N° d'ordre :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR & DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DJILLALI LIABES
FACULTE DES SCIENCES EXACTES
SIDI BEL ABBES

THESE
DE DOCTORAT EN SCIENCES

Présentée par

BENAISSA KADDAR Leila

Spécialité : Informatique
Option : Ingénierie des connaissances

Intitulée

Similarité Sémantique inter ontologies Multi-langage

Soutenue le ***12 Juillet 2022***
Devant le jury composé de :

Président :	*TOUMOUH Adil*	*MCA*	*UDL-SBA*
Examinateurs :	*DEBBAT Fatima*	*Pr*	*Univ de Mascara*
	BENSLIMANE Sidi Mohamed	*Pr*	*ESI-SBA*
Directeur de thèse :	*BEN-NAOUM Farah*	*MCA*	*UDL-SBA*

Année universitaire 2021-2022

Dédicaces

Je dédie cette thèse :

*À l'Homme le plus important de ma vie, l'Homme de référence qui depuis mes premiers pas boiteux n'avait jamais cessé de me guider, de me conseiller, avait toujours été soucieux du moindre détail de ma vie autant personnelle que professionnelle. Cette thèse aurait été la consécration de tous ses efforts… Je regrette tellement de ne pas l'avoir fini en son vivant. Mon Papa Taki !!! Ton sourire, ta chaleur et ton « **Je suis fier de toi ma fille** » me manque tellement…*

À ma mère Setti, la lumière de ma vie, qui n'a cessé de prier pour que j'atteigne mes objectifs dans la vie, pour son amour, sa confiance son soutien moral et surtout ses sacrifices…

À mes sœurs Lamia, Nawel, Narimane, Fatima et A mon frère Moulay.

À mes petits-neveux et nièces : Adem, Nazim, Amira et Chiraz.

À toute ma famille et mes amis …

Leila …

Remerciements

Tout d'abord, mes remerciements sont adressés à ***Allah*** *qui m'a donné la puissance et le courage pour achever ce travail.*

Je tiens à remercier tous ceux qui, par leurs conseils, leurs suggestions et leurs disponibilités, ont contribué à l'aboutissement de ce travail de recherche.

Mes remerciements vont particulièrement à :

Ma directrice de thèse pour son encadrement, sa confiance, son soutien, ses précieux conseils et recommandations m'ont permis de mener ce travail dans de très bonnes conditions. Merci beaucoup Dr. BENAOUM Farah.

Je tiens à exprimer ma vive gratitude au Dr. TOUMOUH Adil, de l'université de Sidi Bel Abbés, pour m'avoir fait le grand honneur d'accepter de présider le Jury d'examen. Qu'il trouve ici l'expression de ma reconnaissance et de mon profond respect.

C'est un agréable devoir d'exprimer mes plus vifs remerciements à ceux qui m'ont fait l'honneur d'être les membres de jury :
Professeur DEBBAT Fatima, de l'université de Mascara. Professeur BENSLIMANE Sidi Mohammed, de L'école Supérieure en Informatique de Sidi Bel Abbés. Qu'ils trouvent ici, le témoignage de ma profonde reconnaissance pour l'intérêt qu'ils ont porté à mes travaux.

Épigraphe

"A pessimist sees the difficulty in every opportunity;
an optimist sees the opportunity in every difficulty."
– Winston Churchill

Résumé

La mesure de la similarité sémantique entre les termes est une étape cruciale de la recherche et de l'intégration d'informations, car elle nécessite la mise en correspondance du contenu sémantique. Bien que plusieurs modèles aient été proposés pour mesurer la similarité sémantique, ces modèles ne sont pas en mesure de quantifier efficacement le poids des termes pertinents qui affectent le processus de jugement de la similarité sémantique. Dans cette étude, nous présentons une nouvelle méthode pour mesurer la similarité sémantique inter ontologies, qui consiste à hybrider les approches basées sur la structure des ontologies telles que Wu & Palmer, Rada, Li et Zargayouna avec le poids de la similarité calculé à l'aide du dictionnaire sémantique WordNet. Le processus que nous allons proposer, passe par quatre phases (i) Le prétraitement : cette phase consiste à obtenir les termes pertinents des deux ontologies sélectionnées dans la langue choisie, pour mesurer leur similarité sémantique en utilisant le lemmatiseur TreeTagger. (ii) Calcul de la mesure de similarité de Wu &Palmer, Li, Rada et Zargayouna. (iii) Calcul des mesures proposées de similarités sémantiques hybride de WWP, WLI, WRA et WZY. (iv) Dans cette dernière phase on procède premièrement à l'expérimentation de notre approche en comparant les résultats obtenus de la phase (ii) et (iii), en l'appliquant sur trois langues (Anglais - Arabe et Français) et deuxièmement à l'évaluation de notre approche en utilisant les deux méthodes la cohésion et la densité. Notre algorithme est appliqué sur plusieurs cas de tests de la compagnie OAEI'2015 et a donné des résultats encourageants.

Mots-clés : Ontologie, Fusion d'ontologies, Similarité sémantique, TreeTagger, WordNet.

Abstract

Measuring semantic similarity between terms is a crucial step in information retrieval and integration, as it requires the mapping of semantic content. Although several models have been proposed to measure semantic similarity, these models are not able to effectively quantify the weight of relevant items that affect the semantic similarity judgment process. In this study, we present a new method for measuring semantic similarity between cross-ontologies that consists of hybridizing ontology structure-based approaches such as Wu &Palmer, Rada, Li, and Zargayouna with the weight similarity computed using the WordNet semantic dictionary. The process that we will propose includes four phases (i) Preprocessing: this phase consists in obtaining the relevant terms of the two ontologies selected in the chosen language, to measure their semantic similarity using the TreeTagger lemmatizer. (ii) Computing the similarity measure of Wu &Palmer, Li, Rada and zargayouna. (iii) Computing the proposed hybrid semantic similarity measures of WWP, WLI, WRA and WZY. (iv) in this last phase, we proceed first to the experimentation of our approach by comparing the results obtained in phase (ii) and (iii), by applying it in three languages (English - Arabic and French) and secondly to the evaluation of our approach by using the two methods cohesion and density. Our algorithm applies to various test cases of the Ontology Alignment Evaluation Initiative campaign, (OAEI'2015) and shows encouraging results.

Keywords: Ontology, Ontology merging, Semantic similarity, TreeTagger, WordNet.

ملخص

يعد قياس التشابه الدلالي بين المصطلحات خطوة حاسمة في استرجاع المعلومات وتكاملها، حيث يتطلب مطابقة المحتوى الدلالي .على الرغم من أنه تم اقتراح العديد من النماذج لقياس التشابه الدلالي، إلا أن هذه النماذج غير قادرة على تحديد وزن المصطلحات ذات الصلة التي تؤثر على عملية حكم التشابه الدلالي بشكل فعال. في هذه الدراسة، نقدم طريقة جديدة لقياس التشابه الدلالي بين الأنطولوجيات، والتي تتمثل في تهجين مقاييس التشابه على أساس هيكل الأنطولوجيا مثل Wu & Palmer و Rada و Li و Zargayouna مع وزن التشابه المحسوب باستخدام القاموس الدلالي WordNet .

تمر العملية التي سنقترحها من خلال أربع مراحل : (1) المعالجة المسبقة : تتكون هذه المرحلة من الحصول على المصطلحات ذات الصلة للاثنتين من الأنطولوجيات المختارتين في اللغة المختارة ، لقياس التشابه الدلالي بينهما باستخدام TreeTagger.(2)حساب مقياس التشابه لكل من Wu & Palmer ، Li ، Rada وZargayouna . (3)حساب مقاييس التشابه الدلالي الهجينة المقترحة لـ WWP ، WLI، WRA وWZY . (4) في هذه المرحلة الأخيرة ، ننتقل أولاً إلى تجربة نهجنا من خلال مقارنة النتائج التي تم الحصول عليها من المرحلة (2) و (3) ، من خلال تطبيقه على ثلاث لغات (الإنجليزية ، العربية والفرنسية) وثانياً لتقييم نهجنا باستخدام طريقتين Cohesion وDensity. إن خوارزميتنا هذه، تم تطبيقها على عدة حالات اختبار من شركة "مبادرة تقييم مطابقة الأنطولوجيات لعام "2015" و أعطت نتائج مشجعة.

الكلمات المفتاحية: الأنطولوجيات، دمج الأنطولوجيات، التشابه الدلالي، TreeTagger، WordNet.

Table des matières

Liste des Figures

Liste des Tables

Chapitre 1

Introduction Générale

1.1. Contexte :

La collaboration et la compétitivité entre les entreprises nécessitent d'importants échanges de données, qui sont nécessaires à l'interopérabilité. À l'aide d'ontologies, les entreprises décrivent la signification de leurs données. Le défi qui rend l'interopérabilité difficile est le manque de cohérence entre les ontologies. Ces dernières années, différents outils et méthodes ont été proposés pour la réconciliation des différentes ontologies. L'évaluation du degré de similarité entre leurs concepts est le point central de la plupart des méthodes existantes. En effet, les mesures de similarité sémantique jouent un rôle important, en particulier dans le processus de désambiguïsation des termes. L'objectif des mesures de similarité est d'estimer la ressemblance entre les concepts (auxquels les termes des requêtes et les documents sont rattachés). Un concept se réfère à un sens particulier d'un terme donné. Dans ce contexte, plusieurs approches pour l'évaluation de la similarité sémantique entre concepts appartenant à une ontologie ou différentes ontologies ont été proposées dans la littérature. Ces approches se basent sur différents aspects, par exemple la structure hiérarchique (arborescente) de l'ontologie, le contenu informatif des différents concepts intégrant des mesures statistiques, ou sur les propriétés des concepts comparés. Bien que plusieurs modèles aient été proposés pour mesurer la similarité sémantique, ces modèles ne sont pas en mesure de quantifier efficacement le poids des éléments pertinents qui affectent le processus de jugement de la similarité sémantique.

1.2. Problématique :

La plupart des mesures de similarité sémantique existante qui utilisent la structure de l'ontologie comme source principale ne peuvent pas mesurer la similarité sémantique entre les termes et les concepts utilisent plusieurs ontologies. Le principe de calcul de similarité avec ces approches est basé sur l'idée suivante : plus le chemin entre deux nœuds est court plus ils sont plus semblables. L'autre notion qui caractérise ces

approches est que les arcs d'une taxonomie représentent des distances uniformes, par conséquent, ces approches présentent l'inconvénient que tous les liens sémantiques possèdent le même poids ce qui impose des difficultés au niveau de la définition et du contrôle des distances des liens sémantiques. Le deuxième problème est que peu d'attention a été accordée à l'étude des impacts des mesures traditionnelles de similarité sémantique sur la langue arabe que ce soit pour le cas intra-ontologie ou inter ontologies**.**

1.3. Objectif général et contributions :

Les techniques de similarité sémantique sont utilisées pour calculer la similarité sémantique (informations communes partagées) entre deux concepts en fonction de certaines ressources linguistiques ou de domaine comme les ontologies, les taxonomies, les corpus, etc. Les techniques de similarité sémantique constituent des composantes importantes de la plupart des systèmes de recherche d'informations et des systèmes basés sur la connaissance. Cette thèse présente une nouvelle technique pour mesurer la similarité sémantique entre les concepts basés sur l'ontologie. Les mesures proposées sont basées sur trois caractéristiques : (1) la longueur du chemin (modifié en fonction du poids de ressemblance entre deux concepts, (2) ces mesures sont une hybridation entre les mesures basées sur la structure des ontologies et le dictionnaire sémantique WordNet, (3) le calcul de la similarité sémantique des concepts dans une ou plusieurs ontologies. La principale contribution est la nouvelle approche pour mesurer la similarité des concepts dispersés dans plusieurs ontologies dans le même domaine et au même temps étudier l'impact d'utilisation de ces mesures sur le même couple d'ontologies écrites dans plusieurs langues (Anglais- Arabe- Français).

1.4. Organisation du manuscrit :

Le reste de la thèse est organisé en deux grandes parties.

La première partie "état de l'art" comporte :

Le chapitre 2 (Généralité sur les ontologies) : ce chapitre présente une vue d'ensemble sur les concepts de base des ontologies. Par la suite, un aperçu sur quelques ontologies les plus connues. Nous présentons les types d'intégration, les langages de description ainsi que les outils pour l'éditeur d'ontologie et nous finissons par le cycle de vie d'une ontologie et les différents moteurs d'inférence de cette dernière.

Le chapitre 3 (Méthodes de calcul de similarité sémantique) : ce chapitre concerne la présentation des différentes mesures de similarités sémantiques existantes, ces mesures sont classées en deux grandes catégories, les méthodes de similarité intra-ontologie et inter-ontologies.

La deuxième partie " Contributions de la thèse " comporte :

Le chapitre 4 (Nouvelle approche pour La similarité sémantique inter-ontologies) **:** L'architecture de notre approche proposée avec les différentes étapes détaillées, est exposée dans ce chapitre**.**

Le chapitre 5 (Expérimentation & Évaluation) : dans ce chapitre, nous évaluons notre approche proposée. Nous verrons donc le travail qui a été accompli ainsi que les résultats obtenus, toute ont détaillons l'ensemble des métriques de qualité sur lesquels se base l'évaluation de nos résultats.

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

PARTIE I

ÉTAT DE L'ART

Chapitre 2

Généralité sur les ontologies

1. Introduction :

Les ontologies sont des systèmes formels dont l'objectif est de représenter les connaissances du domaine par le biais des concepts définis et organisés les uns par rapport aux autres. La représentation ontologique des connaissances fournit un cadre de modélisation à travers une représentation structurée et formelle des connaissances, assurant la cohérence et l'intégrité du système.

Les ontologies sont largement utilisées et ont prouvé leur efficacité dans divers domaines, notamment l'ingénierie des connaissances, l'intelligence artificielle, la recherche d'informations et le commerce électronique, et elles sont au cœur du web sémantique. Cet engouement est motivé par le fait que les ontologies fournissent un mécanisme efficace permettant aux individus et/ou aux systèmes de gérer et de distribuer les connaissances d'un domaine spécifique.

Nous proposons dans ce chapitre d'étudier les ontologies, en premier lieu la notion d'ontologie et les éléments qui la composent. Par la suite, nous présentons quelques ontologies les plus connues. Nous entamons les types d'intégration avec les outils les plus en vue. Nous exposons les langages de description ainsi que les outils pour l'éditeur d'ontologie. Nous présentons le cycle de vie et les différents moteurs d'inférence d'une ontologie.

2. Définition d'une ontologie :

L'ontologie, selon Gruber, est " ***Une ontologie est une spécification formelle et explicite d'une conceptualisation***" [GRU 93], l'expression spécification explicite signifie, que la conceptualisation est représentée dans un langage qu'il soit naturel (Arabe, Français...) ou formel (logique de description, graphes conceptuels...). C'est une approche hiérarchique pour définir les concepts et les relations entre eux. L'ontologie fournit un vocabulaire standardisé pour décrire les entités du domaine. Les ontologies sont divisées en deux catégories en fonction de leur utilisation prévue : les ontologies à usage général et les ontologies spécifiques à un domaine. De nombreuses recherches utilisent les ontologies comme ressources de connaissances pour mesurer la similarité sémantique entre les mots [GAN13].

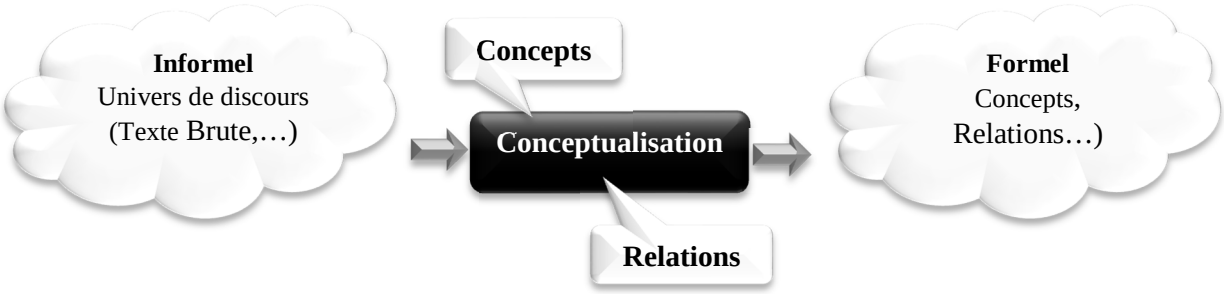


Figure 1. La conceptualisation

En 1997, Borst [BOR97] a modifié légèrement la définition de Gruber en citant qu'une ontologie est définie comme étant ***: « une ontologie est une spécification formelle d'une conceptualisation partagée ».*** Cette définition précise d'une part, le fait que l'ontologie doit être 'formelle', c'est-à-dire exprimée sous forme d'une logique pouvant être exploitable par une machine. D'autre part, elle doit être 'partagée' dans la mesure où elle doit capturer des connaissances partagées entre différents individus.

- ***Spécification explicite*** signifie que les concepts, les propriétés, les relations, les restrictions et les axiomes de l'ontologie sont définis de façon déclarative ;
- ***Formelle*** réfère au fait qu'une ontologie doit être traduite dans un langage interprétable par une machine ;

- ***Conceptualisation*** réfère à un modèle abstrait d'un phénomène du monde en identifiant les concepts appropriés à ce domaine ;
- ***Partagé*** réfère au fait où une ontologie capture la connaissance consensuelle c'est-à-dire non réservée à quelques individus, mais partagée par un groupe ou une communauté.

En 2000 Aussenac-Gilles et ses collègues [AUS00] énonce que :

« Une ontologie organise dans un réseau des concepts représentant un domaine. Son contenu et son degré de formalisation sont choisis en fonction d'une application ».

Cette définition souligne la dépendance entre le degré de formalisation de l'ontologie et son contenu avec l'application dans laquelle elle va être utilisée.

3. Les composants d'une ontologie :

Les ontologies permettent de représenter les connaissances et de les manipuler automatiquement tout en gardant leurs sémantiques. La sémantique des concepts en termes de concepts, propriétés, instances et relations. Dans ce qui suit, nous allons aborder ces notions plus en détails.

3.1 Les classes/ des concepts :

Un concept, également appelé classe dans certains travaux où outils, il représente l'idée que l'on se fait d'un terme : le contenu. Il est porteur d'une connaissance. Il peut désigner un objet concret comme :(حاسوب = ordinateur) ou abstrait comme : (معلومة = information).

3.2 Les propriétés :

La propriété est une caractéristique qui qualifie un concept et qui peut généralement être dotée d'une valeur. Si nous prenons l'exemple précédent (حاسوب =ordinateur) nous pouvons désigner quelques propriétés comme :

(نوع_الحاسوب =Type_de_l'ordinateur),

(سعة _ التخزين =capacité_de_stockage), (سرعة_الحساب = vitesse_de_calcul).

3.3 Les instances :

Les éléments individuels d'un concept concret, également appelés instances dans certains travaux, sont des éléments singuliers du concept. Les instances ne sont nécessaires que lorsque le but de l'ontologie est de servir dans la création d'une base de connaissances.

3.4 Les relations :

Les relations sont deux concepts qui interagissent l'un avec l'autre. La relation la plus souvent utilisée, la relation généralisation-spécialisation (عن_ عبارة = est un), est sans doute celle qui définit la hiérarchie de la structure ontologique. B est _un A, exprime le fait que le concept B est un sous-concept du concept A, en ce sens que B hérite de toutes les propriétés de A en plus d'avoir les siennes. **Exemple:** (الحاسوب عبارة_عن آلة), (الطابعة عبارة_عن آلة) et (الحاسوب الشخصي عبارة _عن حاسوب)

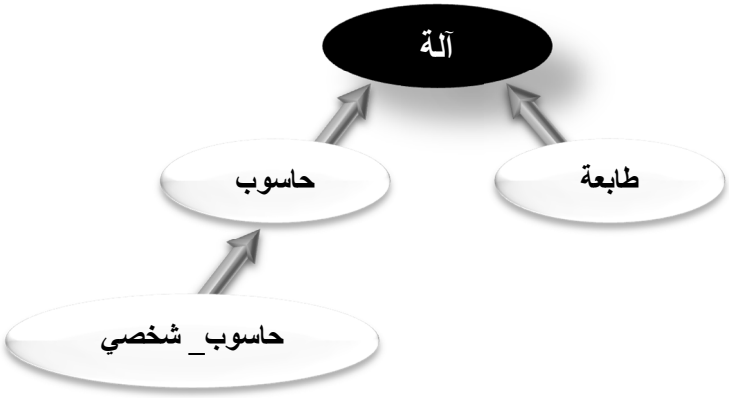


Figure 2. Exemple de la relation de généralisation-spécialisation

Il existe deux types de relations : celles qui sont sans rapport avec le domaine et celles qui lui sont étroitement liées. Les relations indépendantes du domaine peuvent être appliquées à n'importe quel champ de spécialité, les plus connues sont (عن _ عبارة = est un), (جزء_من = partie de), (مرادف_ل = synonyme de) ;

(ضذ=opposé de) etc.

Les relations dépendantes d'un domaine, ont un sens précis dans ce domaine

Exemple : (متصل_ب = est connecté à) dans (الطابعة متصلة بالكمبيوتر)

3.5 Les axiomes :

Les axiomes sont des assertions logiques qui nous permettent de définir la signification de concepts spécifiques, de définir des restrictions sur des propriétés, d'évaluer la cohérence logique d'une ontologie et d'inférer de nouvelles connaissances.

4. Les ontologies les plus connues :

L'un des défis les plus recherchés dans le domaine de la programmation est le traitement du langage naturel. De nombreuses équipes ont tenté de développer des systèmes NLP capables de lire et de comprendre des textes anglais ordinaires, mais le multilinguisme affecte désormais de nombreux systèmes. Un certain nombre de projets issus de différentes époques et équipes sont présentés ci-dessous. Nous présentons dans ce qui suit les ontologies qui ont un lien direct avec notre travail et résumons d'autres.

4.1 WordNet :

WordNet est le produit d'un projet de recherche de l'Université de Princeton [MIL98]. Selon Meng, Huang, & Gu [GRU 93] WordNet est une grande base des données lexicales de l'anglais. Les noms, verbes, adverbes et adjectifs de WordNet sont organisés par ensemble de relations sémantiques en ensemble de synonymes (synsets), qui représentent un concept. Des exemples de relations sémantiques utilisées par WordNet sont la synonymie, l'autonomie, l'hyponymie, le membre, le similaire, le domaine, etc. Ces relations sont représentées sous forme de structure hiérarchique, ce qui en fait un outil utile pour la linguistique informatique et le traitement du langage naturel [MIL90]. WordNet est utilisé par de nombreux chercheurs pour mesurer la similarité sémantique ou la parenté entre une paire de concepts, car il organise les noms et les verbes de manière hiérarchique.

4.2 SUMO:

SUMO [PEA10] (Suggested Upper Merged Ontology). SUMO définit une hiérarchie de classes , avec leurs règles et relations associées. SUMO est encodé dans un langage formel appelé SUO-KIF (Standard Upper Ontology-Knowledge Interchange format).

Le but de SUMO est de favoriser l'interopérabilité de données, la recherche, la récupération de l'information, l'inférence automatisée, et le traitement du langage naturel. SUMO contient des termes choisis pour couvrir les concepts les plus généraux. Il y a environ 4 000 assertions comprenant plus de 800 règles et 1000 concepts (lorsque toutes les ontologies créées pour SUMO sont chargées on obtient 74 693 assertions comprenant 2748 règles et 21223 concepts). De plus SUMO est la seule ontologie formelle qui a été mise en correspondance avec l'ensemble du lexique WordNet. SUMO est libre et appartient à l'IEEE.

4.3 Les ontologies arabes :

Plusieurs ontologies arabes ont été développées pour le traitement du langage naturel arabe. Les ontologies arabes sont très importantes pour mesurer la similarité entre les concepts arabes. L'ontologie arabe la plus connue est le WordNet arabe.

[ALY10] ont proposé un modèle de calcul pour décrire les concepts arabes en utilisant des ontologies. Le modèle a été construit en utilisant des données obtenues à partir du Saint Coran. Le nouveau modèle peut facilement être étendu et lié à d'autres ontologies telles que SUMO. Le modèle a été mis en œuvre sur le vocabulaire de la langue arabe lié au vocabulaire "temps" dans le Saint Coran. Selon les auteurs, les résultats de l'évaluation montrent que le modèle est capable de décrire la sémantique des mots d'une manière qui peut soutenir l'analyse sémantique des mots arabes.

[JAR11] Jarrar a présenté une méthodologie pour le développement d'une ontologie arabe formelle. Le travail proposé a pris en compte les relations sémantiques entre les concepts au lieu des mots. Contrairement à WordNet, l'ontologie arabe proposée se concentre sur les propriétés réelles des concepts. Jarrar souligne que la construction de l'ontologie arabe et la création du contenu arabe doivent être basées sur des principes ontologiques.

[MAZ12] ont proposé une approche de construction automatique sur une ontologie linguistique arabe en utilisant des techniques statistiques pour extraire les entités de

l'ontologie à partir du corpus arabe. L'auteur a utilisé la technique du "segment répété" pour déterminer les éléments connexes qui représentent les principaux concepts du domaine. Ils ont également utilisé la "co-occurrence" des concepts extraits pour définir les relations entre ces concepts dans l'ontologie. Pour accomplir le processus d'extraction, les auteurs ont utilisé corpus arabe précédemment préparé qui a été collecté à partir de livres et d'articles arabes.

[ISH14] ont présenté une ontologie lexicale arabe appelée Azhary. Comme AWN, elle classe les mots arabes en ensemble de synsets. Azhary contient 26 195 mots, regroupés en 13 328 synsets. Cette ontologie a été construite avec un certain nombre de relations entre les mots tels que les synonymes, les hyperonymes, les hyponymes, les antonymes, les holonymes et les relations d'association. Les auteurs s'appuient sur le Saint Coran pour créer les mots d'origine, les relations entre les mots arabes ont été construites manuellement en utilisant des dictionnaires connus. Selon les auteurs, l'ontologie Azhary a plus de mots et de relations entre les mots que l'AWN.

[FAR03] GOLD (General Ontology for Linguistic Description) est une ontologie pour la linguistique descriptive. Elle décrit les concepts et relations de base dans le domaine de la description du langage naturel. Elle a été conçue pour résoudre le problème du marquage des données linguistiques. GOLD se concentre sur les unités de forme, et non sur le sens ou la sémantique. GOLD est un effort pour connecter cette ontologie de domaine linguistique à une ontologie supérieure telle que SUMO.

[BEL09] décrit une représentation ontologique pour la langue arabe. La conception de l'ontologie est basée sur une relation taxonomique entre les principales classes morphologiques de la langue arabe. Les verbes sont classés selon les règles de dérivation de la langue arabe. Bien que la recherche soit pertinente, seul un modèle théorique est décrit et aucune implémentation n'est fournie.

Al-Khalil [ALI10] propose une ontologie OWL qui est basée sur l'ontologie linguistique GOLD [FAR03] .Son objectif est de fournir une référence pour la description de la linguistique arabe en se concentrant sur la grammaire traditionnelle arabe. La conception de l'ontologie suit l'ordre phonétique comme dans le livre Kitab AlAyn de Al-Khalil Ibn Ahmad AlFarahidi. Bien que le projet soit prometteur, il n'y a pas d'implémentation disponible.

[DUK13] Quranic Arabic Corpus fournit une classification ontologique des concepts présents dans le Saint Coran. Le Corpus d'arabe coranique est une ressource linguistique annotée composée de 77 430 mots d'arabe coranique. L'annotation de corpus attribue une étiquette de partie du discours et des caractéristiques morphologiques à chaque mot. Par exemple, l'annotation consiste à déterminer si un mot est un nom ou un verbe, et s'il est infléchi au masculin ou au féminin. La première étape du projet a consisté en un étiquetage automatique de la partie de la parole en appliquant la technologie informatique de la langue arabe au texte.

[BLA06] Arabic WordNet (AWN) est une ressource lexicale gratuite pour l'Arabe standard moderne basée sur le WordNet de Princeton pour l'anglais, largement utilisé. AWN se base sur la conception et le contenu de WordNet pour permettre la traduction automatique. Tout comme WordNet, AWN a également été relié à l'ontologie supérieure fusionnée suggérée (SUMO) [ELK06]. Il suit un processus de développement similaire à celui de WordNet et possède une structure sémantique comparable. Il utilise l'ontologie SUMO pour établir des liens avec des WordNets dans d'autres langues. Le navigateur AWN fournit une représentation arborescente des termes. Il présente les concepts de haut niveau des noms, verbes, adjectifs et adverbes. Les synsets correspondent à un terme général SUMO ou à un terme directement équivalant au synset donné. L'AWN a été utilisé dans un certain nombre d'applications telles que la recherche du passage pour la réponse à des questions.

5. Les Langages de description des ontologies :

Plusieurs formalismes et langages pour représenter les ontologies ont été proposés depuis les années 1990. Le langage OWL reste le standard le plus utilisé pour la représentation des connaissances. Nous allons examiner dans cette section les différents langages et formats qui ont donné naissance à OWL.

5.1. Le langage XML et XML schéma :

Le XML est un langage permettant de décrire des métadonnées et de faciliter leur traitement et leur partage. Il décrit des structures de données ordonnées en utilisant un langage de marquage basé sur le texte. Les définitions de type de document (DTD) qui décrivent la structure des documents XML sont liées aux fichiers XML. Toutefois, les DTD ne sont pas suffisamment expressives lorsqu'il s'agit de décrire des structures de données de haut niveau. XML Schéma est un ensemble plus complet de structures, de type et de contraintes pour spécifier des données. Il inclut des capacités de définition et d'organisation des documents XML à un niveau élevé. Comme le XML manque de sémantique, c'est pour cette raison qu'il nécessite le développement de langages supplémentaires.

5.2. Le langage RDE et RDFS :

Le Ressource Description Framework (RDF) [CYG14] est un langage normalisé de représentation de l'information pour les ressources.

Le W3C (World Wide Web Consortium) l'a choisi comme base du Web sémantique.

RDF est un modèle de graphe utilisant la syntaxe XML pour décrire les ressources Web et leurs métadonnées de manière formelle et pour permettre le traitement automatique de ces descriptions.

Les URI (Uniform Ressource Identifier) sont utilisés pour identifier les ressources dans ce langage. Un triplet sujet-prédicat-objet décrit ces ressources.

Le sujet est la ressource à décrire, le prédicat est un attribut qui exprime un lien avec le sujet, et l'objet est la valeur du prédicat. Les triplets RDF peuvent être représentés graphiquement sous la forme d'un graphe ou décrits en XML.

RDF n'est pas un langage ontologique au sens propre du terme et reste limité, puisqu'il ne définit pas les propriétés des classes, mais sa structure générique a servi de base aux langages d'ontologies présentés ultérieurement, RDFS étant le premier.

RDFS (Ressource Description Framework Schéma) [CAL14] est un langage qui étend à RDF un vocabulaire de termes et leurs relations, tels que Class, Property, type, subClassOf, subPropertyOf, range, et domain. Par conséquent, il est reconnu comme un langage ontologique qui définit les classes, les propriétés, les sous-classes, les super-classes, les sous-propriétés et les super-propriétés.

5.3. Les langages DAML-Oil :

DAML-Oil [HOR02] est le résultat de la combinaison de deux langages, DAML et Oil, pour ajouter plus d'expressivité au langage RDFS. DAML (Darpa Agent Markup Language) [KAL11] est un langage développé par l'Union européenne qui se veut plus efficace que XML pour décrire les objets et leurs relations pour exprimer la sémantique. L'objectif du langage DAML est de fournir les fondations pour la génération du Web sémantique. Parallèlement, un groupe de chercheurs européen ont créé Oil (Ontology Interchange Language), un langage de description d'ontologie basé sur RDF [KAL11].

5.4. Le Langage OWL :

OWL (Ontology Web Langage) est un langage permettant de définir des classes et des types de propriétés, et ainsi à la définition d'ontologies. Le groupe de travail du W3C sur le Web sémantique a produit OWL, qui est un dialecte XML. Il est dérivé du langage ontologique DAML -Oil et constitue une extension du vocabulaire RDF. OWL est construit sur un langage strict qui définit une sémantique formelle. Il inclut des constructeurs pour l'identité, l'équivalence, le contraire, la cardinalité, la symétrie, la

transitivité, la disjonction et d'autres propriétés et classes [LAN15]. En conséquence, parce qu'il possède un vocabulaire plus large et une véritable sémantique formelle, OWL offre aux machines une plus grande capacité d'interprétation du contenu Web que RDF et RDFS. Les triples RDF sont plus fréquemment utilisés pour définir la syntaxe d'un document OWL.

6. Outils de construction d'ontologies :

Il existe principalement deux catégories d'outils de construction d'ontologies.

6.1 Outils pour l'éditeur d'ontologie :

Il s'agit d'éditeurs d'ontologies qui permettent aux utilisateurs de définir de nouveaux concepts, relations et instances. Ces outils ont généralement des capacités d'importation et d'extension d'ontologies existantes.

Les outils de développement comprennent généralement des navigateurs graphiques, des capacités de recherche et de vérification des contraintes. Protégé 2000, OntoEdit, OilEd, WebODE, Ontolingua et Swoop sont quelques exemples d'outils de développement.

- **Protégé :**

Protégé a été développé à l'Université de Stanford [FER97], c'est un éditeur d'ontologie et souvent utilisé dans le Web sémantique et au niveau de la recherche en informatique. Protégé est un logiciel qui permet aux utilisateurs de créer et de modifier des ontologies, il est considéré comme tel, car il s'agit de l'éditeur d'ontologie open source le plus abouti et le plus répandu (utilisée par une communauté de plus de 76000 membres dans le monde). Son interface graphique permet de définir facilement des classes et de les organiser dans une hiérarchie de classes/sous-classes.

Il permet également de définir les propriétés associées aux classes. L'architecture du logiciel permet aux utilisateurs d'insérer des plugins qui ajoutent de nouvelles

fonctionnalités tout en exploitant les avancées les plus récentes de la recherche ontologique (gestion des bases de connaissances, visualisation des ontologies, alignement et fusion, etc.)

Protégé peut lire et enregistrer des ontologies dans divers formats, notamment RDF, RDFS, OWL et autres [MAL11].

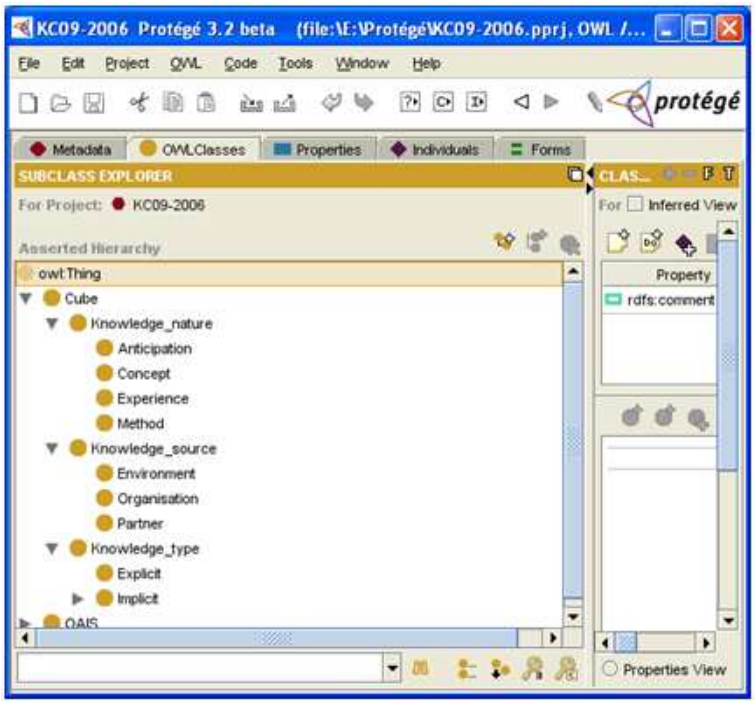


Figure 3. Interface graphique de Protégé

- **Ontolingua :**

Le serveur Ontolingua [COR03, FAR97] a été le premier outil ontologique créé et développé au début des années 90 au Knowledge Systems Laboratory (KSL) de l'Université de Stanford. Il a été construit pour faciliter le développement des ontologies Ontolingua avec des applications basées sur des formulaires. La création d'une nouvelle ontologie est facilitée par la possibilité d'inclure des parties d'ontologies existantes provenant d'un référentiel. Ce référentiel est constitué d'un grand nombre d'ontologies provenant de différents domaines. Une fois l'ontologie nouvellement générée, elle peut être ajoutée au référentiel pour une éventuelle réutilisation. La figure suivante (voir figure 4) donne un aperçu de cet outil.

