Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: *Modèles Intelligents et Décision (M.I.D)*

Thème

Mapping entre WordNets

Réalisé par :

- Billami Mokhtar Boumedyen

Présenté le 04 Juillet 2011 devant la commission composée de :

Président : - *M. Benamar Abdelkrim*

Encadreur : - *M. Bentaallah Mohamed Amine*

Examinateurs : - *M. Benziane Mohammed Yaghmorasan* / - *M. Hadjila Fethallah*

- *Mme. Iles Nawel* / - *M. Smahi Mohammed Ismail*

- *M. Benazzouz Mourtada*

Année universitaire : 2010-2011

Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut…
Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance…
Aussi c'est tout simplement que…

Je dédie ce mémoire ...
A Mes parents,
A mes frères,
Et à tous mes amis

Remerciements

Mes remerciements, les plus vifs, ma profonde gratitude et mes respects s'adressent à mon directeur de recherche *Mr. Bentaallah Med Amine* pour avoir accepté de m'encadrer, pour les conseils et orientations tant précieux qu'il m'a prodigué durant ce Master. Je le remercie aussi vivement pour la démarche fructueuse qu'il a adoptée pour m'introduire dans ce fabuleux domaine de la recherche d'information.

Je présente tous mes respects et mes remerciements aux membres du jury qui ont accepté et m'avoir fait l'honneur d'évaluer ce travail. Je remercie *M. Benamar Abdelkrim* de m'avoir fait l'honneur de présider mon jury. Je remercie *M. Benziane Mohammed Yaghmorasan, Mme. Iles Nawel, M. Benazzouz Mourtada, M. Hadjila Fethallah et M. Smahi Mohammed Ismail,* qui ont bien voulu faire partie de ce jury.

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

Des remerciements qui n'ont pas de limite, qui n'ont pas de mots pour les exprimer, je les adresse aux deux bougies qui m'illuminent ma vie : ma mère, mon père ainsi que ma grande mère « El hadja Kerzabi Amaria », qui m'ont toujours soutenue dans mes choix et qui m'ont toujours encouragée à aller de l'avant. Que Dieu leur donne une longue vie pleine de santé et de joie. Une belle pensée à mes frères et à tous les membres de la famille Billami ainsi que Henaoui, en souhaitant beaucoup de succès à ceux qui sont sur le chemin du savoir.

ملخص

تعتبر التوافقية الدلالية مسألة مهمة، معرفة بشكل واسع في تكنولوجيا التنظيم والمعلومة وفي ميدان البحث في إطار أنظمة المعلومة. إن الاستعمال الواسع للشبكة العنكبوتية (الويب) من أجل الوصول إلى المعلومات المنشورة، يقتضي توافقية الأنظمة التي من شأنها تسيير هذه المعلومات. وتعمل بعض الحلول وردود الأفعال مثل الويب الدلالي على تسهيل تحديد موقع وتناسق المعطيات بصفة أكثر ذكاء عن طريق الأنتولوجيا. الويب الدلالي يعطينا نظرة أكثر دلالية ومفهومة عن الويب، بالرغم من أنه يثير عددا هاما من التحديات في البحث، يتمثل أحد هذه التحديات الهامة في مقارنة واتساق مختلف الأنتولوجيات التي تظهر في عمليات التناسق.

إن التخطيط، أو التنسيق، مقاربة هامة لأنه لا يغير من الأنتولوجيات التي تدخل في إطار عمليات التنسيق، بل يعمل على اكتشاف التوافق الدلالي بين العناصر المتماثلة من مختلف الأنتولوجيات. وهو بالتالي، مفتاح البحث عن التوافق بين الإنتولوجيات. يتم التخطيط في أغلب الأحيان يدويا، مما يتطلب جهدا كبيرا مع أن أدوات النشر المتطورة تعمل على تسهيله، إلا أنه يمكن تحقيقه باستعمال أدوات أوتوماتيكية أو شبه أوتوماتيكية.

من خلال مذكرة التخرج هذه، نقترح مقاربة تخطيط من أجل القيام بتنسيق للأنتولوجيات التي لا تحمل نفس الأسلوب اللغوي الطبيعي. وتستدعي هذه المقاربة استعمال تقنيات بحث المعلومة والمساعدة على الكشف والتمهين الأوتوماتيكي. وهي تعتمد على ترجمة أوتوماتيكية لأنتولوجيا المصدر في الأسلوب اللغوي الطبيعي إلى أنتولوجيا الهدف، وهذا ما يسمح بخلق تواصل عبر تطبيق تقنيات التطابق الأنتولوجي أحادي اللغة. وتمزج المقاربة بين تقنيات ومناهج التطابق والتركيب والبناء اللغوي، هذا من أجل تحديد مقياس التشابه الإجمالي المحسوب عن طريق مزج عدة استراتيجيات لقياس التشابه. وقد أظهرت النتائج الأولية التي توصلنا إليها بأن هذه العملية فعالة وجد واعدة.

الكلمات الرئيسية : الويب الدلالي، أنتولوجيا، التخطيط أو التنسيق للأنتولوجيات.

Résumé

L'interopérabilité sémantique est une question importante, largement identifiée dans les technologies d'organisation et de l'information et dans la communauté de recherche en systèmes d'information. L'adoption large du Web afin d'accéder a des informations distribuées nécessite l'interopérabilité des systèmes qui gèrent ces informations. Des solutions et réflexions comme le Web Sémantique facilitent la localisation et l'intégration des données d'une manière plus intelligente par l'intermédiaire des ontologies. Il offre une vision plus sémantique et compréhensible du web. Pourtant, il soulève un certain nombre de défis de recherche. Un des principaux défis est de comparer et aligner les différentes ontologies qui apparaissent dans des tâches d'intégration.

Le mapping ou bien l'alignement est une approche intéressante parce qu'il ne modifie pas les ontologies qui rentrent dans le processus d'intégration. Il consiste à découvrir la correspondance sémantique entre les éléments similaires de différentes ontologies. C'est la clé de recherche de l'interopérabilité entre les ontologies. Le mapping est souvent génère manuellement, ce qui est extrêmement fastidieux même s'il est facilité par des outils d'édition sophistiqués mais il peut être aussi obtenu par l'utilisation d'outils automatiques ou semi automatiques.

Dans ce mémoire, nous proposons une approche de mapping pour effectuer un alignement des ontologies qui ne partagent pas le même langage naturel. Cette approche utilise des techniques de recherche d'information, des heuristiques et d'apprentissage automatique. Elle est basée sur une traduction automatique de l'ontologie source dans le langage naturel de l'ontologie cible, ce qui permet de générer des correspondances en appliquant des techniques d'appariement d'ontologies monolingues. L'approche combine les techniques et les méthodes d'appariement linguistiques, syntaxiques et structurelles. Ceci afin de définir une mesure de similarité globale calculée en combinant plusieurs stratégies de mesure de similarité. Nos résultats préliminaires ont montré que cette approche est efficace et très prometteuse.

Mots clés : Web sémantique, Ontologie, alignement et mapping d'ontologies.

Abstract

Semantic interoperability is a very important issue; it is widely identified in the organization and information technologies and also in the community of research of information's system. The wide using of the Web to have access to the supplied information needs the interoperability of systems which manage this information. Some solutions and reflections, as the Semantic Web, make the location and the integration of the given data easy in more intelligent way through ontologies, it give more semantic and comprehensive view of the Web. However it provokes some challenges. One of the most important challenges is the fact to compare and align the different ontologies which appear during the integration process.

The mapping or the alignment is an interesting approach since it doesn't change the ontologies which are a part of the integration process. It used to discover the semantic connection between the similar tools and the different ontology; it is the key to find the interoperability between the ontologies. Most of time, mapping is done manually, this is why it is very tedious, although if it is facilitated by sophisticated edition tools, but it can be also obtained by using automatic or semi-automatic tools.

In this project, we propose an approach of mapping to make an alignment of the ontologies which don't share the same natural language. This approach uses research techniques of information, heuristic and automatic learning. It is based on an automatic translation from the source of ontology in the natural language to the target ontology, which allows to generate connections by application of the monolingual ontologies generate. The approach combines the techniques and the methods of the alignment linguistic, syntactic and structural generate, this is to define a measure of a global similarity calculated by gathering lot of strategies of similarity measure. Our preliminary results show that this approach is effective and very promising.

Keywords: Semantic Web, Ontology, alignment and mapping ontologies.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES Tableaux

INTRODUCTION GÉNÉRALE

L'interopérabilité est une question importante, largement identifiée dans plusieurs domaines comme par exemple dans la communauté des systèmes d'information (SI). La dépendance et le partage d'information entre des organismes ont créé un besoin de coopération et de coordination qui facilite aussi bien l'échange et l'accès aux informations distantes qu'aux informations locales. La large adoption de l'internet (WWW : World Wide Web), pour accéder et distribuer l'information, engendre un besoin crucial de l'interopérabilité des systèmes.

Le problème principal de tous les travaux sur l'interopérabilité porte sur la comparaison et le m*apping* des différentes ontologies. Etant donnée la nature décentralisée du développement du Web, le nombre d'ontologies est très important. Pour intégrer les données des ontologies distinctes, nous devons connaitre les correspondances sémantiques entre leurs éléments.

La mise en correspondance peut être traitée entre des ontologies monolingues comme elle peut être traitée entre des ontologies multilingues. Les liens de correspondance peuvent être employés directement après avoir traduit les deux ontologies dans une seule langue afin d'appliquer les techniques de mapping d'ontologies monolingues.

Notre contribution dans ce mémoire consiste en la proposition d'une approche de mapping entre deux ontologies générales de type WordNet tel que WordNet anglais et WordNet espagnol. Cette approche se compose de trois phases : (i) une phase de traduction automatique dont cette dernière consiste à traduire l'ontologie source dans le langage naturel de l'ontologie cible, (ii) une phase de mise en correspondance dont les méthodes de calcul de similarités sont applicables dans le contexte afin de résoudre les hétérogénéités terminologiques et structurelles. Ces méthodes sont basées sur des informations auxiliaires capables d'identifier les éventuels liens linguistiques et hiérarchiques entre deux termes de deux concepts ontologiques et (iii) une phase de test et d'évaluation qui inclut l'expérimentation et l'évaluation des techniques de calcul de similarité. Notre approche de mapping utilise plusieurs techniques telles que la recherche d'information, les heuristiques et l'apprentissage automatique.

1

Ontologies

I. Introduction

La représentation de la sémantique est nécessaire pour l'intégration des systèmes, pour cela on utilise des techniques de représentation de la connaissance, cette dernière étudie comment transformer l'expression du sens en une représentation formelle manipulable par une machine, l'un des moyens les plus utilisés est l'ontologie.

De ce fait, nous avons remarqué que durant la dernière décennie, une attention croissante a été concentrée sur les ontologies. Ces dernières sont largement utilisées de nos jours dans plusieurs domaines comme l'ingénierie de connaissance, les applications liées à la gestion des connaissances, e-commerce, la recherche d'information ainsi que le Web Sémantique.

