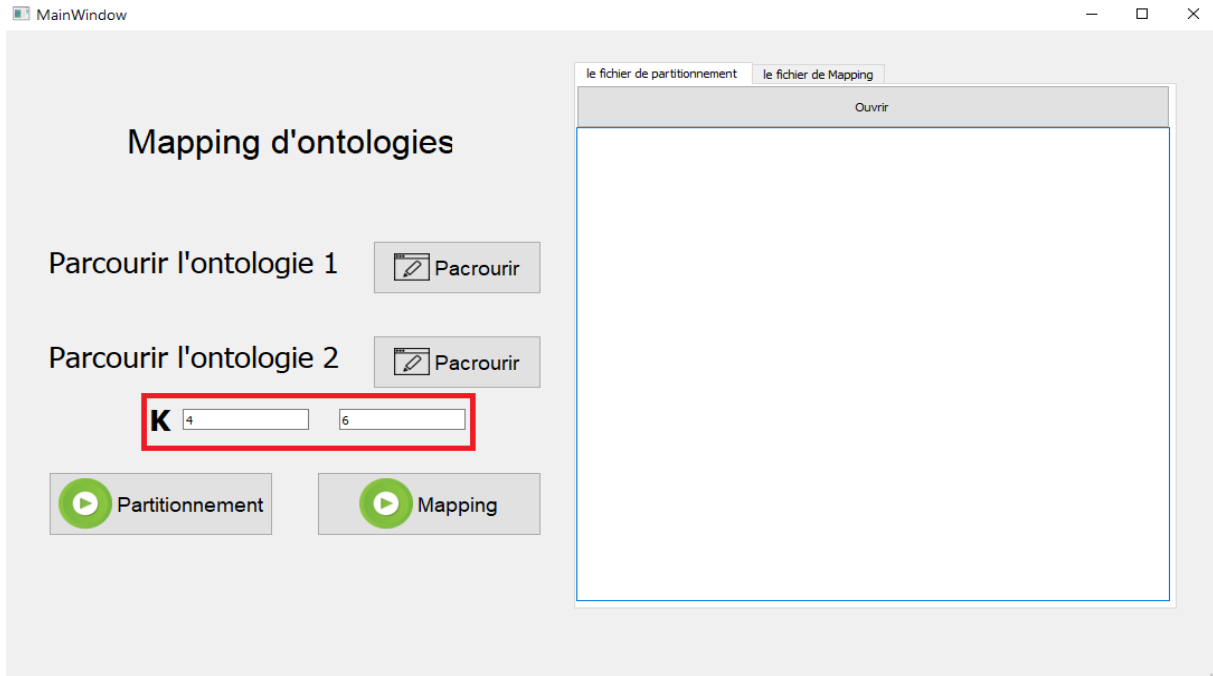


Figure 17: Interface montre les champs pour détermination de nombre de partitions.

L'interface de notre système contient une partie d'affichage pour afficher les résultats de deux opérations partitionnement et mapping :