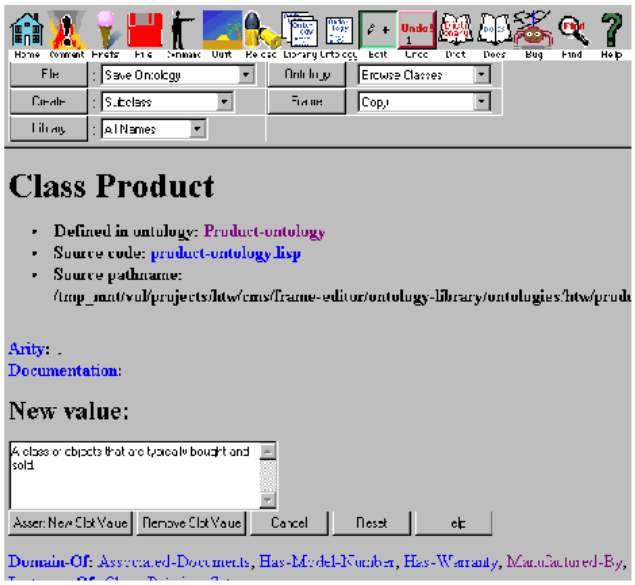


Figure 4. Capture d'écran d'Ontolingua

- **OilEd :**

OilEd a été initialement développé comme un éditeur d'ontologies pour Ontology Inference Layer (OIL) [COR03, FEN01]. OilEd a évolué et il est maintenant éditeur d'ontologies DAML+OIL. Les utilisateurs d'OilEd peuvent se connecter au moteur d'inférence FACT, qui fournit des fonctions de vérification de la cohérence et de classification automatique des concepts. Il supporte également plusieurs options de documentation avec HTML et la visualisation graphique des ontologies. La figure suivante (voir figure 5) montre une capture d'écran de l'OilEd.

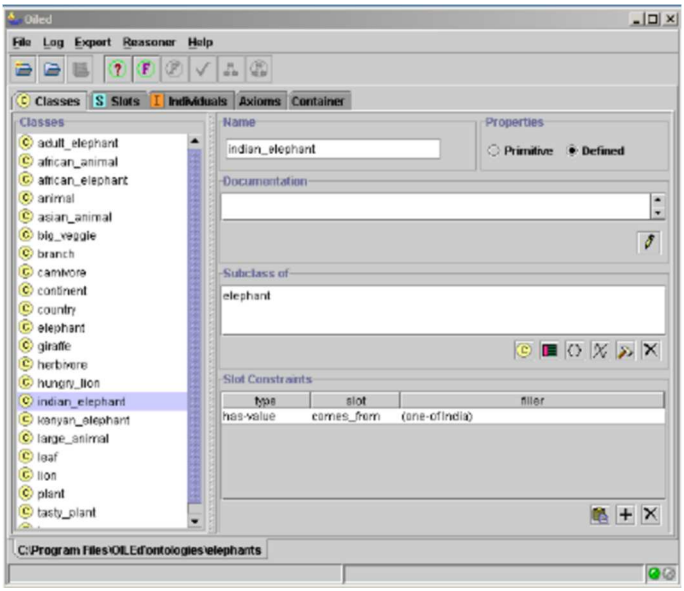


Figure 5. Capture d'écran d'OilEd

• **WebODE :**

WebODE est le successeur de « Ontology Design Environment (ODE) ». Il a été développé dans le laboratoire d'intelligence artificielle de L'Université polytechnique de Madrid. Il n'est pas utilisé comme une application autonome mais comme un serveur web avec une interface web. Le cœur de cet environnement est le service d'accès à l'ontologie, utilisé par tous les services et applications connectés au serveur. Il existe plusieurs services pour l'importation/exportation du langage ontologique, la documentation de l'ontologie, l'évaluation de l'ontologie et la fusion de l'ontologie. Les ontologies de WebODE [COR03, ARP01] sont stockées dans une base de données relationnelles.

Figure 6. Capture d'écran du WebODE

• **OntoEdit :**

OntoEdit a été développé par l'AIFB à l'Université de Karlsruhe. Il s'agit d'un environnement extensible et flexible basé sur une architecture de plugins qui fournit des fonctionnalités pour parcourir et éditer les ontologies [COR03]. Cet outil est disponible en deux versions : gratuite et professionnelle. Il comprend des plugins pour exporter et importer des ontologies dans différents formats. L'outil est basé sur un

cadre flexible de plugins. Il permet tout d'abord d'étendre les fonctionnalités de manière modulaire. L'interface des plugins est ouverte aux tiers, ce qui permet aux utilisateurs d'étendre facilement OntoEdit avec des fonctionnalités supplémentaires. Deuxièmement, le fait de disposer d'un ensemble de plugins, tels qu'un lexique de domaine, un plugin d'inférence et plusieurs plugins d'exportation et d'importation, permet une personnalisation conviviale de l'outil pour différents scénarios d'utilisation. Il permet également le développement collaboratif d'ontologies pour le web sémantique [SUR02]. La capture d'écran d'OntoEdit est présentée comme suit :

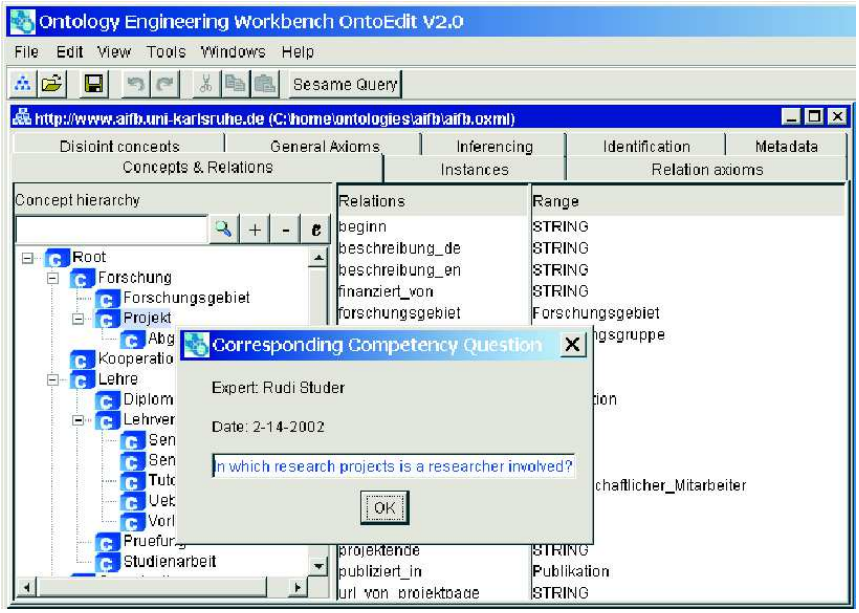


Figure 7. Aperçu d'OntoEdit

- **Apollo :**

Apollo [MAL11] est une application conviviale de développement d'ontologies. Un aperçu d'Apollo est présenté à la figure 8. Une représentation hiérarchique des ontologies existantes est présentée dans le volet supérieur gauche. La hiérarchie des classes et des instances est présentée dans le volet inférieur gauche. Une fois sélectionnée, une classe ou une instance est présentée en détail dans les volets situés à droite de l'écran. Les emplacements et les valeurs d'une classe ou d'une instance peuvent ensuite être ajoutés à l'aide d'une interface de type feuille de calcul.

Apollo prend en charge toutes les primitives de base de la modélisation des connaissances, comme la création d'ontologies par la définition de classes, d'instances, de fonctions et de relations. Une vérification complète de la cohérence est effectuée lors de l'édition. Apollo dispose de son propre langage interne pour stocker les ontologies, mais peut également exporter l'ontologie à partir de différents langages de représentation. Apollo est implémenté en utilisant le langage Java.

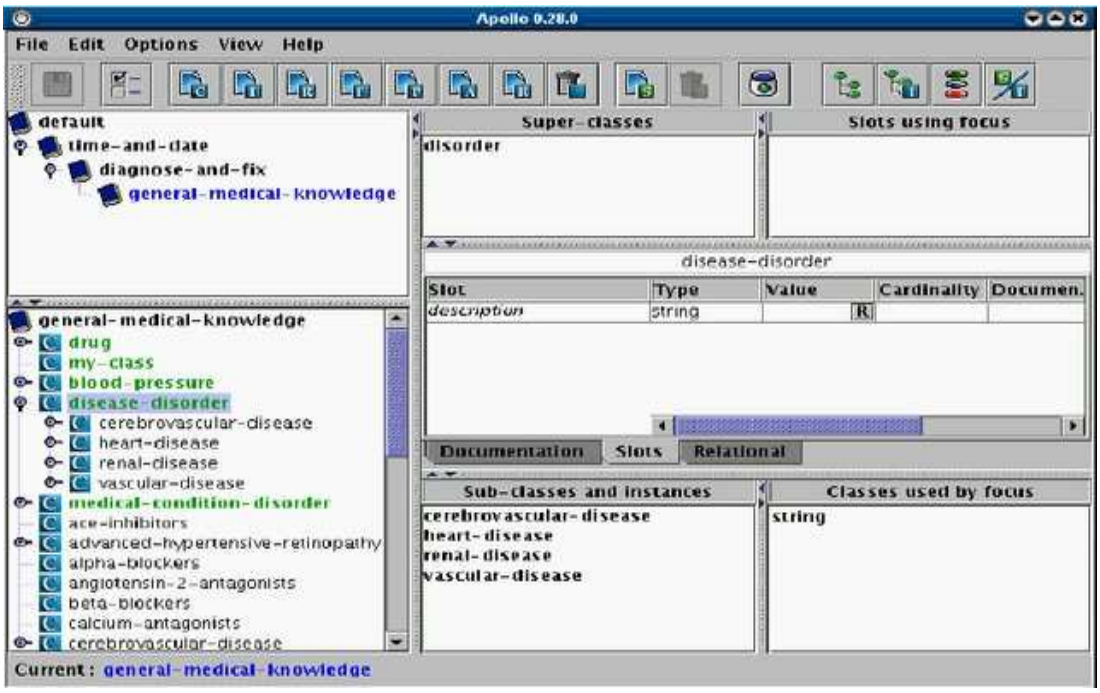


Figure 8. Aperçu d'Apollo

- **Swoop :**

SWOOP [KAL06] est un éditeur et un navigateur d'ontologie OWL basé sur le Web. SWOOP contient la validation OWL et offre diverses vues de la syntaxe de présentation OWL. Il fournit un environnement d'ontologie multiple. Les ontologies peuvent être comparées, éditées et fusionnées. Différentes ontologies peuvent être comparées à leurs définitions basées sur la logique de description, aux propriétés associées et aux instances. L'interface de SWOOP a des capacités d'hyperliens de sorte que la navigation peut être simple et facile. SWOOP ne suit pas de méthodologie pour la construction d'ontologies, mais les utilisateurs peuvent réutiliser des données ontologiques externes. Cela est possible soit en créant un lien vers l'entité externe, soit

en important l'ontologie externe dans son intégralité. Il n'est pas possible de faire des importations partielles d'OWL. Il existe plusieurs façons d'y parvenir, comme un schéma syntaxique de force brute pour copier/coller les parties pertinentes (axiomes) de l'ontologie externe, ou une solution plus élégante qui implique de partitionner l'ontologie externe tout en préservant sa sémantique, puis de réutiliser (importer) uniquement la partition spécifique souhaitée.

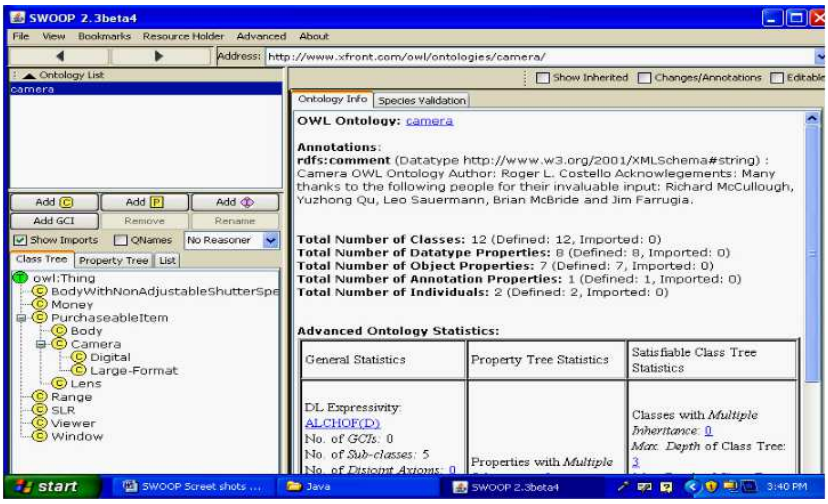


Figure 9. Capture d'écran de SWOOP

6.2. Étude comparative des outils d'ontologie :

Les résultats de la comparaison des outils sont présentés dans le tableau suivant (voir table 1), et qui sont classés sur la base des caractéristiques suivantes :

(i) Description générale des outils.

(ii) Architecture logicielle et évolution des outils.

(iii) Interopérabilité.

(iv) Représentation des connaissances et support méthodologique.

(v) Des services d'inférence qui lui sont attachés.

(vi) Utilisabilité des outils.

Caractéristiques ↓	Protégé	Ontolingua	OilEd	WebODE	OntoEdit	Apollo	Swoop
Disponibilité	Open Source	Web libre accès	Freeware	Web libre accès Licences	Freeware Et Licences	Open Source	Open Source

Développeurs	SMI Université de Stanford	KSL Université de Manchester	Université de Manchester	Laboratoire d'intelligence artificielle Université polytechnique de Madrid	AIFB l'Université de Karlsruhe	KMI L'université ouverte du Royaume-Uni	MND Université de Maryland
S/w Architecture	Standalone Client/ Server	Client/ Server	Standalone	3-tier	Client/ Server et Standalone	Standalone	Web-based Client/ Server
Extensibilité	via Plugins	No	No	via Plugins	via Plugins	via Plugins	via Plugins
Back up	☒	☒	☒	☑	☒	☒	☒
Stockage	Fichiers, DBMS (JDBC)	Fichiers	Fichiers	DBMS (JDBC)	Fichiers, DBMS (JDBC)	Fichiers	Fichiers
Format d'Importation	XML, RDF(S), XML Schema OWL	Ontolingua IDL KIF	RDF(S), OIL, DAML+ OIL	XML, RDF(S), CARIN	XML, RDF(S), FLogic, DAML+ OIL	OCML	RDF (S), OIL, DAML,
Format d'exportation	XML, RDF(S), XML Schem, Java, html	KIF-3.0, CLIPS CML, LOOM	OIL,RDF(S) , DAML+ OIL, SHIQ, Dotty HTML	XML,RD F(S),OIL, DAML+ OIL,CARI N,FLogic ,Prolog,Jess, JAVA	XML, RDF(S), FLogic, DAML+ OIL, SQL-3	OCML	RDF (S), OIL, DAML,
Le langage des axiomes	☑	☑	☑	☑	☑	Sans restrictio n	☑
KR Paradigme	Frames+ FOL+ Meta Classes	Frames+ FOL	DL (DAML+ OIL)	Frames+ FOL	Frames+ FOL	Frames	OWL
support méthodologique	☒	☒	☒	☑	☑	☒	☒
Moteur d'inférence intégré	☑	☒	☑	☑	☑	☒	☒
Moteur d'inférence attaché	JessFaCT FLogic	ATP	☒	Jess	☒	☒	Pellet and RDF
Ontologie bibliothèques	☑	☑	☑	☒	☑	☑	☒
Collaboration avec d'autres outils	☒	☑	☒	☑	☑	☒	☒

☑ ---- indique Oui **☒ ---- indique Non**

Table 1 : Comparaison Entre Les Différents Outils De Développement D'ontologies

6.3. Intégration des ontologies :

Dans cette partie, nous présentons trois types d'intégration d'ontologies, l'alignement, la fusion et le mapping d'ontologies tout en spécifiant quelques outils existants.

6.3.1. Les types d'intégration des ontologies :

6.3.1.1. Alignement d'ontologies :

De nombreuses définitions de l'alignement ont été présentées dans la littérature. Deux d'entre elles sont citées ci-dessous. [RAH01] Propose une méthode d'alignement des schémas de base de données sous la forme d'une fonction considérant deux schémas de bases de données, cette fonction génère une correspondance entre les éléments du premier avec ceux du second schéma.

Le résultat est un ensemble de correspondances exprimant une relation entre les éléments du premier schéma et un élément du second schéma.

Les auteurs dans [SHV05] proposent une définition complète de l'alignement d'ontologies. Des éléments supplémentaires pour la méthode d'alignement sont introduits dans cette définition telle que l'alignement initial, les seuils, les pondérations et les ressources externes. L'approche d'alignement proposée utilise l'alignement initial (ou d'entrée) pour l'enrichir. L'approche utilise les seuils et les poids comme paramètres pendant la procédure d'alignement. Les ressources externes, comme les thésaurus et les connaissances du domaine, servent de support sémantique à la production de l'alignement. Cette définition est plus formellement, exprimée comme suit (voir figure 10). Une procédure d'alignement est une fonction notée *f*. Cette fonction prend en entrée deux ontologies (O et O'), un alignement initial (A), un ensemble de paramètres (p) et un ensemble de ressources (r). La fonction f produit un alignement (A') comme résultat. Formellement, la fonction f est définie comme suit : $A'=f(O, O', A, p, r)$ ou plus simplement $A'=f(O ; O')$ lorsque le contexte est fixe, et l'alignement initial, les paramètres et les ressources sont omis.

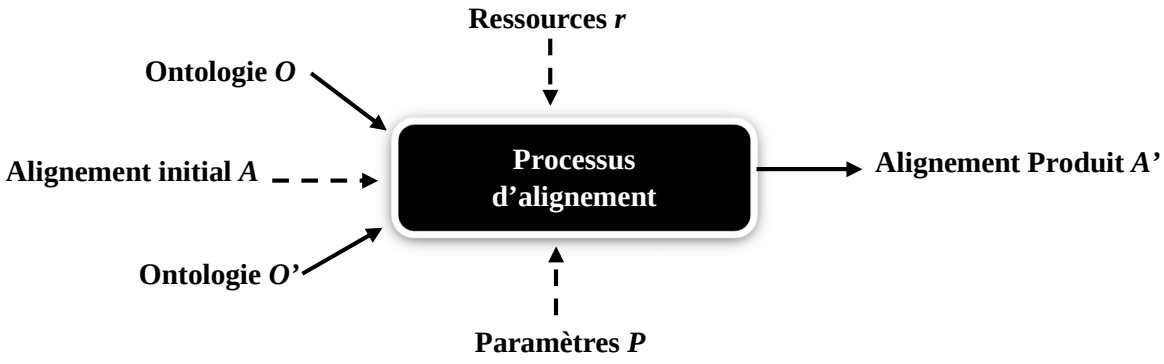


Figure 10. Schéma général d'un processus d'alignement d'ontologies

6.3.1.2. La Fusion d'ontologies :

La fusion d'ontologies est définie par Namyoun et ses collègues [FAQ15] comme suit " **Ontology merging is the process of generating a single, coherent ontology from two or more existing and different ontologies related to the same subject**".

La fusion d'ontologies est le processus qui consiste à combiner les connaissances de deux ou plusieurs ontologies existantes et distinctes qui décrivent le même sujet ou appartiennent au même domaine d'application en une seule ontologie. Les informations de toutes les ontologies sources sont incluses dans l'ontologie produite.

Dans la littérature, il existe deux grands types d'approches de fusion d'ontologies qui reposent principalement sur l'alignement des ontologies : la fusion complète et l'ontologie bridge [BRU06]. (Voir figure 11)

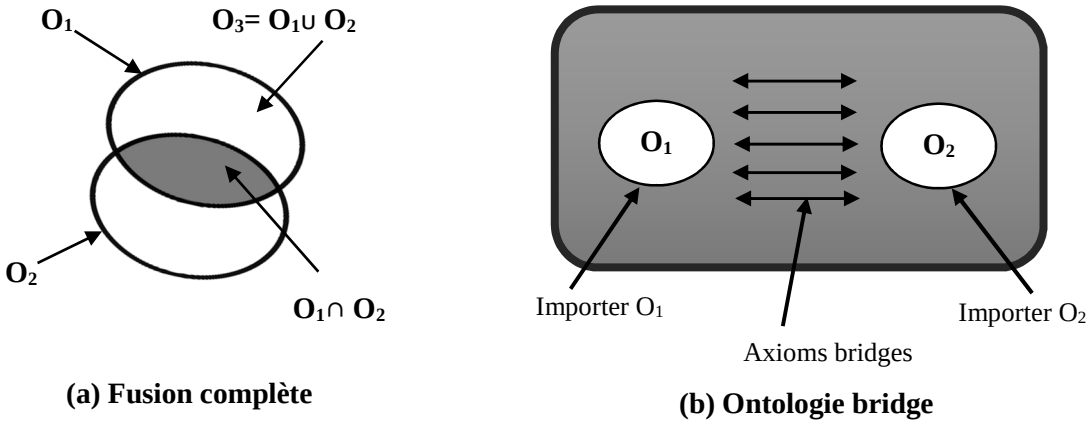


Figure 11. Les approches de fusion d'ontologies.

Le processus de fusion de la première approche donne une seule ontologie de sortie unique qui contient les ontologies sources. Prompt [NOY00], Chimaera [MCG00] et d'autres sont des exemples de cette méthode. Dans la deuxième approche, le processus de fusion résulte en une ontologie bridge qui importe les ontologies sources et associe des axiomes bridges ou des règles d'articulation exprimant les mappages entre les classes des ontologies sources. OntoMerge [DOU05], Onion [MIT02] sont des exemples de cette méthode.

Dans la littérature également, les approches de fusion d'ontologies peuvent être classifiées en deux types, les approches symétriques (full merge) et les approches asymétriques (target-driven mergin).

- **Les approches symétriques :** Elles permettent de fusionner des ontologies tout en conservant tous les éléments des ontologies sources dans l'ontologie globale. Les travaux sur la fusion d'ontologies (tels que Prompt, Chimaera, OntoMerge, etc.,) exploitent une solution symétrique.
- **Les approches asymétriques :** Elles prennent l'une des ontologies d'entrée comme cible et fusionnent les autres ontologies d'entrée dans cette cible, donnant ainsi la préférence à l'ontologie cible. Du coup, cette approche ne préserve que les concepts et les relations de l'ontologie cible et n'intègre que les concepts non redondants de l'autre ontologie.

6.3.1.3. Le mapping entre ontologies :

Il s'agit d'une spécification déclarative de la signification commune de deux ontologies basée sur la description des correspondances entre elles. Ces correspondances sont enregistrées indépendamment des ontologies, formant un composant séparé des ontologies sources et décrivant les liens entre les éléments respectifs via des axiomes énoncés dans un langage de mapping spécifique. Ces correspondances peuvent être de plusieurs types, notamment l'équivalence, la subsomption, l'exclusion et l'incompatibilité. Le processus d'alignement donne lieu à des mappings entre les ontologies sources. Elles peuvent être utilisées pour accomplir une variété d'activités différente du Web sémantique telle que la fusion et le versionning d'ontologies, la gestion des connaissances, l'intégration sémantique, etc.

Selon [MEL07], trois phases principales peuvent être distinguées dans le processus de mapping (voir la figure 12).

- La découverte du mapping ;
- La représentation du mapping ;

- L'exploitation et l'exécution du mapping.

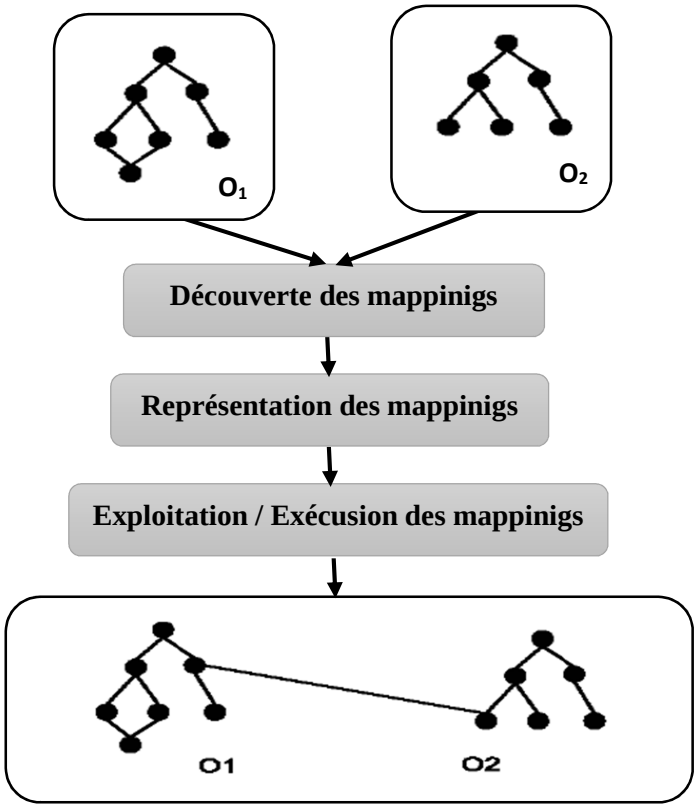


Figure 12. Le mapping des ontologies [MEL07]

6.3.2. Les outils d'intégration :

Pour réaliser l'intégration d'ontologies, un certain nombre d'outils ont été proposés dans la littérature. Elles sont fréquemment utilisées pour permettre le partage de données entre des bases de connaissances et la réutilisation des informations de ces bases, nous présentons un exemple d'outil pour chaque type d'intégration.

6.3.2.1. PROMPT :

PROMPT [NOY00] est un outil avec un processus de fusion interactive. L'ensemble des étapes impliqué dans ce processus comprend les étapes suivantes :

- Les candidats à la fusion sont déterminés en comparant les noms des classes, ensuite l'utilisateur reçoit une liste d'opération de fusion possible à la suite de l'analyse.

- L'utilisateur sélectionne l'une des actions de la liste ou spécifie explicitement l'opération de fusion.
- Le système effectue l'opération souhaitée, puis exécute automatiquement les modifications supplémentaires qui en découlent.
- Sur la base de la nouvelle structure ontologique, le système génère une nouvelle liste d'activités suggérées par l'utilisateur. Il détermine les conflits soulevés par l'action précédente, ainsi que les solutions viables, puis les affiche à l'utilisateur. PROMPT est principalement symétrique mais en cas des conflits, l'utilisateur intervient pour choisir la stratégie de résolution adéquate conduisant ainsi, à une solution partiellement asymétrique.

PROMPT spécifie un ensemble de procédures de fusion d'ontologies (fusion de classes, fusion de slots, fusion de liens, etc.) ainsi qu'un ensemble de conflits possibles consécutifs à l'application de ces opérations (conflits de noms, redondance dans la hiérarchie des classes).

Les outils PROMPT sont disponibles en tant que plugins pour l'outil Protégé.

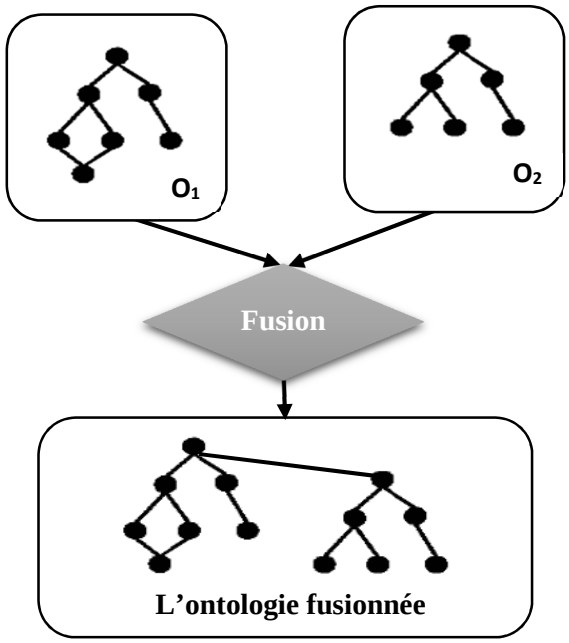


Figure 13. Le principe de la fusion d'ontologies [MEL07]

6.3.2.2. OntoMap :

OntoMap est un "plug-in" pour la plate-forme OntoStudio. Cela vous permet de créer et de gérer des mappings d'ontologies. Ceux-ci sont accessibles via une représentation graphique et avec l'environnement ontologique Schema-View [SCH05]. OntoMap associe des phrases formelles à des déclarations de mapping pour limiter le travail de l'utilisateur dans la compréhension de la sémantique des représentations graphiques (par exemple, une flèche reliant deux concepts).

Les utilisateurs d'OntoMap disposent également d'une fonctionnalité "Drag and Drop» et peuvent vérifier la cohérence des propriétés. OntoMap prend en charge un ensemble de modèles de mapping : concept à concept, attribut à propriété, relation à relation et attribut à concept.

6.3.2.3. Anchor-PROMPT :

Noy et Musen [NOY01] ont développé l'algorithme Anchor-PROMPT pour découvrir automatiquement des termes sémantiquement similaires.

Une ontologie est traitée comme un graphe dans Anchor-PROMPT, les nœuds de ce graphe représentent les classes et les arcs représentent les propriétés. En entrée, l'algorithme prend deux paires de termes relatifs. Il examine les chemins dans le sous-graphe limité par des ancres pour voir quelles classes apparaissent fréquemment dans des positions similaires sur des chemins similaires.

L'algorithme recherche ensuite dans le chemin des termes qui peuvent être similaires au terme d'autres chemins. Ces nouveaux termes relatifs sont identifiés par des similitudes qui peuvent être modifiées lors de l'évaluation des autres chemins dans lesquels ces termes apparaissent. Des termes très similaires sont présentés à l'utilisateur pour améliorer l'ensemble des suggestions possibles.

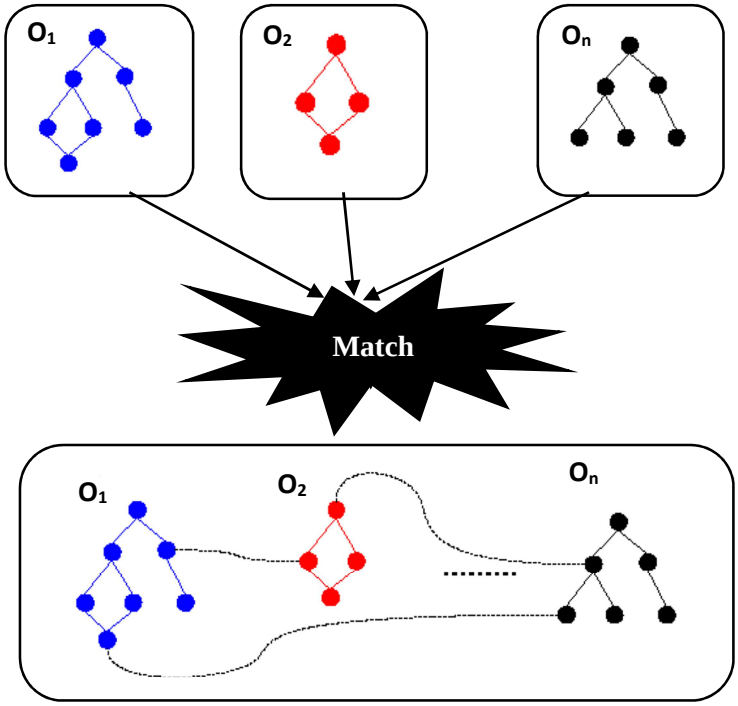


Figure 14. L'opérateur MATCH [MEL07]

7. Le Rôle des ontologies :

Le rôle et l'architecture des ontologies ont un impact important sur le formalisme de la représentation d'une ontologie. Les ontologies sont utilisées pour exprimer la sémantique de l'information source d'une manière formelle. Mais la manière, comment les ontologies sont employées, peut-être différente. En général, trois directions différentes peuvent être identifiées les approches à ontologie **simple**, les approches à ontologies multiples et les approches hybrides.

a) Approche d'une ontologie simple :

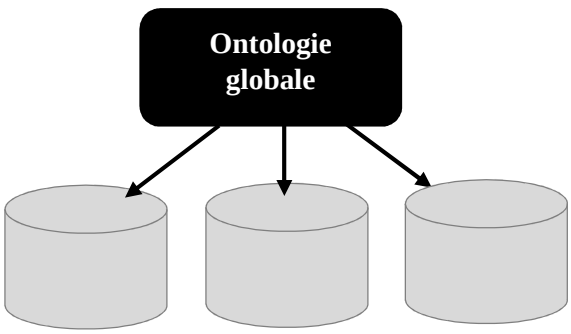


Figure 15. a) Approche d'une ontologie simple

Les approches fondées sur une ontologie simple utilisent une ontologie globale fournissant un vocabulaire partagé pour la spécification de la sémantique (voir figure 15. a). Toutes les sources d'informations sont liées à une ontologie globale. Une approche importante de ce type d'intégration d'ontologie est SIMS [ARE96]. Le modèle SIMS du domaine d'application comprend une base de connaissances terminologique hiérarchique. Chaque source est simplement liée à l'ontologie globale du domaine. Les approches d'ontologie simple peuvent être appliquées aux problèmes d'intégration où toutes les sources d'informations à intégrer fournissent presque la même vue sur un domaine, mais ils sont sensibles aux changements dans les sources d'informations qui peuvent affecter la conceptualisation du domaine représenté dans l'ontologie. Ces inconvénients ont conduit au développement d'approches d'ontologies multiple.

b) Approche à ontologies multiples :

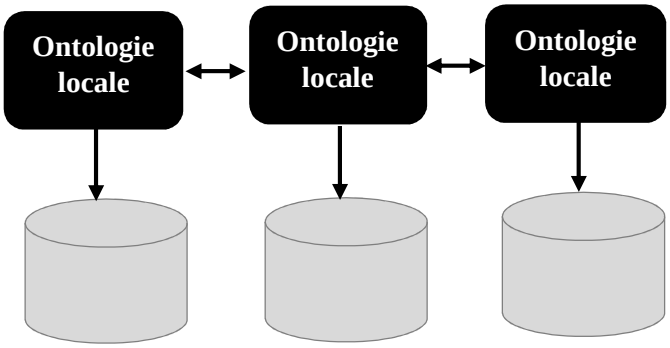


Figure 15. b) Approche à ontologies multiples

Chaque source d'information est décrite par sa propre ontologie. L'avantage est que chaque ontologie source peut être définie sans prendre en considération les autres sources ou les autres ontologies. Mais le manque d'un vocabulaire commun conduit à une difficulté extrême pour comparer différentes ontologies sources. Pour surmonter ce problème, un formalisme de représentation additionnelle définissant le mapping inter-ontologies est fourni. Ce dernier identifie sémantiquement les termes correspondants de différentes ontologies sources. Ce mapping est difficile à définir à

cause de divers problèmes d'hétérogénéité sémantiques qui peuvent se produire. L'approche OBSERVER [MEN00] est utilisée pour ce type.

c)Approche hybride :

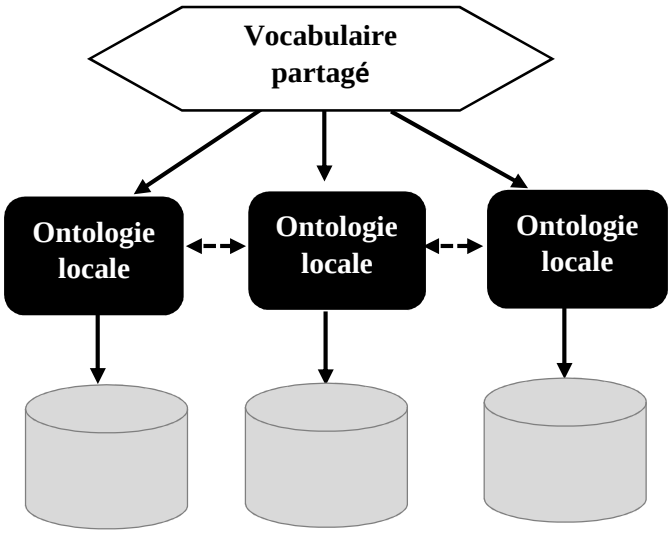


Figure 15. c) Approche hybride

Comme l'approche multiple, la sémantique de chaque source est décrite par sa propre ontologie. Mais pour rendre les sources d'ontologies comparables, elles sont construites sur un vocabulaire partagé global [GOH97, WAC99]. Ce vocabulaire partagé comprend les termes de bases ou primitives d'un domaine. Afin de construire des termes complexes, ces primitives sont combinées par des opérateurs, et les termes seront comparés plus facilement qu'une approche multiple. Parfois, ce vocabulaire partagé est représenté sous forme d'ontologie [VIS00]. Dans COIN [GOH97], la description locale d'une information, qu'on appelle contexte, est simplement un vecteur de valeurs d'attributs. Dans METACOTA [VIS00], chaque source d'information est annotée par un label qui indique la sémantique de l'information. Ce label combine les termes primitifs du vocabulaire partagé. Dans BUSTER [VIS00], le vocabulaire partagé est une ontologie générale. Une source d'ontologie est un raffinement de l'ontologie générale. L'avantage de cette approche est que des nouvelles sources peuvent être ajoutées sans le besoin de la modification du vocabulaire partagé. Elle maintient encore l'acquisition et l'évolution des ontologies.

8. Moteurs d'inférences

La plupart des moteurs d'inférences existante sont conçus pour raisonner sur les logiques de description des informations d'ontologie qui sont implicite, par conséquent, décrire les incohérences. Inférer implique de tirer une conclusion d'une série de propositions reconnues pour ‘’vrai”. Un des principaux services fournis par un raisonneur est la vérification si une classe A est une sous-classe d'une autre classe B. De plus, l'exécution des tests spécifiques sur toutes les classes dans une ontologie, permet à un raisonneur d'acquérir la hiérarchie déduite des classes d'ontologie. Le rôle d'un moteur d'inférence est alors la compilation. Il existe plusieurs moteurs comme : Racer, Pellet, Jena.