Cet engouement est motivé par le fait que les ontologies peuvent fournir un moyen efficace pour gérer les connaissances partagées et communes a un domaine particulier, tout en permettant des échanges tant au niveau syntaxique que sémantique, entre personnes et/ou systèmes. Les ontologies servent à décrire une sémantique et sont au centre des développements du Web sémantique. Elles permettent la modélisation d'informations agréées par une communauté de personnes et accessibles par une machine pour développer des services automatisés. Elles jouent donc un rôle de référence pour décrire la sémantique des informations des différents sites web.

Dans ce chapitre, nous relèverons les différentes définitions qui ont été attribuées à la notion d'ontologie, nous verrons aussi les différents éléments dont elle est composée. Nous présenterons ces différents types. Ensuite, nous détaillons le processus servant à leurs constructions ainsi que les outils utilisés pour leur développement y compris les langages de spécification. Finalement, nous montrerons les différentes approches d'interopérabilité existantes entre les ontologies.

II. Définitions

Nous assistons, ces dernières années, à l'émergence de la notion d'ontologie, qui constitue un sujet de recherche répandu dans diverses communautés notamment celles liées à l'ingénierie des connaissances, à l'ingénierie des systèmes, à l'intégration des systèmes, etc. Parce qu'elle facilite le partage et la réutilisation de connaissances, De ce fait, les ontologies définissent actuellement des vocabulaires structurés, regroupant des concepts utiles d'un domaine et de leurs relations et qui servent à organiser et échanger des informations de façon non ambiguë.

Le terme ontologie vient du mot grec Ontologia qui signifie, parler (logia) au sujet de l'être (onto), l'ontologie est une discipline philosophique qui peut être décrite comme la science de l'existence, ou l'étude de l'être ; Platon (427-347 AJC) était l'un des premiers philosophes qui mentionne explicitement le monde des idées ou des formes contrastées au vrais ou les objets observés qui sont selon sa vue, des réalisations imparfaites des idées ; En fait, Platon a soulevé les idées, les formes ou les abstractions aux entités qu'on peut parler aujourd'hui, donc il a créé les bases d'ontologies.

Plus tard, son étudiant Aristote (384-322 AJC) a formé le fond logique des ontologies, il a présenté des notions telles que la catégorie et la subsumption aussi bien que la distinction entre le superconcept et le subconcept, qui est appliqué sur les espèces pour les classifiés dans différentes catégories ; C'est le principe sur lequel sont basées les notions modernes du concept ontologique [1]. Dans cette section, nous allons décrire les différentes définitions qui ont été attribués à la notion d'ontologie.

− **Définition de Neches et al** 1991 [2] : « *Une ontologie définit les termes et les relations de bases qui composent le vocabulaire d'un domaine, bien que les règles de combinaison des termes et les relations pour définir l'extension du vocabulaire* ».

Cette définition indique en quelques sortes, ce qu'on doit faire pour construire une ontologie, elle identifie les termes de base et les relations entre termes, et les règles pour combiner les termes. Quelques années après, vient la définition qui nous semble être la plus célèbre est celle de Gruber.

− **Définition de Gruber** 1993 [3] : « *Une ontologie est une spécification explicite d'une conceptualisation* ». Le terme « conceptualisation » réfère à un modèle abstrait d'un certain phénomène de la réalité et qui permet d'identifier les concepts pertinents de ce phénomène. Le terme « explicite » signifie que le type des concepts utilisés est réellement défini d'une manière claire et précise.

− **Définition de Borst** 1997 [4] : « *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 manipulée sur machine, et d'autre part, le fait qu'elle doit être partagée dans la mesure où elle doit référer à la notion de groupe qui impose ainsi la mise en place d'un partage de connaissances entre les différents individus.

Les deux dernières définitions ont été fusionnées par Studer pour avoir la définition suivante:

− **Définition de Studer et al** 1998 [5] : « *Une ontologie est une spécification* ***formelle*** *et* ***explicite*** *d'une c****onceptualisation partagée*** ».

III. Constituant des ontologies

Selon Cimiano [1], une ontologie à la structure suivante :

$$O := (C, \leq_c, R, \leq_R, \sigma R, A, \sigma A, T)$$

Elle est composée de :

− Quatre ensembles différents, C, R, A et T dont les éléments s'appellent respectivement, les concepts, les relations, les attributs et les types de données ;

− Un treillis semi-supérieur ≤c sur C avec un élément supérieur qui présente la racine c, appelé la hiérarchie de concepts;

− Un ordre partiel ≤R sur R, appelé la hiérarchie de relation ;

− Une fonction σR: R → C^+ appelé la signature de relation ;

− Une fonction σA: A → C x T, appelé la signature d'attribut ;

− Un ensemble T des types de données tels que String, Integer, etc.

Dans cette section, et en se basant sur [3], nous allons expliquer en détail chacun des éléments précédents plus deux autres composants à savoir les axiomes et les instances.

− **Concepts** : un concept peut représenter un objet matériel, une notion, une idée [6] ; les concepts sont aussi appelés termes ou classe de l'ontologie, constituent les objets de base manipulés par les ontologies. Ils sont présentés dans OWL (Web Ontology Language) par owl : Class.

− **Relations** *:* traduisent les interactions existantes entre les concepts. Ces relations sont formellement définies comme tout sous ensemble d'un produit cartésien de n ensembles, c'est à dire R : C1 x C2x … x Cn et incluent la relation de spécialisation (subsomption), la relation de composition (meronymie), la relation d'instanciation, etc. Elles sont présentées dans OWL par owl : ObjectProperty.

− **Fonctions** : sont des cas particuliers de relations dont lesquelles le nième élément de la relation est défini de manière unique à partir des n-1 éléments précédents.

Formellement, les fonctions sont définies ainsi : F : C1 x C2… x Cn-1 → Cn.

− **Axiomes** : Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation.

− **Instances** : ou individus constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances à propos du domaine du problème. Elles sont définies en OWL par owl:Thing.

IV. Types d'ontologies

Dans cette partie, nous allons présenter les principaux travaux sur la classification des ontologies selon plusieurs types :

− **Guarino 1998 [8] :** il a classifie les ontologies selon le niveau de granularité, il a distingué l'ontologie supérieure, l'ontologie de domaine, l'ontologie de tâche et l'ontologie d'application.

− **Lassila et McGuinness [9] :** ils ont classifie les ontologies selon l'information dont l'ontologie a besoin, et la richesse de sa structure interne, ils ont précisé les catégories suivantes : vocabulaires, glossaires, thésaurus, hiérarchies informelles, hiérarchies formelles, Frame, ontologies avec restriction de valeur et ontologies avec contraintes logiques.

− **Gomez-Perez et al [10]** : ils ont classifie les ontologies selon la richesse de leurs structures internes comme le travail de Lassila [9], et le sujet de conceptualisation qui est une extension des travaux de Van Heijst et al [11] et ceux de Guarino [8].

IV.1. Types d'ontologies basés sur la richesse de leurs structures

• **Vocabulaires contrôlés :** c'est-à-dire un ensemble fini de termes, par exemple les catalogues ;

• **Glossaires :** des listes de termes avec leurs significations, présentés en langage naturel ;

• **Thésaurus :** il fournit plus de la sémantique entre termes, il présente les informations comme des relations entre synonymes, mais il ne fournit aucune hiérarchie explicite ;

• **Hiérarchies informelles :** la structure de ce type d'hiérarchie est basée non pas sur des relations de généralisation mais sur la proximité des concepts ;

• **Hiérarchies formelles :** hiérarchie dont la structure est déterminée par des relations de généralisation ;

• **Hiérarchie formelle avec instances du domaine**: similaire à la catégorie précédente mais incluant des instances du domaine ;

• **Frames** : ontologies incluant des classes avec propriétés pouvant être héritées ;

• **Ontologies avec restrictions de valeur**: ontologies pouvant contenir des restrictions sur les valeurs des propriétés ;

• **Ontologies avec contraintes logiques**: ontologies pouvant contenir des contraintes entre constituants (exemple : relations) définies dans un langage logique.

IV.2. Types d'ontologies basés sur le sujet de conceptualisation

• **Ontologie pour la représentation de connaissances** [11]**:** Selon Van Heijst et al, elle capture les primitifs de représentation employés pour formalisée la connaissance, l'exemple le plus représentatif est celui de Gruber 1993 [3], (the Frame Ontology) et (OKBC- Open Knowledge Base Connectivity), ils ont fourni des définitions formelles des primitives de représentations utilisées principalement dans les langages de Frame ;

• **Ontologie générale** [11] **ou commune** [7]**:** utilisée pour représenter un sens commun de la connaissance réutilisable à travers des domaines, ces ontologies incluent le vocabulaire lié aux : choses, évènements, espace, etc.

• **Les ontologies supérieures (Upper ou Top-level ontology)** [8] **:** modélisent les concepts très généraux auxquels les racines des ontologies de plus bas niveaux devraient être liées.

• **Les ontologies de domaine** [7] [11] **:** ce sont des ontologies qui sont construites sur un domaine particulier de la connaissance. Elles fournissent le vocabulaire des concepts du domaine de connaissance et les relations entre ces derniers.

• **Les ontologies de tâche** [7] [8] **:** Ces ontologies sont utilisées pour gérer des tâches spécifiques liées à la résolution de problèmes dans les systèmes, telles que les tâches de diagnostic, de planification, de configuration, etc.

• **Les ontologies d'application** [11]**:** Elles permettent de décrire des concepts dépendants à la fois d'un domaine et d'une tâche. Dans cette classification, la notion d'ontologie d'application définit le contexte d'une application qui décrit la sémantique des informations et les services manipulés par les d'applications sur un même domaine.

V. Construction des ontologies

A l'heure actuelle, on peut recenser dans la littérature une multitude de méthodologies, un peu plus d'une trentaine selon Gomez-Perez [10]. Les auteurs s'accordent à reconnaitre qu'il n'existe pas encore de consensus en matière de normes de construction. Donc, il n'existe pas encore de méthode universellement reconnue pour la construction d'ontologies. Ceci relève beaucoup plus du savoir-faire que de l'ingénierie.

V.1. Cycle de vie d'une ontologie

La figure 1.1 montrée ci-dessous illustre les différentes étapes de la vie d'une ontologie. On peut découper le cycle de vie d'une ontologie en sept étapes. La première étape consiste à détecter et spécifier les besoins auxquels la création de l'ontologie est sensée répondre, ainsi qu'à analyser la situation à modéliser.