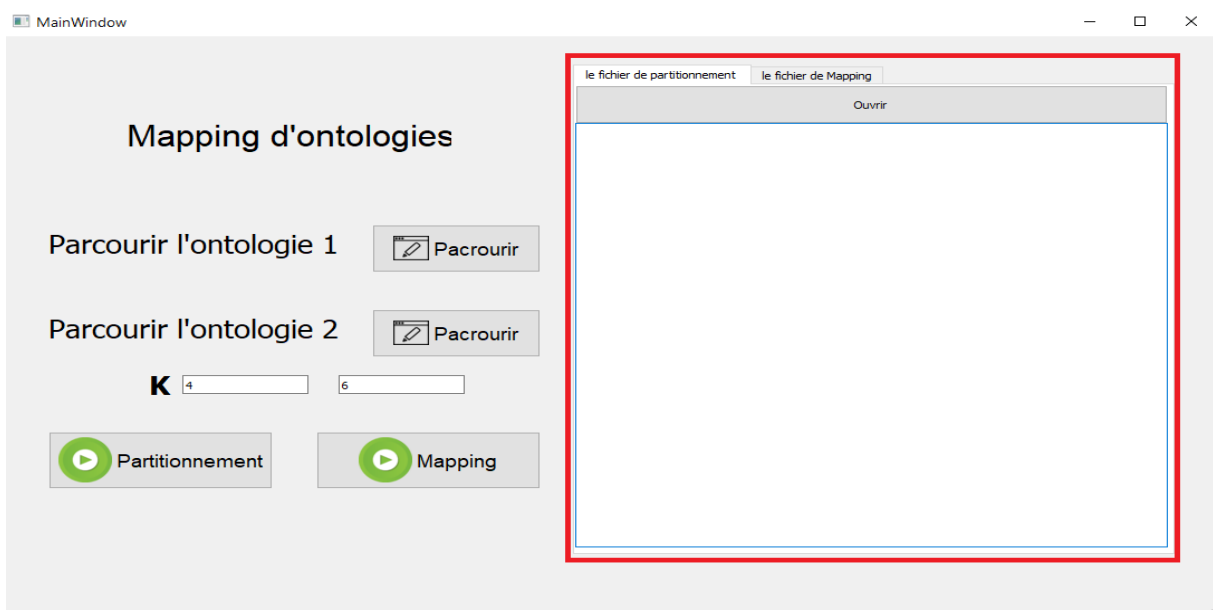


Figure 18: Onglet d'affichage de résultat

le système génère deux fichiers texte qui sont enregistrés dans le bureau, les fichiers montrent le résultat de partitionnement et de mapping, les résultats seront affichés dans la partie d'affichage

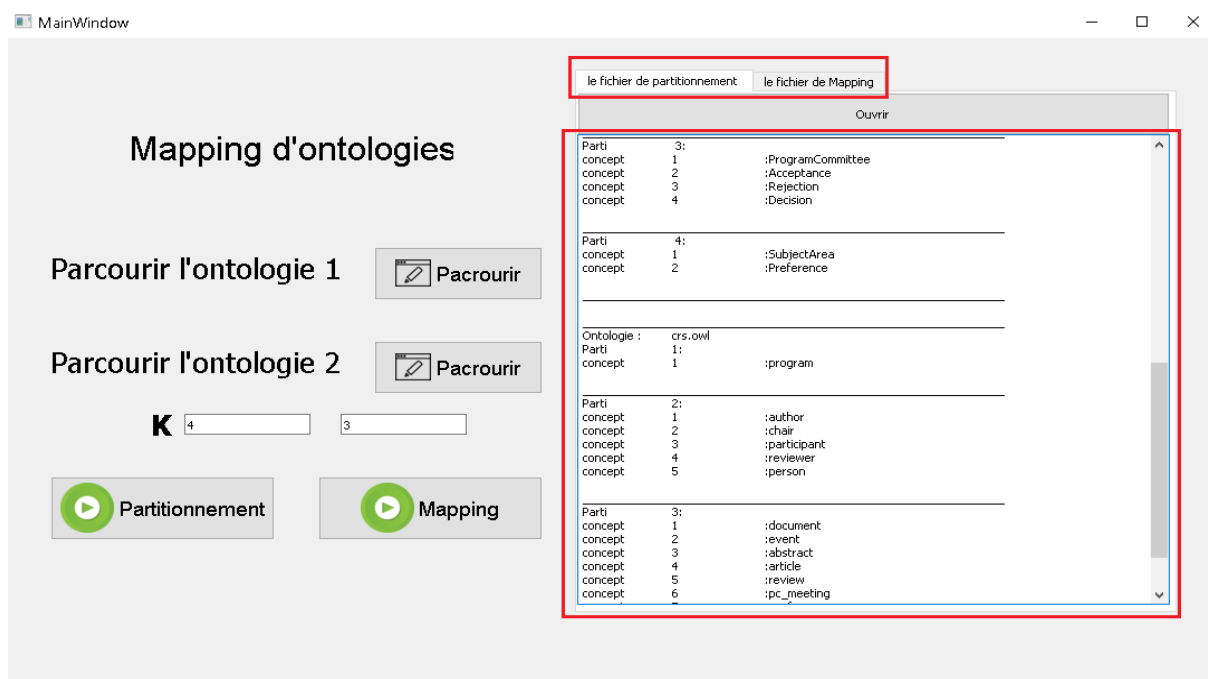


Figure 19: Affichage de résultats.

4. Tests et validation

4.1. Evaluation

Afin de valider notre approche, nous avons effectué des tests sur des ontologies, selon le tableau suivant :

Ontologie (OWL)	ConfOf	conference	Ekaw	Edas	mouse	human
Nombre de classes	39	60	74	104	2744	3304

Tableau 3: Les ontologies de tests

Nous rappelons que le calcul de similarité entre les différents concepts de l'ontologie se fait en combinant deux mesures de similarité (Terminologique, structurelle).

- **Evaluation du mapping**

Dans cette section on procède à l'évaluation de mapping avec les deux ontologies « conférence » et « crs »

- **Mesures d'évaluation utilisées**

Pour mesurer la qualité d'un mapping, différentes mesures d'évaluation existent. Les plus connues et les plus utilisées sont :

- ❖ **Précision**

La précision est une mesure qualitative d'un système de mapping qui se calcul par rapport un alignement de référence (ensemble des correspondances considérées correctes généralement par un humain)

Soit H l'alignement de référence. La précision d'un alignement A produit par un système S est une fonction qui renvoie une valeur appartenant à l'intervalle [0,1].