- **Racer :**

Le moteur d'inférence sans ambigüité, le plus connu et le plus utilisé dans le domaine est Racer [HAA99], grâce à ses performances et sa stabilité. Racer prend en considération les ontologies modélisées par son langage, mais il accepte aussi des ontologies définies en RDF ou OWL. Ceci à cause de la traduction de ces ontologies vers le langage utilisé par Racer. Les principaux services d'inférences de Racer :

1. Le test de satisfiabilité d'un concept (vérifier qu'un concept C accepte des instances),
2. Le test de subsomption de concepts (vérifier qu'un concept A est subsumé par un concept B),
3. Le test d'instanciation (vérifier qu'un individu I est instance d'un concept C, si seulement si $I \in C$).

Racer possède quelques points négatifs :

1. Racer n'autorise pas l'utilisation de type de données définies par l'utilisateur, car il dispose de ses propres types de données et il exécute une conversion avec les types de base.
2. Racer est un produit commercial, il n'existe pas de version libre d'utilisation.

- **Pellet :**

Pellet [SIR07] est le moteur le plus récent. Il est un des projets du MINDSWAP Group, « un groupe de recherche sur le Web sémantique de l'université du Maryland ». Pellet manipule des ontologies décrites en RDF ou OWL.

Les atouts de Pellet sont :

1. Pellet est open-source et réalisé en Java.
2. Pellet est un raisonneur OWL DL complet.
3. Pellet présente en cas d'incohérence dans l'ontologie des rétablissements possibles.

Les points négatifs de Pellet sont :

1. Il dispose d'une documentation pauvre par rapport à celle de Racer.
2. Actuellement Pellet ne permet pas l'utilisation de règles SWRL.
3. Pellet n'expose pas de système de souscription à un concept.

- **JENA :**

JENA est une bibliothèque de classes Java développée par HP qui simplifie le déploiement d'application pour le Web sémantique. Il permet de gérer des ontologies (RDFSchema, DAML + OIL, OWL) et de raisonner en utilisant les connaissances de l'ontologie. Il permet :

1. La création d'une classe : createClass retourne une OntClass (OntClass est une spécialisation de Ressource).
2. La création d'une propriété : createObjectProperty retourne une ObjectProperty (ObjectProperty est une spécialisation de Ressource).
3. L'utilisation de déclarations RDF.
4. Lecture et écriture RDF/XML.
5. Le stockage en mémoire ou sur disque de connaissances RDF.
6. La gestion d'ontologies : RDF-Schema, DAML+OIL, OWL.

Il existe de nombreux raisonneurs, dont certains sont spécifiques à OWL. Pour démontrer l'apport de chaque moteur d'inférence, nous avons présenté ce tableau (voir Table 2) qui englobe les caractéristiques de chaque moteur.

Jena	Racer	Pellet
1. API Java le plus largement utilisées pour RDF et OWL. 2. Représentation du modèle. 3. L'analyse syntaxique. 4. Les requêtes et quelques outils de visualisation. 5. Raisonner sur les instances et les concepts.	1. Raisonner sur les instances de concepts. 2. Permet la vérification de la hiérarchie entre concepts (subsomption de concepts).	1. Raisonner sur les instances de concepts. 2. Le premier raisonneur OWLDL sûr et complet. 3. Pas de raisonnement sur les concepts. 4. Ne peut pas être intégré dans JAVA

Table 2 : Caractéristiques des moteurs d'inférence

9. Cycle de vie d'ontologie :

Étant donné que l'ontologie est destinée à être utilisée en tant que composants logiciels dans des systèmes qui répondent à différents objectifs opérationnels, la conception doit être basée sur les mêmes principes que ceux utilisés en génie logiciel.

Par conséquent, l'ontologie doit être considérée comme ayant un cycle de vie dans lequel des entités techniques doivent être développées et définies. Dans ce contexte, les activités liées à l'ontologie sont des activités de gestion de projet (planification, contrôle, assurance qualité) d'une part et des activités de développement (spécification, conceptualisation, formulation) d'autre part. Il existe également des activités de support transversal tel que l'évaluation, la documentation et la gestion de la configuration.

Le cycle de vie d'une ontologie comprend une étape initiale de détection et de spécification des besoins, une étape de conception dans laquelle le domaine de connaissances peut notamment être précisément défini, qui se divise en trois phases : la phase de déploiement, de diffusion et une étape d'utilisation, une étape inévitable d'évaluation, et enfin, une sixième étape consacrée à l'évaluation et à la maintenance du modèle. Après chaque utilisation significative, l'ontologie doit être réévaluée, et

l'ontologie peut être étendue et partiellement reconstruite si nécessaire. La validation du modèle de connaissance est au cœur du processus et se fait de manière itérative. Le processus de construction peut être intégré dans le cycle de vie d'une ontologie comme l'indique la Figure 16. [KHA09]

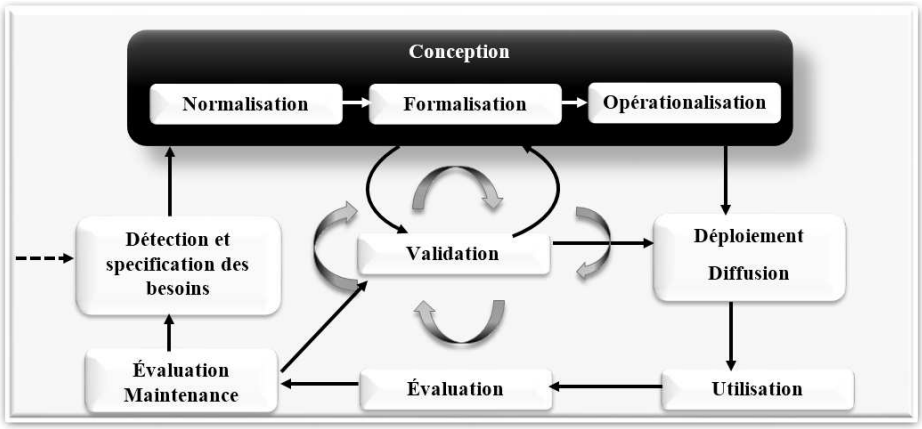


Figure 16. Cycle de vie d'une ontologie.

10. Conclusion :

Avant l'apparition des concepts ontologiques, les connaissances étaient principalement stockées dans des bases de données. Ils ont été confrontés à de nombreux défis et difficultés dans le partage et la réutilisation des informations et des connaissances qu'il contient.

Les ontologies sont utilisées pour résoudre ces problèmes et d'autres types de problèmes en tant que méthodes potentielles de représentation, de stockage qui facilitent le partage, la réutilisation des connaissances et des informations sont d'après ce que nous avons présenté dans ce chapitre, le concept d'ontologie apparaît comme une approche très efficace pour exprimer les connaissances.

Dans ce chapitre, nous avons essayé de clarifier le concept d'ontologie en présentant certaines définitions, nous avons exposé les composantes d'une ontologie, quelques ontologies les plus connues, ensuite nous avons présenté les types d'intégration avec les outils les plus en vue, nous avons montré les langages de description ainsi que les outils pour l'éditeur d'ontologie. Et finalement, nous avons présenté les différents moteurs d'inférence et le cycle de vie d'une ontologie.

Chapitre 3

Méthodes de calcul de Similarité Sémantique

1. Introduction :

La mesure de la similarité sémantique est exploitée dans plusieurs domaines de recherche, notamment l'intelligence artificielle, la recherche d'informations, la traduction automatique, l'extraction d'information ou la détection de plagiat.
La similitude entre deux concepts est identifiée par les humains en comparant leurs attributs communs et différents. Ces attributs sont considérés pour simuler le processus des jugements humains. Par conséquent, la similarité entre deux mots (anglais, Français ou Arabe) est calculée en fonction de la relation entre les attributs différents et communs des mots comparés dans la base de connaissances sémantiques approprier.

Dans ce chapitre, nous présentons une classification des principales approches de la mesure de la similarité sémantique. Ensuite, nous donnons un aperçu des mesures de similarité sémantique existant selon cette classification, toute on détaillons-les plus utilisées dans notre étude.

2. Définition de la mesure de similarité sémantique :

L'objectif des mesures de similarité sémantique est de déterminer la proximité sémantique entre les concepts. Le calcul de la similarité entre deux concepts permet de déterminer s'ils sont similaires, c'est-à-dire s'ils atteignent un niveau de similarité ou de dissimilarité, par exemple, dans la phrase : «صليت المغرب بالمغرب», le premier mot représente la prière et le deuxième signifie le lieu Maroc, Syntaxiquement, ces deux mots peuvent être considérés comme très proches, mais leur sémantique est différente. De ce fait, la similarité sémantique recouvre notamment l'étude du sens lexical (ou sens des mots) et l'étude du sens de combinaisons de mots.

Que veut dire une distance sémantique :

Une distance Sémantique '*d'* est une distance qui vérifier les propriétés suivantes :

- La distance entre deux concepts équivalents est nulle : *Si* $A \equiv B$ *alors d (A, B) =0*
- La distance entre un concept avec subsume directe et inférieure à sa distance avec n'importe quel autre subsumer : Si $A \subseteq B \subseteq C$ alors $d\ (A, B) \leq d\ (A, C)$
- La distance entre deux concepts incompatibles est infinie : Si $A \cap B \sqsubseteq \perp$ alors $d\ (A, B) = \infty$ ($\perp$ concept le plus spécifique)

Cette distance *'d'*est sémantique, car elle permet d'exprimer certains liens sémantiques intuitifs entre les concepts, comme le fait que la distance entre deux concepts égaux est nulle.

3. Classification des approches de similarité sémantiques en fonction de l'ontologie :

Plusieurs approches pour déterminer la similarité sémantique ont été proposées. Dans l'ontologie, l'approche de la similarité sémantique peut être classée en deux catégories intra-ontologie et inter-ontologies [ELA14], [SAR11]. La classification est basée sur la façon dont la mesure de similarité sémantique est quantifiée, cette dernière est soit basée sur la structure ontologique ou sur le contenu de l'information.

3.1. Les approches pour la similarité sémantique intra-ontologie :

Ce sont des approches qui supposent que les termes, qui sont comparés, sont issus de la même ontologie. Ces mesures de similarité sémantique utilisent l'ontologie comme source d'information principale. La figure présente une classification de ces approches.

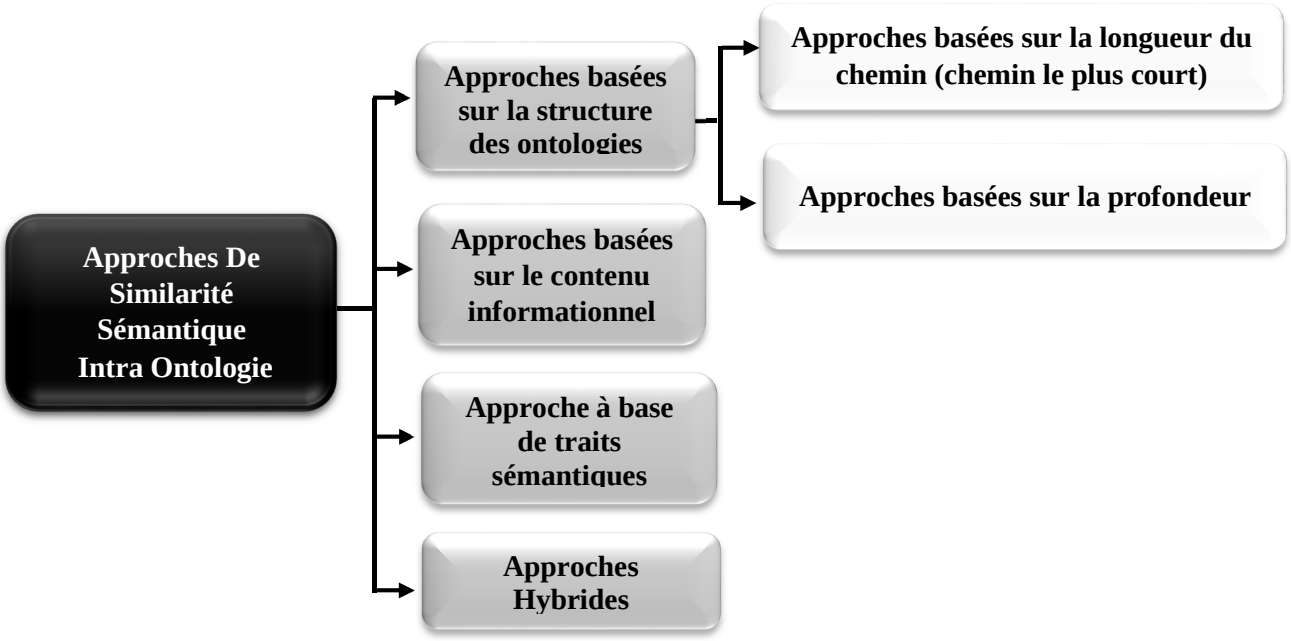


Figure 17. Classification des approches pour la similarité sémantique intra ontologies

3.1.1. Approches basées sur la structure des ontologies :

La plupart des mesures qui sont basées sur la structure de l'ontologie sont en fait basées sur : la longueur/distance du chemin (longueur du chemin le plus court) entre les deux nœuds de concept, et la profondeur des nœuds de concept dans l'ontologie/l'arbre hiérarchique.

3.1.1.1. Approches basées sur la longueur du chemin (chemin le plus court) :

La mesure de la similarité entre les concepts est basée sur la distance de chemin séparant les concepts. Ces mesures calculent la similarité en fonction du chemin le plus court entre les concepts cibles (groupe de synonymes) dans la taxonomie.

Rada et al. [RAD89] est adoptée dans un réseau sémantique et elle est fondée sur le fait qu'on peut calculer la similarité en se basant sur les liens hiérarchiques «is-a». Dans cette mesure, la distance sémantique est calculée en comptant le nombre d'arcs minimal

qui séparent deux concepts. La définition de la similarité donnée par ces auteurs pour cette approche est ainsi établie : « soient C_1 et C_2 deux concepts représentés par les nœuds c_1 et c_2 dans une hiérarchie « is-a », la distance conceptuelle entre C_1 et C_2 : *dist* (c_1, c_2), est représentée par le minimum du nombre d'arcs séparant les nœuds c_1 et c_2 ». Formellement, cela voudrait dire que la distance entre les deux concepts est représentée par la longueur du plus court chemin séparant les nœuds qui les représentent dans la hiérarchie. L'expression suivante définit la mesure de Rada :

$$Sim_{Ra}(C_1, C_2) = \frac{1}{1+dist(C_1,C_2)} \qquad (1)$$

Sachant que, *dist* (c_1, c_2) correspond au nombre d'arcs qui doivent être traversés dans l'ontologie pour connecter les concepts C_1 et C_2.

HSO [HIR98] calcule la similarité entre les concepts en utilisant la distance du chemin entre les nœuds des concepts, le nombre de changements de direction du chemin reliant deux concepts et le caractère acceptable du chemin. S'il existe une relation étroite entre les significations de deux concepts ou mots, alors on dit que les concepts sont sémantiquement liés les uns aux autres [CHO12]. Un chemin admissible est un chemin qui ne s'éloigne pas de la signification du concept source et qui doit donc être pris en compte dans le calcul de la similarité. Soit, d le nombre de changements de direction dans le chemin qui relie deux concepts C_1 et C_2, et C, k sont des constantes dont les valeurs sont dérivées par des expériences, *SPD* (*Shortest Path Distance*) est la distance du plus court chemin en nombre d'arcs. La fonction de similarité de HSO est formulée comme suit :

$$Sim_{HSO}(C_1, C_2) = C - SPD - k * d \quad (2)$$

Zhong [ZHO02] évalue l'absence de ressemblance entre les concepts. La méthode proposée est basée sur la distance entre les concepts. La mesure de similarité s'exprime ainsi :

$$Sim_{\text{Zhong}}(C_i, C_j) = 1 - dist(C_i, C_j) \qquad (3)$$

La distance est donnée par l'expression suivante :

$$dist(C_i, C_j) = \frac{1}{2^{P_{pppc}}} - \frac{1}{2^{P_{i+1}}} - \frac{1}{2^{P_{j+1}}} \qquad (4)$$

Où *Cpppc* désigne le concept communément appelé le plus petit parent commun de *Ci* et *Cj* . *Ppppc*, *Pi* et *Pj* représentent respectives les profondeurs de *Cpppc*, *Ci* et *Cj*.

3.1.1.2. Approches basées sur la profondeur :

Les approches basées sur la profondeur sont fondamentalement les chemins les plus courts, mais elles considèrent la profondeur des nœuds qui relient les deux concepts dans la structure globale de l'ontologie pour quantifier la similarité. Elle calcule la profondeur depuis la racine de la taxonomie jusqu'au concept cible.

Wu et Palmer [WUP94] proposent une mesure de similarité définie comme suit : (voir figure 18)

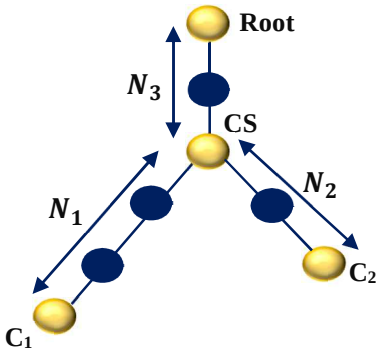


Figure 18. Calcul de similarité en employant l'approche de Wu et Palmer [WUP94]

Où C_1 et C_2 sont deux concepts de l'ontologie pour lesquels ils ont calculé la similarité. Les distances (N_1 et N_2) entre les nœuds C_1 et C_2 et le concept subsumant (CS) de C_1 et de C_2 au nœud Racine (Root), sont utilisées pour calculer la similarité.

La mesure de Wu et Palmer est définie par la formule suivante (voir formule 5) :

$$Sim_{wp}(C_1, C_2) = \frac{2*N_3}{N_1+N_2+2*N_3} \qquad (5)$$

Cette mesure a été utilisée par plusieurs chercheurs parmi eux [HAL03] il a utilisé pour organiser des documents web dans des clusters. Elle a aussi été exploitée dans [DES01]

pour évaluer la proximité sémantique de deux concepts d'une page HTML relativement à un thésaurus dans le cadre d'une indexation d'un site web par des ontologies [DEN13]. [LIN98] a effectué une comparaison entre les méthodes de mesure de similarité, il en ressort que la mesure de [WUP94] a l'avantage d'être simple à calculer en plus des performances qu'elle présente, tout en restant aussi expressive que les autres.

Slimani et al. [SLI06] a proposé une extension de la mesure de Wu et Palmer avec la tentative d'améliorer les résultats du comptage d'arêtes, car calculer la similarité de deux termes dans une hiérarchie ne donne généralement pas de résultats satisfaisants, en particulier lorsque cette mesure offre une similarité plus élevée entre un concept et son voisinage par rapport à ce même concept et un concept contenu dans le même chemin. S'il existe un extrait de hiérarchie avec le format suivant :

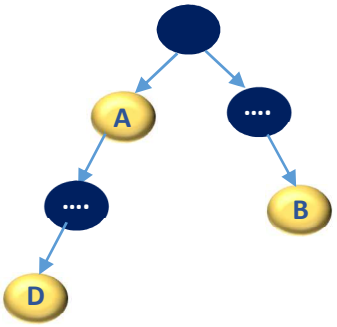


Figure 19. Exemple d'extrait de hiérarchie.

En appliquant la mesure de Wu et Palmer, il est possible d'obtenir Simwp (A, D) < Simwp (A, B), où D est un descendant de A et B un des descendants des frères de A. Cette situation est inadéquate dans le cadre de la recherche d'informations où il est nécessaire de retrouver tous les descendants d'un concept (i.e. requête) avant son voisinage. Pour faire face à ce problème, les travaux de [SLI06] proposent un facteur de pénalisation de deux concepts C_1 et C_2 placés dans le voisinage à multiplier par la mesure de Wu et Palmer. Cette fonction vise à pénaliser ou à réduire la valeur de la mesure de similarité lorsque deux concepts ne sont pas dans la même hiérarchie. La similarité entre les deux concepts C_1 *et* C_2 par cette mesure est donnée par la formule suivante :

$$Sim_{tbk}(C_1, C_2) = \frac{2*profondeur(C)}{profondeur(C1)+profondeur(C2)} * fp(C_1, C_2) \quad (6)$$

fp (C_1, C_2) est la fonction de pénalisation des concepts C_1 et C_2. L'utilité de cette fonction est de pénaliser la similarité de deux concepts éloignés qui ne sont pas situés dans une même hiérarchie.

$$\begin{cases} fp(C_1, C_2) = \frac{1}{|profondeur(C1)-profondeur(C2)|+1} \\ Si\ C1\ et\ C2\ sont\ \subset\ dans\ le\ même\ \ chemin \\ fp(C_1, C_2) = 1 \quad Autrement \end{cases} \quad (7)$$

Tel que Profondeur(C_1) est la distance en nombre d'arcs entre C et le concept subsumant commun (CS). Le rapport *fp* (C_1, C_2)=1 si C_1 est ancêtre de C_2 ou l'inverse. Avec cette formule, on va pénaliser uniquement les nœuds voisins de C_1 ou de C_2.

Leacock et Chodorow [LEA98] se basent sur la longueur *length* (C_1, C_2) du chemin le plus court entre deux concepts de *WordNet* pour mesurer la similarité. Ils ont utilisé une seule relation dans une hiérarchie « *is-a* » et ont modifié la formule de longueur du chemin pour refléter le fait que les arcs les plus bas dans la hiérarchie d'hyponymie correspondent à la plus petite distance sémantique et mettent à l'échelle la longueur du chemin par la profondeur globale D de l'ensemble des synonymes. La similarité entre les deux concepts C_1 *et* C_2 est :

$$Sim_{lch}(C_1, C_2) = -log \frac{length(C_1, C_2)}{2*D} \quad (8)$$

Où *length* (C_1, C_2) représente le plus court chemin entre deux concepts.

Zargayouna [ZAR04] propose une extension de la mesure Wu & Palmer en prenant en compte le concept le plus bas de la taxonomie qu'il nomme le bottom. Il ajoute à la mesure de Wu & Palmer une mesure de spécificité qui prend en considération le degré de spécificité du concept. En d'autres termes, c'est le nombre d'arcs qui le séparent de *bottom*.

Cette mesure s'exprime par la formule (9) :

$$\begin{cases} Sim(C_1, C_2) = \frac{2 * depth(C)}{depth_c(C1) + depth_c(C2) + spec(C_1, C_2)} \\ spec(C_1, C_2) = depth_b(C) * dist(C, C_1) * dist(C, C_2) \end{cases} \quad (9)$$

avec *depthb* (C) est le nombre d'arcs qui sépare C de bottom et (C,C_i) la distance en nombre d'arcs entre C et c_i et *spec*(C_1, C_2) une fonction qui calcule la spécificité de deux concepts par rapport au concept le plus bas de l'ontologie (bottom, concept virtuel qui symbolise la fin de l'ontologie) comme le montre la figure 20.

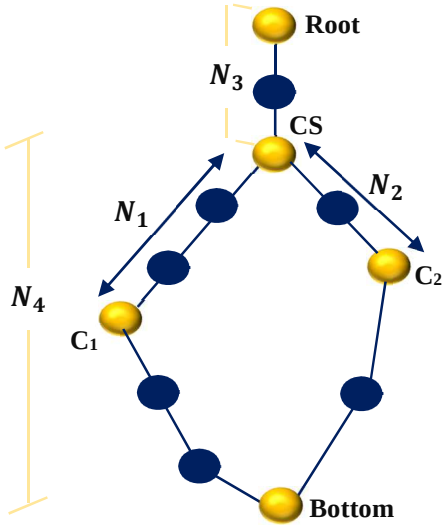


Figure 20. Les relations conceptuelles [ZAR04]

3.1.2. Approches basées sur le contenu informationnel (IC) :

La notion du contenu de l'information (ou contenu informatif) « IC » a été introduite pour la première fois par Resnik [RES99] qui a suggéré que le contenu informatif d'un concept traduit la pertinence de celui-ci dans un corpus en tenant compte de la fréquence d'apparition des mots auxquels il se réfère ainsi que la fréquence d'apparition des concepts qu'il généralise. Plus précisément, le contenu informatif se calcul par la formule suivante (voir formule 10) :

$$IC(c) = -\log(p(c)) \quad \textbf{(10)}$$

Où $p(c)$ est la probabilité de retrouver qu'un mot du corpus soit une instance du concept c (un des mots référés par le concept c ou par un de ses descendants).

Plusieurs auteurs ont mis en place des approches basées sur le contenu informatif des concepts dans une hiérarchie afin d'évaluer la similarité entre eux. L'intuition derrière l'utilisation de la notion du contenu informatif est que la similarité entre deux concepts est la portion d'information qu'ils ont en commun qui, dans le cadre d'une ontologie, peut être déterminée par le plus petit généralisant commun qui les subsume (LCS). Cette intuition est indirectement appliquée par les mesures présentées dans la section précédente qui calculent la similarité avec le nombre d'arcs qui séparent deux concepts.

Resnik [RES99] propose une mesure qui utilise le contenu informatif des parents partagés. Le principe de cette mesure est le suivant : deux concepts sont plus similaires s'ils présentent une information plus partagée, et l'information partagée par deux concepts C_1 et C_2 est évaluée numériquement par le contenu informatif du plus petit généralisant commun aux deux concepts comparés (LCS) comme suit :

$$Sim_{Resnik}(C_1, C_2) = IC(LCS(C_1, C_2)) \qquad (11)$$

Ainsi, si deux concepts sont très éloignés et ont comme racine le LCS, leur similarité est nulle.

Jiang & Conrath [JIA97] proposent une distance qui est non seulement dépendante du chemin parcouru, mais aussi, de la quantité d'informations véhiculées par les nœuds traversés par ce chemin. Le calcul de la longueur du chemin entre les deux concepts s'appuie sur la recherche du plus petit généralisant commun subsumant de deux concepts LCS (c_1, c_2). Le plus court chemin entre c_1, c_2 est sp (c_1, c_2) l'unique chemin passant par LCS (c_1, c_2). Ensuite, pour chaque arête de la hiérarchie reliant deux concepts C_1 et C_2, les auteurs proposent de définir un poids TC (C_1, C_2) $\in$ [0,1]. La distance de Jiang et Conrath [JIA97] a été alors définie comme suit.

$$dist(c_1, c_2) = \sum_{c \epsilon SP(C_1,C_2) \setminus LCS(C_1,C_2)} [IC(c) - IC(parent(c)] * TC(c, parent(c)) \qquad (12)$$

Selon [BUD06], cette mesure de distance est probablement la plus utilisée actuellement et la plus efficace pour déterminer la proximité sémantique entre deux concepts.

Comme la plupart des mesures de similarités elle se limite à la relation de subsomption, mais elle offre la possibilité de prendre en compte des valeurs différentes pour chaque arête de la hiérarchie des concepts, les TC(c, parent(c)). Une fonctionnalité qui a rarement été prise en compte. En effet, les auteurs ont effectué leur évaluation avec un poids d'arête constant (TC=1). Depuis, les systèmes utilisant la formule de Jiang et Conrath se servent de la version simplifiée sans poids sur les arêtes formulées pas les auteurs comme suit :

$$dist_{JC_{simple}}(c_1, c_2) = \big(IC(c_1) * IC(c_2)\big) - 2 * IC(LCS(c_1, c_2) \qquad (13)$$

La similarité entre les deux concepts est alors calculée par l'inverse de la distance qui les sépare (voir formule 14)

$$Sim_{JC}(c_1, c_2) = \frac{1}{dist_{JC_{simple}}(c_1, c_2)} \qquad (14)$$

Lin et al. [LIN93] les auteurs de cette approche proposent d'évaluer la similarité entre deux concepts en tenant compte à la fois de leur contenu d'information (IC), et le contenu d'information de leur concept commun le plus spécifique comme suit :

$$Sim_{lin}(C_1, C_2) = \frac{2 * IC(LCS(C_1, C_2))}{IC(C_1) + IC(C_2)} \qquad (15)$$

Ceci peut être vu comme le contenu d'information de l'intersection des deux concepts (multipliés par deux) qui est divisé sur leur somme.

Les valeurs retournées par cette mesure varient entre 1 (concepts complètement similaires) et 0. Dans ce cas, la mesure d'un concept comparé à lui-même aura la valeur 1. Cette mesure semble combiner les propriétés des mesures présentées ci-dessus, c'est-à-dire elle offre à la fois des informations sur la taille de l'ontologie et le classement des paires de termes différents.

3.1.3. Approche à base de traits sémantiques :

L'étude des traits d'un terme est très importante, car elle contient des informations précieuses concernant la connaissance du terme. La mesure basée sur les traits suppose que chaque terme est décrit par un ensemble de termes indiquant ses propriétés ou ses

traits. La mesure de similarité entre deux termes est définie comme une fonction de leurs propriétés (par exemple, leurs définitions ou "glosses" dans WordNet) ou basée sur leurs relations avec d'autres termes similaires dans une structure hiérarchique.

Tversky [TVE77] prend en compte le nombre de traits communs et les différences entre les deux concepts à comparer. En effet, cette approche est basée sur la description des concepts. Selon l'auteur, la similarité entre deux concepts est évaluée par le nombre pondéré de traits en commun auquel est retiré le nombre de traits spécifiques à chacun des concepts. Ce qui voudrait dire que, plus les concepts ont des traits communs et moins de non-communs, plus ils sont similaires. La mesure a été alors exprimée par la formule suivante :

$$Sim_{Tversky}(c_1, c_2) = \alpha.comm(c_1, c_2) - \beta.diff(c_1, c_2) - \gamma.diff(c_1, c_2) \quad (16)$$

Où : α, β, γ sont des constantes qui mènent à différents types des similarités.

Dans cette mesure, si $\beta = \gamma = 0$ et $\alpha = 1$, la similarité entre c_1 et c_2 correspond à la quantité de propriétés en commun. Si $\alpha = 0$, $\beta > 0$ et $\gamma > 0$, les concepts c_1 et c_2 sont évalués selon ce qui les différencie ce qui en fait une mesure de dissimilarité non de similarité.

Cette mesure est donc dépendante du cardinal des propriétés de chaque concept. Une seconde version de la formule précédente a été donnée par son auteur dans [TVE77] comme suit :

$$Sim_{tversky}(c_1, c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \beta|C_1 \setminus C_2| + \gamma|C_2 \setminus C_1|} \quad (17)$$

Où C_1 et C_2 sont les ensembles de descriptions (propriétés) de c_1 et c_2 respectivement. β, γ définissent l'importance relative des propriétés non communes.

3.1.4. Approches Hybrides :

Les mesures hybrides combinent les caractéristiques structurelles décrites ci-dessus (telles que la longueur des chemins, la profondeur et la densité locale)

Li et al [**LI 03**] a proposé d'incorporer le vecteur sémantique et l'ordre des mots pour calculer la similarité des phrases. Cette mesure de similarité combine la longueur du plus court chemin (SP) entre deux concepts C_1 et C_2, et la profondeur dans la taxonomie (N) du concept commun le plus spécifique C, dans une fonction non-linéaire.

$$Sim_{li}(C_1, C_2) = e^{-\propto * SP} \frac{e^{\beta * N} - e^{-\beta * N}}{e^{\beta * N} + e^{-\beta * N}} \qquad (18)$$

Où $\alpha \geq 0$ et $\beta \geq 0$ sont des paramètres mesurant la contribution de la longueur du plus court chemin et de la profondeur respectivement. Les paramètres optimaux sont α= 0,2 et $\beta = 0{,}6$ sur la base de [LI 03]. Il est donc évident que cette mesure obtient un score compris entre 1 (pour les concepts similaires) et 0.

Knappe [**KNA03**] définit une mesure de similarité utilisant les informations de généralisation et de spécification de deux concepts comparés. Cette mesure est principalement basée sur l'aspect qu'il peut y avoir plusieurs chemins reliant deux concepts. L'expression de la mesure proposée est formulée comme suit :

$$Sim_{Knappe}(C_1, C_2) = p * \frac{|Ans(C_1) \cap Ans(C_2)|}{|Ans(C_1)|} + (1 - p) * \frac{|Ans(C_1) \cap Ans(C_2)|}{|Ans(C_2)|} \qquad (19)$$

Où p est une valeur dans [0,1] qui détermine le degré d'influence de la généralisation. $Ans(C_1)$ et $Ans(C_2)$ correspondent aux ensembles de description (les nœuds ancêtres) des termes C_1 et C_2 respectivement. Les nœuds atteignables partagés par C_1 et C_2 sont $Ans(C_1) \cap Ans(C_2)$.

Zhou et al. [**ZHO08**] ont proposé une mesure prenant en compte les mesures basées sur le contenu de l'information et les mesures basées sur les chemins comme paramètre. La mesure proposée est exprimée par la formule suivante :

$$Sim_{zhou}(C_1, C_2) = 1 - k\left(\frac{\ln(len(C_1, C_2) + 1}{\ln(2 * (deep_{max} - 1))}\right) - (1 - k) * ((IC(C1) + IC(C2) - 2 * \frac{IC(lso(C1, C2))}{2}) \qquad (20)$$

Où *lso* (C_1, C_2) est la plus petite super ordonnée de C_1 et C_2. D'après la formule précédente, on remarque que l'IC et le chemin ont été pris en compte pour le calcul de

la mesure de similarité. Le paramètre k doit être adapté manuellement pour obtenir de bonnes performances. Si k=1, la formule 16 est basée sur le chemin ; si k=0, la formule 16 est une mesure basée sur l'IC.

3.2. Les approches pour la similarité sémantique inter-ontologies :

Ce sont des approches qui comparent les termes de différentes ontologies. Dans ce cas, les concepts pour lesquels la similarité doit être évaluée appartiennent à deux ontologies différentes. L'ontologie secondaire est connectée à l'ontologie primaire par les nœuds communs. Deux nœuds dans deux ontologies sont équivalents s'ils font référence au même concept. Ces approches peuvent être classées comme suit : (voir figure 21)

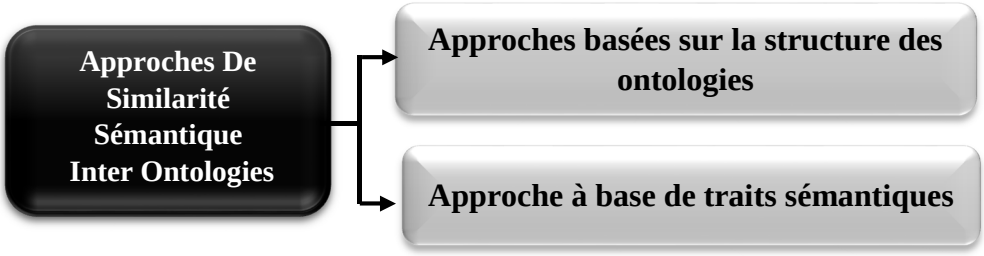


Figure 21. Classification des approches pour la similarité sémantique inter ontologies

3.2.1. Approches basées sur la structure inter ontologies :

Al-Mubaid and Nguyen[NGU06] ont proposé une mesure basée sur le chemin utilisant la structure d'ontologie pour mesurer la similarité des concepts entre plusieurs ontologies dans le domaine biomédicaux MeSH (Medical Subject Headings) et SNOMED-CT (Systemized Nomenclature of Medicine Clinical Term) dans le cadre du Système de langage médical unifié (UMLS). La similarité sémantique entre termes inter-ontologies est mesurée en considérant une ontologie comme primaire et une autre comme ontologie secondaire. Pour mesurer la similarité inter-ontologies des concepts, la granularité des ontologies est prise en considération. Cette méthode prend en considération la spécificité du concept, la densité locale (force et poids du lien) du concept et la spécificité locale des concepts dans une ontologie.

Cette méthode a pour but de trouver la similarité entre les concepts de deux ontologies qui ont des nœuds de concepts communs. La mesure de similarité est généralement quantifiée sur la base des caractéristiques communes partagées par les concepts.

Cette caractéristique commune est quantifiée par le facteur de spécificité commun qui est donné par :

$$CSpec(C_1, C_2) = D - Depth(LCS(C_1, C_2)) \qquad (21).$$

Où : D est la profondeur de l'ontologie. $Depth\ (LCS\ (c_1,\ c_2))$ est la profondeur du concept commun le plus spécifique entre c_1 et c_2. Plus la valeur *CSpec* est faible, plus l'information partagée entre les deux concepts est importante.