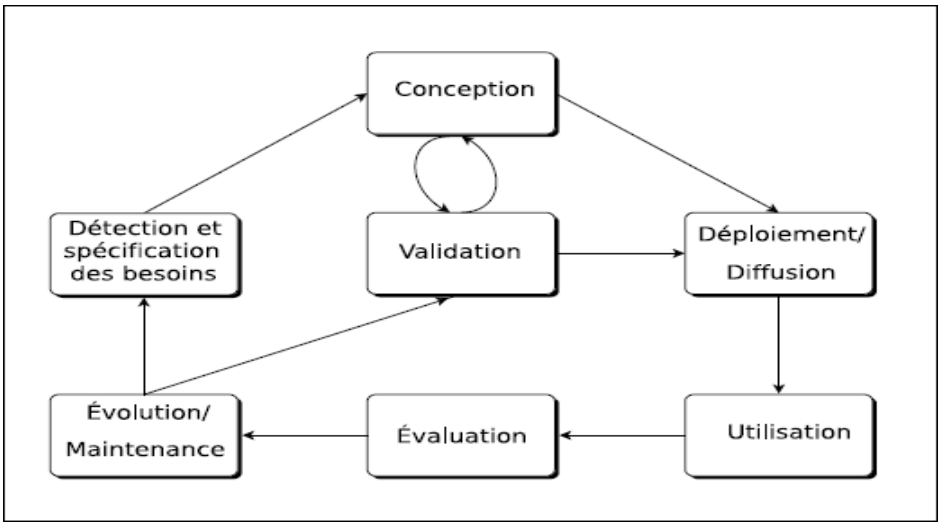


Figure 1.1 Cycle de vie d'une ontologie

Une fois les besoins définis, on passe à la conception, qui est l'étape majeure du cycle. Elle peut être divisée en trois sous-parties, la normalisation, la formalisation et enfin la population. La normalisation consiste à définir le vocabulaire utilisé, la formalisation consiste à définir l'ensemble des concepts et propriétés utiles et leurs relations. Enfin, l'étape optionnelle de population sert à associer des instances à l'ontologie créée.

Le déploiement permet de mettre à disposition l'ontologie à ses utilisateurs. L'utilisation correspond à l'exploitation par ces utilisateurs des informations fournies.

L'évaluation correspond à un retour des utilisateurs vis-à-vis de la qualité de l'ontologie et des résultats fournis. Elle permet de mettre à jour des problèmes de modélisation qui auraient pu être occultés lors de la première analyse.

L'étape suivante est une étape d'évolution. On peut la considérer comme une phase de maintenance si les modifications sont mineures, dans ce cas on peut passer directement à l'étape de validation. Des modifications plus importantes conduisent à l'exécution d'un nouveau cycle de développement complet, depuis l'analyse des besoins.

L'étape de validation permet de vérifier la validité de tout changement fait dans la structure de l'ontologie afin d'assurer une cohérence d'ensemble.

V.2. Les outils de développement des ontologies

Parmi les nombreux outils de développement d'ontologies, nous citons :

− **ONTOLINGUA** [12] : qui est un serveur d'édition d'ontologies au niveau symbolique, une ontologie est directement exprimée dans un formalisme également nommé ONTOLINGUA, qui constitue en fait une extension du langage KIF [21].

− **ODE** (Ontology Design Environment) [13] : développé au laboratoire d'Intelligence Artificielle de l'Université de Madrid, permet de construire des ontologies au niveau connaissance. La formalisation avec ODE s'effectue avec un langage de frames.

− **PROTEGE OWL** [14] : Le modèle de connaissances sous-jacent pour PROTEGE2000 est issu du modèle des frames et contient des classes, des slots (attributs) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés pour permettre le contrôle et la visualisation d'ontologies.

VI. Formalismes de représentation

Plusieurs formalismes de représentation des connaissances ont été proposés, seuls ceux qui sont les plus utilisés pour la représentation des ontologies seront présentés.

− **Frames :**

Le modèle des frames [15] est un classique de l'intelligence artificielle et a été initialement proposé comme langage de représentation d'ontologies par Gruber [3].

Le principe de ce modèle est de décomposer les connaissances en classes (ou frames) qui représentent les concepts du domaine. A un frame est rattaché un certain nombre d'attributs (slots), chaque attribut peut prendre ses valeurs parmi un ensemble de facettes (facets) [16].

− **Graphes conceptuels :**

Le modèle des graphes conceptuels [17] appartient à la famille des réseaux sémantiques [18]. Les réseaux sémantiques modélisent les connaissances sous forme de graphes, les nœuds étant associés à des concepts et les arrêtes à des relations.

- Le modèle des graphes conceptuels se décompose en deux parties :

• Une partie terminologique dédiée au vocabulaire conceptuel des connaissances a représenté, c'est-à-dire les types de concepts, les types de relations et les instances. Cette partie inclut également la hiérarchisation des types de concepts et de relations.

• Une partie assertion dédiée à la représentation des assertions du domaine de connaissance étudie.

− **Logiques de description :**

La LD [19] est une logique basée sur un formalisme de représentation des connaissances, elle s'apparente à la logique du premier ordre. Elle contient deux catégories d'éléments qui sont les concepts et les rôles. Les concepts sont considères comme des ensembles d'objet. Les rôles sont des relations binaires sur les objets. La logique de description permet par l'utilisation de différents constructeurs de créer des concepts complexes à partir de concepts plus simples.

VII. Langages de représentation des ontologies

Au début des années 90, un ensemble de langages d'ontologies basées Intelligence Artificielle était créée, nous trouvons la logique du premier ordre (exemple : KIF), les Frames combinées avec la logique du premier ordre (exemple Cycl, Ontolingua,OCML, Flogic), et la logique de description (exemple LOOM), OKBC (Open Knoweldge Base Connectivity) qui a été également crée comme un protocole pour accéder à des ontologies implémentées dans différents langages avec une représentation de connaissances basé frame. La figure 1.2 présente la disposition globale de ces langages.

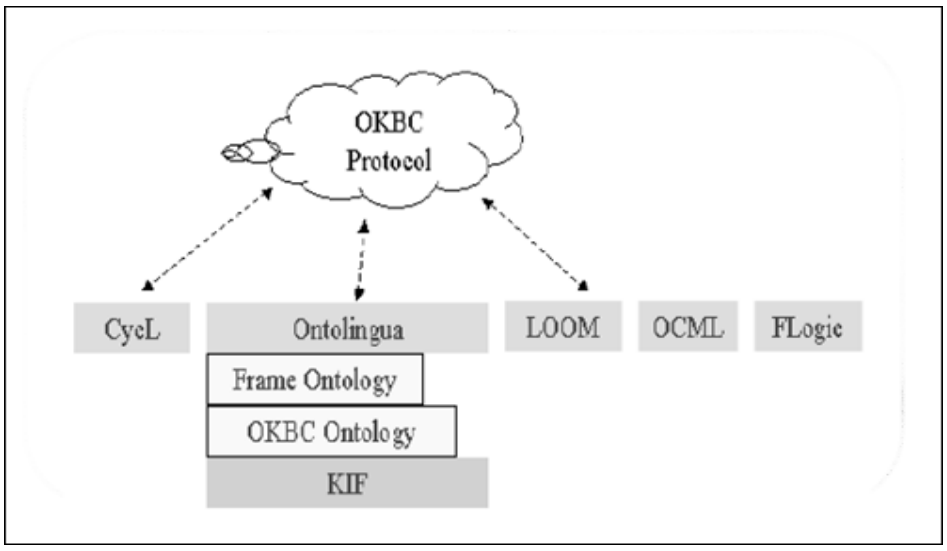


Figure 1.2 Langages traditionnels d'ontologies [10]

− **Cycl** de Lenat et Guha 1990 [20] : c'est le premier langage qui a été créer, il se base sur les Frames et la logique du premier ordre ;

− **KIF (Knowledge Interchange Format)** de Genesereth et Fikes 1992 [21] : il a été conçu comme un format d'échange de la connaissance, KIF est basé sur la logique du premier ordre. Il est difficile de créer directement des ontologies dans ce langage, cependant Ontolingua a été créé dans ce langage, d'ailleurs KIF est un langage supporter par le serveur Ontolingua.

− **LOOM** de MacGregor 1991 [22] : ce langage n'est pas conçu pour le développement des ontologies mais pour les bases de connaissances, il est basé sur la logique de description et les règles de production, il fournit une classification automatique des concepts.

− **OCML** de Motta 1999 [23]: il a été développé avant en 1993 et conçu comme une extension du langage Ontolingua, dans la mesure où il comble les lacunes de ce dernier en prenant en charge les règles de production, ce qui permet d'améliorer les mécanismes de raisonnement d'Ontolingua.

− **F-Logic (Framework-Logic)** de Kifer et al 1995 [16] : combine aussi bien les frames que la logique du premier ordre. Il permet de représenter des connaissances avec les concepts, taxonomies (sorte d'ontologies servant à décrire des liens sémantiques entre concepts), relations, axiomes et règles de déduction. F-Logic possède un moteur d'inférence, Ontobroker, qui peut être utilisé pour dériver de nouvelles connaissances.

- L'explosion des technologies d'Internet a mené à la création des langages pour l'exploitation des caractéristiques du Web, ces langages sont appelés généralement les langages basés Web ou les langages d'annotation d'ontologies, leurs syntaxe est basée sur l'existence d'annotation comme HTML et XML, dans le but n'est pas le développement d'ontologies mais la représentation et l'échange des données [10]. La figure 1.3 présente la relation entre ces langages.

− **SHOE** (Simple HTML Ontology Extensions) : est le premier langage d'annotation d'ontologies. Il a été créé en 1996 comme une extension de HTML. Il utilise des balises particulières qui permettent d'insérer des ontologies dans des documents HTML. Ce langage combine les Frames et les règles de production.

− **XML (eXtensible Markup Language) :** Ce langage a été très vite adopté comme un standard pour les échanges d'informations sur le Web par le W3C (World Wide Web Consortium) en 1998, SHOE a été modifié de telle sorte qu'il puisse supporter des documents structures décrits en XML.

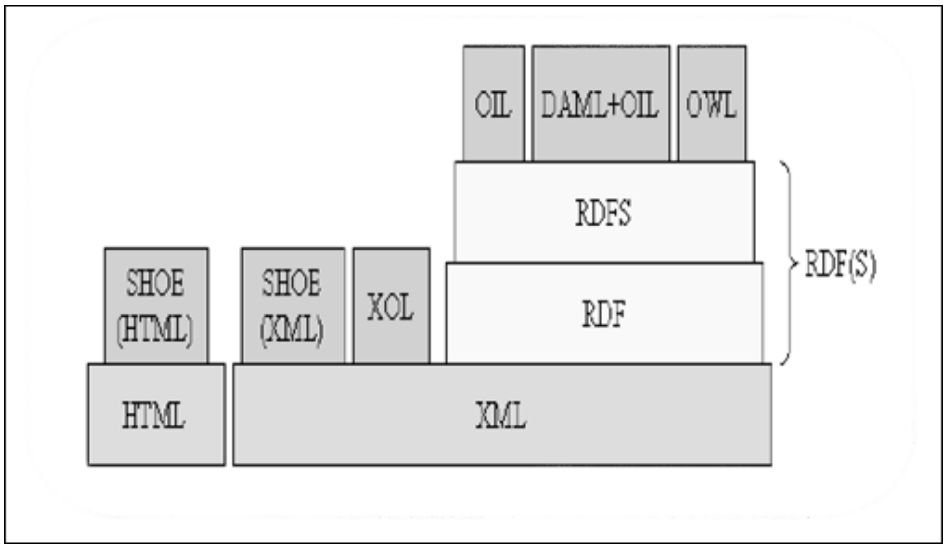


Figure 1.3 Langages d'annotation d'ontologies

− **XOL** (XML Ontology Language) : a été développé en 1999 comme un langage basé XML qui permet de spécifier des concepts et des relations binaires. Bien qu'il ne permette pas d'effectuer des raisonnements, ce langage a été longtemps utilisé dans les échanges d'ontologies dans le domaine biomédical.