La précision est le rapport entre les correspondances correctes (True positives) sur la totalité des correspondances retournées par le système S (True positives et False positives). Sa formule est donnée Comme suit :

$$\mathbf{Precision} = \frac{|A \cap H|}{|A|}$$

- ❖ **Rappel :**

Le Rappel est le rapport des correspondances correctes produites par le système sur le nombre total de correspondances considérées correctes (Alignements de référence H)

Il est considéré comme étant un paramètre de performance.

Une valeur élevée de Recall signifie qu'un nombre important de correspondance a été trouvé, mais aucune information sur les mappings incorrects trouvés n'est donnée.

Le calcul de Rappel est donné par la formule suivante :

$$\mathbf{Rappel} = \frac{|H \cap A|}{|H|}$$

4.2. Résultats expérimentaux et discussion

1. Dans cette section nous présentons les résultats obtenus. Les séries de tests ont été faite sur les ontologies : *crs*, *cmt*, *conférence* et *Confof*.
2. Dans le cadre des expérimentations menées, les tests de comparaison ont été réalisés avec l'aide d'un expert du domaine.
3. Nous avons comparé nos résultats avec le résultat des experts du domaine pour calculer les mesures de rappel et de précision.
4. Il faut prendre en compte que le mapping de test ne précise pas le nombre de clusters lors de partitionnement et ne précise pas les mesures de similarités utilisées.
5. Le tableau ci-dessous présente les valeurs de rappel et de précision avec **4** comme nombre de partition :

	edas/ekaw	Conference/Confof	edas/Confof	ekaw/Conference	Human/mouse
La précision	$\frac{14}{17} = 0.73$	$\frac{10}{14} = 0.71$	$\frac{15}{24} = 0.62$	$\frac{20}{33} = 0.60$	$\frac{1002}{1671} = 0.59$
Le rappel	$\frac{14}{22} = 0.63$	$\frac{10}{15} = 0.66$	$\frac{1154}{22} = 0.68$	$\frac{20}{27} = 0.74$	$\frac{1002}{1516} = 0.66$

Tableau 4: Valeurs résultantes de rappel et de précision.

5. Conclusion

Dans ce chapitre, nous avons évalué notre algorithme, cette évaluation nous a permis en premier lieu de calculer les paramètres qualitatifs (le rappel et la précision). Nous avons constaté que ces derniers sont les paramètres les plus déterminants pour l'évaluation de tels systèmes.

Conclusion générale

L'alignement des ontologies volumineuses propose un problème appelé « passage à l'échelle » des méthodes d'alignement l'une des solutions pour résoudre et traiter ce problème, en suppose un algorithme de partitionnement (Partitionner en des ontologies de petite taille en quelques centaines de concepts) avant réaliser l'alignement. Dans notre travail nous avons développé un système d'alignement d'ontologies volumineuses, nous avons considéré comme entrée des ontologies OWL. Nous avons commencé d'abord par l'explication de la notion d'ontologie, ses composants, une définition de processus d'alignement, les domaines d'application de l'alignement, les méthodes de base utilisées dans l'alignement. Ensuite on a fait une comparaison entre quelques approches de l'alignement d'ontologies. Enfin nous avons présenté la conception de notre application qui décrit la méthode proposée pour aligner les ontologies et explique le fonctionnement de l'algorithme exploité en suivant une procédure pour mesurer les similarités entre les entités des ontologies.

Quand les ontologies sont de très grandes tailles, par exemple en agronomie et en médecine, des ontologies comportent plusieurs dizaines de milliers de concepts, l'efficacité des méthodes classiques pour toute opération réalisée sur les ontologies diminue considérablement en termes de la précision et de la qualité des résultats obtenus. Le partitionnement, donc, est la meilleure solution pour résoudre ce problème.

Notre objectif était de réduire l'espace de recherche, pour ce faire on a partitionné l'ontologie sans passer par les mesures de similarité en utilisant une méthodologie basée sur les critères d'ingénierie ontologique et la structure d'ontologie pour le réaliser. La désignation de représentant de chaque bloc se fait en calculant les similarités entre chaque concept et les autres concepts d'un même bloc, le concept qui porte la valeur maximale de somme de similarités est le point le plus central de bloc.

Le résultat de partitionnement est l'entrée de la phase de mapping, nous calculons les mesures de similarité (terminologique et structurelle) pour chaque centre de partition de la première ontologie avec tous les autres centres de la deuxième ontologie, on obtient un résultat de mapping qui a pour but la découverte des correspondances sémantiques entre des 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.