Al-Mubaid et Nguyen [NGU06] classent les ontologies en ontologies primaire et secondaire. L'ontologie avec la plus grande densité de concepts est appelée ontologie primaire et l'ontologie avec des concepts moins denses est appelée ontologie secondaire. Les auteurs ont alors soulevé quatre cas possibles comme suit : les concepts comparés appartiennent tous à l'ontologie primaire (cas 1), l'un des concepts appartient à l'ontologie primaire et l'autre à la secondaire (cas2), deux concepts comparés appartiennent à l'ontologie secondaire (cas 3), les concepts appartiennent à de multiples ontologies secondaires (cas 4). Alors d'après les auteurs, la similarité sémantique entre deux concepts est estimée pour les quatre cas en suivant l'un des processus suivants :

Cas 1 : Al-Mubaid définit la similarité entre les concepts appartenant à une même ontologie comme une distance sémantique entre les concepts, elle est donnée par la formule suivante :

$$SemDist(C_1, C_2) = \log((path - 1)^{\alpha} * (CSpes)^{\beta} + k \quad (22).$$

Où $\alpha > 0$ et $\beta > 0$ et ils sont les facteurs de contribution de *path* et de *CSpec*, k est une constante, *path* est la longueur du chemin le plus court entre les concepts et *CSpec* est calculé à l'aide de l'équation (21).

Cas 2 : similarité inter-ontologies

Dans ce cas, les concepts pour lesquels la similarité doit être évaluée appartiennent à deux ontologies différentes. L'ontologie secondaire est connectée à l'ontologie primaire

par l'intermédiaire des nœuds communs. Deux nœuds dans deux ontologies sont équivalents s'ils font référence au même concept. Ces nœuds communs sont fusionnés et sont appelés nœuds de pont (bridge) voir figure 22.

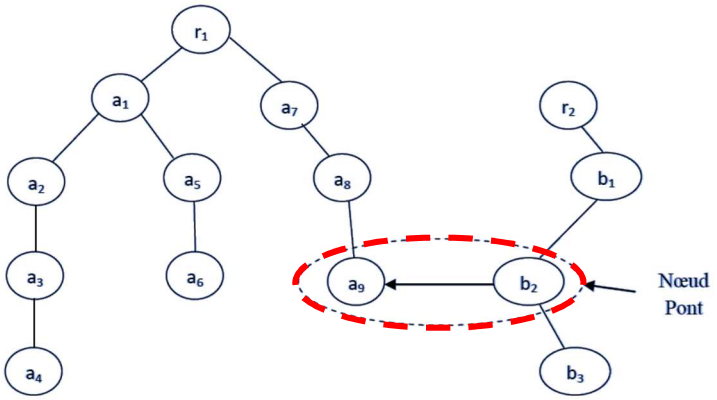


Figure 22. Deux fragments d'ontologies connectés par un nœud pont [NGU06]

Le concept de subsomption le moins commun est calculé comme indiqué ci-dessous.

$$LCS(C_1, C_2) = LCS(C_1, bridge_n) \qquad (23)$$

Où C_1 appartient à l'ontologie primaire et C_2 appartient à l'ontologie secondaire. La longueur du chemin de deux concepts appartenant à deux ontologies différentes est calculée en additionnant les longueurs de chemin de chacun des nœuds de concept jusqu'au nœud de pont est donné par :

$$Path(C_1, C_2) = d_1 + d_2 - 1 \qquad (24)$$

Tel que d_1 est le chemin le plus court entre le concept C_1 et le nœud de pont, d_2 est le chemin le plus court entre C_2 et le nœud de pont. Ensuite, la spécificité commune et la longueur du chemin de l'ontologie secondaire sont mises à l'échelle de l'ontologie primaire en définissant le rapport de granularité de la caractéristique de spécificité commune ($CSpecRate$) et le rapport de granularité de la caractéristique de longueur du chemin ($PathRate$). Le $CSpecRate$ et le $PathRate$ sont calculés comme indiqué par les équations (25) et (26).

$$CSpecRate = \frac{(D_1 - 1)}{(D_2 - 1)} \qquad (25)$$

$$PathRate = \frac{(2D_1 - 1)}{(2D_2 - 1)} \qquad (26)$$

Où D_1 et D_2 sont les profondeurs de l'ontologie primaire et de l'ontologie secondaire. Afin d'éviter la détérioration des valeurs de similarité dans l'ontologie primaire, la longueur du chemin d_2 de l'ontologie secondaire est mise à l'échelle de la profondeur de l'ontologie primaire et la nouvelle longueur du chemin du second concept est indiquée comme suit.

$$d_2' = PathRate * d_2 \qquad (27)$$

En mettant à l'échelle les valeurs de l'ontologie secondaire par rapport à l'ontologie primaire, la distance sémantique entre les concepts est donnée par la formule suivante :

$$SemDist(C_1, C_2) = \log((path - 1)^{\alpha} * (CSpec_i)^{\beta} + k \quad (28)$$

Où $path_i$ dénote des chemins multiples et est la longueur du chemin entre deux concepts calculée par le pont. Le $CSpec_i$ est la spécificité commune associée au $i^{ème}$ chemin. S'il existe plusieurs chemins reliant les concepts, la distance minimale entre les chemins possibles est mesurée comme la distance sémantique [NGU06].

Cas 3 : Dans ce cas, les deux concepts comparés appartiennent à l'ontologie secondaire. Par conséquent, selon les auteurs, la distance sémantique entre ces concepts peut être calculée seulement quand le chemin les reliant $Path\ (C_1, C_2)$ et leur spécificité commune $CSpec(C_1, C_2)$ sont traduits à l'échelle de l'ontologie primaire. Ses deux caractéristiques sont alors données par les formules suivantes :

$$Path(C_1, C_2) = Path(C_1, C_2)_{secondaire} * PathRate \qquad (29)$$

$$CSpec(C_1, C_2) = CSpec(C_1, C_2)_{secondaire} * CSpecRate \quad (30)$$

Où : $Path\ (C_1,\ C_2)_{secondaire}$ est le plus court chemin entre C_1 *et* C_2 dans l'ontologie secondaire. $CSpec\ (C_1,\ C_2)_{secondaire}$ est calculé en se basant sur l'ontologie secondaire en utilisant l'équation ($CSpec(C_1,\ C_2) = D\text{-}Depth(LCS(C_1,\ C_2))$), La distance sémantique entre les concepts dans l'ontologie secondaire est alors calculée par l'équation (28).

Cas 4 : les concepts appartiennent à deux ontologies secondaires différentes, parmi lesquelles une agit temporairement comme une ontologie primaire, et l'autre agit comme une ontologie secondaire. Pour cela, l'ontologie secondaire qui a le plus grand

nombre de concepts est choisie comme ontologie primaire. La mesure de similarité sémantique est alors calculée en utilisant la longueur du chemin entre les deux concepts dans l'ontologie secondaire et leur spécificité commune *CSpec* en les ramenant à l'échelle de l'ontologie primaire suivant les formules données dans le cas 3.

3.2.2. Approche à base de traits sémantiques :

Rodriguez et al. [ROD03] ont proposé une mesure qui peut être utilisée pour des similarités intra -ontologie ou inter-ontologies. Selon Rodriguez et Egenhofer, un concept est considéré comme une classe d'entité. La recherche de la similarité entre les ensembles de synonymes des classes d'entités, la similarité entre les traits distinctifs des classes d'entités et la similarité entre les voisinages sémantiques des classes d'entités sont utilisées pour identifier la similarité entre les classes d'entités. L'agrégation pondérée de la similarité entre les trois composantes spécifiées (ensemble de synonymes, traits et voisinages) constitue la fonction de similarité entre les classes d'entités. Par conséquent, la similarité entre les classes d'entités de l'ontologie p et de l'ontologie q, est donnée comme suit :

$$Sim_{Rod}(C1^p, C2^q) = W_w S_w(C1^p, C2^q) + W_u S_u(C1^p, C2^q) + W_n S_n(C1^p, C2^q) \quad (31)$$

Où S_w, S_u et S_n sont respectivement la mesure de la similarité entre les ensembles de synonymes, les traits et les voisinages sémantiques parmi les classes C_1 de l'ontologie p et les classes C_2 de l'ontologie q sont calculée à l'aide de l'équation (16) (basée sur le modèle de correspondance des caractéristiques de Tversky). W_w, W_u et W_n sont les poids respectifs de la similarité de chaque composant de spécification. W_w, W_u et W_n doivent être ≥0 et la somme de W_w, W_u et W_n doivent être égale à 1.

La fonction de similarité S_w est une fonction de correspondance des mots utilisée pour déterminer le nombre de mots communs et de mots différents dans les ensembles de synonymes. La fonction de similarité S_u est une fonction de correspondance des caractéristiques utilisée pour trouver la similarité entre les traits distinctifs de la classe

d'entités. La fonction de similarité S_n est une fonction de similarité qui mesure la similarité entre les voisinages sémantiques.

Petrakis et al. [PET06] ont proposé une nouvelle méthode de similarité d'ontologies multiples développée basée sur les traits appelés X-similarité qui repose sur la correspondance entre les concepts et les ensembles de description des termes.

[ROD03] ont utilisé des paramètres pour calculer la profondeur du concept dans les deux ontologies, alors que selon [PET06] la mise en correspondance d'ontologies multiples ne devrait pas dépendre des informations sur la structure de l'ontologie. Deux termes sont similaires si les concepts des mots et les concepts de leurs voisinages (basés sur les relations sémantiques) sont lexicalement similaires. Soit A et B deux concepts ou ensemble de description de termes. Puisque tous les termes dans le voisinage d'un terme ne présentent pas une connexion avec la même relation, les similarités d'ensemble sont calculées par type de relation sémantique (SR) (par exemple, Is-A et Part-Of). La mesure de similarité proposée est exprimée comme suit :

$$Sim_{xsim}(A,B) = \begin{cases} 1 \quad if\ S_{synset}(A,B) > 0 \\ \max\{S_{neighb}(A,B), S_{descr}(A,B)\}, if\ S_{synset}(A,B) = 0 \end{cases} \qquad (32)$$

Soit i un type de relation, la similarité pour les voisins sémantiques S_{neighb} est formulée comme suit : $$S_{neighb}(A,B) = max_{i \subset SR} \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \qquad (33)$$

De la même manière, si A et B désignent l'ensemble des synonymes ou des descriptions pour le terme a et b, la similarité des descriptions S_{descr} et des synonymes $S_{synsets}$ est calculée comme suit : $$S_{\text{descr}}(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (34)$$

Batet [SAN13] proposent une méthode permettant d'estimer la similarité entre plusieurs ontologies. Ils identifient différents cas selon l'appartenance des termes d'entrée aux ontologies, il propose plusieurs heuristiques pour traiter chaque cas, visant à résoudre les valeurs manquantes, lorsque des connaissances partielles sont disponibles, et à capturer la preuve sémantique la plus forte qui aboutit à l'évaluation de similarité

la plus précise, lorsqu'il s'agit de connaissances qui se chevauchent. Cette méthode fonctionne dans trois cas qui sont :

1 - Si le concept de la paire apparaît dans une ontologie unique, la similarité est calculée comme une mono ontologie

2- Ces deux concepts apparaissent dans plusieurs ontologies, chacune d'entre elles modélisant les connaissances dans un domaine différent mais qui se chevauche.

3- Deux concepts appartiennent à des ontologies différentes, chacune modélisant la connaissance d'un point de vue différent.

Sur la base de toutes les situations ci-dessus, la dernière situation est la même que celle des ontologies multiples.

Dans cette situation, le chercheur n'utilise pas la correspondance terminologique pour comparer le concept, mais il utilise le subsumer du concept pour obtenir la similarité du concept comparé. Subsumer signifie ici l'élément de l'hyperonyme, du super-concept ou un élément de l'ancêtre taxonomique du concept. Cette méthode utilise une mesure de similarité sémantique comme l'équation ci-dessous :

$$Sim(C_1, C_2) = -\log 2 \frac{|T(C_1) \cup T(C_2)| - |T(C_1) \cap T(C_2)|}{|T(C_1) \cup T(C_2)|} \quad (35)$$

Où T (C_i) est défini comme l'ensemble des super-concepts du concept I incluant le concept lui-même.

4. Étude comparative entre les approches de similarité sémantiques en fonction de l'ontologie :

Le tableau suivant représente une comparaison entre différentes mesures pour les deux classes d'ontologies intra-ontologie et inter-ontologies, cette comparaison permet de démontrer les avantages et les inconvénients de chacune des mesures, permettant ainsi au chercheur de mieux choisir la mesure adéquate à utiliser selon le problème traité.

Similarité Sémantique Intra Ontologie	Méthodes		L'année	Les avantages	Les inconvénients
Approches basées sur la structure des ontologies	Approches basées sur la longueur du chemin	**Rada**	1989	Simple et facile à implémenter.	Ne prend pas en compte la profondeur des concepts
		Hirst & ST Onge	1998	Permet d'évaluer la similarité entre nom et verbe sur WordNet.	Limitée par des restrictions sur le nombre de chemins.
		Zhong	2002	Donne une meilleure similarité entre "père" et "fils" qu'entre deux "frère" dans une taxonomie.	Pas de garantie de l'unicité du plus petit parent commun
	Approches basées sur la profondeur	**Wu & Palmer**	1994	Simple et facile à implémenter ; prend en compte la profondeur des concepts.	Ne donne pas une bonne similarité entre concepts voisins et concepts de la même hiérarchie.
		Leacock & Chodorow	1998	Simple à implémenter.	Ne prend en compte que la relation is-a ; moins performante que Wu & Palmer sur WordNet.
		Zargayouna	2004	Simple et facile à implémenter ; prend en compte la profondeur des concepts et la similarité entre concepts voisins et concepts de la même hiérarchie	Elle suppose disposer d'une ontologie de concepts reliée au corpus
		Slimani	2006	Les mêmes avantages que celle de Zargayouna	Trop dépendante à l'organisation des concepts dans la taxonomie

Similarité Sémantique Inter Ontologie	Méthodes	L'année	Les avantages	Les inconvénients
Approches basées sur la structure des ontologies	**Al-Mubaid et Nguyen**	2006	dépend du modèle de graphe qui nécessite un faible niveau de calcul. Simple et facile à implémenter	-la structure de chaque ontologie est différente et ne peut pas être comparée directement -le nombre et la répartition des ancêtres taxonomiques communs et non communs ne sont pas pris en compte.
Approche à base de traits sémantiques	**Rodriguez et Egenhofer**	2003	Prendre en compte les voisinages sémantiques dans le calcul de la similarité.	-Une partie incomplète pour le calcul sera la cause d'une une faible précision. -Le paramètre γ prend en compte la profondeur du concept dans les deux ontologies.
	X-similarity	2006	- Ne dépend pas du paramètre de poids. -La similarité maximale fournie par chaque trait est prise en compte.	-La contribution des autres traits est omise si -seule la valeur maximale est prise à chaque instant.
	Batet et al.	2013	-Utiliser un subsumer du concept. -Sans se reposer sur le concept comparé	-Toujours en s'appuyant sur les stratégies de correspondance terminologique -Se concentrer uniquement sur l'ancêtre taxonomique pour obtenir le LCS

Table 3 : Récapitulation de l'étude chronologique pour les mesures de Similarité Sémantique Intra-ontologie et Inter-Ontologies

5. Conclusion :

Ce chapitre a abordé les bases de la mesure de similarité sémantique, la classification des mesures de similarité intra-ontologie et des mesures de similarité inter-ontologies. Une brève introduction des différentes mesures de similarité sémantique par les chercheurs est présentée, qui a montré que les approches basées sur la structure de l'ontologie, où la similarité est déterminée soit par le calcul du plus court chemin tandis que le degré de similarité est déterminé en fonction de la longueur du chemin, soit par la profondeur des bords reliant deux concepts dans l'ontologie de structure, ont l'inconvénient de ne pas prendre en compte l'aspect sémantique . Les approches basées sur le contenu de l'information sont dépendantes du corpus et peu de chercheurs ont tenté de calculer le contenu informationnel indépendamment du corpus en considérant les relations taxonomiques de l'ontologie. Les approches basées sur les traits ont le potentiel d'augmenter l'efficacité et la précision de la similarité sans utiliser les informations de structure. Les approches hybrides nécessitent de multiples sources d'information pour quantifier la similarité et prennent donc beaucoup de temps. Il y a peu de tentatives faites pour mesurer la similarité inter-ontologies. La quantification de la similarité est soit basée sur les chemins, soit sur les traits. Les avantages et les inconvénients de chaque approche sont également décrits, ce qui peut aider à l'évaluation de la sélection des meilleures approches que ce soit pour le cas intra-ontologie ou inter-ontologies. Le chapitre suivant aborde en détail les approches basées sur la structure de l'ontologie, sur lesquelles cette recherche a défini une nouvelle façon de calculer ces approches par une hybridation avec WordNet qui représente une ressource la plus utilisée pour le calcul de la similarité sémantique entre termes.

PARTIE II

Contributions De La Thèse

Chapitre 4

Nouvelle Approche Pour La Similarité Sémantique Inter Ontologies

1. Introduction :

Dans ce chapitre, nous proposons notre approche hybride pour calculer la similarité sémantique entre les concepts de différentes ontologies écrites en OWL, mais du même domaine. Nous hybridons quelques mesures à base de structure d'ontologie avec WordNet, cette combinaison est nécessaire pour intégrer et renforcer le facteur sémantique. Ces mesures vont être appliquées à trois langues (Anglais, Français et arabe). Nous présentons notre architecture globale de notre approche proposée et nous détaillons les différents algorithmes utilisés.

2. Architecture globale de notre système :

L'architecture globale de notre approche proposée est illustrée dans la figure 23 :

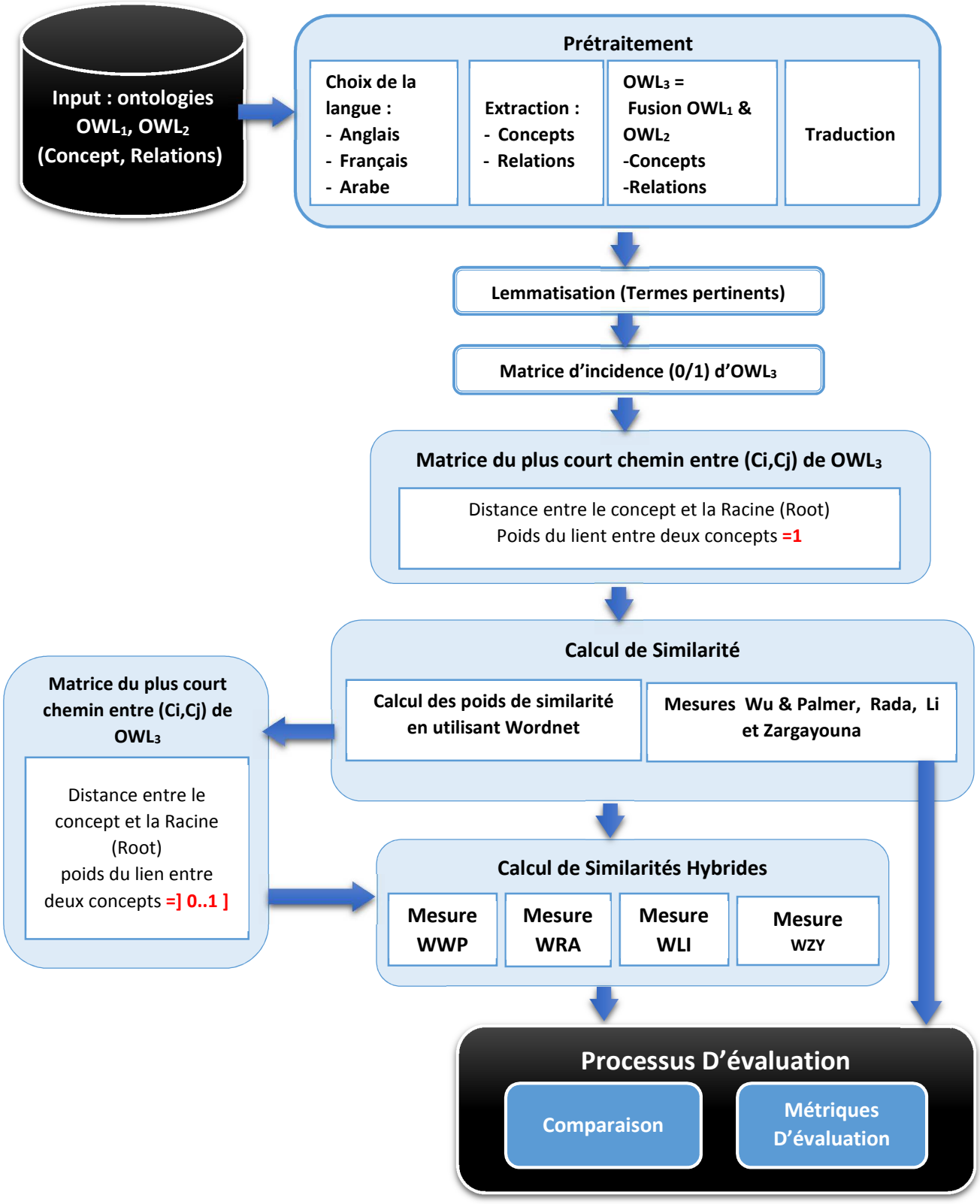


Figure 23. L'approche proposée

3. Description de notre approche :

Notre approche est composée de quatre phases, qui sont :

- Le prétraitement.
- Calcul de la mesure de similarité de Wu & Palmer, Li, Rada et Zargayouna.
- Calcul de la mesure de similarité hybride de WWP, WLI, WRA et WZY.
- L'évaluation de notre approche en utilisant la méthode de cohésion et de densité.

Chaque étape est composée de plusieurs phases, qui sont expliquées dans les sections suivantes.

3.1. Entrée owl1 &owl2 (dataset Benchmark):

Pour le jeu de données en différentes langues, nous avons utilisé une collection d'ontologies décrivant le domaine de l'organisation de conférences, trouvée dans The OAEI [CHE14] organise des campagnes d'évaluation visant à évaluer les techniques de correspondance d'ontologies. Ces ontologies décrivent le domaine des conférences.

Nous avons utilisé aussi le jeu de données ADOM(Arabic Dataset for Evaluating Arabic and Cross-lingual Ontology Alignement Systems) [KHI15] qui contient les ontologies de la piste des conférences de l'OAEI [CHE14] écrites en langue arabe. ADOM a été intégré dans le jeu de données multilingues de la piste multifarm de l'OAEI 2015. Nous justifions le choix de cet ensemble de données par les points suivants :

(1) La campagne d'évaluation la plus connue pour tester les performances des systèmes de correspondance d'ontologies est celle de l'OAEI.

(2) Dans notre approche, nous devons appliquer des critères de similarité à des ontologies différentes mais issues du même domaine, ce qui est possible avec ce jeu de données.

(3) Ce jeu de données propose cinq différentes ontologies en plusieurs langues.

3.2. Prétraitement : Premièrement, afin de réaliser notre implémentation, nous chargeons deux ontologies à partir d'un fichier de données ; ce fichier contient les mêmes ontologies, mais

écrites en plusieurs langues, sachant que lorsque en veut consulter ces ontologies en utilisant l'éditeur protégé, tous les concepts sont apparus sous forme de phrase sauf pour la langue arabe où il est sous forme de code d'où la nécessiter de renommer chaque code par son concept. La figure 24 montre le fichier d'origine (avant) et le fichier renommé en utilisant protégé (après)

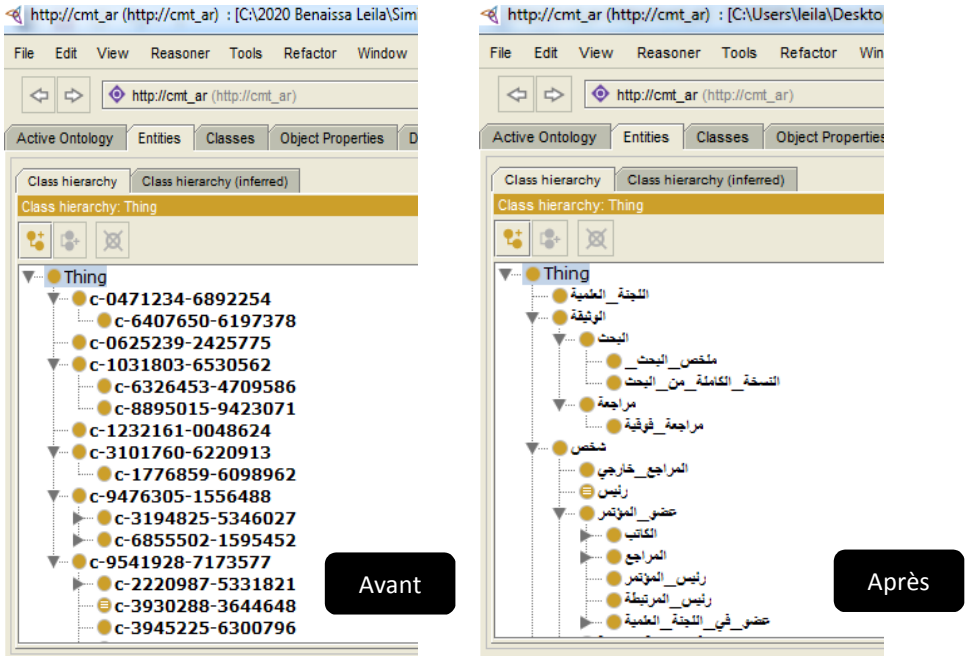


Figure 24. Ontologie cmt-arabe avant et après décodage

- Après que le système extrait tous les constructeurs du fichier d'ontologie OWL (concepts et relations entre chaque concept dans la même ontologie), nous fusionnons deux ontologies sélectionnées à l'étape précédente afin de trouver des chemins de connexion entre les concepts de la première ontologie avec un concept de la seconde ontologie en utilisant Protégé 2000. Protégé 2000 est l'un des meilleurs logiciels de gestion d'ontologies actuellement disponibles. La performance de cette application est attribuable à l'efficacité de ses outils intégrés, tels que PROMPT Suite [NOY03]. Elle est constituée d'un ensemble d'outils utiles pour la fusion et la mise en correspondance des ontologies. L'un des outils PROMPT est iPROMPT, qui effectue des opérations de fusion d'ontologies de base. La première étape de l'algorithme requiert deux ontologies en entrée et renvoie une liste de

premières suggestions de correspondances basées sur l'équivalence lexicale des noms de concepts [NOY00]. L'algorithme passe ensuite à l'étape suivant, dans laquelle les utilisateurs effectuent une action de leur choix.

Cette opération est une tâche de l'algorithme, qui est effectuée après une intervention humaine. Le choix de l'opération se fait en sélectionnant une des suggestions ou en spécifiant l'opération requise en utilisant l'environnement d'édition de l'ontologie. L'étape suivante d'iPROMPT exécute automatiquement les modifications selon l'opération précédemment sélectionnée. Ensuite, iPROMPT génère à nouveau une liste de suggestions basées sur la structure de l'ontologie, les incohérences et les problèmes résolus après l'exécution de l'opération. Enfin, iPROMPT propose des solutions à ces problèmes et génère l'ontologie de fusion.

Cependant, les outils iPROMPT présentent certaines limites :

- La semi-automatisation de l'algorithme de fusion.
- L'iPROMPT considère la structure de l'ontologie, mais ne considère pas le traitement des relations entre les concepts et la pertinence des concepts.

La figure suivante est une capture d'écran de notre implémentation représentant les graphes de nœuds des deux ontologies sélectionnées ainsi leur graphe de fusion (voir figure 25).

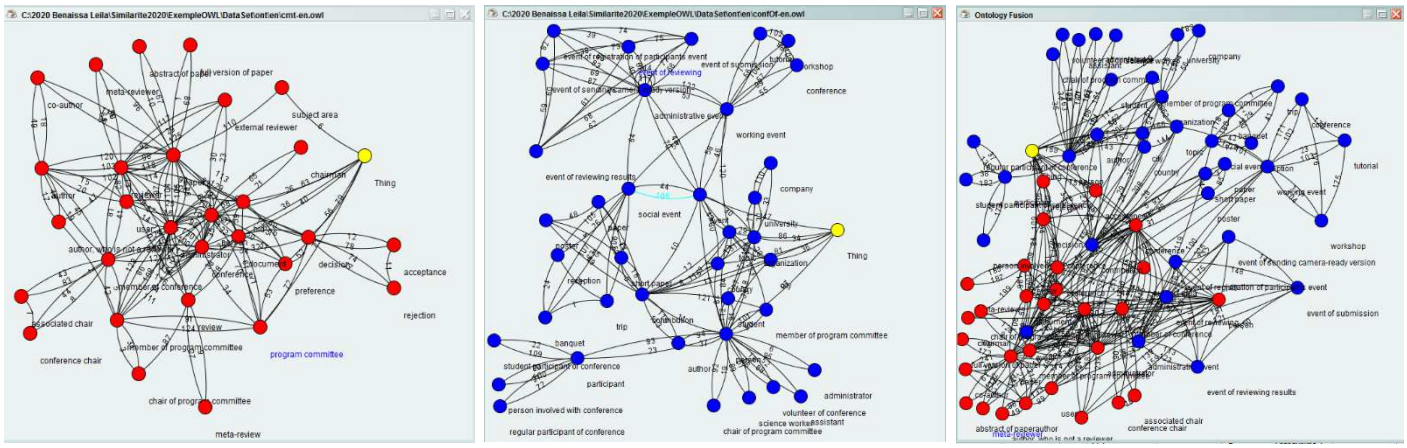


Figure 25. Interface de nos deux ontologies OWL et leur fusion sous forme de graphe de nœuds

3.3. Traduction et lemmatisation :

Dans cette étape pour la langue arabe et française, les concepts des deux ontologies doivent être traduits en anglais, car par la suite, nous utiliserons WordNet qui prend les termes uniquement en cette langue.

Après avoir analysé les concepts des deux ontologies, nous remarquons que chaque concept est une phrase (ensemble de mots). Avant d'appliquer la mesure de similarité, nous avons utilisé TreeTagger [SCH94] pour lemmatiser tous les concepts afin de ne garder que les mots pertinents de chaque concept, comme le montre la Figure 26.

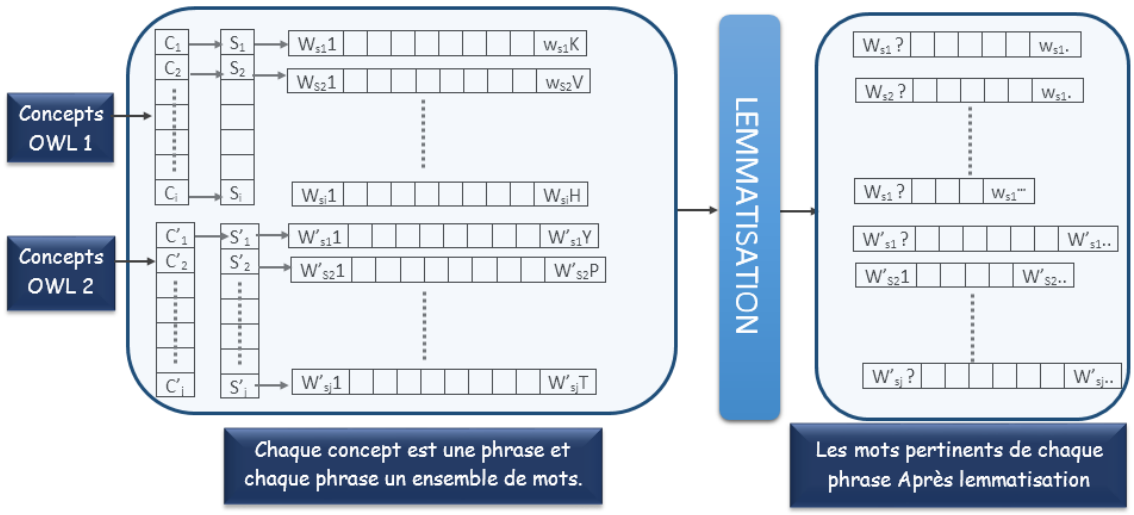


Figure 26. Prétraitement des concepts

Pour la langue anglaise en passe directement à l'étape de lemmatisation par contre pour l'arabe et le français en commence par la traduction et ensuite la lemmatisation, la figure 27 est une capture d'écran de l'exécution de cette étape dans notre application.

La traduction est faite dans notre travail, ont utilisons l'API Google Translate qui est un outil de traductions bien connu, rapide et dynamique, adaptable à divers besoins en matière de contenu, développé par Google. Grâce à cette API, les blocs linguistiques du texte peuvent être facilement détectés et traduits dans d'autres langues préférées. L'API est conçue pour être un outil simple et facile pour détecter ou traduire des langues (voir Annexe A).

Le TreeTagger est un outil permettant d'annoter des textes avec des informations sur les parties du discours et les lemmes. Helmut Schmid [SCH94] l'a créé dans le cadre du projet TC de l'Institut de linguistique informatique de l'Université de Stuttgart. Le TreeTagger a été utilisé avec succès pour annoter des textes en allemand, anglais, français, italien, danois, suédois, norvégien, néerlandais, espagnol, bulgare, russe, portugais, galicien, grec, chinois, swahili, slovaque, slovène, latin, estonien, polonais, persan, roumain, tchèque, copte. Pour plus de détails sur le lemmatiseur **TreeTagger** voir (Annexe B). Nous avons choisi les termes pertinents dans notre approche (Nom, Adjectif, Verbe), pour calculer les poids de similarité entre les concepts (Ci, Cj). En utilisant l'algorithme suivant :

Algorithme (1) de lemmatisation à l'aide de TreeTagger

Entrées : Ensemble de concepts Ci // sachant que chaque concept est une phrase et chaque phrase un ensemble de mots.