− **RDF** (Ressource Description Framework) : il a été développé par W3C comme un langage basé sur les réseaux sémantiques pour décrire les ressources du Web. Afin de renforcer ce langage, RDF Schema a été construit par W3C pour comme extension de RDF comportant des primitives basées sur des frames. RDF Schema permet notamment de déclarer les propriétés des ressources ainsi que le type des ressources. La combinaison de RDF et RDF Schema est connue sous le nom RDF(S).

− **OIL** (Ontology Inference Layer) : basé sur RDF(S), le langage OIL permet de décrire les ontologies en combinant les primitives de modélisation utilisées dans les langages de frames. Il est limité du point de vue expressivité, puisque par exemple il ne supporte pas les types concrets.

− **DAML+OIL** : d'après son nom, ce langage est la combinaison des deux langages DAML et OIL. Il hérite des avantages de ces deux derniers. DAML (*Darpa Agent Markup Language*) est conçu pour permettre l'expression d'ontologies dans une extension du langage RDF. En conséquence, DAML+OIL est un langage très expressif et lisible par la machine ainsi que par un être humain avec une syntaxe basée sur RDF.

− **OWL** (Web Ontology Language): il est défini par le W3C [24], et construit sur DAML+OIL. OWL est développé pour pallier les lacunes de DAML+OIL, et plus précisément pour être utilisé dans des situations où les informations contenues dans les

documents Web doivent être traitées par des applications logicielles. Le langage OWL se base sur la recherche effectuée dans le domaine de la logique de description.

Le langage OWL se compose de trois sous langages qui proposent une expressivité croissante, tel que : OWL Lite, OWL DL et OWL Full.

VIII. Approches de l'interopérabilité sémantique

Les ontologies peuvent être utilisées dans les tâches d'intégration pour décrire la sémantique des différentes sources d'information et pour rendre leurs contenus explicite. La zone la plus importante d'applications des ontologies est l'intégration des systèmes existants et la capacité d'échanger l'information au temps de l'exécution qui est également connue sous le nom de l'interopérabilité dont l'objectif est de décrire un domaine unifié et accomplir une tâche commune. En général, il existe trois approches pour l'explication de la sémantique définies entre les ontologies [25] :

− **Approche mono ontologique** (single ontology approach) : (voir Figure 1.4 a) cette approche utilise une ontologie globale qui fournit un vocabulaire partagé pour la spécification de la sémantique. Toutes les sources d'informations sont reliées par une seule ontologie globale. Ce type d'approche est le plus simple à mettre en œuvre lorsque les sources de données se réfèrent à des domaines similaires.

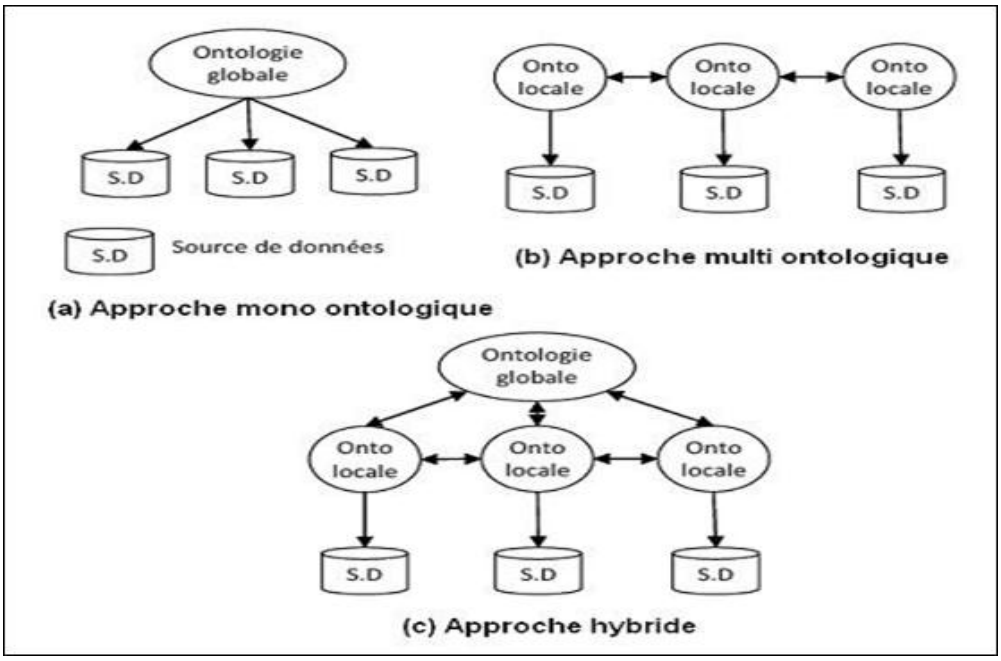


Figure 1.4 Les trois approches d'explication du contenu [26]

− **Approche multi ontologique** (multiple ontology approach) [26]: (voir Figure 1.4 b) Dans cette approche, chaque source d'information est décrite par sa propre ontologie,

cette approche convient dans le cas où il devient difficile de trouver une ontologie commune résultant des grandes différences sémantiques existant entre les systèmes.

Des mises en correspondance inter-ontologiques seront alors nécessaires afin d'établir une interprétation commune des données.

− **Approche hybride** (hybrid ontology approach) [26]: (voir Figure 1.4 c) cette approche a été développée pour éviter les lacunes des approche mono et multi ontologique. L'avantage de cette approche est que l'on peut facilement ajouter une nouvelle source sans modifier le vocabulaire partagé ainsi elle supporte l'acquisition et l'évolution des ontologies alors que leur inconvénient réside dans le fait que les ontologies existantes ne peuvent pas être réutilisées facilement mais elles doivent être reconstruites à zéro pour se référencier à un vocabulaire partagé.

IX. Conclusion

A travers ce que nous avons présenté dans ce chapitre, il ressort que la notion d'ontologie constitue l'une des approches les plus efficaces pour représenter les connaissances.

Dans le cadre de notre mémoire. Notre objectif sera de faire un appariement entre les ontologies et de sortir avec les meilleures correspondances. Il est généralement admis que la démarche d'alignement d'ontologies est basée sur trois processus [27] : le processus d'extraction de la sémantique qui présente une étape de découverte de la sémantique, le processus de représentation et enfin le processus d'alignement à proprement dit. Le processus d'extraction correspond à la modélisation de la sémantique effectuée sur la base d'ontologies, le processus de représentation d'ontologies permet de quantifier le degré de similarité des concepts des ontologies, et ce en se basant sur des calculs et des mesures de similarité, et enfin, le dernier processus permet d'aligner les ontologies concernées.

Après ce panorama des technologies qui constituent le paysage technique de ce mémoire, le chapitre suivant se focalise sur les travaux de recherche autour de l'alignement d'ontologies.

2

Etat de l’art

I. Introduction

Comme le Web est un environnement fortement distribué, évolutif et hétérogène, une seule ontologie ne peut jamais être suffisante pour représenter toutes les connaissances d’un tel environnement, ce qui nécessite l’utilisation de plusieurs ontologies. Celles-ci doivent parfois être intégrées pour fournir une vision globale sur le domaine de connaissances ou pour assurer l’interopérabilité des systèmes pour les quels ces dernières sont conçues.

Le mapping d’ontologies permet de chercher une méthodologie pour un échange efficace et correcte entre les ontologies. La recherche en mapping d’ontologies est organisée autour de trois axes, la découverte qui constitue l’étape fondamentale de ce processus, la représentation et le raisonnement (exploitation/exécution) avec le mapping qui sont des nouveaux axes de recherche. Le mapping permet de découvrir les correspondances entre les entités (concepts, propriétés, instances) des différentes ontologies, il peut se faire manuellement mais avec la complexité et la taille d’ontologies, cette manière devienne fastidieuse, par contre automatiquement ou semi automatiquement sont les meilleures façons de le faire.

L’objectif de ce chapitre est de décrire d’une manière claire la notion du mapping, en commençant d’abord par la description de la terminologie liée à ce domaine. Nous décrivons le processus du mapping suivant une étude détaillée sur ses différentes étapes ensuite nous présenterons les mesures de similarité sémantique. Nous présenterons un panorama pour la classification des techniques du mapping selon plusieurs vues ainsi que les diverses compositions et combinaisons de ses techniques. Finalement, nous exposerons les principales méthodes, outils et Framework existants pour le mapping d’ontologies monolingues ainsi que le mapping d’ontologies multilingue.

II. Terminologies

Le terme mapping a été utilisé de façon abusive dans la littérature dans la mesure où il est associé à une multitude des situations. Dans ce mémoire, le mapping désigne la découverte de correspondances entre les ontologies. La littérature concernant l'alignement n'est pas assez claire [28, 29, 30, 31], elle est très hétérogène. Pour pallier à cette multitude de définitions concernant le mapping et l'alignement, nous présenterons quelques définitions des concepts relatifs à ces notions.

II.1. Mapping/Matching d'ontologies

Appelé de même « correspondance entre ontologies ». Ces concepts sont considérés similaires. L'établissement du mapping entre deux ontologies signifie que pour chaque entité dans une ontologie, il faut trouver l'entité correspondante dans la deuxième ontologie avec le sens équivalent ou le sens le plus proche. La caractéristique importante de ce dernier est qu'il ne modifie pas les ontologies impliquées et qu'il produit en sortie un ensemble de correspondances.

II.2. Méthodes de comparaison ou matchers

Un matcher est une fonction utilisée pour calculer la distance entre deux entités. Les matchers peuvent être combinées dans le processus de matching (mapping).

II.3. Alignement d'ontologies

Il est appliqué si les ontologies concernées deviennent homogènes entre elles et ceci tout en les gardant séparées. Cette catégorie de *mapping* d'ontologies est faite habituellement quand les ontologies sources appartiennent à des domaines complémentaires.

II.4. Transformation d'ontologies

La transformation d'ontologies est un processus qui permet de changer la structure des ontologies en conservant au maximum la sémantique de cette structure. Selon que la transformation se fait sans pertes ou avec perte d'informations, on parle de transformation sans pertes (lossless) et de transformation avec pertes (lossy).

II.5. Fusion d'ontologies (merging ontologies)

La fusion d'ontologies est le processus de création d'une seule ontologie rassemblant les connaissances de deux ou plusieurs ontologies existantes et différentes qui décrivent

le même sujet ou appartiennent au même domaine d'application L'ontologie générée inclut les informations de toutes les ontologies sources.

II.6. Intégration d'ontologies

L'intégration d'ontologies est un processus de construction d'une nouvelle ontologie qui n'est pas forcement destinée à remplacer les autres. Ces différentes ontologies peuvent être connexes (ce qui veut dire que les ontologies peuvent être étroitement liées par rapport à leur domaine utilisé).