Références

- Algergawy, A., Moawed, S., Sarhan, A., Eldosouky, A., & Saake, G. (2014). Improving clustering-based schema matching using latent semantic indexing. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XV* (pp. 102-123). Springer, Berlin, Heidelberg.
- Bach, T. L. (2006). *Construction d'un Web sémantique multi-points de vue* (Doctoral dissertation, Paris, ENMP).
- Connolly, D., Harmelen, F. V., Horrocks, I., McGuinness, D. L., Patel-Schneider, P.F., et Stein, L. A. (2001). DAML-OIL Guide Reference <http://www3c.org/fr/lciaml+oil-reference>.
- David, J. (2007). *AROMA: une méthode pour la découverte d'alignements orientés entre ontologies à partir de règles d'association* (Doctoral dissertation, Université de Nantes). de Doctorat. Annecy Le Vieux : Polytech Savoie.
- Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M. C. A., Broekstra, J., Erdmann, M., et Horrocks, I. (2000). The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing* pages 63-74, <http://citeseer.ist.psu.edu/decker00semanrk.html>.
- Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002, May). Learning to map between ontologies on the semantic web. In *Proceedings of the 11th international conference on World Wide Web* (pp. 662-673). ACM.
- Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 29(2-3), 103-130.
- Ehrig, M., & Staab, S. (2004, November). QOM—quick ontology mapping. In *International Semantic Web Conference* (pp. 683-697). Springer, Berlin, Heidelberg.
- Elbyed, A. (2009). *ROMIE, une approche d'alignement d'ontologies à base d'instances* (Doctoral dissertation, Evry, Institut national des télécommunications).
- Euzenat, J., Valtchev, P. (2004, August). Similarity-based ontology alignment in OWL-lite. In *ECAI* (Vol. 16, p. 333).
- Euzenat, J., Shvaiko, P. (2007). *Ontology matching* (Vol. 18). Heidelberg: Springer.
- Furst, F. (2004, May). Opérationnalisation d'une ontologie : une méthode et un outil. In *15èmes Journées francophones d'Ingénierie des Connaissances* (pp. 199-210). Presses universitaires de Grenoble.

- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220.
- Jean-Mary; Asmov: Ontology alignment with semantic validation. In Joint SWDB-ODBIS Workshop.15-20.
- Giunchiglia, F., Shvaiko, P. (2003). Semantic matching. workshop on Ontologies and Distributed Systems. IJCAI.
- Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004, May). S-Match: an algorithm and an implementation of semantic matching. In *European semantic web symposium* (pp. 61-75). Springer, Berlin, Heidelberg.
- Kasri, S. (2010). Etude des techniques d'alignement des ontologies : proposition d'un algorithme de passage à l'échelle. Skikda, Algérie.
- Kachroudi, M., Zghal, S., & Ben Yahia, S. (2013). OntoPart: at the cross-roads of ontology partitioning and scalable ontology alignment systems. *International Journal of Metadata, Semantics and Ontologies*, 8(3), 215-225.
- Leacock C, Chodorow M. Combining local context and WordNet similarity for word sense identification. In: WordNet: an electronic lexical database. Cambridge (USA): MIT Press, 1998.
- Massmann, S., Raunich, S., Aumüller, D., Arnold, P., & Rahm, E. (2011, October). Evolution of the COMA match system. In *Proceedings of the 6th International Conference on Ontology Matching-Volume 814* (pp. 49-60). CEUR-WS. org.
- Mellal N. (2007). Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information. Thèse de Doctorat. Annecy Le Vieux : Polytech Savoie.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*.38(11).39-41.
- Monge A, Elkan C. The field-matching problem: algorithm and applications. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996, p. 267-70.
- Noy; Using non--local context for semantic matching. workshop on ontology and information sharing, Seattle (WA US).63-70.
- Ogden, C. K., & Richards, I. A. (1923). The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism (Vol. 29). K. Paul, Trench, Trubner & Company, Limited.

- Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4), 334-350.
- Setti, Ahmed,S., Malki, M., et Benslimane, S. M. (2015). Ontology partitioning: Clustering based approach. *International Journal of Information Technology and Computer Science*, 7(6), 1-11.
- Shvaiko, P., & Euzenat, J. (2005). A survey of schema-based matching approaches. In *Journal on data semantics IV* (pp. 146-171). Springer, Berlin, Heidelberg.
- Smith, M. K., Welty, C., & McGuinness, D. L. (2004). *OWL Web Ontology Language Guide* <http://www.w3.org/TR/owl-guide>.
- Uschold, M., & Gruninger, M. (1996). Ontologies : Principles, methods and applications. *The knowledge engineering review*, 11(2), 93-136.
- Visser, P. R., Jones, D. M., Beer, M. D., Bench-Capon, T. J. M., Diaz, B. M., & Shave, M. J. R. (1999, August). Resolving ontological heterogeneity. In *International Conference on Database and Expert Systems Applications* (pp. 668-677). Springer, Berlin, Heidelberg.
- Ziani M, Boulanger D, Talens G. Système d'aide à l'alignement d'ontologies métier. 28 congrès INFORSID, Marseille 2010 : 345-60.