Sorties : $\mathbf{V_{rt}}$: vecteur contenant les termes pertinents pour chaque concept

Méthode :

$\mathbf{V_{rt}} \leftarrow \phi$

pour chaque Ci **faire**

début

Soit $\mathbf{V_{token}}$ Vecteur des termes pertinents Ti du concept Ci

pour chaque terme $\mathbf{T_i}$ du vecteur $\mathbf{V_{token}}$ **faire**

début

Lemmatisation ($V_{token}(\mathbf{T_i})$)

Soit **token**← terme $\mathbf{T_i}$

Soit **pos**← type $\mathbf{T_i}$

Soit **lemma**← original $\mathbf{T_i}$

Si pos ∈ {Nom, adjectif, verbe}

$\mathbf{V_{rt}} \leftarrow \mathbf{V_{rt}} \cup$ **lemma**

fin

Retourner $\mathbf{V_{rt}}$

Fin

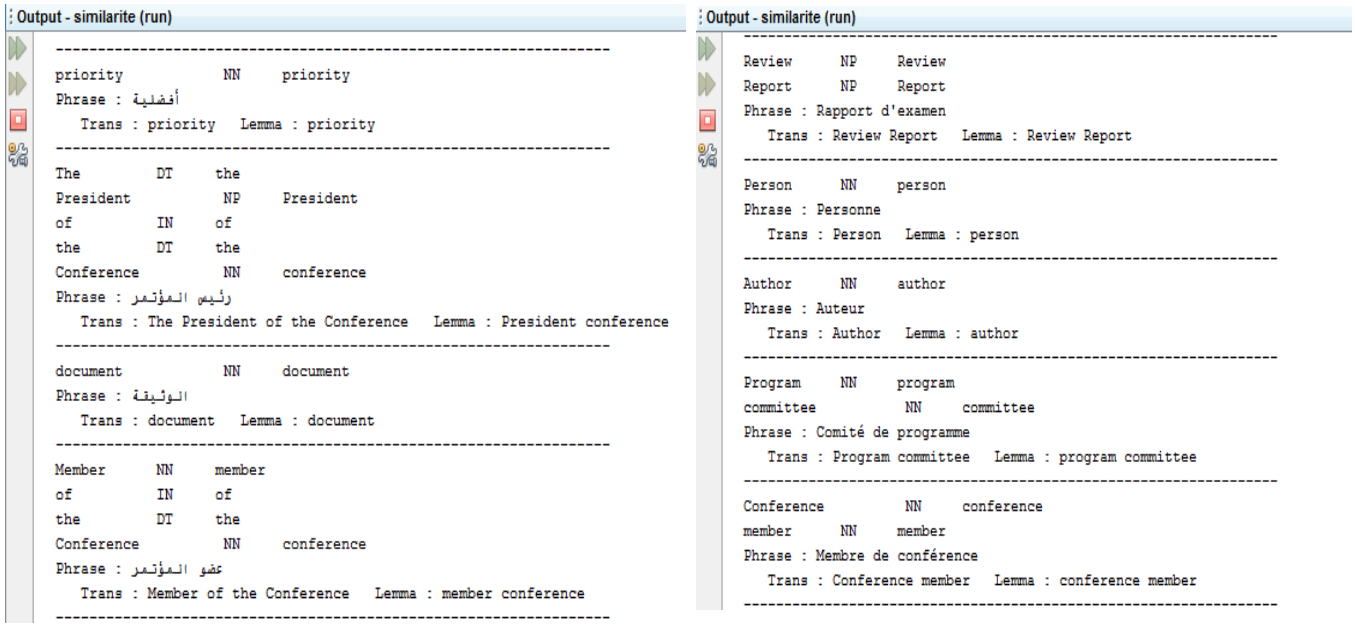


Figure 27. Exécution de la Traduction et la lemmatisation
(Arabe - Français)

3.4. Le calcul des matrices :

Dans cette étape, nous calculons la matrice d'incidence sur *OWL3* en mettant "0" s'il n'y a pas de chemin entre deux concepts (*Ci, Cj*), sinon nous mettons "1". En fonction de cette matrice, nous appliquons l'algorithme de *Dijkstra*, qui est l'un des algorithmes les plus courants qui résolvent le problème de la recherche des chemins les plus courts entre un nœud source particulier et tout autre nœud, où le poids du lien est égal à "1". Nous calculons toutes les distances entre chaque concept et la racine (*racine R*), la raison est que certaines mesures de similarité utilisées dans ce travail sont basées sur le calcul des distances qui séparent les nœuds désirés du nœud racine *R* et la distance qui sépare le concept subsumant (*CS*) de ces nœuds. Cette phase génère une distance minimale entre un concept et son nœud *racine R.* Voir l'algorithme suivant (Algorithme (2)) :

Algorithme (2) pour trouver le plus court chemin entre les concepts

Entrées :

- Max : nombre de tous les concepts dans l'ontologie de fusion OWL3,
- Mat : matrice entière de taille (max) x (max),
- MI : c'est une matrice de lien entre les concepts, égale à 1 s'il y a un lien sinon -1

Sorties : Mpath : matrice de tous les chemins de chaque départ (d) à chaque arrivée (a)

Méthode :

Pour i=0 : Max **faire**
Pour j=0 : Max **faire**
si (MI[i,j] == -1)
Mat[i,j] ← 100000
d← 0
Tant que (d< Nombre de concepts) **faire**
début
Dijkstrada // da est une instance de Dijkstra
da ensemble de tous les chemins du départ d à l'arrivée a dans la matrice Mat
Pour a =1 : Max **faire**
début
Soit V un vecteur de tous les chemins de d pour arriver à a
Soit N soit un vecteur entier de taille V.
pathda← Ø //path est un ensemble de nœuds où d est le premier nœud et a est le dernier nœud initialisé à l'ensemble vide.
Pour i =0 : sizeof(N) **faire**
début
N[i] ←Get(V (i)) // reçoit la valeur de V[i]
pathda←pathda∪ {N[i]}
Fin
Stockage du chemin da dans la matrice Mpath à la ligne d colonne a
Fin
d←d+1
Fin de Tant **que**

Après avoir trouvé le plus court chemin entre les concepts, en cherche le concept subsumant (*CS*) entre deux nœuds *Ci* et *Cj,* en appliquant l'algorithme suivant :

Algorithme (3) pour trouver le CS entre deux concepts C_i et C_j

Entrées : Matrice des chemins entre les concepts Mpath, Ci, Cj // Mpath c'est la matrice de tous des plus cours chemin entre l'ensemble des concepts de ow_3
Sorties : CS et $dist_{min}$ // CS c'est le concept commun entre de concepts Ci et Cj, et $dist_{min}$ et al distance minimal entre **CS** et Racine
Méthode :
CS ←" "
$dist_{min}$**←** infini
Soit N1=0
Soit N2=0
Pour chaque **i=0 à Taille (**Mpath**) faire**
début
Si Mpath [i, 0] = Ci alors **N1←** i
Si Mpath [i, 0] = Cj alors **N2←** i
Fin pour
Soit path← Mpath [**N1**, **N2**]
Soit vect ← Split (path, (", "))
Soit Mat matrice [ ][ ] **→**[Len(vect)] [2] //Len la taille du vecteur
Pour chaque **i=0 à** Len(vect) **faire**
début
Mat[i][0] **←** Position (vect[i])
Mat[i][1] **←** 0
Fin pour
Pour chaque **i=0 à** Len (vect) **faire** //Len la taille du vecteur
début
Mat[i][1] **←** Distance(Mat[i][0],Racine)// Distance entre le concept Mat[i][0], et la Racine
Si Mat[i][1] < >0 **alors**
Si Mat[i][1]< $dist_{min}$ **alors**
début
$dist_{min}$**←** Mat[i][1]
CS ← Mpath (Mat[i][0], 0)
Fin
Fin si
Fin si
Fin pour
Retourner CS, $dist_{min}$
Fin

3.5. Calcul de la mesure de similarité :

Nous avons implémenté les différentes mesures de similarité, les formules des quatre mesures de similarité (Wu & Palmer [WUP94], Rada [RAD89], Li [LI 03]) sont détaillées dans chapitre 3 afin de les utiliser pour la comparaison avec nos mesures

hybrides proposées. Concernant la méthode de Zargayouna [ZAR04] parmi les paramètres d'entrer utilisé pour appliquer cette mesure et le calcul N_4 le nombre maximum d'arcs qui séparent *CS* du Bottom (voir figure 20 chapitre 3), ce dernier est calculé suivant cet algorithme (voir algorithme 4) :

Algorithme(4) pour calculer N_4

Entrées : Matrice des chemins entre les concepts Mpath, Ci, Cj, CS
Sorties : N_4 // **le** nombre maximum d'arcs qui séparent *CS* du Bottom

Méthode :
ordeC$_i$←0
ordeC$_j$←0
ordeC$_S$←0
N_4←0
Pour chaque **i=0 a Taille** (Mpath) **faire**
début
si Mpath[i,0] = Ci alors **ordeC**$_i$← i
si Mpath[i,0] = Cj alors **ordeC**$_j$← i
si Mpath[i,0] = **CS** alors **ordeC**$_S$← i
Fin pour
Pour chaque **i=0 a Taille** (Mpath) **faire**
début
Soit **path**← Mpath[i, **ordeC**$_s$]
Soit **vect** ← Split (path,(", "))
Si Taille (vect)>0 alors
début
Pour chaque **j=0 a Taille** (vect) **faire**
Si vect(j)= ordeC$_i$ **ou vect(j)= ordeCj alors**
si Taille (vect)> N_4 **alors**
N_4 **= Taille** (vect)
Fin si
Fin si
Fin pour
Fin si
Fin pour
Retourner N_4
Fin

3.6. La matrice des poids de similarité :

L'objectif de cette étape est de calculer le poids de similarité entre chaque concept d'OWL1 avec les concepts d'OWL2. Sachant que les concepts de nos deux ontologies sont sous forme des phrases, afin de calculer le poids de similarité entre ces concepts, nous utilisons l'étape de lemmatisation, qui a été appliquée précédemment. Cela nous a permis d'obtenir un ensemble de termes pertinents pour chaque concept, afin de calculer d'abord le poids de similarité entre les paires de mots avant de passer à la phrase. Nous avons utilisé WorldNet, ce dernier est un dictionnaire sémantique électronique anglais gratuit développé par des chercheurs en sciences cognitives, dirigés par Miller à l'Université de Princeton [MIL95]. Le WorldNet 2.0 comprend totalement 150 000 mots, qui sont organisés en 115 000 ensembles de synonymes. Il existe 207 000 groupes de sens de mots. Chaque synonyme représente un concept sémantique de base et est lié à des relations conceptuelles sémantiques et lexicales. Nous utilisons WorldNet pour calculer le poids de la similarité entre deux mots. La valeur obtenue sera utilisée dans la formule (36) pour calculer le poids de la similarité entre les concepts.

Après avoir utilisé la formule [YAZ19] pour calculer la similarité entre deux concepts, le premier concept provient d'OWL1, le second d'OWL2, Où w_i représente les mots du concept d'OWL1, wj représente les mots du concept d'OWL2, n le nombre de mot du concept d'OWL1 et m le nombre de mot du concept d'OWL2.

$$Sim(C_{O1}, C_{O2}) = \frac{\sum_{i=0}^{n} MaxSim(wi(w'1,\ldots,w'm))}{2n} + \frac{\sum_{j=0}^{m} MaxSim(w'j,(w1,\ldots,wn))}{2m} \qquad (36)$$

OWL 1 \ OWL 2					C_1			C_2		C_3	C_4				C_5			C_6	C_7		C_8
					w_1	w_2	w_3	w_1	w_2	w_1	w_1	w_2	w_3	w_4	w_1	w_2	w_3	w_1	w_1	w_2	w_1
C'_1	$w'1$	$w'2$	$w'3$	$w'4$																	
C'_2	$w'1$		$w'2$																		
C'_3	$w'1$																				
C'_4	$w'1$		$w'2$	$w'3$																	
C'_5	$w'1$	$w'2$	$w'3$	$w'4$	$w'5$																
C'_6	$w'1$																				
C'_7	$w'1$		$w'2$																		

Calcul de la similarité pour chaque deux concepts en parcourant OWL 3

Figure 28. Matrice des poids de similarité

Le poids de la similarité entre les concepts permet de construire la matrice des poids de similarité (figure 28), qui représente la similarité sémantique entre ces deux ontologies.

3.7. Calcul de la matrice hybride :

Dans cette étape, la matrice d'incidence est mise à jour en calculant l'union de la matrice d'incidence calculée à la section 3.4 avec la matrice du poids de similarité calculée à la section 3.6. Ensuite, l'algorithme est modifié pour intégrer la mise à jour de la matrice d'incidence.

Algorithme (5) d'optimisation du plus court chemin entre les concepts

Entrées :
- Max : nombre de tous les concepts dans l'ontologie de fusion OWL3,
- Mat : matrice entière de taille (max) x (max),
- MI : c'est une matrice de lien entre les concepts, égale à 1 s'il existe un lien sinon -1
- MP : matrice des poids de la similarité entre tous les concepts.

Sorties : $Mpath_{sh}$ Mise à jour de la matrice de similarité hybride de tous les chemins de chaque départ (d) à chaque arrivée (a)

Méthode :

Soit MIup : matrice d'union entre MI et MP
Pour i=0 : Max **faire**
Pour j=0 : Max **faire**
si (MIup [i,j] == -1)
Mat [i,j] ← 100000
d← 0
Tant que (d< Nombre de concepts) **faire**
début
Dijkstrada // da est une instance de Dijkstra
da ensemble de tous les chemins du départ d à l'arrivée a dans la matrice Mat
Pour a =1 : Max **faire**
début
Soit V un vecteur de tous les chemins depuis d pour arriver à a
Soit N est un vecteur entier de taille V
pathda← Ø //path est un ensemble de noeuds où d est le premier noeud et a est le dernier noeud initialisé à l'ensemble vide.
Pour i =0 : sizeof(N) **faire**
début
N[i] ←Get(V (i)) // reçoit la valeur de V[i]
pathda←pathda∪ {N[i]}
Fin
stockage du chemin da dans la matrice $Mpath_{sh}$ à la ligne d colonne a
Fin

d←d+1
Fin Tant que

Pour trouver le nouveau concept subsumant (*CSsh*) entre deux nœuds *Ci* et *Cj,* en appliquent le même algorithme (3) le changement est uniquement dans la matrice d'entrer où en utilise la nouvelle matrice d'optimisation du plus court chemin entre les concepts **Mpath**$_{sh}$

3.8. Les mesures hybrides réalisées :

Le principe du calcul de la similarité sémantique avec les approches basées sur la structure de l'ontologie, telles que Wu & Palmer[WUP94] ,Rada [RAD89], Li[LI 03] et Zargayouna [ZAR04] qui repose sur l'idée qu'un chemin plus court entre deux nœuds les rend plus similaires. Un autre point concernant ces approches est que les arcs représentent des distances uniformes. Par conséquent, cette approche présente l'inconvénient que tous les liens sémantiques ont le même poids, ce qui rend difficile la définition et le contrôle des distances de liaison. C'est la raison pour laquelle nous avons choisi d'hybrider ces approches avec le poids de la similarité pour renforcer l'aspect sémantique.

3.8.1. La mesure WWP :

Dans cette approche, nous avons utilisé la matrice hybride calculée en 3.7, et nous l'avons appliquée à la mesure Wu & Palmer [WUP94]. Cela est fait en recalculant N_3' qui représente la distance entre CS_{sh} le nouveau subsumant commun (sh : similarité hybride) et la racine, N_1' le nouveau plus court chemin entre le concept C_1 et CS_{sh} et N_2' le nouveau plus court chemin entre le concept C_2 et CS_{sh}, en appliquant la formule 37.

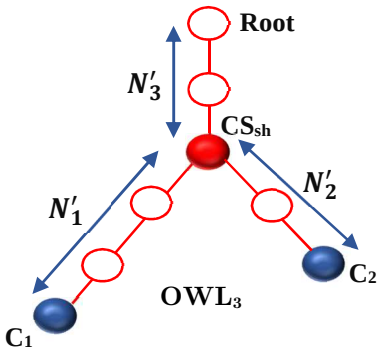


Figure 29. Exemple de hiérarchie de concepts dans WWP

$$Sim_{WWP}(C_1, C_2) = \frac{2*N_3'}{N_1'+N_2'+2*N_3'} \qquad (37)$$

3.8.2. La mesure WRA:

Dans cette approche, nous utilisons la formule de Rada [RAD89], mais avec un calcul actualisé de la distance entre *C1* et *C2* (voir formule 38). Considérant que *distsh*(C_1, C_2) correspondent au nombre d'arcs, en tenant compte du poids de similarité entre les concepts qui doivent être traversés pour connecter les concepts *C1* et *C2*.

$$Sim_{wra}(C_1, C_2) = \frac{1}{1+dist_{sh}(C_1,C_2)} \qquad (38)$$

3.8.3. La mesure WLI :

On a appliqué le principe de la mesure de Li [LI 03], notre mesure de similarité WLI combine la longueur du nouveau plus court chemin (SP') entre deux concepts C_1 et C_2, et la profondeur dans la taxonomie (N') du concept commun le plus spécifique CS_{sh}, dans une fonction non linéaire. En utilisant la formule 39 :

$$Sim_{WLI}(C_1, C_2) = e^{-\alpha*SP'} \frac{e^{\beta*N'}-e^{-\beta*N'}}{e^{\beta*N'}+e^{-\beta*N'}} \qquad (39)$$

Les paramètres optimaux sont α= 0,2 et β = 0,6 sur la base de [LI 03].

3.8.4. La mesure WZY :

De la même manière que les autres mesures nous avons utilisé la formule de la mesure de Zargayouna [ZAR04], le changement c'est dans le calcul du chemin plus court entre deux concepts. N_3' qui représente la distance entre CS_{sh} le nouveau subsumant commun (sh : similarité hybride) et la racine, N_1' le nouveau plus court chemin entre le concept C_1 et CS_{sh} et N_2' le nouveau plus court chemin entre le concept C_2 et CS_{sh} et N_4' le nouveau nombre maximum d'arcs qui séparent CS_{sh} de Bottom, en appliquant la formule suivante :

$$\begin{cases} Sim_{wzy}(C_1, C_2) = \dfrac{2 * N_3'}{N_1' + N_2' + 2 * N_3' + spec(C_1, C_2)} \\ spec(C_1, C_2) = N_4' * N_1' * N_2' \end{cases}$$

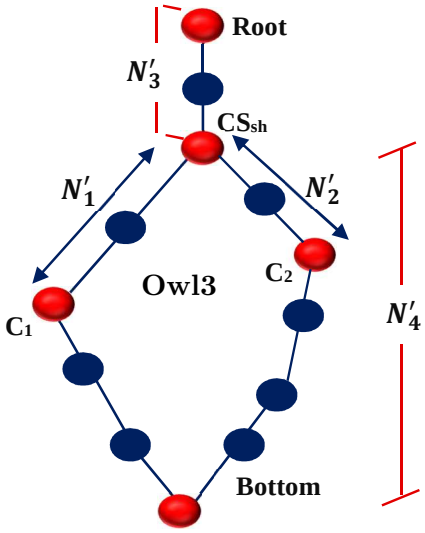


Figure 30. Exemple de hiérarchie de concepts dans WZY

Pour trouver le nouveau N_4', en applique le même Algorithme (4) le changement est dans la matrice d'entrer où en utilise la nouvelle matrice d'optimisation du plus court chemin entre les concepts $Mpath_{sh}$ et concernant le concept commun en utilise CSsh. Dans la figure 32, nous voulons montrer premièrement l'impact d'intégration du poids de similitude sur le calcul du chemin plus cours entre deux concepts. Nous remarquant pour les mêmes deux concepts C_i et C_j que les *Ccommun* a changé, que la distance *N1* et *N2* a également changer (voir la figure 31).

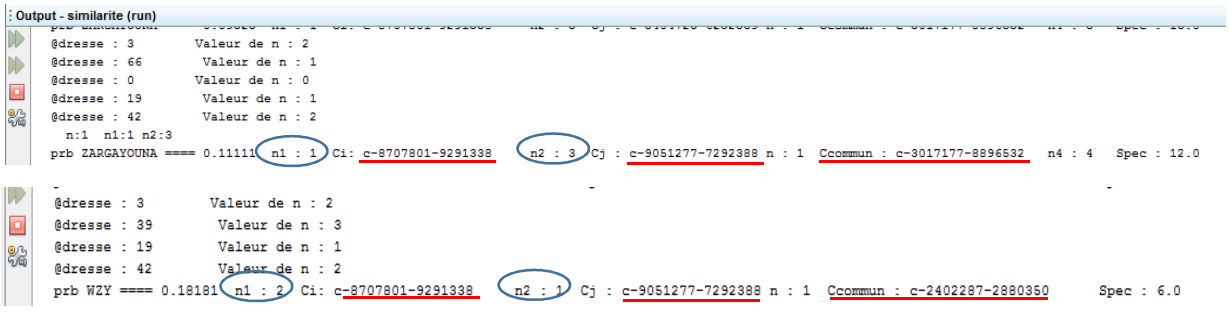


Figure 31. Exécution des mesures Zargayouna et WZY

3.9. Processus D'évaluation :

Cette section sera détaillée dans le chapitre suivant, elle comprend deux parties, la première concerne la comparaison entre les différentes mesures basées sur la structure avec nos mesures hybrides proposées, et la deuxième partie concerne l'évaluation de notre approche en utilisant deux métriques, la densité et la cohésion.

4. Conclusion :

Dans ce chapitre, nous avons présenté en détail les étapes suivies pour arriver à la réalisation de notre approche qui est l'intégration du facteur sémantique dans le calcul des mesures basé sur la structure de l'ontologie.

Le principe de calcul de similarité des approches tel que [WUP94], [RAD89], [ZAR04] et [LI 03] est basé sur l'idée suivante : plus le chemin entre deux nœuds est court plus ils sont plus semblables. L'autre notion qui caractérise cette deuxième approche est que les arcs d'une taxonomie représentent des distances uniformes, par conséquent, cette approche présente l'inconvénient que tous les liens sémantiques possèdent le même poids ce qui impose des difficultés au niveau de la définition et du contrôle des distances des liens, cela est l'un des points faibles de ces mesures. Pour contourner ce problème l'idée était, au lieu que le poids soit uniforme entre les nœuds en le remplace par le poids réel de similarité sémantique entre les nœuds. Cela nous a été possible en utilisant WordNet qui possède parmi ces fonctions le calcul de poids de similitude.

Dans le chapitre suivant, nous présentons la réalisation de notre approche ont comparons les résultats obtenus par l'utilisation des mesures basée sur la structure des ontologies et nos mesures hybride. Nous évaluerons notre étude en utilisant des métriques à savoir : la densité et la cohésion.

Chapitre 5

Expérimentation & Évaluation

1. Introduction :

Nous avons présenté dans le chapitre précédant notre approche hybride pour calculer la similarité sémantique entre les concepts de différentes ontologies OWL. Dans ce chapitre, nous allons présenter l'implémentation de notre approche, on commence tout d'abord par présenter l'environnement de développement ainsi que les différents outils utilisés, puis nous donnons une description de notre application à travers quelques fenêtres de capture qui représentent les interfaces de ce dernier, qui sont conçues de manière à être conviviales et simples d'utilisation. Cette étape nous a aussi permis de nous familiariser avec les outils utilisés pour le développement de notre application. Ensuite, une étude expérimentale est réalisée en appliquant notre approche sur différentes mesures, couple d'ontologies et langue. Nous évaluons finalement les résultats en utilisant deux métriques d'évaluation la cohésion et la densité et nous conclurons ce chapitre.

2. Environnement de développement :

Tous les tests de notre application sont exécutés sur un ordinateur portable équipé d'un processeur Intel Core (TM) i3 Duo 2,30 GHz et de 4 Go de RAM, sous le système d'exploitation Windows 7, mais comme elle est développée en langage de programmation Java, elle peut être utilisée avec tout autre système d'exploitation supportant la machine virtuelle Java (Linux, ...). Concernant les outils, l'éditeur Protégé (Voir chapitre 2 sections 5.1.1) et NetBeans IDE 8.2, ce dernier est un environnement de développement intégré (EDI), placé en open source en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multilangages, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme, il s'enrichit à l'aide de plugins. La page d'accueil de NetBeans, illustrée ci-dessous.

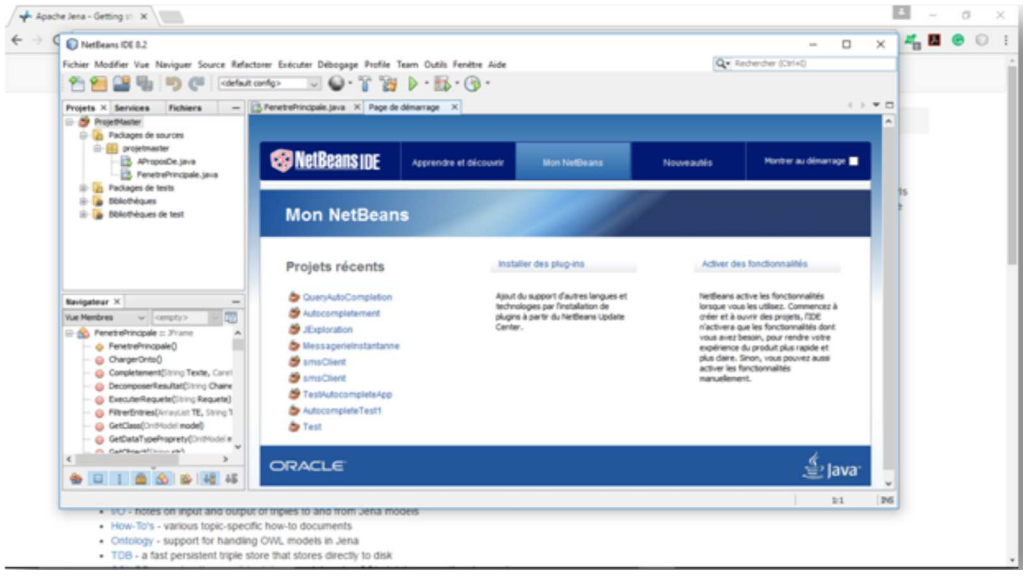


Figure 32. Fenêtre principale de NetBeans.

3. Description de l'implémentation :

L'interface homme/machine représente l'élément-clé dans l'utilisation de toute application informatique. Les interfaces de notre application ont été conçus de manière à être simples, naturelles, de compréhension et d'utilisation faciles. Pour pouvoir utiliser notre application, l'utilisateur doit d'abord charger les fichiers de l'ontologie OWL après validation par protégé (toutes les ontologies utilisées ici sont validées par l'outil protégé).

3.1. Interface principale :

L'interface illustrée par la figure ci-dessous représente l'interface principale de notre application.

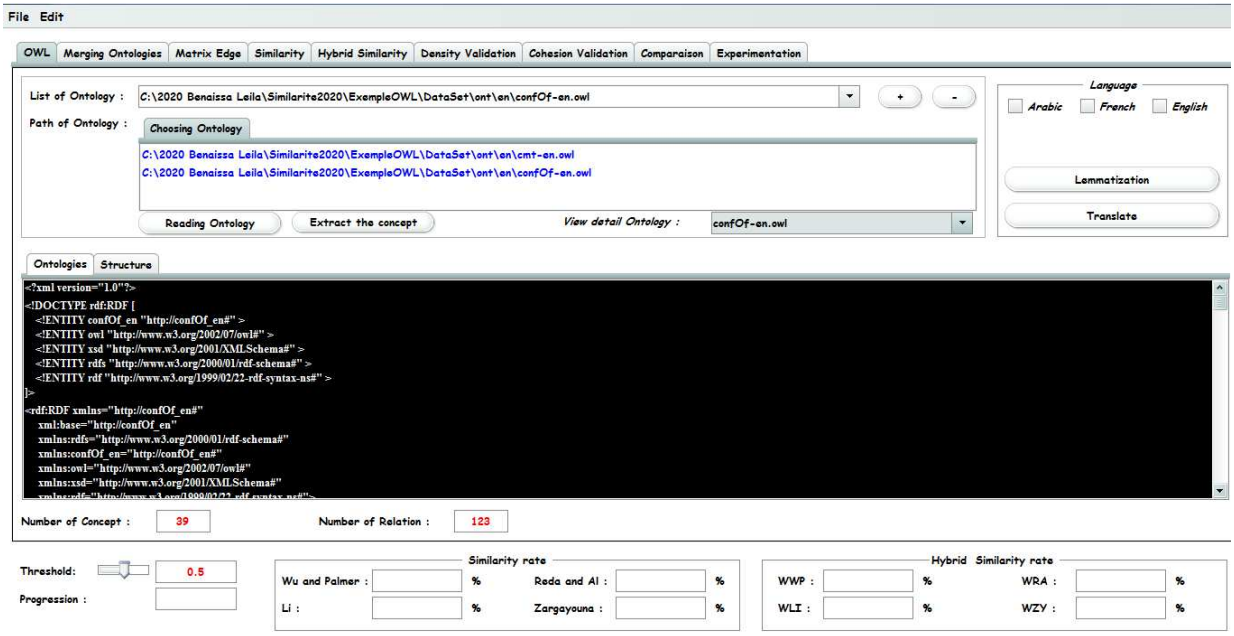


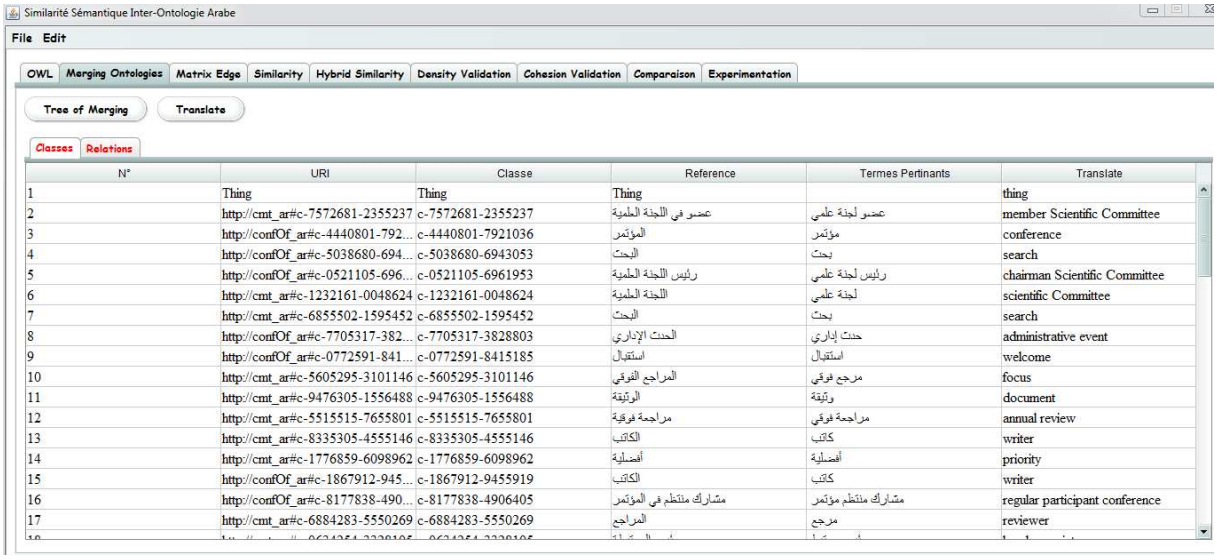
Figure 33. Interface principale de notre Application

Après le chargement du fichier .owl de nos deux ontologies (voir figure 33), on lance le module d'analyse et d'extraction d'ontologie qui permet d'extraire les informations utiles de notre ontologie telques les concepts, les relations et d'éliminer tout ce qui est bruit tel que les tags. Car dans un fichier OWL, les constructeurs sont délimités par des balises. Pour réaliser cette étape, nous avons utilisé un algorithme de cleansing.

3.2. Prétraitement des données :

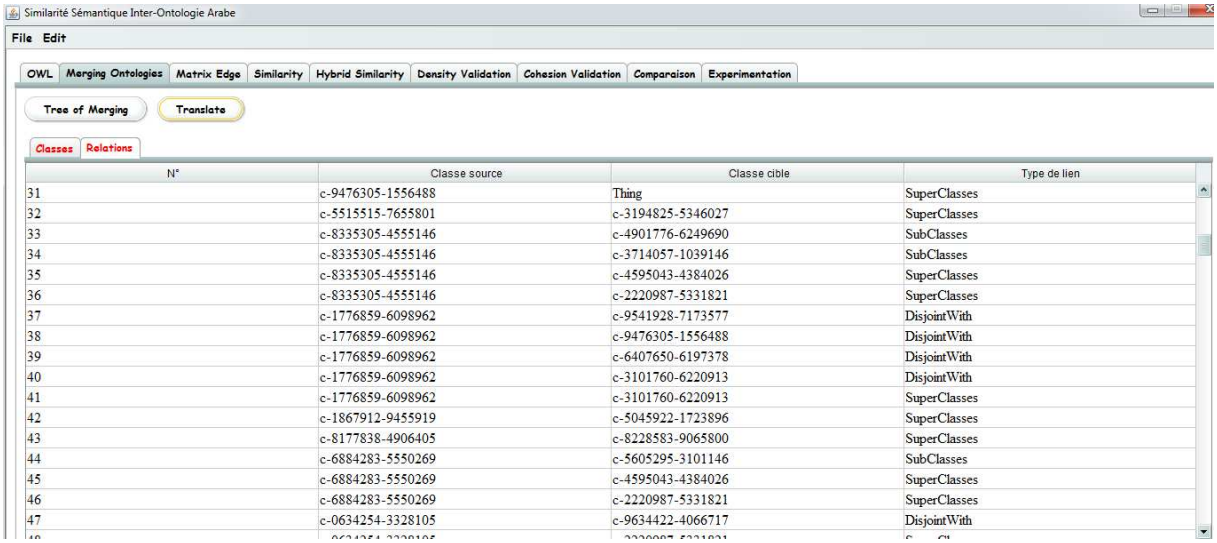
Sachant que nos concepts sont sous-forme de phrase, si nos ontologies OWL1 et OWL2 sont en anglais en lance directement l'opération de lemmatisation et si elles sont en arabe ou français en passe tout d'abord par l'étape de traduction ensuite la lemmatisation.

Après la lemmatisation en passe à la fusion d'OWL1 et OWL2 pour construire OWL3 (voir figure 34). Ces étapes permettent de préparer les données pour appliquer les mesures Wu & Palmer, Rada, Li et Zargayouna et nos mesures proposées WWP, WRA, WLI et WZY.



N°	URI	Classe	Reference	Termes Pertinants	Translate
1	Thing	Thing	Thing		thing
2	http://cmt_ar#c-7572681-2355237	c-7572681-2355237	عضو في اللجنة العلمية	عضو لجنة علمي	member Scientific Committee
3	http://confOf_ar#c-4440801-792...	c-4440801-7921036	المؤتمر	مؤتمر	conference
4	http://confOf_ar#c-5038680-694...	c-5038680-6943053	البحث	بحث	search
5	http://confOf_ar#c-0521105-696...	c-0521105-6961953	رئيس اللجنة العلمية	رئيس لجنة علمي	chairman Scientific Committee
6	http://cmt_ar#c-1232161-0048624	c-1232161-0048624	اللجنة العلمية	لجنة علمي	scientific Committee
7	http://confOf_ar#c-6855502-1595452	c-6855502-1595452	البحث	بحث	search
8	http://confOf_ar#c-7705317-382...	c-7705317-3828803	الحدث الإداري	حدث إداري	administrative event
9	http://confOf_ar#c-0772591-841...	c-0772591-8415185	استقبال	استقبال	welcome
10	http://cmt_ar#c-5605295-3101146	c-5605295-3101146	المراجع الفوقي	مرجع فوقي	focus
11	http://cmt_ar#c-9476305-1556488	c-9476305-1556488	الوثيقة	وثيقة	document
12	http://cmt_ar#c-5515515-7655801	c-5515515-7655801	مراجعة فوقية	مراجعة فوقي	annual review
13	http://cmt_ar#c-8335305-4555146	c-8335305-4555146	الكاتب	كاتب	writer
14	http://cmt_ar#c-1776859-6098962	c-1776859-6098962	أفضلية	أفضلية	priority
15	http://confOf_ar#c-1867912-945...	c-1867912-9455919	الكاتب	كاتب	writer
16	http://confOf_ar#c-8177838-490...	c-8177838-4906405	مشارك منتظم في المؤتمر	مشارك منتظم مؤتمر	regular participant conference
17	http://cmt_ar#c-6884283-5550269	c-6884283-5550269	المراجع	مرجع	reviewer

Figure 34.a) Interface présentant les classes d'OWL3



N°	Classe source	Classe cible	Type de lien
31	c-9476305-1556488	Thing	SuperClasses
32	c-5515515-7655801	c-3194825-5346027	SuperClasses
33	c-8335305-4555146	c-4901776-6249690	SubClasses
34	c-8335305-4555146	c-3714057-1039146	SubClasses
35	c-8335305-4555146	c-4595043-4384026	SuperClasses
36	c-8335305-4555146	c-2220987-5331821	SuperClasses
37	c-1776859-6098962	c-9541928-7173577	DisjointWith
38	c-1776859-6098962	c-9476305-1556488	DisjointWith
39	c-1776859-6098962	c-6407650-6197378	DisjointWith
40	c-1776859-6098962	c-3101760-6220913	DisjointWith
41	c-1776859-6098962	c-3101760-6220913	SuperClasses
42	c-1867912-9455919	c-5045922-1723896	SuperClasses
43	c-8177838-4906405	c-8228583-9065800	SuperClasses
44	c-6884283-5550269	c-5605295-3101146	SubClasses
45	c-6884283-5550269	c-4595043-4384026	SuperClasses
46	c-6884283-5550269	c-2220987-5331821	SuperClasses
47	c-0634254-3328105	c-9634422-4066717	DisjointWith

Figure 34.b) Interface présentant les relations d'OWL3

4. Expérimentations et évaluations :

Pour tester la fiabilité et montrer la faisabilité de notre approche, nous l'avons expérimenté sur plusieurs couples d'ontologies trouvées dans OAEI [CHE14] décrit le domaine des conférences, ce database contient cinq différentes ontologies écrites en plusieurs langues, dans notre application en a fait les tests sur trois langues (Anglais-Arabe et Français). La figure 35 représente des graphes de comparaison des huit mesures implantées, cette comparaison est faite sur deux ontologies **cmt** contenant 30 concepts, 131 relations et **ConfOf** contenant 39 concepts, 123 relations.

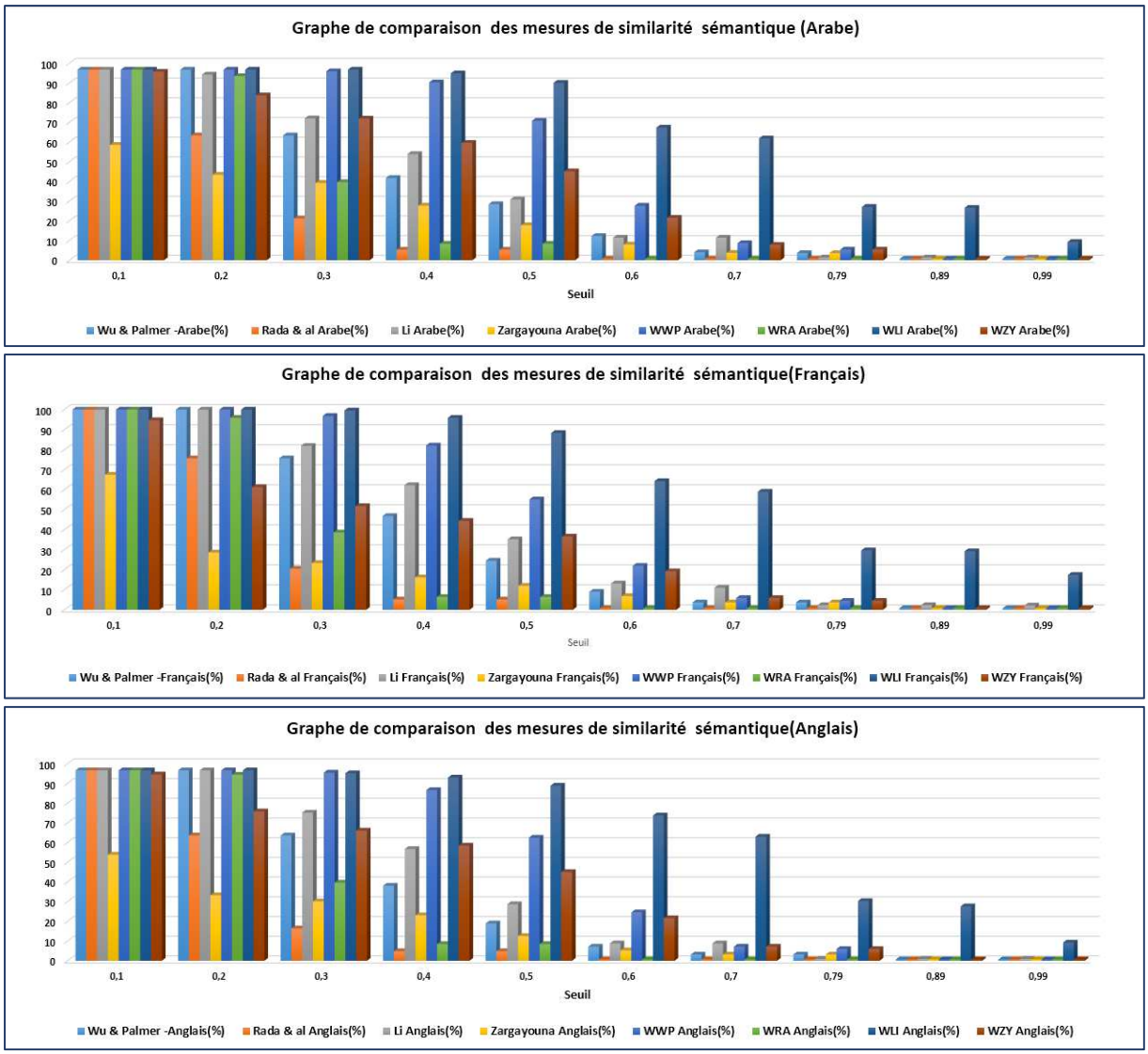


Figure 35. Graphe de comparaison des mesures de similarité sémantique

En examinant ces graphes de loin, du point de vue de langue en estime que ces mesures

donnent les mêmes résultats pour les trois langues, mais de près en étudiant les chiffres (voir Table 4, Table 5 et Table 6) en constatent que les résultats obtenus pour la langue arabe son meilleur que celle des autres langues, mais l'information pertinente qui sort de cette comparaison et que la mesure WLI donne le meilleur résultat par rapport aux autres mesures dans toutes les langues.