III. Le processus du mapping

Il s'agit de la combinaison d'un ensemble de méthodes de comparaison (des matchers) qui calculent le niveau de similarité entre deux entités. Ces méthodes nous permettent de calculer la meilleure correspondance entre des couples d'entités. Elles peuvent maximiser la découverte du nombre de couples similaires et réduire le nombre de ceux qui sont dissimilaires. Plusieurs définitions ont été attribués au processus du mapping comme dans [31, 32, 33, 34]. Parmi les quelles la plus spécifique est celle d'Ehrig et Staab [31,34] qui ont proposé un modèle de mapping englobant toutes les approches de ce dernier dont le procédé est comme suit :

- Ingénierie de caractéristique : qui transforme les ontologies sources dans un format pour calculer la similarité.
- Sélection de la prochaine étape de recherche *:* la dérivation des mappings a lieu dans un espace de recherche des mappings candidats. Cette étape permet de choisir de calculer la similarité d'un sous-ensemble restreint des concepts.
- Calcul de similarité : détermine la valeur de similarité entre concepts candidats.
- Agrégation de la similarité : généralement il peut y avoir plusieurs valeurs de similarité pour une paire de concepts candidats. Ces différentes valeurs doivent être agrégées dans une seule valeur de similarité.
- Interprétation : l'interprétation exploite la valeur de similarité agrégée ou individuelle pour dériver le mapping final, par l'utilisation des seuils par exemple.
- Itération : plusieurs algorithmes effectuent des itérations sur tout le processus afin de recouvrir toutes les connaissances, l'itération s'arrête quand il n'y a aucun mapping proposé.

IV. Similarité sémantique

Il existe plusieurs techniques pour évaluer la similarité entre deux entités, la plus courante consiste à définir une mesure de similarité. La similarité joue un rôle central dans le processus de mapping ; Elle se rapporte à la comparaison des éléments d'ontologies. Elle renvoie une valeur numérique indiquant si les deux éléments ont un degré élevé ou bas de similitude. Cette similarité peut être calculée selon plusieurs façons (terminologique « syntaxique /linguistique », structurelle, etc.). Selon [27], la similarité peut être formellement définie comme suit.

- ***Définition de la mesure de similarité***

C'est une fonction définie par f : O×O→R (O ensemble des objets et R ensemble des réels). Elle retourne une valeur numérique qui exprime le degré de similarité entre deux objets. Cette fonction satisfait les propriétés suivantes :

- Positive : $\forall (x, y) \in O\times O\ f(x, y) \geq 0$;
- Maximale : $\forall (x, y, z) \in O\times O\times O\ f(x, x) \geq f(y, z)$;
- Symétrique : $\forall (x, y) \in O\times O\ f(x, y) = f(y, x)$.

- ***Définition de la mesure de dis-similarité***

Cette mesure est l'opposée de la mesure de similarité. Elle est définie par f : O×O→R. Elle retourne une valeur numérique qui exprime le degré de dis-similarité (ou la divergence) entre deux objets. Cette fonction satisfait les propriétés suivantes:

- Positive : $\forall (x, y) \in O\times O\ f(x, y) \geq 0$;
- Minimale : $\forall x \in O\ f(x, x) = 0$;
- Symétrique: $\forall (x, y) \in O\times O\ f(x, y) = f(y, x)$.

- ***Définition de la distance ou encore de la métrique***

C'est une fonction qui calcule la dis-similarité entre deux objets f : O×O→R. Elle satisfait les propriétés suivantes :

- $\forall (x, y) \in O\times O\ f(x, y)=0$ Si seulement si x=y ;
- Inégalité triangulaire : $\forall (x, y, z) \in O\times O\times O\ f(x, y) + f(y, z) \geq f(x, z)$.

V. Classification des techniques de mapping

Le problème du mapping n'est pas limité aux ontologies, mais touche également les domaines de l'intégration de schémas ou des entrepôts de données. En effet, dans tous ces domaines, une étape d'analyse puis de mise en correspondance des structures de données est un passage obligé avant de pouvoir exploiter l'architecture considérée.

V.1. Pré-traitement

L'étape de pré-traitement traite les données en entrée pour produire un résultat utilisé par le reste du programme. Ce résultat est une version retouchée des données d'origine afin de faciliter l'exécution des tâches subséquentes. Les méthodes de pré-traitement, sans être des méthodes de similarité à proprement parler, sont utilisées pour améliorer les résultats des diverses méthodes de mapping qui sont bien détaillées dans les sections VII et VIII.

- **Le pré-traitement basé sur des chaines de caractères**

Il est également appelé « pré-traitement syntaxique ». Il permet de normaliser les données en les considérants comme des chaines de caractères. Pour cette normalisation, on peut supprimer les différences de casse, supprimer les accents, supprimer les valeurs numériques, remplacer les soulignés et les tirets par des espaces, supprimer la ponctuation, ou encore supprimer tous les autres caractères non alphanumériques. Cette normalisation permet de supprimer des différences entre chaines de caractères semblables, augmentant ainsi la valeur de similarité obtenue.

Malheureusement, la suppression de ces éléments peut conduire à une perte de sens, et par conséquent l'augmentation de la valeur de similarité obtenue peut concerner des chaines qui ne sont pas sémantiquement proches. Un exemple de similarité inappropriée causé par la suppression des accents : *tâche* est normalisé en *tache*. La perte de sens peut se retrouver par exemple dans le cas de la chaine de caractère, *CodeBase64*, transformée en codebase, qui n'a plus la même signification sémantique. Enfin, elle n'est souvent applicable qu'à des données décrites dans un langage occidental.

- **Le pré-traitement linguistique**

Il cherche à réduire les termes rencontrés à leur forme de base, de façon à pouvoir facilement retrouver les termes identiques. Les manipulations effectuées par cette normalisation consistent en : l'extraction de termes et la récupération des lemmes des mots rencontrés (stemming). L'extraction de termes consiste à transformer les chaines de caractères analysées en liste de mots. La récupération de lemmes, permet d'ignorer les variations dues à la conjugaison, entre autres. L'objectif de base de la lemmatisation (stemming) est de récupérer le radical des mots, même s'il ne correspond pas à un mot en soi. Par exemple le stemming du mot *disambiguation* donne *ambigu*.

- **L'expansion de termes**

Elle vise à remplacer une chaine de caractères correspondant à une abréviation par le ou les mots complets correspondants. Ce type de pré-traitement correspond également à l'expansion d'acronymes. Cette méthode repose généralement sur l'utilisation d'un lexique qui liste les abréviations susceptibles d'être rencontrées ainsi que le ou les mots qui doivent les remplacer.

V.2. Les différentes techniques de mapping

Pour présenter les différentes méthodes de mapping existantes, nous adoptons ici la classification définie par Bernstein et Rahm [35] et remaniée par Euzenat et Shvaiko [27], qui organise les méthodes en fonction du type d'information utilisé pour générer les mappings. Cette classification repose sur deux dimensions. D'une part, les méthodes sont classifiées selon les critères de niveau de granularité des données prises en compte et selon les critères de niveau d'interprétation des données. D'autre part on utilise comme critère le type des informations considérées. La figure 2.1 illustre ces deux dimensions.

La première dimension, qui correspond au type des données utilisées, est divisée en quatre catégories. Les méthodes utilisant le nom des données sont dites terminologiques. Les méthodes prenant en compte la structure des ontologies, que ce soit les types de données au niveau des éléments ou de graphe au niveau des ontologies dans leur globalité sont dites structurelles. Les méthodes utilisant les instances sont qualifiées d'extensionnelles. La dernière catégorie concerne les méthodes sémantiques.

Dans l'autre dimension, la première caractéristique est le niveau de granularité des données qui peut être de type *élémentaire* ou *structurel*. Un niveau élémentaire signifie que l'on ne tient compte que des éléments à la granularité la plus basse, par exemple les colonnes dans le cas d'une base de données, ou les attributs pour le cas des fichiers XML. Le niveau structurel prend en compte les interactions entre les entités ou leurs instances par rapport au reste de la structure. La seconde caractéristique est le critère du niveau d'interprétation des données qui peut prendre trois valeurs, *syntaxique*, *externe* ou *sémantique*. Une interprétation syntaxique correspond à l'utilisation des données seules en leur appliquant un algorithme précis. Une interprétation externe permet l'utilisation de ressources externes pour interpréter les données et une interprétation sémantique dont les données en entrée sont interprétées par une sémantique formelle.

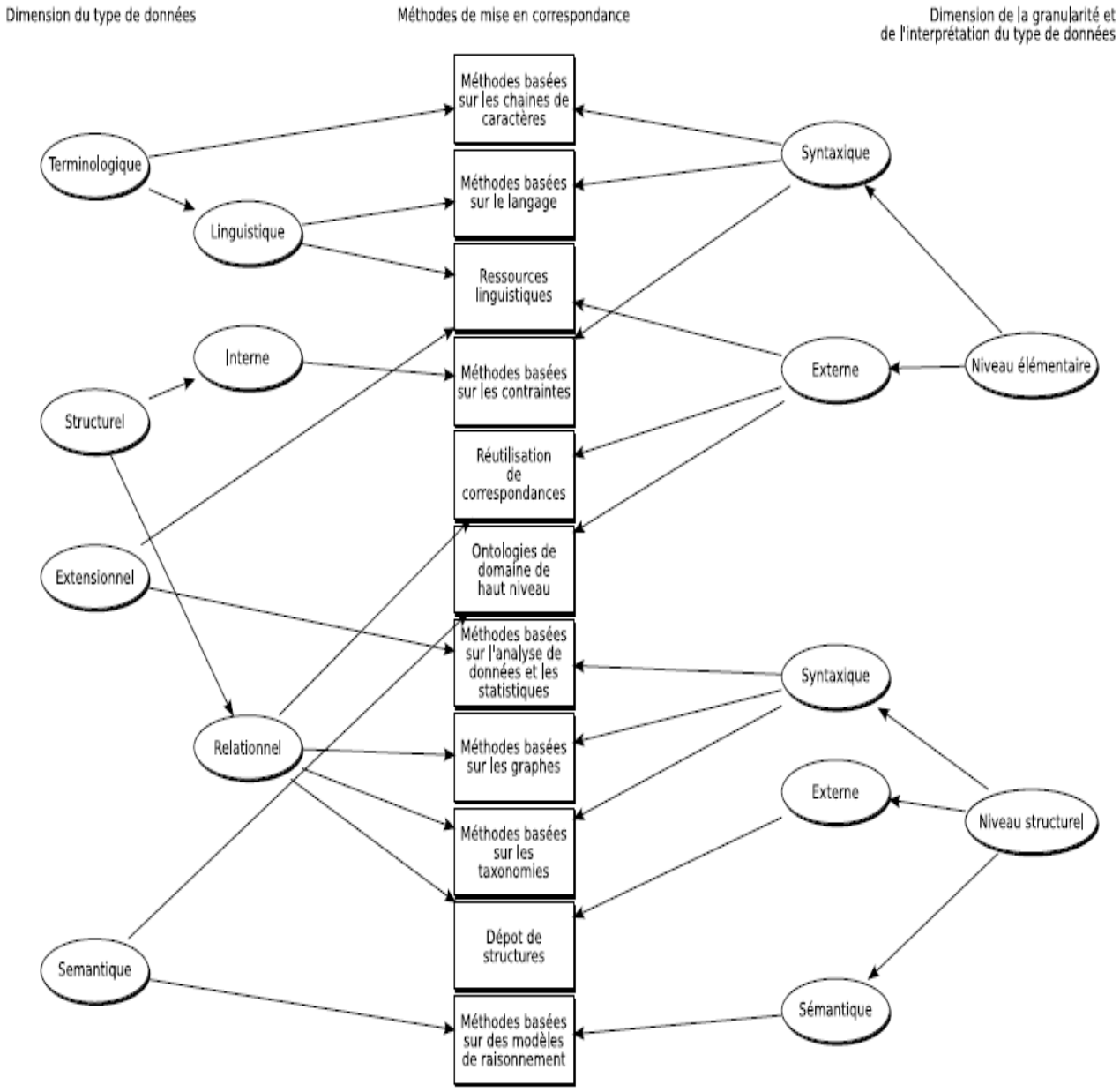


Figure 2.1 Classification des méthodes de mapping selon Euzenat et Shvaiko

V.2.1. Méthodes basées sur l'analyse des chaines de caractères

On peut distinguer deux grandes catégories d'analyse des chaines de caractères, d'une part les méthodes qui prennent en compte deux chaines de caractères simples, et d'autre part celles qui comparent des ensembles de chaines.