N°	Seuil	Wu & Palmer - Arabe(%)	Rada & al Arabe(%)	Li Arabe(%)	Zargayouna Arabe(%)	WWP Arabe(%)	WRA Arabe(%)	WLI Arabe(%)	WZY Arabe(%)
1	0,1	96,774	96,774	96,774	58,548	96,774	96,774	96,774	95,725
2	0,2	96,774	63,387	94,274	43,387	96,774	93,467	96,774	83,709
3	0,3	63,387	21,209	72,016	39,274	95,967	39,677	96,774	71,935
4	0,4	41,774	5,322	53,951	27,741	90,322	8,387	94,919	59,596
5	0,5	28,467	5,322	30,887	17,741	70,806	8,387	90,08	45,161
6	0,6	12,258	0,769	11,451	7,983	27,661	0,769	67,338	21,532
7	0,7	4,032	0,769	11,451	3,909	8,629	0,769	64,854	7,822
8	0,79	3,629	0,769	1,37	3,729	5,903	0,769	29,096	5,403
9	0,89	0,769	0,769	1,37	0,769	0,769	0,769	26,612	0,769
10	0,99	0,769	0,769	1,37	0,769	0,769	0,769	9,274	0,769

Table 4. Expérimentations des différentes mesures entre Cmt et ConfOf écrite en Arabe

N°	Seuil	Wu & Palmer - Français(%)	Rada & al Français(%)	Li Français(%)	Zargayouna Français(%)	WWP Français(%)	WRA Français(%)	WLI Français(%)	WZY Français(%)
1	0,1	97,884	97,884	97,884	67,435	97,884	97,884	97,884	94,7
2	0,2	97,884	75,641	100	28,632	97,884	95,897	97,884	61,282
3	0,3	75,641	20,512	81,88	23,333	96,837	38,632	95,572	51,794
4	0,4	46,837	5,128	62,222	16,153	82,051	6,41	94,897	44,444
5	0,5	24,529	5,128	35,213	11,965	55,128	6,41	88,29	36,581
6	0,6	8,974	0,725	13,162	6,837	21,965	0,725	64,273	19,23
7	0,7	3,675	0,725	11,025	3,675	5,897	0,725	58,974	5,897
8	0,79	3,675	0,725	2,307	3,675	4,444	0,725	29,658	4,444
9	0,89	0,725	0,725	2,307	0,725	0,725	0,725	29,23	0,725
10	0,99	0,725	0,725	2,136	0,725	0,725	0,725	17,435	0,725

Table 5. Expérimentations des différentes mesures entre Cmt et ConfOf écrite en Français

N°	Seuil	Wu & Palmer - Anglais(%)	Rada & al Anglais(%)	Li Anglais(%)	Zargayouna Anglais(%)	WWP Anglais(%)	WRA Anglais(%)	WLI Anglais(%)	WZY Anglais(%)
1	0,1	96,666	96,666	96,666	53,846	96,666	96,666	96,666	94,615
2	0,2	96,666	63,589	96,666	33,247	96,666	94,444	96,666	75,811
3	0,3	63,589	16,41	75,213	30,17	95,555	39,658	95,213	66,068
4	0,4	38,119	4,871	56,752	23,076	86,666	8,461	92,991	58,461
5	0,5	18,974	4,871	28,717	12,564	62,478	8,461	88,888	45,042
6	0,6	7,179	0,684	8,803	5,299	24,615	0,690	73,76	21,623
7	0,7	3,162	0,672	8,803	3,162	7,179	0,683	62,991	7,179
8	0,79	3,162	0,672	0,94	3,162	5,982	0,683	30,341	5,982
9	0,89	0,670	0,672	0,94	0,683	0,683	0,683	27,606	0,693
10	0,99	0,670	0,672	0,94	0,683	0,683	0,683	9,23	0,693

Table 6. Expérimentations des différentes mesures entre Cmt et ConfOf écrite en Anglais

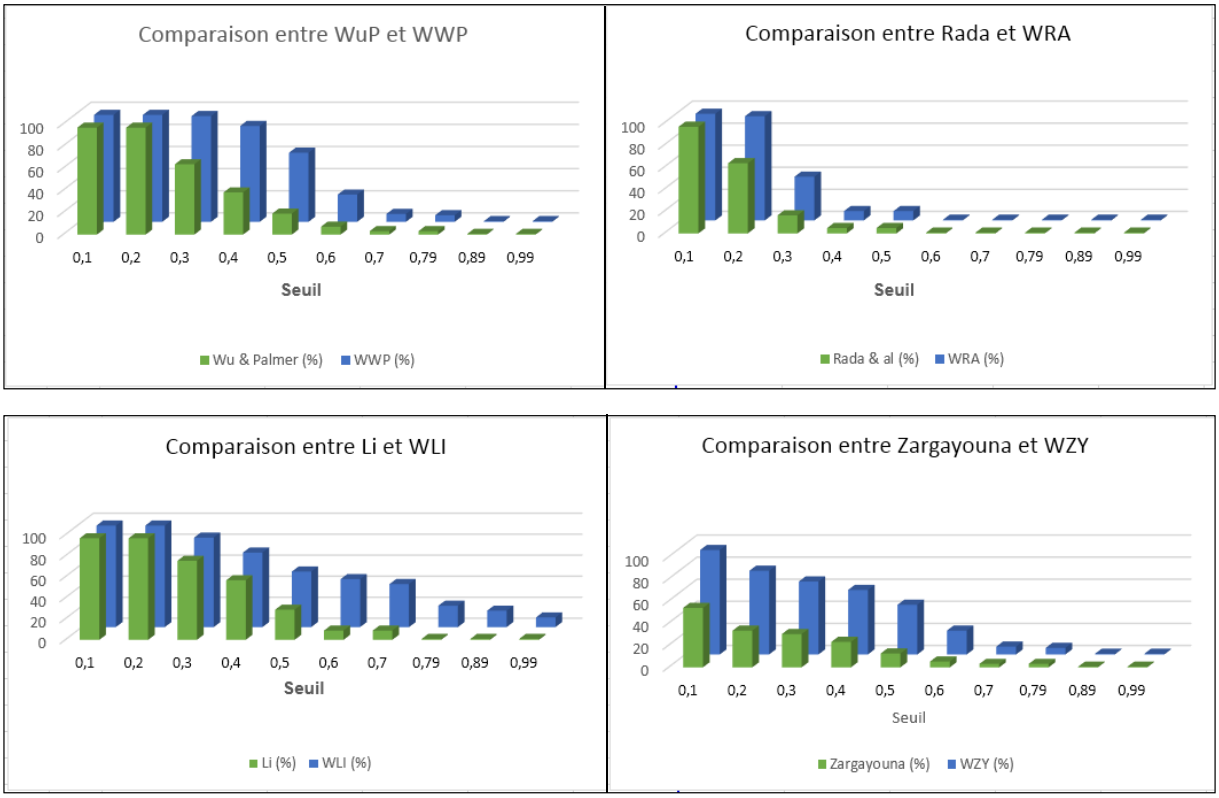


Figure 36. Comparaison entre les approches basées sur la structure avec nos approches hybrides

À travers la comparaison entre toutes les mesures Wu & Palmer, Rada, Li et Zargayouna avec WWP, WRA, WLI et WZY respectivement, représenter dans la figure 36, en constate que les mesures hybrides proposées donnent de meilleurs résultats, mais avec un pourcentage diffèrent.

En étudiant la comparaison des approches basées sur la structure avec chacune de nos approches hybride séparément (voir figure 37) avec un seuil élever égal à 0.8 en confirme que l'approche WLI donne le meilleur résultat.

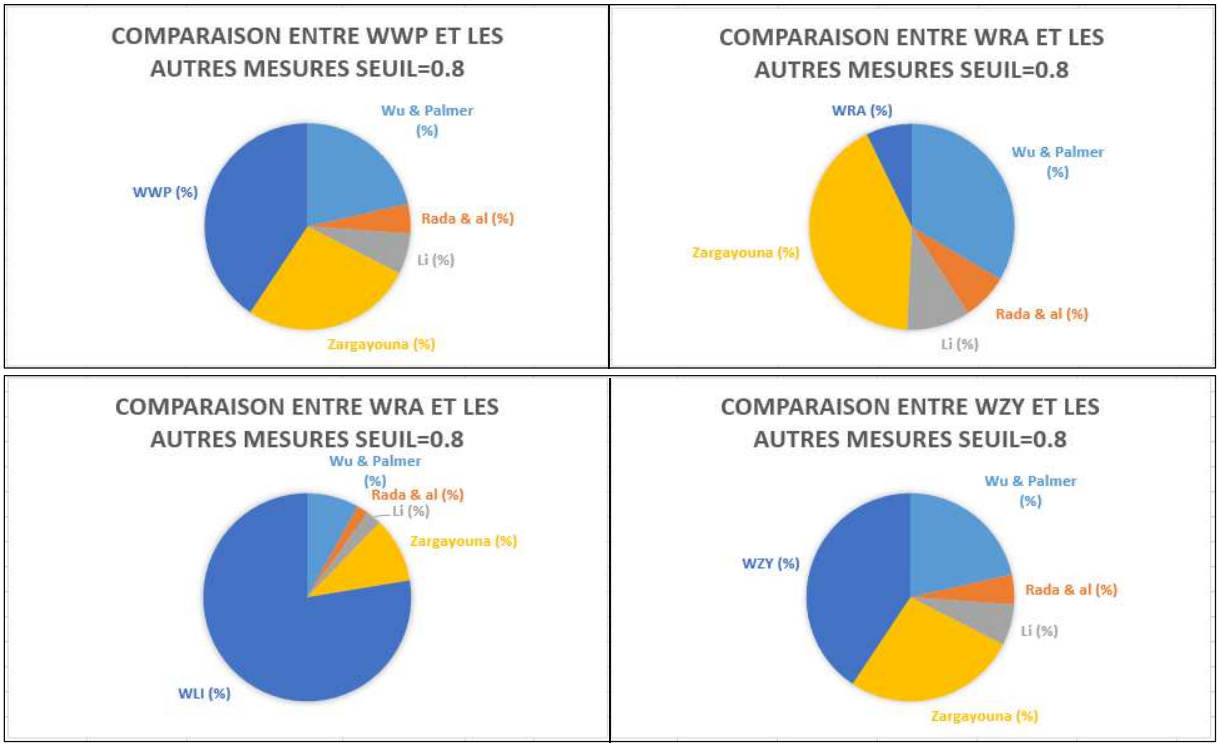


Figure 37. Comparaison entre les approches basées sur la structure avec chaque une de nos approches hybride séparément

Selon la comparaison entre toutes les mesures Wu & Palmer, Rada, Li et Zargayouna avec WWP, WRA, WLI et WZY respectivement dans les trois langues (Anglais-Arabe-Français) voir Figure 38 on constate que nos approches hybrides les remportent pour :

(1) La meilleure mesure est WWP-Arabe avec le seuil >=0.4.

(2) La meilleure mesure WRA- Français

(3) Les meilleures mesures WLI-Anglais avec le seuil entre 0.5 et 0.79 et WLI-Français >=0.79

(4) Les meilleures mesures WZA-Arabe avec le seuil <=0.79 et WZA- Français>0.7

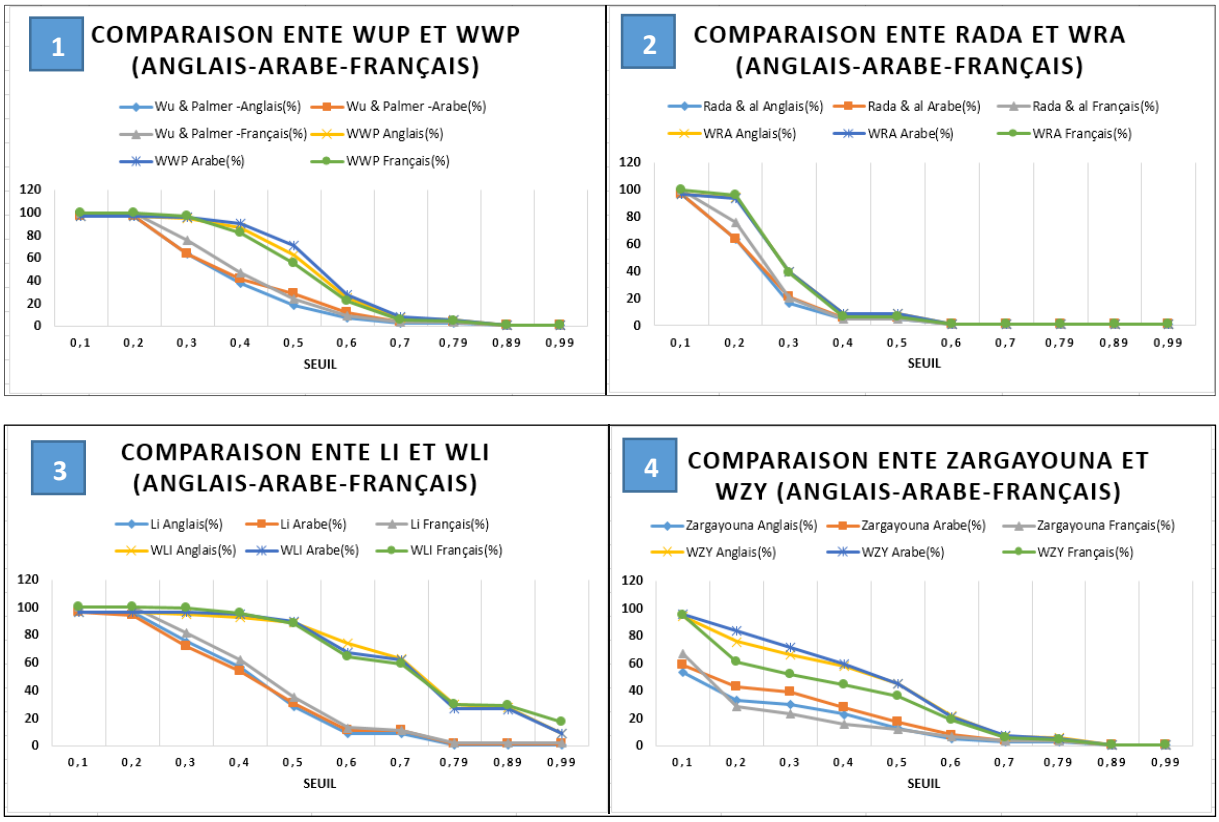


Figure 38. Comparaison entre les approches basées sur la structure avec nos approches hybrides (Anglais-Arabe-Français)

Bien qu'il existe aujourd'hui de nombreuses méthodes d'évaluation, ce sont les méthodes d'évaluation de la cohésion et de la densité qui sont utilisées dans notre travail. Les figures 39 et 40 montrent l'impact de la cohésion et de la densité entre nos mesures hybrides proposées et d'autres méthodes (Wu & Palmer, Rada, Li et Zaragayouna). Nous calculons la densité et la cohésion entre des paires de concepts (Ci, Cj) en traitant ces concepts comme des unités de contexte. En ce qui concerne la densité, nous utilisons la probabilité d'apparition des termes dans ces unités ainsi que les relations sémantiques explicites pour déterminer la densité. En comparant les résultats obtenus, nous remarquons que la densité est plus élevée dans nos mesures hybrides que dans les autres mesures (voir figure 39). Le degré de parenté des concepts OWL, qui sont sémantiquement liés par la parenté des propriétés des éléments dans les ontologies, est appelé cohésion, et nous remarquons que la cohésion dans nos mesures

est plus forte que celle de (Wu & Palmer, Rada, Li et Zaragayouna), comme le montre la figure 40.

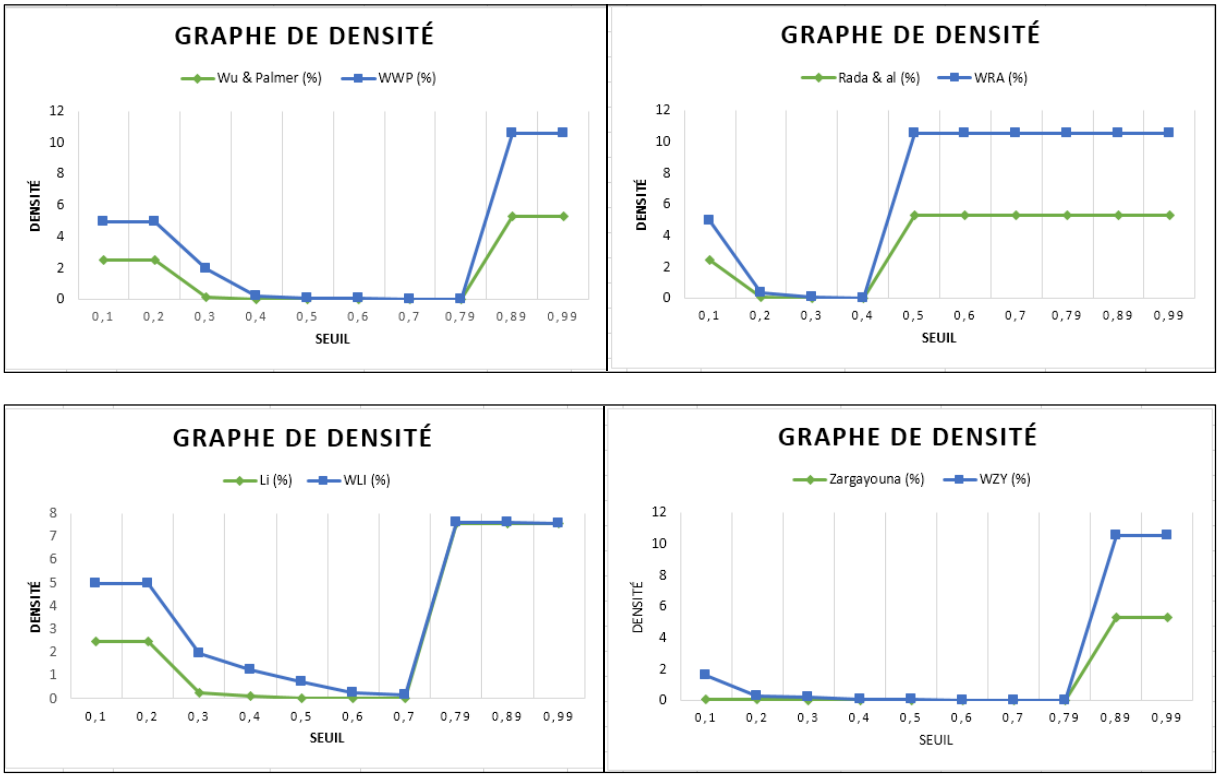


Figure 39. Graphiques d'évaluation de la Densité

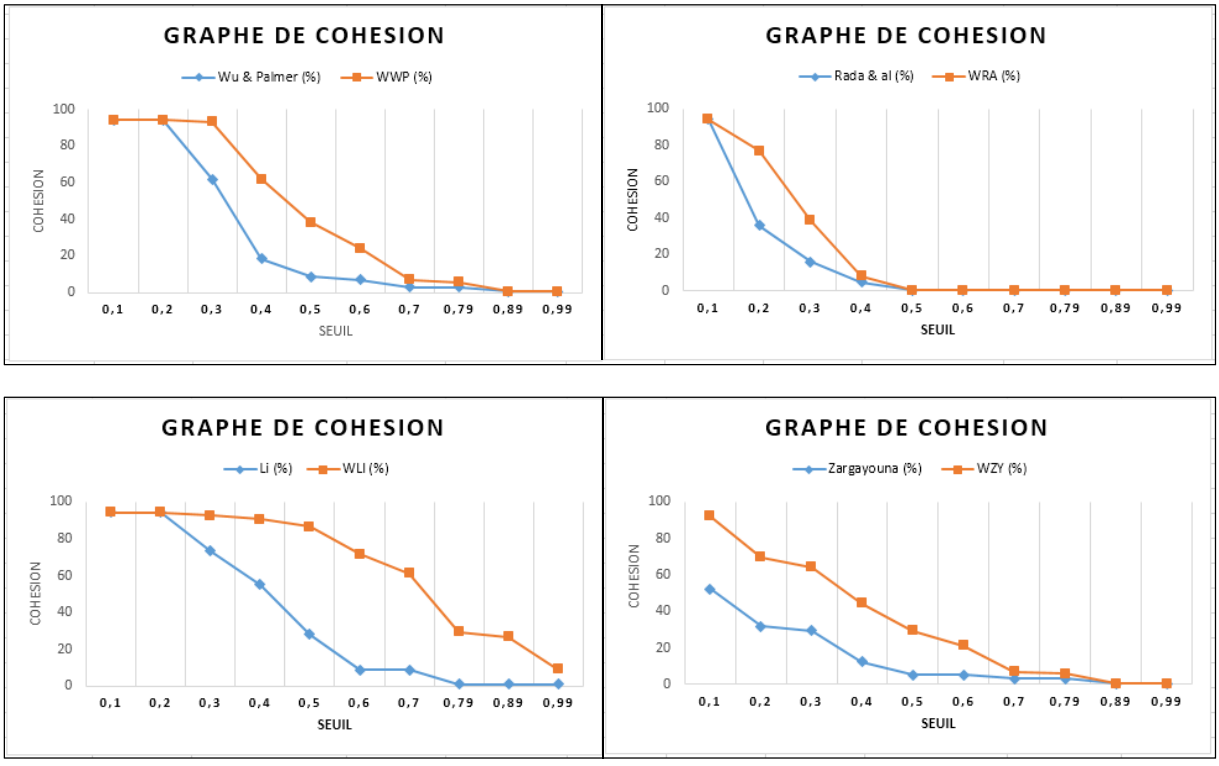


Figure 40. Graphiques d'évaluation de la Cohésion

La comparaison représentée dans la figure 41 confirme que la mesure WLI donne une meilleure cohésion.

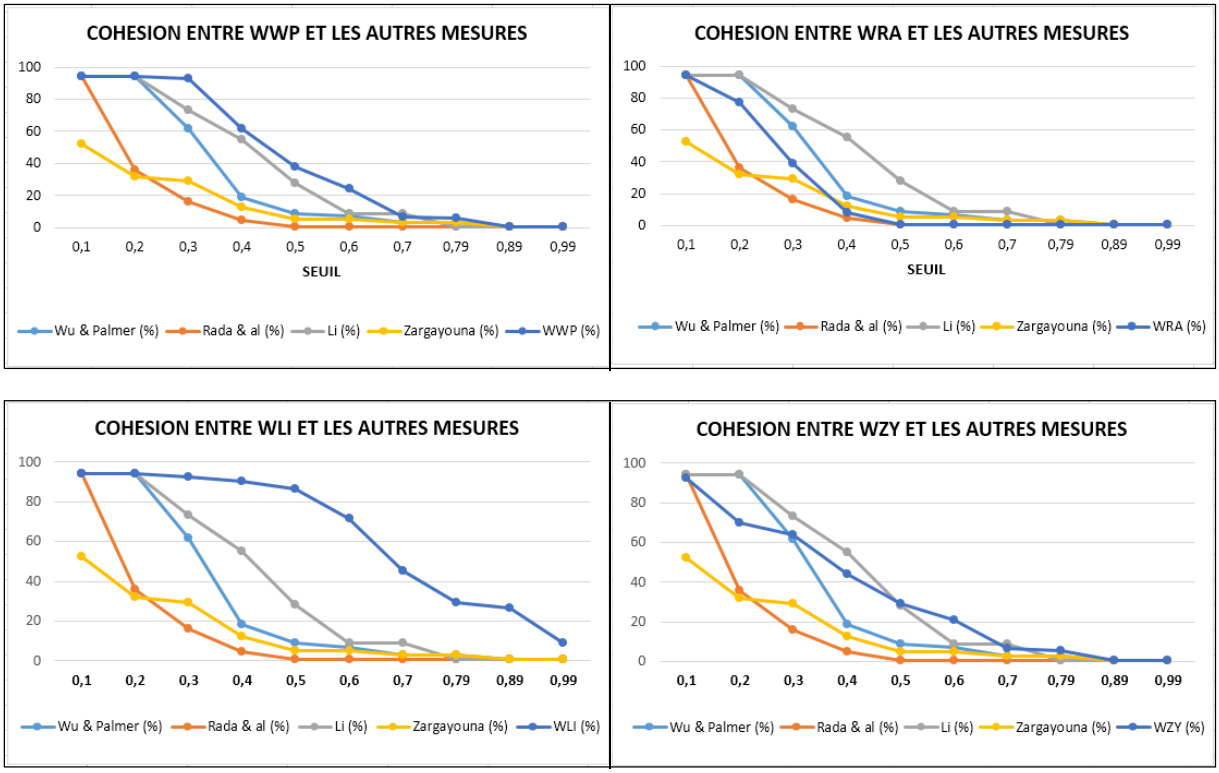


Figure 41. Comparaisons des cohésions entre les approches basées sur la structure avec chaque une de nos approches hybride séparément

5. Conclusion :

Dans ce chapitre, nous avons présenté le cadre applicatif de nos travaux. Ensuite, nous avons présenté et discuté quelques expérimentations mises en place. Le but principal de cette implémentation, est d'évaluer nos propositions, de tester la faisabilité de nos mesures hybrides, et de démontrer que l'approche proposée donne des meilleurs résultats en les comparant avec les mesures existantes, et qu'elle peut être appliquée à plusieurs langues Anglais, Arabe ou Français.

Dans le chapitre suivant nous conclurons notre thèse en ouvrant quelques perspectives et pistes de recherche.

Chapitre 6

Conclusion & Perspectives

1. Conclusion :

Le défi de la mesure de la similarité sémantique entre les concepts est de trouver une méthode qui peut simuler le processus de réflexion de l'homme. L'utilisation d'ordinateurs pour quantifier et comparer les similarités sémantiques est devenue un domaine de recherche important dans divers domaines, notamment l'intelligence artificielle, la gestion des connaissances, la recherche d'informations et le traitement du langage naturel.

Dans cette thèse, dans la première partie, nous nous sommes tous d'abord intéressés à ce qu'était une ontologie. Pour cela, nous avons étudié la notion d'ontologie et les éléments qui la composent. Nous avons présenté les types d'intégration avec les outils les plus en vue. Nous avons exposé les langages de description ainsi que les outils pour l'éditeur d'ontologie. Nous avons présenté le cycle de vie d'une ontologie et les différents moteurs d'inférence de cette dernière. Par la suite nous avons présenté une classification des différentes approches de la mesure de la similarité, nous avons donné un aperçu des mesures de similarité sémantique existant selon cette classification.

Dans la deuxième partie qui était consacrée à notre contribution nous avons présenté une nouvelle approche pour mesurer la similarité sémantique inter-ontologies, qui consiste à hybrider les approches basées sur la structure des ontologies telles que Wu & Palmer, Rada, Li et Zargayouna avec le poids de la similarité calculé à l'aide du

dictionnaire sémantique WordNet. Cette combinaison était nécessaire pour intégrer et renforcer le facteur sémantique. Ensuite une étude expérimentale était réalisée en appliquant notre approche sur différentes mesures, couple d'ontologies et langues (anglais, français et arabe). Nous avons évalué les résultats en utilisant deux métriques, la cohésion et la densité. Nous avons constaté que parmi les différentes mesures hybrides proposées, WLI donnait les meilleurs résultats en les comparants avec les mesures existantes et que ces mesures peuvent être appliquées à la langue arabe.

2. Perspectives :

Nos perspectives et celles de ceux qui s'intéressent à cette recherche se résument en quatre points, qui nous paraissent, importants :

1-Utilisation d'autres approches pour créer le lien entre les ontologies telles que le mapping ou le matching pour appliquer les mesures de similarité sémantique entre plusieurs ontologies.

2- Faire une extension du dictionnaire sémantique arabe « Arabic Wordnet » pour pouvoir l'utiliser pour le calcul du poids de similarité entre les termes arabes directement.

3. Faire une adaptation des mesures proposées entre plusieurs ontologies mais de différents domaines toute en étudiant l'interopérabilité des ontologies.

4. Étudier comment ces mesures influence l'efficacité de la recherche dans les applications de recherche d'informations.

Bibliographie

[**ALI10**] Aliane, H., Alimazighi, Z., & Mazari, A. C. (2010, May). Al—Khalil : The Arabic Linguistic Ontology Project. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).

[**ALY10**] Al-Yahya, M., Al-Khalifa, H., Bahanshal, A., Al-Odah, I., & Al-Helwah, N. (2010). An ontological model for representing semantic lexicons: an application on time nouns in the holy Quran. Arabian Journal for Science and Engineering, 35(2), 21.

[**ARE96**] Arens, Y., Hsu, C. N., & Knoblock, C. A. (1996). Query processing in the SIMS information mediator. Advanced Planning Technology, 32, 78-93.

[**ARP01**] Arpírez, J. C., Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2001, October). WebODE : a scalable workbench for ontological engineering. In Proceedings of the 1st international conference on Knowledge capture, 6-13, https://doi.org/10.1145/500737.500743

[**AUS00**] Aussenac-Gilles, N., Biebow, B., & Szulman, S. (2000, October). Revisiting ontology design : a method based on corpus analysis. In International Conference on Knowledge Engineering and Knowledge Management, 172-188. Springer, Berlin, Heidelberg, https://doi.org/10.1007/3-540-39967-4_13.

[**BEL09**] Belkredim, F. Z., El Sebai, A., & Bouali, U. H. B. (2009). An ontology based formalism for the Arabic language using verbs and their derivatives. Communications of the IBIMA, 11(5), 44-52.

[**BLA06**] Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C. (2006, January). Introducing the Arabic wordnet project. In Proceedings of the third international WordNet conference, 295-300.

[**BOR97**] Borst, P., Akkermans, H., & Top, J. (1997). Engineering ontologies. International journal of human-computer studies, 46(2-3), 365-406, https://doi.org/10.1006/ijhc.1996.0096.

[**BRU06**] Bruijn, J., Ehrig, M., Feier, C., Martín-Recuerda, F., Scharffe, F., & Weiten, M. (2006). Ontology mediation, merging and aligning. Semantic web technologies, 95-113, https://doi.org/10.1002/047003033X.ch6.

[**BUD06**] BUDANITSKY A. & HIRST G. (2006). Evaluating wordnet-based measures of semantic distance. Computational Linguistics, 32(1), 13–47.

[**CAL14**] Calvanese, D., Fischl, W., Pichler, R., Sallinger, E., & Simkus, M. (2014, June). Capturing relational schemas and functional dependencies in RDFS. In Proceedings of the AAAI Conference on Artificial Intelligence, 28 (1), 1003-1011.

[**CHE14**] Cheatham, M., & Hitzler, P. (2014, October). Conference v2. 0: An uncertain version of the oaei conference benchmark. In International Semantic Web Conference, 33-48. Springer, Cham.

[**CHO12**] Choudhari, M. (2012). Extending the hirst and St-Onge measure of semantic relatedness for the unified medical language system. Master Thesis.

[**COR03**] Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point ? Data & knowledge engineering, 46(1), 41-64, https://doi.org/10.1016/S0169-023X(02)00195-7

[**CYG14**] Cyganiak, R., Galway, N., Wood, D., Stones, R., and Lanthaler, M., (2014) 'Rdf 1.1 concepts and abstract syntax'. Technical report, World Wide Web Consortium.

[**DEN13**] Dennai, A., & Benslimane, S. M. (2013, November). Toward an update of a similarity measurement for a better calculation of the semantic distance between ontology concepts. In The second international conference on informatics engineering & information science (ICIEIS2013) ,197-207.

[**DES01**] Desmontils, E., & Jacquin, C. (2001). Des ontologies pour indexer un site web. actes des journées francophones d'Ingénierie des Connaissances.

[**DOU05**] Dou, D., McDermott, D., & Qi, P. (2005). Ontology translation by ontology merging and automated reasoning. In Ontologies for agents: Theory and experiences, 73-94. Birkhäuser Basel. https://doi.org/10.1007/3-7643-7361-X_4.

[**DUK13**] Dukes, K., Atwell, E., & Habash, N. (2013). Supervised collaboration for syntactic annotation of Quranic Arabic. Language resources and evaluation, 47(1), 33-62, https://doi.org/10.1007/s10579-011-9167-7.

[**ELA14**] Elavarasi, S. A., Akilandeswari, J., & Menaga, K. (2014). A survey on semantic similarity measure. International Journal of Research in Advent Technology, 2(3), 389-398. https://doi.org/ 10.4236/ajcm.2015.52017

[**ELK06**] Elkateb, S., Black, W. J., Vossen, P., Farwell, D., Rodríguez, H., Pease, A., & Alkhalifa, M., (2006). Arabic WordNet and the challenges of Arabic. In Proceedings of the International Conference on the Challenge of Arabic for NLP/MT.

[**FAQ15**] Faqihi, B., Daoudi, N., & Ajhoun, R. (2015). From the Collaboration of Companies to the Interoperability of Information Systems (Concepts and Perspectives). International Journal of Information and Education Technology, 5(3), 208. https://doi.org/10.7763/IJIET.

[**FAR03**] Farrar, S., & Langendoen, D. T. (2003). A linguistic ontology for the semantic web. GLOT international, 7(3), 97-100.

[**FAR97**] Farquhar, A., Fikes, R., & Rice, J. (1997). The ontolingua server : A tool for collaborative ontology construction. International journal of human-computer studies, 46(6), 707-727, https://doi.org/10.1006/ijhc.1996.0121.

[**FEN01**] Fensel, D., Van Harmelen, F., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. (2001). OIL : An ontology infrastructure for the semantic web. IEEE intelligent systems, 16(2), 38-45. https://doi.org/10.1109/5254.920598.

[**FER97**] Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In Proceedings of the American Association for the Artificial Intelligence, Springer, 33-40.

[**GAN13**] Gan, M., Dou, X., & Jiang, R. (2013). From ontology to semantic similarity: calculation of ontology-based semantic similarity. The Scientific World Journal, https://doi.org/10.1155/2013/793091.

[**GOH97**] Goh, C. H. (1997). Representing and reasoning about semantic conflicts in heterogeneous information systems (Doctoral dissertation, Massachusetts Institute of Technology).

[**GRU 93**] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220. https://doi.org/10.1006/knac.1993.1008.

[**HAA99**] Haarslev, V., Möller, R., & Turhan, A. Y. (2001). Racer user's guide and reference manual version 1.6. Technical report, University of Hamburg, Computer Science Department.

[**HAL03**] Halkidi, M., Nguyen, B., Varlamis, I., & Vazirgiannis, M. (2003). THESUS: Organizing Web document collections based on link semantics. The VLDB Journal, 12(4), 320-332. https://doi.org/10.1007/s00778-003-0100-6.

[**HIR98**] Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, *305*, 305-332.

[**HOR02**] Horrocks, I. (2002). DAML+OIL: A Description Logic for the Semantic Web. IEEE Data Eng. Bull., 25(1), 4-9.

[**ISH14**] Ishkewy, H., Harb, H., & Farahat, H. (2014). Azhary: An Arabic Lexical Ontology. International Journal of Web & Semantic Technology, 5(4), 71–82, https://doi.org/10.5121/ijwest.2014.5405

[**JAR11**] Jarrar, M. (2011). Building a Formal Arabic Ontology (Invited Paper). in proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks. Alecso, Arab League. Tunis, 26-28.

[**JIA97**] Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the International Conference on Research in Computational Linguistic, Taiwan.

[**KAL06**] Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., & Hendler, J. (2006). Swoop: A web ontology editing browser. Journal of Web Semantics, 4(2), 144-153, https://doi.org/10.1016/j.websem.2005.10.001

[**KAL11**] Kalibatiene, D., & Vasilecas, O. (2011, October). Survey on ontology languages. In International Conference on Business Informatics Research, 124-141. Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-24511-4_10.

[**KHA09**] Khalfi, S., & Zarour, N. (2009). Construction d'une ontologie pour la prise en charge des patients à domicile (Doctoral dissertation, Constantine : Université Mentouri Constantine).

[**KHI15**] [KHI15] Khiat, A., Benaissa, M., & Jiménez-Ruiz, E. (2015). ADOM: arabic dataset for evaluating arabic and cross-lingual ontology alignment systems. OM, 1545, 50-54.

[**KNA03**] Knappe, R., Bulskov, H., Andreasen, T., & Kaynak, O. (2003). On similarity measures for content-based querying. In 10th International Fuzzy Systems Association World Congress, IFSA, 400-403.

[**LAN15**] Lantow, B., & Sandkuhl, K. (2015, November). From Visual Language to Ontology Representation: Using OWL for Transitivity Analysis in 4EM. In PoEM (Short Papers), 51-60.

[**LEA98**] Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. WordNet: An electronic lexical database, 49(2), 265-283.

[**LI 03**] Li, Y., Bandar, Z. A., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on knowledge and data engineering, 15(4), 871-882. https://doi.org/ 10.1109/TKDE.2003.1209005

[**LIN93**] Lin, D. (1993, June). Principle-based parsing without overgeneration. In 31st annual meeting of the association for computational linguistics, 112-120.

[**LIN98**] Lin, D. (1998, July). An information-theoretic definition of similarity. In *Icml*, 98(1998), 296-304, Kaufmann, San Francisco USA.

[**MAL11**] Malviya, N., Mishra, N., & Sahu, S. (2011). Developing university ontology using protégé owl tool : Process and reasoning. International Journal of Scientific & Engineering Research, 2(9), 1-8.

[**MAZ12**] Mazari, A. C., Aliane, H., & Alimazighi, Z. (2012). Automatic Construction of Ontology from Arabic Texts. Proceeding of ICWIT, 193-202.

[**MCG00**] McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). The chimaera ontology environment. AAAI/IAAI, 2000, 1123-1124.

[**MEL07**] Mellal, N. (2007). Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information, (Doctoral dissertation, Chambéry).

[**MEN00**] Mena, E., Illarramendi, A., Kashyap, V., & Sheth, A. P. (2000). OBSERVER : An approach for query processing in global information systems based on interoperation across pre-existing ontologies. Distributed and parallel Databases, 8(2), 223-271. https://doi.org/10.1023/A:1008741824956

[**MIL90**] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet : An on-line lexical database. International journal of lexicography, 3(4), 235-244, https://doi.org/10.1093/ijl/3.4.235

[**MIL95**] Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41, https://doi.org/10.1145/219717.219748

[**MIL98**] Miller, G. A. (1998). WordNet: An electronic lexical database. MIT press.

[**MIT02**] Mitra, P., & Wiederhold, G. (2002, July). Resolving terminological heterogeneity in ontologies. In Proceedings of the ECAI workshop on Ontologies and Semantic Interoperability, 45-50.

[**NGU06**] Nguyen, H. A., & Al-Mubaid, H. (2006, May). New ontology-based semantic similarity measure for the biomedical domain. In 2006 IEEE International Conference on Granular Computing, 623-628. IEEE. https://doi.org/10.1109/GRC.2006.1635880.

[**NOY00**] Noy, N. F., & Musen, M. A. (2000, August). Algorithm and tool for automated ontology merging and alignment. In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00), 450-455.

[**NOY01**] Noy, N. F., & Musen, M. A. (2001, January). Anchor-PROMPT: Using non-local context for semantic matching. Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence, Seattle, 63-70.

[**NOY03**] Noy, N. F., & Musen, M. A. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. International journal of human-computer studies, 59(6), 983-1024. http://dx.doi.org/10.1016/j.ijhcs.2003.08.002.

[**PEA10**] Pease, A., & Benzmüller, C. (2010). Sigma : an integrated development environment for logical theories. 19 th EuropeanConferenceon ArtificialIntelligence, 7.

[**PET06**] Petrakis, E. G., Varelas, G., Hliaoutakis, A., & Raftopoulou, P. (2006). X-similarity: Computing semantic similarity between concepts from different ontologies. Journal of Digital Information Management, 4(4).

[**RAD89**] Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. IEEE transactions on systems, man, and cybernetics, 19(1), 17-30, https://doi.org/10.1109/21.24528.

[**RAH01**] Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. The VLDB Journal, 10(4), 334-350, https://doi.org/10.1007/s007780100057

[**RES99**] Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of artificial intelligence research, 11, 95-130. https://doi.org/10.1613/jair.514.

[**ROD03**] Rodriguez, M. A., & Egenhofer, M. J. (2003). Determining semantic similarity among entity classes from different ontologies. IEEE transactions on knowledge and data engineering, 15(2), 442-456. https://doi.org/10.1109/TKDE.2003.1185844.

[**SAN13**] Sánchez, D., & Batet, M. (2013). A semantic similarity method based on information content exploiting multiple ontologies. Expert Systems with Applications, 40(4), 1393-1399. https://doi.org/10.1016/j.eswa.2012.08.049

[**SAR11**] Saruladha, K., Aghila, G., & Bhuvaneswary, A. (2011, June). COSS: Cross Ontology Semantic Similarity measure—An information content based approach. In 2011 International Conference on Recent Trends in Information Technology (ICRTIT), 485-490. IEEE. https://doi.org/10.1109/ICRTIT.2011.5972360.

[**SCH05**] Schnurr, H. P., & Angele, J. (2005, November). Do not use this gear with a switching lever ! Automotive industry experience with semantic guides. In International Semantic Web Conference 1029-1040. Springer, Berlin, Heidelberg.

[**SCH94**] Schmid, H., (1994). Probabilistic part-of-speech tagging using decision trees. In Proceedings of International Conference on New Methods in Language Processing.

[**SHV05**] Shvaiko, P., & Euzenat, J. (2005). A survey of schema-based matching approaches. In Journal on data semantics IV, 146-171. Springer, Berlin, Heidelberg, https://doi.org/10.1007/11603412_5.

[**SIR07**] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet : A practical owl-dl reasoner. Journal of Web Semantics, 5(2), 51-53. https://doi.org/10.1016/j.websem.2007.03.004.

[**SLI06**] Slimani, T., Yaghlane, B. B., & Mellouli, K. (2006). A new similarity measure based on edge counting. World academy of science, engineering and technology, 23(2006), 34-38.

[**SUR02**] Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., & Wenke, D. (2002, June). OntoEdit : Collaborative ontology development for the semantic web. In International semantic web conference, 221-235. Springer, Berlin, Heidelberg, https://doi.org/10.1007/3-540-48005-6_18.

[**TVE77**] Tversky, A. 1977. Features of Similarity. Psycological Review, 84(4):327-352

[**VIS00**] Visser, U., Stuckenschmidt, H., Wache, H., & Vögele, T. (2000). Enabling technologies for interoperability. In Workshop on the 14th International Symposium of Computer Science for Environmental Protection, 35-46.

[**WAC99**] Wache, H., Scholz, T., Stieghahn, H., & Konig-Ries, B. (1999, November). An integration method for the specification of rule-oriented mediators. In Proceedings 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), 109-112. IEEE. https://doi.org/10.1109/DANTE.1999.844947.

[**WUP94**] Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. In Proceeding of the 32nd annual meeting on Association for Computational Linguistics, 133–138, 1994, https://doi.org/10.3115/981732.981751

[**YAZ19**] Yazid, B., Mourad, O., & Abdelmalik, T. (2019). Semantic similarity approach between two sentences. In Proceedings of the 5th International Conference on the Image and Signal Processing and their Applications.

[**ZAR04**] Zargayouna, H. (2004). Contexte et sémantique pour une indexation de documents semi-structurés. CORIA, 4, 161-177.

[**ZHO02**] Zhong, J., Zhu, H., Li, J., & Yu, Y. (2002, July). Conceptual graph matching for semantic search. In International conference on conceptual structures, 92-106. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45483-7_8.

[**ZHO08**] Zhou, Z., Wang, Y., & Gu, J. (2008, November). New model of semantic similarity measuring in wordnet. In 2008 3rd International Conference on Intelligent System and Knowledge Engineering (Vol. 1, pp. 256-261). IEEE. https://doi.org/ 10.1109/ISKE.2008.4730937.

Annexes

Annexe A

Cette Annexe contient le code source de Google translate API, c'est la version utiliser dans notre travail.

```
/*****************************************************************************************
*********************
 * An API for a Google Translation service in Java.
 * The translator allows for language detection and translation.
 * Recommended for translation of user interfaces or speech commands.
 * All translation services provided via Google Translate
 * @author Aaron Gokaslan (Skylion)

  ****************************************************************************************
  ******************/
public final class GoogleTranslate { //Class marked as final since all methods are static
  /**
   * URL to query for Translation
   */
private final static String GOOGLE_TRANSLATE_URL =
"http://translate.google.com/translate_a/single";
  /**
   * Private to prevent instantiation
   */
  private GoogleTranslate(){};

  /**
   * Converts the ISO-639 code into a friendly language code in the user's default language
   * For example, if the language is English and the default locale is French, it will return "anglais"
   * Useful for UI Strings
   * @param languageCode The ISO639-1
   * @return The language in the user's default language
   */
  public static String getDisplayLanguage(String languageCode){
        return (new Locale(languageCode)).getDisplayLanguage();
  }

  /** Completes the complicated process of generating the URL
   * @param sourceLanguage The source language
   * @param targetLanguage The target language
   * @param text The text that you wish to generate
   * @return The generated URL as a string.
   */
  private static String generateURL(String sourceLanguage, String targetLanguage, String
  text)throws UnsupportedEncodingException{
        String encoded = URLEncoder.encode(text, "UTF-8"); //Encode
```

```
        StringBuilder sb = new StringBuilder();
        sb.append(GOOGLE_TRANSLATE_URL);
        sb.append("?client=webapp"); //The client parameter
        sb.append("&hl=en"); //The language of the UI?
        sb.append("&sl="); //Source language
        sb.append(sourceLanguage);
        sb.append("&tl="); //Target language
        sb.append(targetLanguage);
        sb.append("&q=");
        sb.append(encoded);
        sb.append("&multires=1");//Necessary but unknown parameters
        sb.append("&otf=0");
        sb.append("&pc=0");
        sb.append("&trs=1");
        sb.append("&ssel=0");
        sb.append("&tsel=0");
        sb.append("&kc=1");
        sb.append("&dt=t");//This parameters requests the translated text back.
        //Other dt parameters request additional information such as pronunciation, and so on.
        //TODO Modify API so that the user may request this additional information.
        sb.append("&ie=UTF-8"); //Input encoding
        sb.append("&oe=UTF-8"); //Output encoding
        sb.append("&tk="); //Token authentication parameter
        sb.append(generateToken(text));
       return sb.toString();
}

/**
 * Automatically determines the language of the original text
 * @param text represents the text you want to check the language of
 * @return The ISO-639 code for the language
 * @throws IOException if it cannot complete the request
 */
public static String detectLanguage(String text) throws IOException{
        String urlText = generateURL("auto", "fr", text);
        URL url = new URL(urlText); //Generates URL
        String rawData = urlToText(url);//Gets text from Google
        return findLanguage(rawData);
}

/**
 * Automatically translates text to a system's default language according to its locale
 * Useful for creating international applications as you can translate UI strings
 * @see GoogleTranslate#translate(String, String, String)
 * @param text The text you want to translate
 * @return The translated text
 * @throws IOException if cannot complete request
 */
public static String translate(String text) throws IOException{
        return translate(Locale.getDefault().getLanguage(), text);
```

```
}
/**
 * Automatically detects language and translate to the targetLanguage.
 * Allows Google to determine source language
 * @see GoogleTranslate#translate(String, String, String)
 * @param targetLanguage The language you want to translate into in ISO-639 format
 * @param text The text you actually want to translate
 * @return The translated text.
 * @throws IOException if it cannot complete the request
 */
public static String translate(String targetLanguage, String text) throws IOException{
	return translate("auto",targetLanguage, text);
}

/**
 * Translate text from sourceLanguage to targetLanguage
 * Specifying the sourceLanguage greatly improves accuracy over short Strings
 * @param sourceLanguage The language you want to translate from in ISO-639 format
 * @param targetLanguage The language you want to translate into in ISO-639 format
 * @param text The text you actually want to translate
 * @return the translated text.
 * @throws IOException if it cannot complete the request
 */
public static String translate(String sourceLanguage, String targetLanguage, String text)
throws IOException{
	String urlText = generateURL(sourceLanguage, targetLanguage, text);
	URL url = new URL(urlText);
	String rawData = urlToText(url);//Gets text from Google
	if(rawData==null){
		return null;
	}
	String[ ] raw =  rawData.split("\"");//Parses the JSON
	if(raw.length<2){
		return null;
	}
	return raw[1];//Returns the translation
}

/**
 * Converts a URL to Text
 * @param url that you want to generate a String from
 * @return The generated String
 * @throws IOException if it cannot complete the request
 */
private static String urlToText(URL url) throws IOException{
	URLConnection urlConn = url.openConnection(); //Open connection
	//Adding header for user agent is required. Otherwise, Google rejects the request
	urlConn.addRequestProperty("User-Agent", "Mozilla/5.0 (Windows NT 6.1; WOW64;
rv:2.0) Gecko/20100101 Firefox/4.0");
	Reader r = new java.io.InputStreamReader(urlConn.getInputStream(),
Charset.forName("UTF-8"));//Gets Data Converts to string
```

```
        StringBuilder buf = new StringBuilder();
        while (true) {//Reads String from buffer
                int ch = r.read();
                if (ch < 0)
                        break;
                buf.append((char) ch);
        }
        String str = buf.toString();
        return str;
}

/**
 * Searches RAWData for Language
 * @param RAWData the raw String directly from Google you want to search through
 * @return The language parsed from the rawData or en-US (English-United States) if Google cannot
determine it.
 */
private static String findLanguage(String rawData){
        for(int i = 0; i+5<rawData.length(); i++){
                boolean dashDetected = rawData.charAt(i+4)=='-';
                if(rawData.charAt(i)==','  && rawData.charAt(i+1)== '"'
                                && ((rawData.charAt(i+4)=='"' &&
rawData.charAt(i+5)==',')
                                                || dashDetected)){
                        if(dashDetected){
                                int lastQuote = rawData.substring(i+2).indexOf('"');
                                if(lastQuote>0)
                                        return rawData.substring(i+2,i+2+lastQuote);
                        }
                        else{
                                String possible = rawData.substring(i+2,i+4);
                                if(containsLettersOnly(possible)){//Required due to Google's
inconsistent formatting.
                                        return possible;
                                }
                        }
                }
        }
        return null;
}

/**
 * Checks if all characters in text are letters.
 * @param text The text you want to determine the validity of.
 * @return True if all characters are letter, otherwise false.
 */
private static boolean containsLettersOnly(String text){
        for(int i = 0; i<text.length(); i++){
                if(!Character.isLetter(text.charAt(i))){
                        return false;
```

```
                }
        }
        return true;
}

/************************** Cryptography section ************************/


//TODO Possibly refactor code as utility class

/**
 * This function generates the int array for translation acting as the seed for the hashing
algorithm.
 */
private static int[] TKK() {
        int[ ] tkk = { 0x6337E, 0x217A58DC + 0x5AF91132};
        return tkk;
}

/**
 * An implementation of an unsigned right shift.
 * Necessary since Java does not have unsigned ints.
 * @param x The number you wish to shift.
 * @param bits The number of bytes you wish to shift.
 * @return The shifted number, unsigned.
 */
private static int shr32(int x, int bits) {
        if (x < 0) {
                long x_l = 0xffffffffl + x + 1;
                return (int) (x_l >> bits);
        }
        return x >> bits;
}

private static int RL(int a, String b) {//I am not entirely sure what this magic does.
        for (int c = 0; c < b.length() - 2; c += 3) {
                int d = b.charAt(c + 2);
                d = d >= 65 ? d - 87 : d - 48;
                d = b.charAt(c + 1) == '+' ? shr32(a, d) : (a << d);
                a = b.charAt(c) == '+' ? (a + (d & 0xFFFFFFFF)) : a ^ d;
        }
        return a;
}

/**
 * Generates the token needed for translation.
 * @param text The text you want to generate the token for.
 * @return The generated token as a string.
 */
private static String generateToken(String text) {
```

```
        int tkk[ ] = TKK();
        int b = tkk[0];
        int e = 0;
        int f = 0;
        List<Integer> d = new ArrayList<Integer>();
        for (; f < text.length(); f++) {
                int g = text.charAt(f);
                if (0x80 > g) {
                        d.add(e++, g);
                } else {
                        if (0x800 > g) {
                                d.add(e++, g >> 6 | 0xC0);
                        } else {
                                if (0xD800 == (g & 0xFC00) && f + 1 < text.length() &&
                                                0xDC00 == (text.charAt(f + 1) & 0xFC00)) {
                                        g = 0x10000 + ((g & 0x3FF) << 10) +
  (text.charAt(++f) & 0x3FF);
                                        d.add(e++, g >> 18 | 0xF0);
                                        d.add(e++, g >> 12 & 0x3F | 0x80);
                                } else {
                                        d.add(e++, g >> 12 | 0xE0);
                                        d.add(e++, g >> 6 & 0x3F | 0x80);
                                }
                        }
                        d.add(e++, g & 63 | 128);
                }
        }

        int a_i = b;
        for (e = 0; e < d.size(); e++) {
                a_i += d.get(e);
                a_i = RL(a_i, "+-a^+6");
        }
        a_i = RL(a_i, "+-3^+b+-f");
        a_i ^= tkk[1];
        long a_l;
        if (0 > a_i) {
                a_l = 0x80000000l + (a_i & 0x7FFFFFFF);
        } else {
                a_l = a_i;
        }
        a_l %= Math.pow(10, 6);
        return String.format(Locale.US, "%d.%d", a_l, a_l ^ b);
  }
//------------------------------------------------------------------------
      public static String translateArabe(String txt)
      {
          String arabe="";
           try {
```

```
            System.out.println("Entrer : "+txt);
            //System.out.println("Translated text: " +translate("ar",text));
            arabe = translate("ar",txt);
            System.out.println("Sortie : "+txt);

        } catch (IOException ex) {
            Logger.getLogger(GoogleTranslate.class.getName()).log(Level.SEVERE, null, ex);
        }

        return arabe;
    }
    //----------------------------------------------------------------------

    public static String TranslateArabe(String text)
    {
       String srt ="";
        try {
             srt = translate("ar",text);
             //-------------------------------------------------------


        } catch (IOException ex) {
            Logger.getLogger(GoogleTranslate.class.getName()).log(Level.SEVERE, null, ex);
        }
        return srt;
    }
}
```

Annexe B

TreeTagger : Comment lemmatiser une chaîne de caractères ?

B.1. Comprendre la lemmatisation :

Les mots (lemmes) d'une langue peuvent prendre plusieurs états en fonction de leur genre (masculin ou féminin), leur nombre (un ou plusieurs), leur personne (moi, toi, eux…), leur mode (indicatif, impératif…) donnant ainsi naissance à plusieurs formes pour un même lemme. La lemmatisation d'une forme d'un mot consiste à en prendre sa forme canonique. Celle-ci est définie comme suit :

- Pour un verbe : ce verbe à l'infinitif.
- Pour les autres mots : le mot au masculin singulier.

Par exemple, l'adjectif petit existe sous quatre formes : petit, petite, petits et petites. La forme canonique de tous ces mots est "petit". Il existe beaucoup plus de formes du verbe avoir : ai, as, a, avons, ais, avons eu, ayez eu, eussions eu, aurions eu, etc. La forme canonique de eussions eu est avoir.

Supposons que nous devons visualiser la fréquence d'apparition de certains mots-clés contenus dans le texte suivant :

« **Nous aurions dû prendre le train depuis la gare d'Oran pour rentrer chez nous, mais tous les trains provenant d'Alger étaient en retard de 5 heures ! Donc nous avons décidé de laisser tomber les trains. Nous avons ainsi fait du covoiturage jusqu'à Sidi Bel Abbes, à une heure de chez nous, et mes parents sont venus nous chercher** ».

Combien de fois il y a les mots "train" et "heure" dans ce texte ? Une seule fois, pour chacun des deux. Oui, car les deux sont présents dans leurs formes singulières et plurales : train (1 fois) et trains (2 fois), heure (1 fois) et heures (1 fois).

Afin de bien déterminer la fréquence des mots contenus dans ce texte, nous devons tout d'abord les transformer dans leur lemme. Grâce à ce type de traitement, nous arrivons ainsi à bien comptabiliser 3 fois, le mot "train" et 2 fois le mot "heure".

B.2. Qu'est-ce que c'est TreeTagger ?

TreeTagger est un logiciel de lemmatisation développé par Helmut Schmid [SCH94]. TreeTagger permet l'étiquetage de l'Allemand, l'Anglais, le Français, l'Italien, le Deutsch, l'Espagnol, le Bulgare, Le Russe, le Grec, le Portugais, le chinois et les textes français anciens. Il est adaptable à d'autres langages si des lexiques et des corpus étiquetés manuellement sont disponibles.

Il fonctionne selon deux modes :

1. Le premier mode apprend un modèle linguistique de lemmatisation à partir d'un corpus d'apprentissage et d'un lexique morphologique (composé de triplets 'forme graphique du mot / catégorie / lemme').
2. Le second mode projette un modèle linguistique sur un texte brut pour lemmatiser ses mots.

Nous l'utilisons parce qu'il dispose d'une grande quantité de modèles linguistiques différents pour lemmatiser beaucoup de langues.

Il s'agit d'un étiqueteur de type probabiliste, qui utilise un arbre de décision qui a pour feuilles des listes de probabilités et des dictionnaires de référence (un dictionnaire par défaut, un de suffixes et un des mots étiquetés dans la phase d'entraînement).

Son output complet contient trois colonnes (voir Table 4) :

1. Le mot.
2. L'étiquette : on a deux options, soit la meilleure étiquette (Best tag) soit toutes les étiquettes avec probabilité minimum, paramétrable par l'utilisateur. On peut afficher ou pas la probabilité de ces étiquettes alternativement.

3. Le lemme.

Le mot	L'étiquette	Le lemme
On	PRO :PER	On
ne	ADV	ne
change	VER : pres	changer
pas	ADV	pas
une	DET : ART	un
équipe	NOM	équipe
qui	PRO : REL	qui
gagne	VER : pres	gagner

Table 4. Exemple d'étiquetage d'une phrase à l'aide de TreeTagger.

B.3.Comment utiliser TreeTagger ?

TreeTagger peut être directement utilisé depuis le terminal, mais aussi dans de nombreux langages comme Python, Java, R ou encore PHP.

Dans notre cas, nous avons travaillé avec le langage Java, nous avons utilisé TreeTagger avec le package TT4J (TreeTagger for JAVA), sa classe principale est TreeTaggerWrapper (voir figure 42). Un processus TreeTagger sera créé et maintenu pour chaque instance de cette classe. Le processus associé sera interrompu et redémarré automatiquement si le modèle est modifié (setModel(String)). Sinon, le processus reste en cours d'exécution, en arrière-plan une fois qu'il est lancé, ce qui permet de gagner beaucoup de temps. Pendant l'analyse, deux threads sont utilisés pour communiquer avec le TreeTagger. Un processus écrit les tokens au processus TreeTagger, tandis que l'autre reçoit les tokens analysés.

- **Analyse des tokens :**

  Pour une intégration facile dans l'application, cette classe prend n'importe quel objet contenant des informations sur le token et utilise soit sa méthode toString(), soit un TokenAdapter défini à l'aide de setAdapter(TokenAdapter)

pour extraire le token réel. Pour recevoir le token analysé, il faut définir un TokenHandler personnalisé en utilisant setHandler(TokenHandler).

- **Obtenir des probabilités :**

  TT4J permet d'extraire des probabilités de TreeTagger. Pour utiliser cette fonctionnalité, un TokenHandler implémentant l'interface ProbabilityHandler doit être passé à setHandler(TokenHandler) et un seuil de probabilité doit être défini à l'aide de setProbabilityThreshold(Double). Cela correspond à la spécification des arguments -prob -threshold <value> de TreeTagger. Pendant le traitement, TT4J invoque d'abord TokenHandler.token() avec le token, le meilleur tag et le meilleur lemme. Ensuite, ProbabilityHandler.probability() est invoqué pour chaque étiquette/lemme/probabilité retourné par TreeTagger.

```
public List<String> tag(String str) {
    final List<String> tagLemme = new ArrayList<String>();
    String[] tokens =tokenizer.tokenize(str);
      System.setProperty("treetagger.home", "parametresTreeTagger/TreeTagger");
    TreeTaggerWrapper tt = new TreeTaggerWrapper<String>();
    try {
        tt.setModel("parametresTreeTagger/english/english.par");
        tt.setHandler(new TokenHandler<String>(){
                public void token(String token, String pos, String lemma) {
                        tagLemme.add(token + "_" + pos + "_" + lemma);
                        //System.out.println(token + "_" + pos + "_" + lemma);
                }
        });
        tt.process(asList(tokens));
     } catch (IOException e) {
        e.printStackTrace();
      } catch (TreeTaggerException e) {
        e.printStackTrace();
    }
finally {
        tt.destroy();
}
    return tagLemme;
}
```

Figure 42. Exemple de code Java utilisant TreeTagger

B.4. Les étiquette de TreeTagger

L'étiquetage morphologique se réalise par le biais d'un catégoriseur, un outil qui a comme input des phrases et comme output les mots accompagnés par des étiquettes

qui précisent leur catégorie grammaticale. Les étiquettes applicables sont définies dans des jeux d'étiquettes (tag-set) qui contiennent la liste des catégories grammaticales (nom, verbe, adjectif, etc). Dans ce qui suit nous présentant des exemples d'étiquettes pour la langue anglaise et française.

B.4.1. Pour la langue anglaise : The penn Treebank

Tagset

CC Coordinating conjunction (and,but,or...)
CD Cardinal Number
DT Determiner
EX Existential *there*
FW Foreign Word
IN Preposition or subordinating conjunction
JJ Adjective
JJR Adjective, comparative
JJS Adjective, superlative
LS List Item Marker
MD Modal (can, could, might, may...)
NN Noun, singular or mass
NNP Proper Noun, singular
NNPS Proper Noun, plural
NNS Noun, plural
PDT Predeterminer (all, both ... when they precede an article)
POS Possessive Ending (Nouns ending in 's)
PRP Personal Pronoun (I, me, you, he...)
PRP$ Possessive Pronoun (my, your, mine, yours...)
RB Adverb (Most words that end in -ly as well as degree words like quite, too and very)
RBR Adverb, comparative (Adverbs with the comparative ending -er, with a strictly comparative meaning)
RBS Adverb, superlative
RP Particle
SYM Symbol (Should be used for mathematical, scientific or technical symbols)
TO *to*
UH Interjection (uh, well, yes, my...)
VB Verb, base form (subsumes imperatives, infinitives and subjunctives)
VBD Verb, past tense (includes the conditional form of the verb to be)
VBG Verb, gerund or persent participle
VBN Verb, past participle
VBP Verb, non-3rd person singular present
VBZ Verb, 3rd person singular present
WDT Wh-determiner (which, and *that* when it is used as a relative pronoun)
WP Wh-pronoun (what, who, whom...)
WP$ Possessive wh-pronoun **(**w, where why)
WRB Wh-adverb ho

B.4.2. Punctuation Tags:

#
$
''

(
)
,
.
:
``

B.4.3. Pour la langue française :

Tagset

```
ABR Abreviation
ADJ Adjectif
ADV Adverbe
DET:ART Article
DET:POS Pronom Possessif (ma, ta, ...)
INT Interjection
KON Conjunction
NAM Nom Propre
NOM Nom
NUM Numéral
PRO Pronom
PRO:DEM Pronom Démonstratif
PRO:IND Pronom Indefini
PRO:PER Pronom Personnel
PRO:POS Pronom Possessif (mien, tien, ...)
PRO:REL Pronom Relatif
PRP Préposition
PRP:det Préposition + Article (au,du,aux,des)
PUN Ponctuation
PUN:cit Ponctuation de citation
SENT Balise de phrase
SYM Symbole
VER:cond Verbe au conditionnel
VER:futu Verbe au futur
VER:impe Verbe à l'impératif
VER:impf Verbe à l'imparfait
VER:infi Verbe à infinitif
VER:pper Verbe au participe passé
VER:ppre Verbe au participe présent
VER:pres Verbe au présent
VER:simp Verbe au passé simple
VER:subi Verbe à l'imparfait du subjunctif
VER:subp Verbe au présent du subjunctif
```

Annexe C

Quelques interfaces de notre implémentation

Dans cette Annexe, nous présentons quelques interfaces de l'implémentation de notre approche.

Après le chargement du fichier .owl de nos deux ontologies [KHI15] (voir figure 43), on lance le module d'analyse et d'extraction d'ontologie qui permet d'extraire les informations utiles de notre ontologie telle que les classes, les références (concepts) et les relations (voir figure 44 et figure 45), ce test est fait sur deux ontologies **Conférence-ar** contenant 62 concepts, 165 relations et **Sigkdd-ar** contenant 51 concepts, 106 relations.

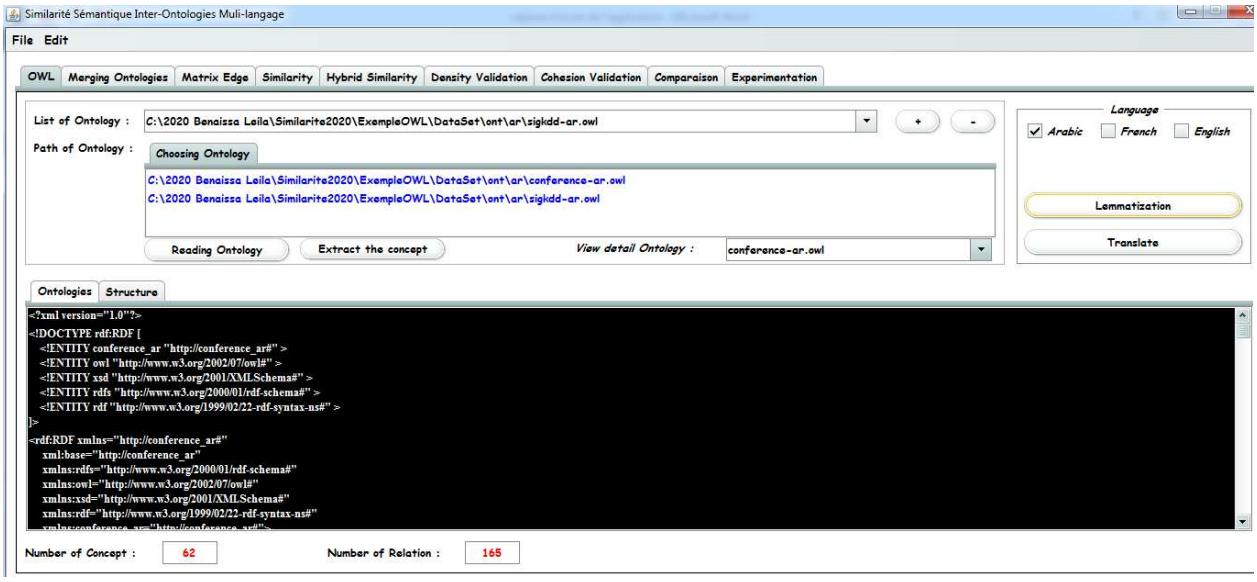


Figure 43. Chargement de l'ontologie

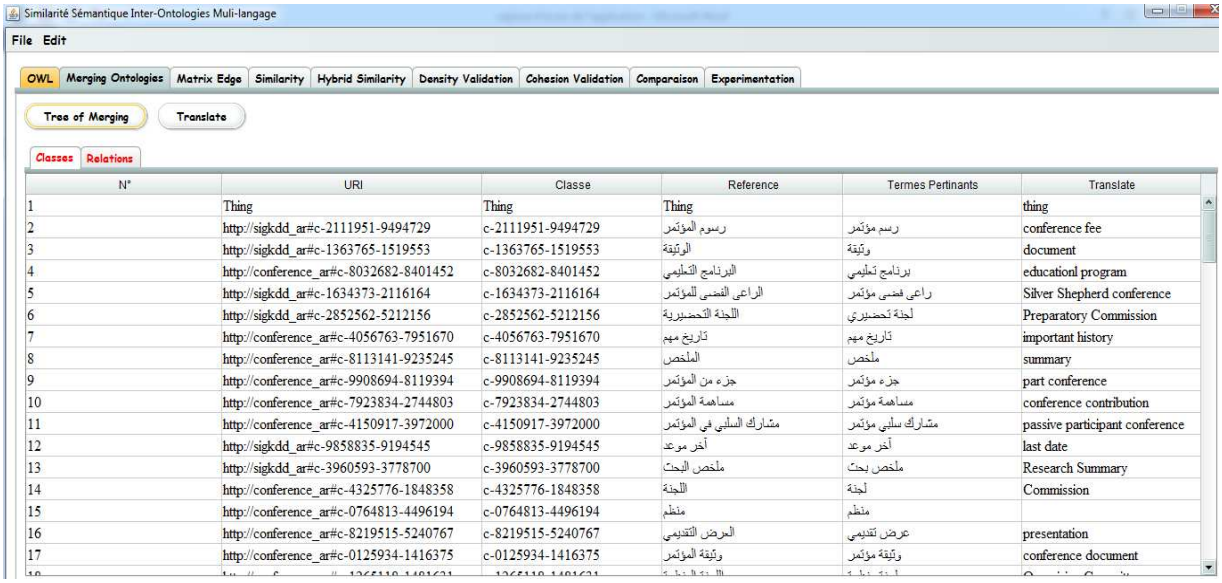


Figure 44. Interface d'analyse et d'extraction de nos ontologies (extraction des concepts)

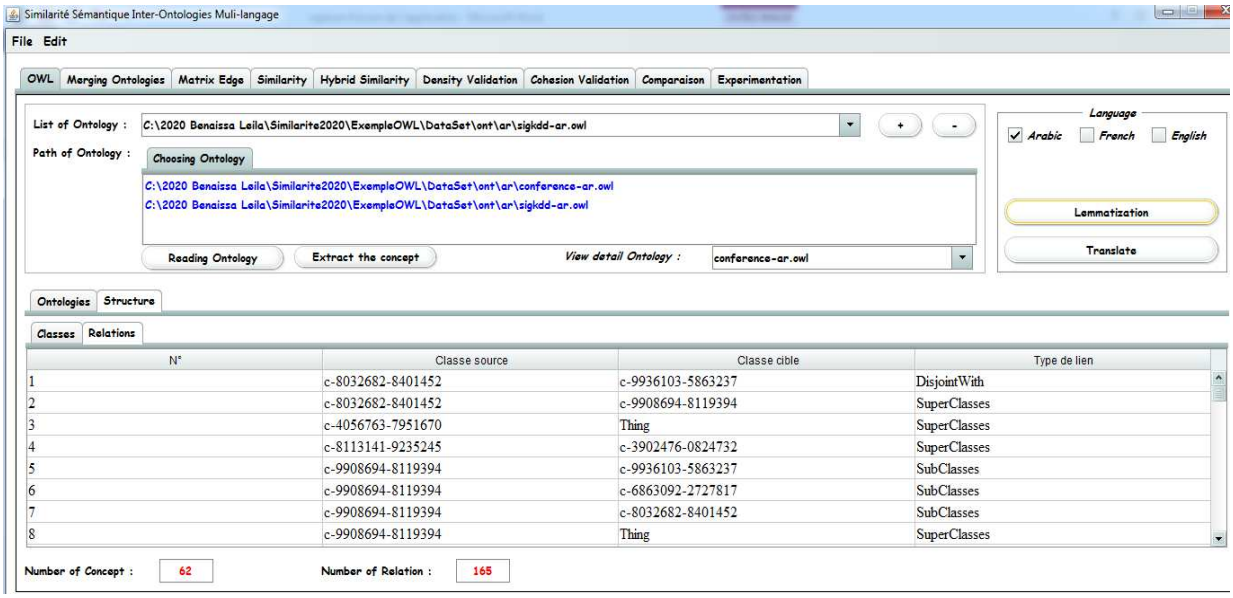


Figure 45. Interface de type de relation entre concepts

L'étape suivante consiste à fusionner les deux ontologies, la figure suivante montre les graphes de nœuds des deux ontologies choisies et le graphe de leur fusion (voir figure 46). Par la suite en calcule la matrice d'incidence sur OWL_3 ontologie de fusion entre OWL_1 (ontologie conférence) et OWL_2 (ontologie Sigkdd) voir figure 47.