- La méthode de base

Elle consiste simplement en une comparaison des chaines de caractères, résultant en une similarité (valeur 1) en cas d'identité et en une dis-similarité (valeur 0) sinon.

Une autre méthode de base consiste à tester si l'une des deux chaines comparées est une sous-partie de l'autre. D'autres méthodes d'analyse de chaines de caractères existent comme la méthode des n-grammes [36] et la distance de Hamming [37] qui se basent sur le nombre de caractères différents entre deux chaines de caractères.

Pour les méthodes qui prennent en compte un nombre plus important de paramètres, on trouve les distances basées sur des tokens et qui reposent sur l'utilisation d'un ou plusieurs groupes de chaines de caractères non organisées (sacs de mots). Ces ensembles peuvent être comparés selon diverses méthodes. Dans ce titre, on peut citer le coefficient de Dice [38] ou la distance euclidienne qui se base sur un espace géométrique à *n* dimensions.

V.2.2. Méthodes linguistiques

Ces méthodes reposent sur l'utilisation de ressources externes. Une description de ces ressources externes est présentée comme suit.

- Un lexique ou dictionnaire

Il présente un ensemble de mots auxquels sont associées des définitions écrites en langage naturel. Un mot donné peut avoir plusieurs définitions s'il possède plusieurs sens. Un dictionnaire multilingue est constitué de la même façon qu'un lexique mais la définition est remplacée par son équivalent dans l'autre langage considéré.

- Une taxonomie

Elle présente une hiérarchie de termes. Un thésaurus étant la taxonomie en y ajoutant des relations de synonymie entre termes équivalents

- Une terminologie

Elle est considérée comme étant un thésaurus qui contient également des expressions en plus de mots simples. De plus une terminologie se limite à décrire le vocabulaire d'un domaine précis, contrairement aux autres structures présentées ci-dessus.

- Il existe diverses méthodes qui reposent sur l'utilisation de telles ressources. La méthode la plus simple utilise la synonymie, et consiste à déclarer que deux termes sont équivalents simplement s'ils sont présentés comme étant des synonymes.

V.2.3. Méthodes extensionnelles

Les instances associées aux concepts d'une ontologie sont un autre type de données qui peuvent être analysées pour définir une similarité. Elles cherchent à trouver

l'intersection entre les instances des deux ontologies. Plus une telle intersection existe, plus on a de chances d'être en face de concepts similaires.

V.2.4. Méthodes structurelles

Les techniques structurelles reposent sur l'analyse des relations et des propriétés des concepts des ontologies comparées. Elles peuvent être internes ou relationnelles.

- Les méthodes internes

Elles se basent sur l'analyse des caractéristiques propres d'un concept, comme ses attributs. Dans l'analyse de données, on prend en compte la structure interne des concepts, ce qui correspond à l'ensemble de leurs propriétés. On peut comparer les types de données grâce à une égalité stricte, par exemple en considérant que le type *string* est équivalent à lui-même et complètement disjoint de tous les autres types.

- Les méthodes relationnelles

Elles peuvent comparer les ontologies en les assimilant à des graphes. Les comparaisons qui se basent sur la structure taxonomique sont seulement des cas spécifiques.

Ce type de comparaison peut se faire au niveau des concepts voisins dans la hiérarchie, ou en comparant les ensembles de fils directs, ou de feuilles des concepts, c'est-à-dire les fils à n'importe quel niveau de profondeur qui ne possèdent pas de fils.

V.2.5. Méthodes basées sur la sémantique

Ces méthodes passent par l'utilisation d'une ontologie de référence qui sert d'agrément intermédiaire pour générer des correspondances. Les ontologies de références sont constituées par des ressources externes, décrivant une connaissance de domaine, dans un contexte commun aux deux ontologies que l'on souhaite mettre en relation. Il existe de nombreux types d'ontologies de référence pouvant être utilisées pour servir dans ce genre de situation comme exemple DOLCE [39].

VI. Composition des techniques de mapping

Du point de vue de Rahm et Bernstein [35], il existe deux manières de combiner les différentes techniques de mapping au sein de l'approche à mettre au point.

− L'approche hybride

Elle permet de combiner, en un seul algorithme, plusieurs techniques basées sur différents critères et types d'information. Ces méthodes ont l'avantage d'obtenir de meilleures performances que l'exécution de plusieurs algorithmes individuels.

– L'approche composite

Elle permet de combiner les résultats produits par plusieurs algorithmes exécutés de manière indépendante. Les méthodes composites ont l'avantage d'être modulaires et ainsi d'être adaptables plus facilement à différentes structures d'entrée des ontologies.

Cependant, ces méthodes nécessitent de fusionner les résultats produits par les différents algorithmes. Nous distinguons deux approches principales :

- ✓ La composition parallèle, comme décrit dans la figure 2.2, où les résultats des algorithmes individuels sont obtenus de manière indépendante, puis agrégés pour former le mapping final;

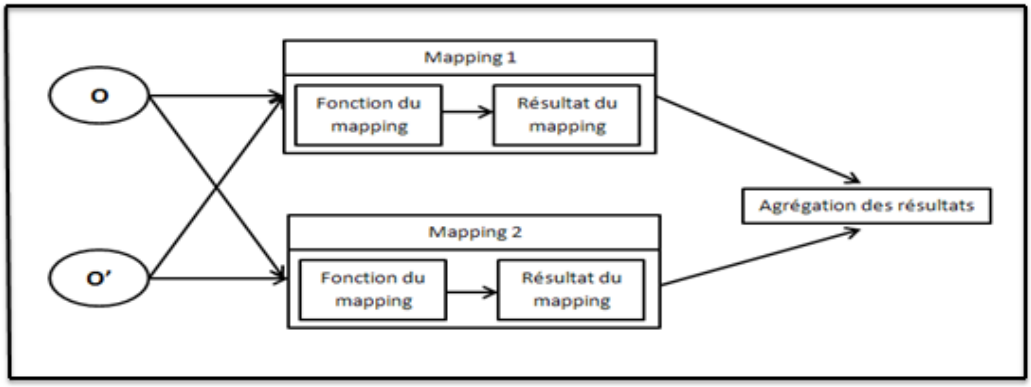


Figure 2.2 Composition parallèle des techniques de mapping

- ✓ La composition linéaire ou séquentielle, comme décrit dans la figure 2.3, où les résultats produits par un algorithme servent d'entrée à un suivant et ainsi de suite. La production du mapping final est ainsi successivement raffinée.

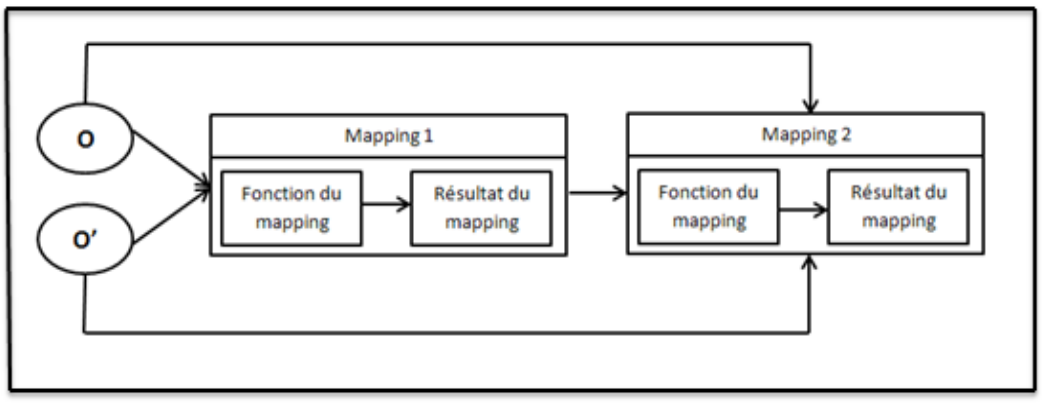


Figure 2.3 Composition linéaire (séquentielle) des techniques de mapping

VII. Classification des méthodes, outils et Framework existants

Une comparaison des méthodes de *mapping* d'ontologies monolingues est présentée dans les prochaines sous-sections selon deux critères :

1. Les techniques de *matching* supportées, afin d'étudier et de comparer l'impact des différentes méthodes de comparaison sur les résultats d'alignement ;
2. Les langages de représentation de l'ontologie et du *mapping*. L'étude de ce critère a pour objectif d'étudier d'une part, le problème d'hétérogénéité des langages de représentation d'ontologie et d'autre part, les langages de représentation des correspondances (*mapping*) ainsi que leurs réutilisations dans les processus d'intégration d'ontologies ;

VII.1. Les techniques de *mapping* supportées

Une classification de quelques approches de *mapping* existantes dans la littérature est présentée dans le Tableau II.1.

Deux critères de comparaison ont été utilisés pour distinguer les différents systèmes de mapping :

1. Les techniques de *matching* employées par les outils de *mapping*, par exemple l'exploitation des termes, de la structure des ontologies ou encore les instances ;
2. Le mode d'intégration, par exemple l'alignement, la fusion ou le *mapping*.

VII.2. Langages d'ontologies et de *mapping*

Le Tableau II.2 énumère les langages de représentation d'ontologies et des *mappings*. En analysant ce tableau nous pouvons constater qu'il y a trois types de représentation :

1. L'ontologie et le *mapping* sont représentés par le même langage. C'est le cas de MOMIS et OntoMap.
2. Le langage de *mapping* est diffèrent du langage de l'ontologie. C'est le cas de MAFRA et RDFT, qui emploient une méta-ontologie pour décrire les *mappings*.
3. Il n'y a pas un vrai langage de *mapping* : soit parce que l'objectif de l'outil est juste de découvrir des mesures de similarité entre les concepts des deux ontologies et non pas de réaliser un *mapping*, c'est le cas pour GLUE et S-Match ; soit parce qu'il s'agit d'un outil de fusion et par conséquent, nous avons une ontologie globale au lieu de liens de *mapping*, comme exemple PROMPT.