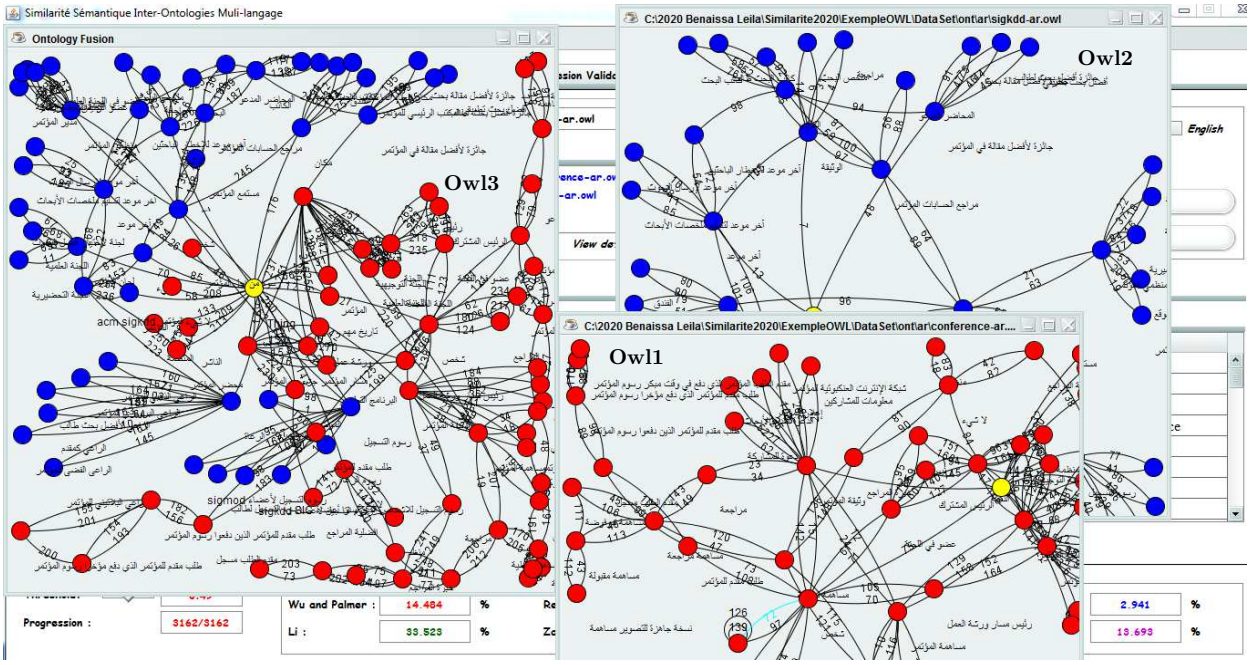
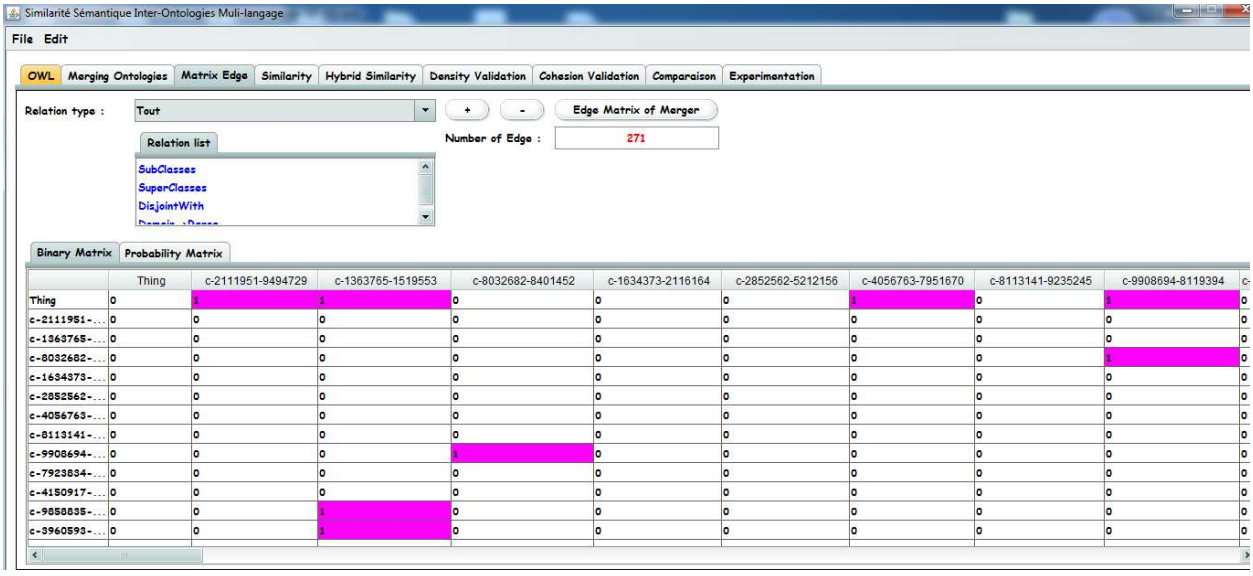


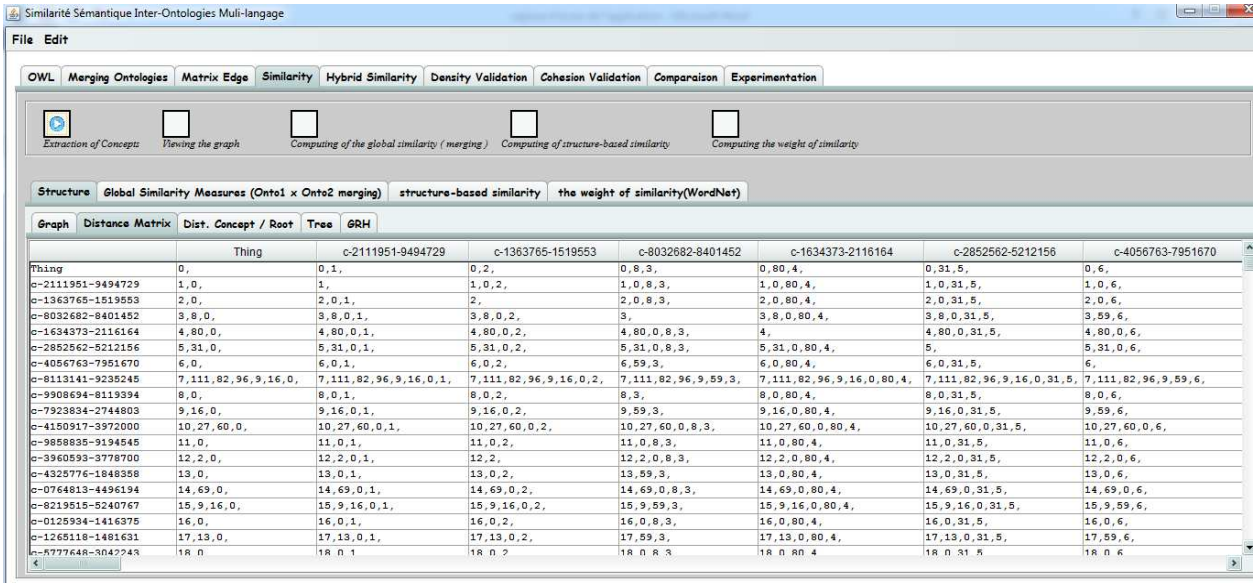
Figure 46. Graphes des nœuds de Conférence et Sigkdd et leur ontologie de fusion



	Thing	c-2111951-9494729	c-1363765-1519553	c-8032682-8401452	c-1634373-2116164	c-2852562-5212156	c-4056763-7951670	c-8113141-9235245	c-9908694-8119394
Thing	0	1	1	0	0	0	1	0	1
c-2111951-...	0	0	0	0	0	0	0	0	0
c-1363765-...	0	0	0	0	0	0	0	0	0
c-8032682-...	0	0	0	0	0	0	0	0	1
c-1634373-...	0	0	0	0	0	0	0	0	0
c-2852562-...	0	0	0	0	0	0	0	0	0
c-4056763-...	0	0	0	0	0	0	0	0	0
c-8113141-...	0	0	0	0	0	0	0	0	0
c-9908694-...	0	0	0	1	0	0	0	0	0
c-7923834-...	0	0	0	0	0	0	0	0	0
c-4150917-...	0	0	0	0	0	0	0	0	0
c-9858835-...	0	0	1	0	0	0	0	0	0
c-3960593-...	0	0	1	0	0	0	0	0	0

Figure 47. La matrice d'incidence appliqué sur OWL3

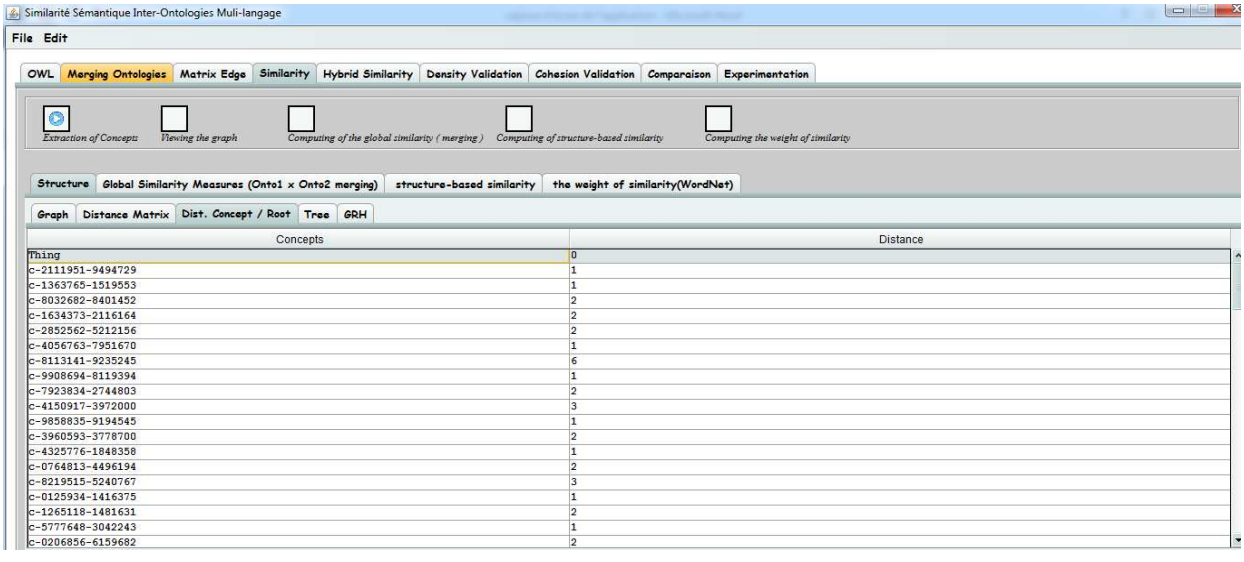
Après cette étape, le processus de calcul de matrice de distance est lancé. Le résultat est illustré dans la figure suivante (voir figure 48).



	Thing	c-2111951-9494729	c-1363765-1519553	c-8032682-8401452	c-1634373-2116164	c-2852562-5212156	c-4056763-7951670
Thing	0,	0,1,	0,2,	0,8,3,	0,80,4,	0,31,5,	0,6,
c-2111951-9494729	1,0,	1,	1,0,2,	1,0,8,3,	1,0,80,4,	1,0,31,5,	1,0,6,
c-1363765-1519553	2,0,	2,0,1,	2,	2,0,8,3,	2,0,80,4,	2,0,31,5,	2,0,6,
c-8032682-8401452	3,8,0,	3,8,0,1,	3,8,0,2,	3,	3,8,0,80,4,	3,8,0,31,5,	3,59,6,
c-1634373-2116164	4,80,0,	4,80,0,1,	4,80,0,2,	4,80,0,8,3,	4,	4,80,0,31,5,	4,80,0,6,
c-2852562-5212156	5,31,0,	5,31,0,1,	5,31,0,2,	5,31,0,8,3,	5,31,0,80,4,	5,	5,31,0,6,
c-4056763-7951670	6,0,	6,0,1,	6,0,2,	6,59,3,	6,0,80,4,	6,0,31,5,	6,
c-8113141-9235245	7,111,82,96,9,16,0,	7,111,82,96,9,16,0,1,	7,111,82,96,9,16,0,2,	7,111,82,96,9,59,3,	7,111,82,96,9,16,0,80,4,	7,111,82,96,9,16,0,31,5,	7,111,82,96,9,59,6,
c-9908694-8119394	8,0,	8,0,1,	8,0,2,	8,3,	8,0,80,4,	8,0,31,5,	8,0,6,
c-7923834-2744803	9,16,0,	9,16,0,1,	9,16,0,2,	9,59,3,	9,16,0,80,4,	9,16,0,31,5,	9,59,6,
c-4150917-3972000	10,27,60,0,	10,27,60,0,1,	10,27,60,0,2,	10,27,60,0,8,3,	10,27,60,0,80,4,	10,27,60,0,31,5,	10,27,60,0,6,
c-9858835-9194545	11,0,	11,0,1,	11,0,2,	11,0,8,3,	11,0,80,4,	11,0,31,5,	11,0,6,
c-3960593-3778700	12,2,0,	12,2,0,1,	12,2,	12,2,0,8,3,	12,2,0,80,4,	12,2,0,31,5,	12,2,0,6,
c-4325776-1848358	13,0,	13,0,1,	13,0,2,	13,59,3,	13,0,80,4,	13,0,31,5,	13,0,6,
c-0764813-4496194	14,69,0,	14,69,0,1,	14,69,0,2,	14,69,0,8,3,	14,69,0,80,4,	14,69,0,31,5,	14,69,0,6,
c-8219515-5240767	15,9,16,0,	15,9,16,0,1,	15,9,16,0,2,	15,9,59,3,	15,9,16,0,80,4,	15,9,16,0,31,5,	15,9,59,6,
c-0125934-1416375	16,0,	16,0,1,	16,0,2,	16,0,8,3,	16,0,80,4,	16,0,31,5,	16,0,6,
c-1265118-1481631	17,13,0,	17,13,0,1,	17,13,0,2,	17,59,3,	17,13,0,80,4,	17,13,0,31,5,	17,59,6,
c-5777648-3042243	18,0	18,0,1	18,0,2	18,0,8,3	18,0,80,4	18,0,31,5	18,0,6

Figure 48. Interface de la matrice de distance après calcul.

Comme nous avons mentionné dans les chapitres précédents que certaines mesures de similarités utilisées dans notre travail nécessitent le calcul de la distance minimale entre un concept et la racine (root), nous avons un processus de calcul de distance de n'importe quel concept vers la racine R. Cela est illustré par la figure suivante (Figure.49).



Concepts	Distance
Thing	0
c-2111951-9494729	1
c-1363765-1519553	1
c-8032682-8401452	2
c-1634373-2116164	2
c-2852562-5212156	2
c-4056763-7951670	1
c-8113141-9235245	6
c-9908694-8119394	1
c-7923834-2744803	2
c-4150917-3972000	3
c-9858835-9194545	1
c-3960593-3778700	2
c-4325776-1848358	1
c-0764813-4496194	2
c-8219515-5240767	3
c-0125934-1416375	1
c-1265118-1481631	2
c-5777648-3042243	1
c-0206856-6159682	2

Figure 49. Interface de calcul de distance concept/racine

Ensuite, il suffit de cliquer sur le bouton calcul de similarité à base de structure et le calcul des quatre mesures (Wu and Palmer, Rada, Li et Zargayouna) se fait (voir figure 50).

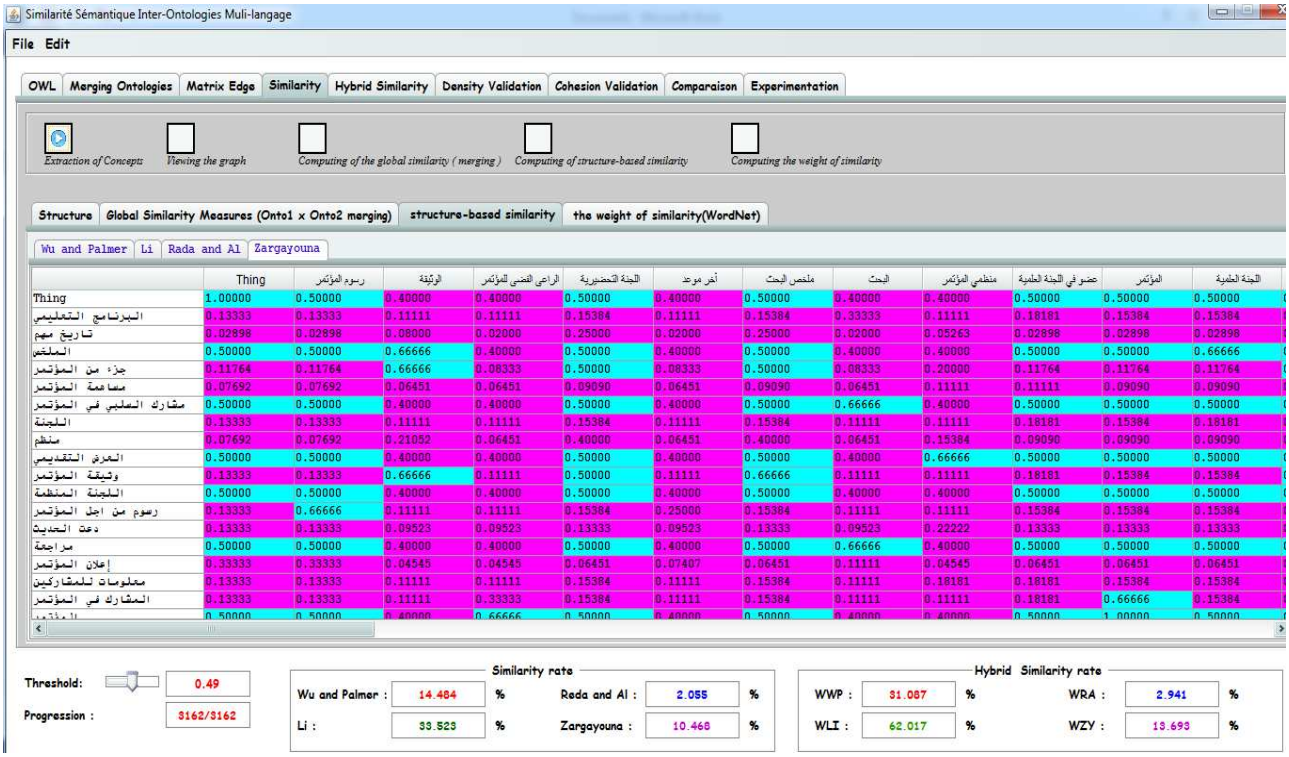


Figure 50. Interface de visualisation des résultats de calcul des quatre mesures de similarités (Wu and Palmer, Rada, Li et Zargayouna).

Par la suite en procède au calcul du poids de similarité calculé en utilisant WordNet (voir Figure 51)

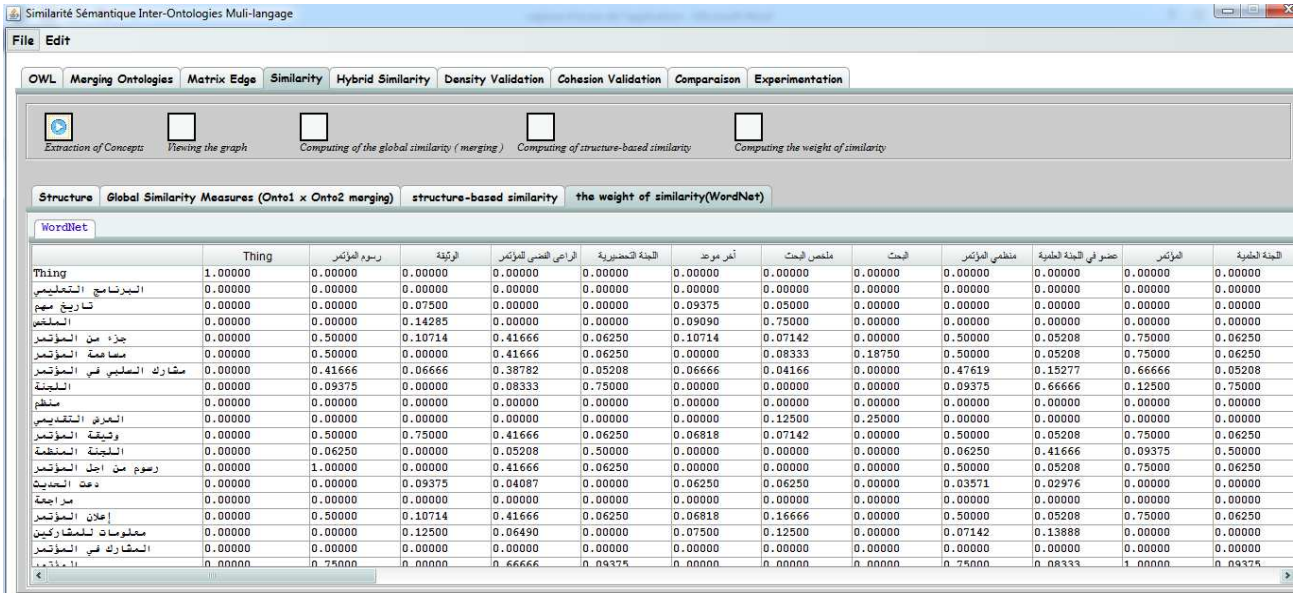


Figure 51. Interface de visualisation du résultat de calcul de poids de similarité utilisant WordNet

L'étape suivante consiste à calculer la nouvelle matrice d'incidence quand l'appel Matrice des poids de similarité où la valeur des arcs n'est pas 1, mais le poids de similarité entre deux concepts (voir figure 52).

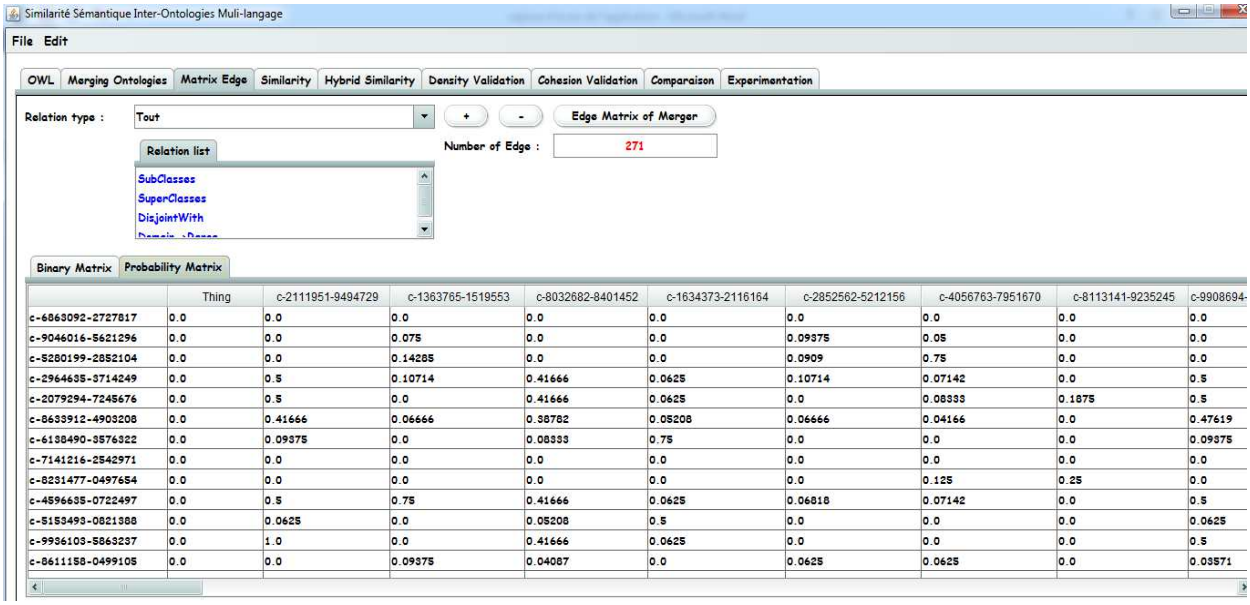


Figure 52. Interface de visualisation de la nouvelle matrice d'incidence appliqué sur OWL3

Cette modification affecte le calcul de la matrice de distance, cela permettra d'avoir des nouveaux plus cours chemins entre les concepts (voir figure 53).

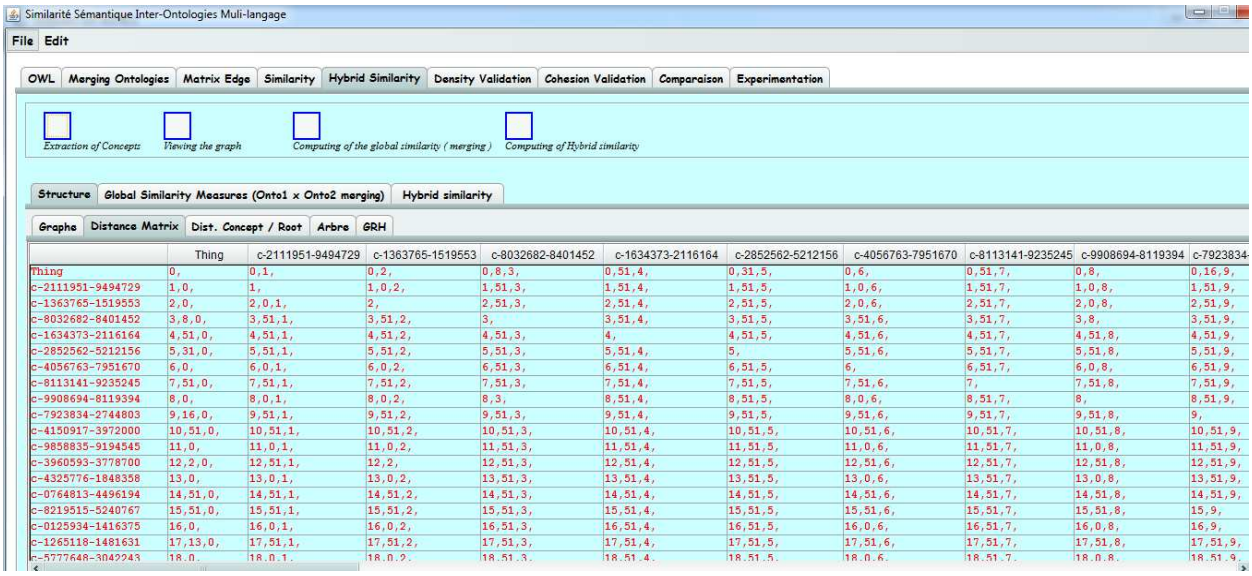


Figure 53. Interface de la nouvelle matrice de distance.

La figure suivante montre l'impact sur le calcul de la matrice concept / racine (voir figure 54)

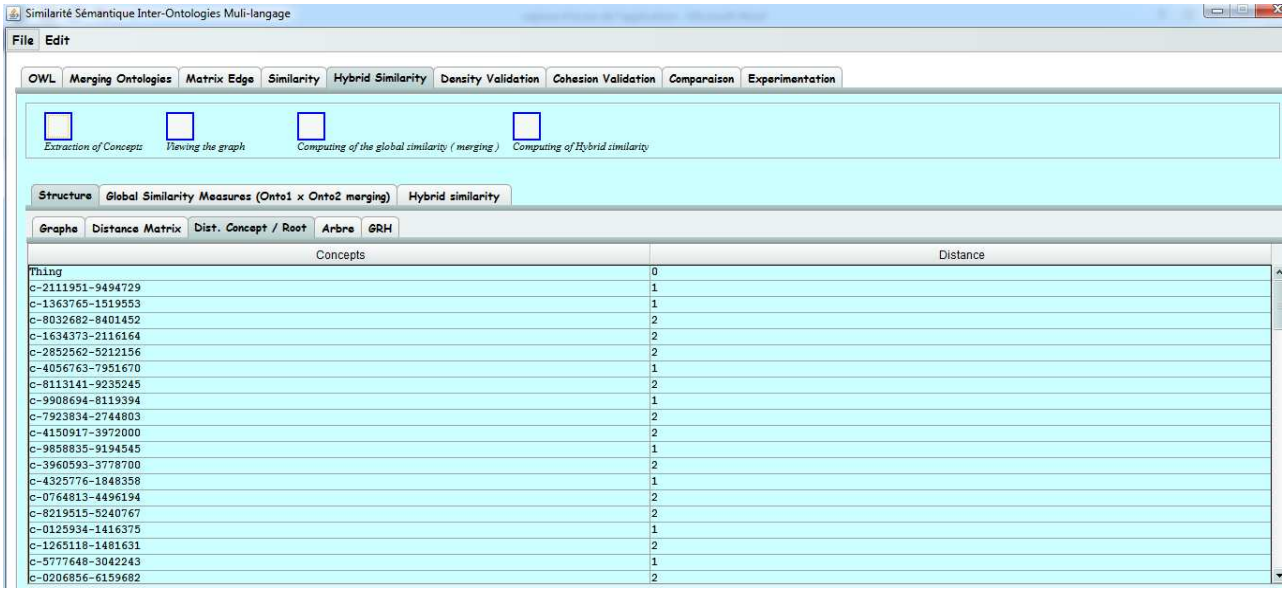


Figure 54. Interface de l'impact sur le calcul de distance Concept/racine

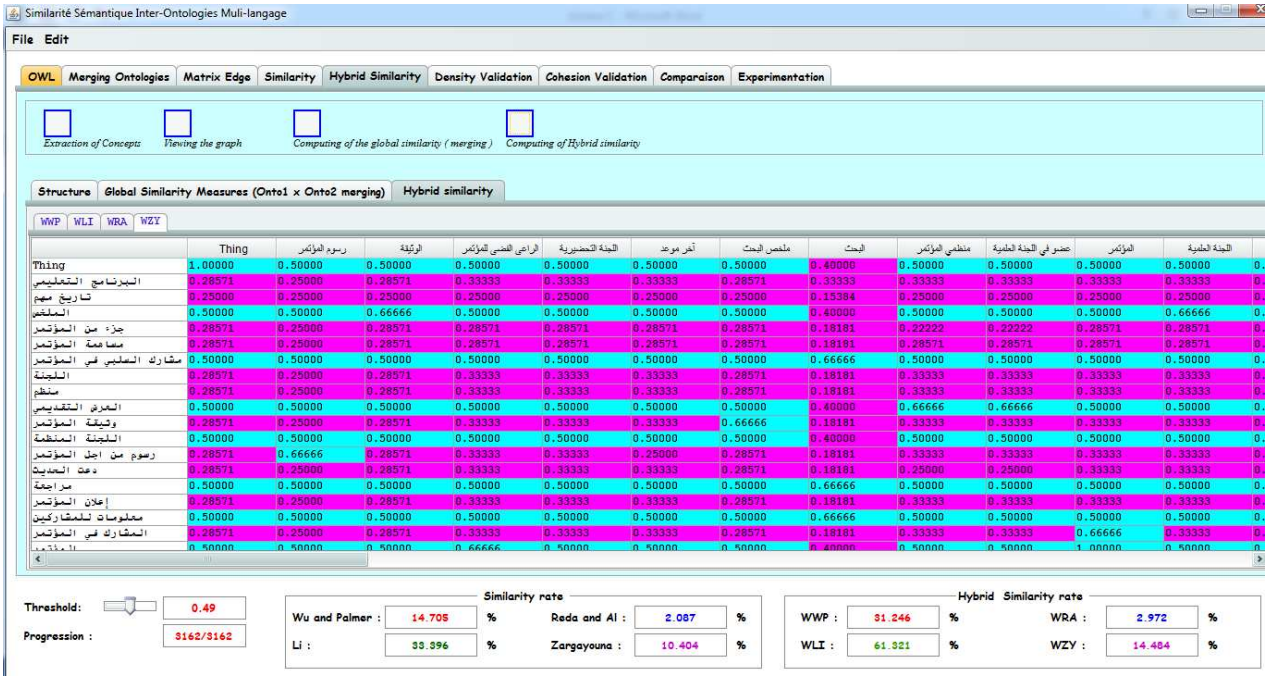


Figure 55. Interface de visualisation des résultats de calcul des quatre mesures de similarités (WWP, WRA, WLi et WZY).

La figure suivante montre une comparaison entre les mesures de similarités sémantiques existant avec chacune de nos mesures proposées (voir figure 56).

Figure 56. Interface de visualisation de la comparaison entre les mesures à base de structure avec nos mesures pour un seuil =0.8

Dans cette Annexe C, nous avons vu les différentes interfaces de notre implémentation, ils sont conçus pour être simples et faciles à utiliser.

Résumé **:** La mesure de la similarité sémantique entre les termes est une étape cruciale de la recherche et de l'intégration d'informations, car elle nécessite la mise en correspondance du contenu sémantique. Bien que plusieurs modèles aient été proposés pour mesurer la similarité sémantique, ces modèles ne sont pas en mesure de quantifier efficacement le poids des termes pertinents qui affectent le processus de jugement de la similarité sémantique. Dans cette étude, nous présentons une nouvelle méthode pour mesurer la similarité sémantique inter ontologies, qui consiste à hybrider les approches basées sur la structure des ontologies telles que Wu & Palmer, Rada, Li et Zargayouna avec le poids de la similarité calculé à l'aide du dictionnaire sémantique WordNet. Le processus que nous allons proposer, passe par quatre phases (i)Le prétraitement : cette phase consiste à obtenir les termes pertinents des deux ontologies sélectionnées dans la langue choisie, pour mesurer leur similarité sémantique en utilisant le lemmatiseur TreeTagger. (ii) Calcul de la mesure de similarité de Wu &Palmer, Li, Rada et Zargayouna. (iii) Calcul des mesures proposées de similarités sémantiques hybride de WWP, WLI, WRA et WZY. (iv) Dans cette dernière phase on procède premièrement à l'expérimentation de notre approche en comparant les résultats obtenus de la phase (ii) et (iii), en l'appliquant sur trois langues (Anglais - Arabe et Français) et deuxièmement à l'évaluation de notre approche en utilisant les deux méthodes la cohésion et la densité. Notre algorithme est appliqué sur plusieurs cas de tests de la compagnie OAEI'2015 et a donné des résultats encourageants.

Mots-clés : Ontologie, Fusion d'ontologies, Similarité sémantique, TreeTagger, WordNet.

Abstract: Measuring semantic similarity between terms is a crucial step in information retrieval and integration, as it requires the mapping of semantic content. Although several models have been proposed to measure semantic similarity, these models are not able to effectively quantify the weight of relevant items that affect the semantic similarity judgment process. In this study, we present a new method for measuring semantic similarity between cross-ontologies that consists of hybridizing ontology structure-based approaches such as Wu &Palmer, Rada, Li, and Zargayouna with the weight similarity computed using the WordNet semantic dictionary. The process that we will propose includes four phases (i) Preprocessing: this phase consists in obtaining the relevant terms of the two ontologies selected in the chosen language, to measure their semantic similarity using the TreeTagger lemmatizer. (ii) Computing the similarity measure of Wu &Palmer, Li, Rada and zargayouna. (iii) Computing the proposed hybrid semantic similarity measures of WWP, WLI, WRA and WZY. (iv) in this last phase, we proceed first to the experimentation of our approach by comparing the results obtained in phase (ii) and (iii), by applying it in three languages (English - Arabic and French) and secondly to the evaluation of our approach by using the two methods cohesion and density. Our algorithm applies to various test cases of the Ontology Alignment Evaluation Initiative campaign, (OAEI'2015) and shows encouraging results.

Keywords: Ontology, Ontology merging, Semantic similarity, TreeTagger, WordNet.

ملخص: يعد قياس التشابه الدلالي بين المصطلحات خطوة حاسمة في استرجاع المعلومات وتكاملها، حيث يتطلب مطابقة المحتوى الدلالي .على الرغم من أنه تم اقتراح العديد من النماذج لقياس التشابه الدلالي، إلا أن هذه النماذج غير قادرة على تحديد وزن المصطلحات ذات الصلة التي تؤثر على عملية حكم التشابه الدلالي بشكل فعال. في هذه الدراسة، نقدم طريقة جديدة لقياس التشابه الدلالي بين الأنطولوجيات، والتي تتمثل في تهجين مقاييس التشابه على أساس هيكل الأنطولوجيا مثل Wu&Palmer و Rada و Li و Zargayouna مع وزن التشابه المحسوب باستخدام القاموس الدلالي WordNet .

تمر العملية التي سنقترحها من خلال أربع مراحل : (1) المعالجة المسبقة : تتكون هذه المرحلة من الحصول على المصطلحات ذات الصلة للاثنتين من الأنطولوجيات المختارتين في اللغة المختارة ، لقياس التشابه الدلالي بينهما باستخدام TreeTagger.(2) حساب مقياس التشابه لكل من Wu & Palmer ، Li ، Rada وZargayouna . (3)حساب مقاييس التشابه الدلالي الهجينة المقترحة لـ WWP ، WLI، WRA وWZY . (4) في هذه المرحلة الأخيرة ، ننتقل أولاً إلى تجربة نهجنا من خلال مقارنة النتائج التي تم الحصول عليها من المرحلة (2) و (3) ، من خلال تطبيقه على ثلاث لغات (الإنجليزية ، العربية والفرنسية) وثانياً لتقييم نهجنا باستخدام طريقتين Cohesion وDensity. إن خوارزميتنا هذه، تم تطبيقها على عدة حالات اختبار من شركة "مبادرة تقييم مطابقة الأنطولوجيات لعام "2015" و أعطت نتائج مشجعة.

الكلمات المفتاحية: الأنطولوجيات، دمج الأنطولوجيات، التشابه الدلالي، TreeTagger، WordNet.