Tableau II.1 Les techniques de *mapping* utilisées dans les systèmes existants

Approche	Mode d'intégration	Technique de mapping	Fonctionnement
Chimerae [40]	Fusion d'ontologies	Chimerae aide l'utilisateur à trouver le bon terme en lui proposant une liste de termes utilisés (et aide à résoudre les difficultés d'ordre terminologique)	Environnement basé sur le Web. Effectue la traduction au niveau langage depuis plusieurs formalismes. Chimerae utilise des heuristiques pour trouver les parties de l'ontologie à réorganiser.
OntoMorph [41]	Système de traduction et de transformation d'ontologies	OntoMorph utilise la plupart des techniques d'appariement disponibles.	OntoMorph emploie deux mécanismes de réécriture : syntaxique (pattern matching influencé par PLisp) et sémantique (basé sur PowerLoom, système de représentation de la connaissance permettant l'inférence). OntoMorph est très proche de Chimerae.
Prompt [42]	Fusion et alignement d'ontologies	Prompt commence par chercher les appariements possibles par similarité linguistique, mais se base surtout sur la structure et vérifie les actions de l'utilisateur pour détecter des éventuels conflits.	Proche de Chimerae et d'OntoMorph, Prompt est un module dans l'éditeur d'ontologies [Protege]. Il permet des mises à jour automatiques, se rend compte de conflits et propose une aide pour les résoudre
FCA-Merge [43]	Fusion d'ontologies	Fait une comparaison basée sur les instances : FCA-Merge utilise des techniques linguistiques ; relie les termes clés des instances avec les concepts de l'ontologie (utilise un dictionnaire)	Il transforme les ontologies (parcours des feuilles vers la racine) en des contextes formels (techniques d'analyse de concepts formels) puis ajoute les termes extraits des documents (les instances).
Similarity Flooding [44]	*mapping* de sources génériques	Il est basé sur le calcul de la distance entre les termes et tient compte de la structure (l'appariement de deux concepts influence positivement	La source est convertie en un graphe. Pour chaque appariement réalisé, la similarité augmente pour les paires voisines. Une mesure juge de la qualité des appariements

		leur voisinage). L'appariement est robuste aux cycles.	
Anchor-prompt [45]	*Mapping* d'ontologies	Compare les labels (définition ou exemples pour un concept), la typologie des attributs, la structure.	Nécessite l'introduction préalable d'un ensemble d'appariements pour différents concepts des deux ontologies.
COMA [46]	*matching* des schémas	Combine 13 algorithmes (implémentant les techniques pour l'appariement). Compare les labels, leurs similarités phonétiques, compare la structure, la typologie des attributs, vérifie les synonymes. Utilise un historique des appariements effectués.	L'utilisateur peut interagir pour sélectionner le mode de combinaison des *matchers*.
GLUE [47]	*Mapping* d'ontologies	Utilise les informations sur les instances (nom, taille, …) et sur la fréquence des mots contenus. Permet de prendre en compte le sens commun, les contraintes du domaine et la structure (prise en compte du voisinage).	Met en œuvre trois différentes stratégies d'apprentissage (adaptées au type d'information à acquérir) : une pour les noms, une pour les concepts et une autre pour combiner les deux approches (de manière probabiliste)
Cupid [48]	*matching* des schémas	Approche hybride ; compare les labels (thesaurus, etc.), la typologie des attributs, la structure (transforme le graphe en un arbre pour préserver le contexte défini par le chemin depuis la racine).	Parcourt les ontologies des feuilles vers la racine pour donner plus d'importance aux feuilles afin de pouvoir mettre en correspondance des schémas dont la structure intermédiaire varie mais peu.

Tableau II.2 Classification des systèmes suivant le langage de l'ontologie et de *mapping*

Approche	Langage d'ontologie	Langage du mapping	Commentaires
MAFRA [49]	RDFS	SBO : Semantic Bridge Ontology	SBO est une méta-ontologie. Il permet de présenter les *mappings* entre les concepts, les relations, et les attributs.
RDFT [50]	RDFS	RDFT	RDFT est une méta-ontologie qui décrit les types de *mapping*. Il permet seulement de présenter les *mappings* entre les concepts et entre les propriétés
Prompt [42]	Protege-2000	N'existe pas : Prompt est un outil de Fusion	Supporte le langage RDFS et OWL
GLUE [47]	Taxonomies	mesures de similarité	
S-Match [51]	DAGs	mesures de similarité	
OntoMap [52]	Similaire à OWL Lite	RDFS	Supporte le RDFS
InfoSleuth [53]	OKBC (Open Knowledge Base Connectivity)	N'existe pas	Il génère les *mappings* entre une ontologie et un schéma de donnée
OBSERVER [54]	Description logique	ERA: Extended Relational Algebra	Il génère les *mappings* entre une ontologie et un schéma de données
MOMIS [55]	$ODL_{/3}$	$ODL_{/3}$	Les bases de données relationnelles et semi structurelles (ex. XML) sont traduites par un adaptateur à l'ODL/3
ONION [56]	Taxonomies	Règles	Les schémas des sources de données sont traduits en utilisant des adaptateurs (wrappers)

Par ailleurs, il existe aujourd'hui de nombreuses méthodes automatiques permettant d'aligner des ontologies. Ces méthodes d'alignement sont basées sur des techniques très variées et obtiennent des performances très différentes en fonction des caractéristiques des ontologies à aligner. Dans ce contexte, il existe une campagne annuelle d'évaluation des outils d'alignement, appelée OAEI (*Ontology Alignement Evaluation Initiative*) [57], qui permet de comparer les résultats obtenus par les méthodes d'alignement participantes sur différents jeux d'ontologies. Cette campagne OAEI tente d'évaluer les algorithmes de mapping pour normaliser et améliorer le travail sur l'alignement d'ontologie. Parmi les principaux objectifs de cette initiative nous citons :

- L'évaluation des systèmes d'alignement et d'appariement ;
- La comparaison de la performance des techniques de mapping ;
- L'amélioration des techniques d'évaluation.

VIII. Mapping pour des ontologies multilingues

Les ontologies sont au cœur de la gestion des connaissances et l'utilisation de l'information qui n'est pas seulement écrite en anglais, mais aussi dans de nombreuses autres langues naturelles. Afin de permettre la découverte de connaissances, le partage et la réutilisation des ontologies multilingues, il est nécessaire de soutenir le mapping.

Dans [58], une approche générique a été utilisée qui implique des outils de traduction automatique pour ensuite appliquer des techniques d'appariement vers les ontologies monolingues décrites suivant le langage RDF Shema et qui présente une première étape vers CLOM (Cross Lingual Ontology Mapping), c'est-à-dire vers la réalisation des mappings d'ontologies multilingues dans les domaines des connaissances génériques, qui peuvent être améliorés pour accueillir plus sophistiquement des stratégies de mapping. En particulier, des résultats expérimentaux provenant d'études de cas ont réalisés des mappages ontologiques indépendants qui sont étiquetés suivant deux langues différentes en anglais et en chinois. CLOM est obtenue par les traductions des étiquettes désignant les concepts d'une hiérarchie de l'ontologie source vers le langage naturel de l'ontologie cible en utilisant librement des outils de traduction automatique (Machine Translation (MT)).

Sur la base de cette conclusion, une sémantique orientée CLOM appelée SOCOM (Semantic Oriented Cross lingual Ontology Mapping) a été proposée, qui est spécifiquement conçu pour réduire le bruit introduit par les outils de MT.

Benjamins et al [59] ont identifié le multilinguisme comme l'un des défis pour le Web sémantique, et ont proposé des solutions au niveau ontologique, au niveau annotation et au niveau d'interface. Au niveau ontologique, le soutien devrait être accordé pour les ingénieurs d'ontologies afin de créer des représentations de connaissances dans diverses langues naturelles. Au niveau d'annotation, les outils doivent être développés pour aider les utilisateurs dans l'annotation des ontologies indépendamment des langues naturelles utilisées dans les ontologies données. Enfin, au niveau de l'interface, les utilisateurs doivent pouvoir accéder à l'information dans les langues naturelles de leur choix. Cette approche vise à relever les défis au niveau de l'annotation en particulier.

Considérées comme ontologies poids léger, les thésaurus contiennent souvent de grandes collections de mots. Selon l'Association mondiale WordNet, plusieurs versions de WordNets ont été proposées pour différentes langues afin de rendre l'utilisation de ces bases de connaissances énormes, la recherche a été menée dans le domaine de la fusion des thesaurus. Ceci est exploré lorsque Carpuat et al [60] ont fusionnées des thésaurus qui ont été écrits en anglais et en chinois dans un thésaurus bilingue afin de minimiser le travail répétitif tout en contenant la construction des ontologies des ressources multilingues. Pour des langues indépendantes, une approche basée sur les corpus parallèles a été employée pour fusionner WordNet et HowNet (HowNet présente le WordNet pour la langue chinoise) par alignements des synsets entre les WordNets.

Le Plurilinguisme n'est pas seulement présenté que dans les thésaurus, mais aussi évident en RDF / OWL. Par exemple en 2009, l'ontologie OntoSelect Library a apportée plus de 25% de ses indexés, 1530 ontologies ont été étiquetées dans les langues naturelle autres que l'anglais afin de permettre la découverte de connaissances.

Une approche pour faciliter le partage des connaissances entre les diverses langues naturelles s'appuie sur la notion d'enrichissement d'ontologies avec des ressources linguistiques. [61] vise à soutenir le processus d'enrichissement linguistique des concepts ontologiques au cours du développement d'ontologies. Un outil OntoLing est développé comme un plug-in pour l'éditeur d'ontologie PROTEGE afin de réaliser un tel processus de mapping. Cet enrichissement d'ontologies fournit aux ingénieurs de la connaissance, des données linguistiques riches qui peuvent être utilisés dans CLOM. Cependant, pour les applications informatiques de faire usage de ces données, la standardisation de l'enrichissement est nécessaire. Comme cette exigence n'est actuellement pas inclus dans OWL, il serait difficile de faire usage du grand nombre de techniques d'appariement d'ontologies monolingues qui existent déjà.

Dans [62], une approche d'appariement entre WordNet de Princeton et WordNet Hindi a été soulevée afin de permettre le partage de connaissances du langage naturel. Cette approche se base sur l'utilisation d'un dictionnaire bilingue anglais-hindi afin d'avoir les deux ontologies dans la même langue. Le niveau hiérarchique des relations de généralisation entre concepts est pris en considération. Ce dernier a mené à un rendement de précision plus important par rapport à l'utilisation seulement des relations de synonymes. Notre travail s'inspire sur l'approche proposée dans [62]. Cette approche exploite plusieurs disciplines telles que la recherche d'information, l'apprentissage automatique, et les heuristiques qui se basent sur l'extraction des différentes informations contenues dans les concepts.

Bien qu'il existe déjà un domaine établi de la recherche dans les outils et les techniques de correspondances d'ontologies monolingues [27], seulement que le mapping d'ontologies ne peut plus être limitée à des environnements monolingues, les outils et les techniques doivent être développés pour aider les mappages dans les scénarios inter-langues. Compte tenu de l'ontologie cible et l'ontologie source traduite - maintenant tous les deux représentés dans la même langue naturelle - les techniques d'appariement d'ontologies monolingues peuvent être appliqués tels que proposés dans Alignment API [63].

IX. Conclusion

Comme nous venons de le voir, le mapping d'ontologies constitue à notre sens le meilleur moyen pour mettre en œuvre l'intégration des ontologies. Cette approche constitue le moyen le plus flexible pour lier des domaines sémantiques. La découverte de correspondances prend la grande partie dans les recherches concernant ce domaine due à leur importance incontournable.

Tout au long de ce chapitre, nous avons essayé d'éclaircir la notion du mapping en présentant les différentes définitions des concepts liées au contexte du mapping. Nous avons présenté aussi les différentes étapes de ce processus ainsi que les classifications des techniques du mapping selon plusieurs vues. Les différents systèmes de mapping existants monolingues ainsi que multilingues étaient décrits à la fin de ce chapitre.

Nous avons constaté qu'un bon processus de mapping est celui qui permet de trouver les correspondances les plus pertinentes. Dans le chapitre qui suit, nous essayerons de décrire notre contribution au problème posé par ce mémoire, à savoir la proposition d'un processus de mapping pour l'alignement d'ontologies.

3

Une approche de mapping pour l'alignement d'ontologies

I. Introduction

Le problème actuel lié aux ontologies est qu'étant donné un même domaine ou des domaines connexes, il est possible que plusieurs ontologies soient disponibles, car elles sont développées simultanément par plusieurs communautés différentes. Le choix d'une ontologie particulière et/ou l'exploitation de plusieurs ontologies en même temps devient difficile et surtout si elles sont présentées dans différents langues du langage naturel. Le besoin de comparer les termes des ontologies ou de passer de l'une à l'autre devient donc nécessaire.

Le mapping consiste à découvrir la correspondance sémantique entre les éléments similaires de différentes ontologies. Pour les ontologies multilingues, le passage vers l'utilisation des ressources externes comme un traducteur ou bien un dictionnaire qui soit bilingue ou multilingue est nécessaire. Notre objectif est de réaliser une intégration au niveau structurel des ontologies. L'approche proposée utilise des techniques de recherche d'information, des heuristiques pour définir une mesure de similarité globale calculée en se basant sur des stratégies de mesure de similarité.

Ce chapitre est organisé comme suit. La section II décrit l'algorithme de Lesk qui se base sur la désambiguïsation des sens de mots, nous décrivons son principe générale ainsi que l'algorithme proposé dont lequel notre approche se base et qui est décrite dans la section qui suit. Dans la suite, les différents matchers utilisés seront présentés tel que les matchers linguistiques, syntaxiques et structurels définies pour mesurer les similarités selon plusieurs niveaux hiérarchiques entre les ontologies. Un exemple illustratif de notre approche sera présenté par la suite. La section qui suit dans ce chapitre sera consacrée à la partie expérimentation et évaluation de l'approche proposée.

II. Algorithme de Lesk

La désambiguïsation sémantique des mots est une tâche fondamentale pour la plupart des applications de traitement automatique du langage telles que la traduction automatique, la recherche d'information, l'acquisition automatique de connaissances, la compréhension automatique, l'interaction homme-machine, le traitement de la parole, etc.

L'ambiguïté inhérente aux langues naturelles est un problème récurrent dans le domaine du Traitement Automatique du Langage (TAL). On peut, en effet, rencontrer différents types d'ambiguïté en fonction du niveau d'analyse linguistique où l'on se situe: au niveau syntaxique, du fait des différentes manières possibles d'agencer ceux-ci dans une même langue (catégories syntaxiques et structurelles), au niveau sémantique, avec les différents types d'ambiguïtés lexicales dues aux différents sens des mots (homonymies, polysémie, etc.), etc. S'ajoute à cela le fait qu'une langue peut faire l'objet de différents types d'usages, avec des conséquences importantes sur la manière de gérer les informations.

Parmi les nombreuses méthodes de désambiguïsation existantes. La méthode de Lesk [64] a attiré notre attention dans le cadre de notre travail.

II.1. Principe générale

[64] représente une méthode supervisée qui calcule la similarité entre mots sur la base du chevauchement de leurs définitions respectives dans un dictionnaire électronique. C'est une méthode de désambiguïsation automatique dont le but est de discriminer les sens des mots polysémiques à l'aide d'un dictionnaire électronique. Cette méthode consiste alors à compter les mots communs entre les définitions des sens d'un mot ambigu et les définitions des mots apparaissant dans le contexte (définition) du mot à désambiguïser. Lesk a testé son approche sur quatre MRD (Machine Readable Dictionary) : le Webster's 7th Collegiate (W7), le Collins English Dictionary (CED), *Oxford English Dictionnary* (OED) et le Oxford Advanced Learner's Dictionary of Current English (OALDCE). Le principe de base de cette méthode est de mesurer le chevauchement entre les différentes définitions, dans le dictionnaire, d'un mot ambigu et les définitions de ses voisins immédiats. Par exemple, suivant le dictionnaire OALDCE, Lesk a dit : « Si on considère la cooccurrence des mots anglais *pine* et *cone* dans le même contexte, un programme de désambiguïsation automatique, basé sur cette idée

simple, serait capable de choisir le sens *arbre* du mot *pine* en comptant les intersections entre les différentes définitions de sens des deux mots », une illustration de ce programme entre les deux mots « pine » et « cone » est présentée comme suite :

- Pine

1. *kind of evergreen tree with needle-shaped leaves …;*
2. *Waste away through sorrow or illness*.

- Cone

1. *Solid body which narrows to a point … ;*
2. *Something of this shape whether solid or hollow …;*
3. *Fruit of certain evergreen tree …".*

Dans ce cas, le nombre maximal de mots communs (*evergreen*, *tree*) est donné par l'intersection entre les définitions 1 et 3 respectivement de *pine* et *cone*, ce qui détermine le choix du sens correspondant pour le mot *pine* soumis à l'analyse. Lesk mentionne que ses programmes traitent de manière séquentielle les mots à désambiguïser *(à un moment donné, on compare les définitions des sens d'un mot cible avec toutes les définitions de chaque mot du contexte).* Il suggère cependant que, une fois la décision prise sur le sens d'un mot, seulement la définition de ce sens soit prise en compte pour les désambiguïsations ultérieures, des autres mots.

II.2. Informations syntaxiques. Contexte local/global

Le caractère non-syntaxique de la méthode, et donc son éventuelle utilisation comme supplément d'une méthode de désambiguïsation par la syntaxe, est présenté par Lesk comme son premier avantage. La combinaison des deux types de contextes (les voisins et les co-occurrents syntaxiques) est effectivement plus intéressante. Le second atout de cette méthode est son indépendance par rapport à l'information tirée de contextes plus larges (par exemple, le fait qu'un mot apparait souvent dans le voisinage du mot ambigu relativement au nombre total de ses occurrences dans la base de ressources linguistiques utilisé).

Les performances d'un tel système reposent avant tout sur le choix du dictionnaire utilisé, pour lequel la principale caractéristique à prendre en compte serait le volume d'informations fournies pour chaque définition, c'est-à-dire la longueur des entrées, la fréquence de chevauchement entre les définitions étant corrélée au nombre de termes utilisés pour les décrire. Lesk pose d'ailleurs la question (sans y répondre) de savoir si cette fréquence doit, ou non, être pondérée par la longueur des entrées.

Cette méthode permet de désambiguïser correctement dans 50% à 70% des cas. Cependant, elle présente l'inconvénient d'être très sensible aux mots qui se trouvent dans chaque définition. En effet, le choix des sens basés sur un nombre restreint de mots communs peut être source d'erreurs. Ainsi par exemple, bien que sémantiquement liés, les mots *sandwich* et *Breakfast «Petit déjeuner »* n'ont pas de mots en commun dans leurs définitions respectives suivantes :

- *Two (or more) slices of bread a filling between them.*
- *The first meal of the day (usually in the morning).*

L'algorithme de Lesk les considérant de ce fait comme totalement sémantiquement indépendants. Par ailleurs, la présence ou l'absence d'un mot donné peut radicalement changer le résultat. En effet, dans les cas où aucun mot ne co-occurre entre le contexte et les définitions ambiguës de l'occurrence à désambiguïser, l'approche de Lesk ne permet pas de désambiguïser. La méthode de Lesk servit tout de même de base pour la plupart des travaux postérieurs en désambiguïsation basée sur les dictionnaires informatisés.

II.3. Description de l'algorithme

Le principe de désambiguïsation de Lesk peut être défini comme suit :

1. Pour chaque occurrence de mot ambigu, retrouver tous les sens du mot dans un dictionnaire.
2. Pour chaque sens *S* du mot à désambiguïser :

(a). Consulter sa définition ;

(b). *Score (S)* = le nombre de mots en commun entre la définition du mot à désambiguïser et les définitions des mots co-occurrents dans son contexte ;

(c). Retenir le sens *S* qui maximise *Score(S).*

L'intérêt de cette méthode pour notre approche serait d'en utiliser une variante supervisée qui se baserait non pas sur un dictionnaire traditionnel mais sur des thésaurus de type WordNets après avoir traduit les deux ressources linguistiques dans la même langue comme décrit dans [62]. L'intégration de la désambiguïsation du sens dans notre cas ne se fait pas en se basent sur le gloss d'un concept (définition du concept) mais plutôt sur les mots représentant le synset (concept). Ainsi l'algorithme peut être vu autrement, comme décrit ci-dessous.

1. Pour chaque premier mot d'un concept C d'une ontologie source, retrouver tous les sens du mot dans l'ontologie cible après avoir fait une traduction suivant un dictionnaire bilingue. Ces sens retrouvés représentent les concepts candidats.
2. Pour chaque concept candidat S contenant les traductions du premier mot du concept C :
   - ✓ Consulter les mots contenus dans toute la hiérarchie du concept candidat.
   - ✓ Score (S) = le nombre de mots en commun entre la hiérarchie du concept C et la hiérarchie du concept candidat S.
   - ✓ Retenir le sens S qui maximise Score(S).

La mise en correspondance entre le concept C et les différents concepts candidats S se fait suivant un niveau dont lequel on monte dans la hiérarchie d'hyperonymie (relation de généralisation entre concepts). Ce niveau peut influencer sur l'équivalence des deux concepts mais les résultats sont plus importants si on se base seulement sur la relation de synonymie entre concepts.

III. Approche proposée

Différentes techniques de manipulation d'ontologies existent (fusion, intégration, alignement). Toutes ces techniques consistent à trouver les correspondances entre les concepts des différentes ontologies. Ce travail de génération de *mapping* statique peut très vite s'avérer couteux, car il faut faire toutes les combinaisons possibles entre tous les concepts d'ontologies. Il peut même s'avérer inutile dans certains cas, car certains *mappings* risquent de ne jamais servir.

Cette section a pour but la définition du modèle conceptuel du *mapping*. Nous commençons tout d'abord par une présentation du principe général pour le fonctionnement du mapping abordé. Le reste de la section est dédiée aux différentes étapes de l'algorithme ainsi qu'aux différents *matchers* utilisés pour déterminer la similarité entre les différents éléments impliqués de deux ontologies.

III.1. Principe générale

Afin de produire des mappings entre deux ontologies, il faut que les deux ontologies appartiennent à la même langue pour appliquer des techniques de correspondances monolingues, notre approche se base sur plusieurs étapes, chacune contient un ou plusieurs mécanismes (Voir Figure 3.1).

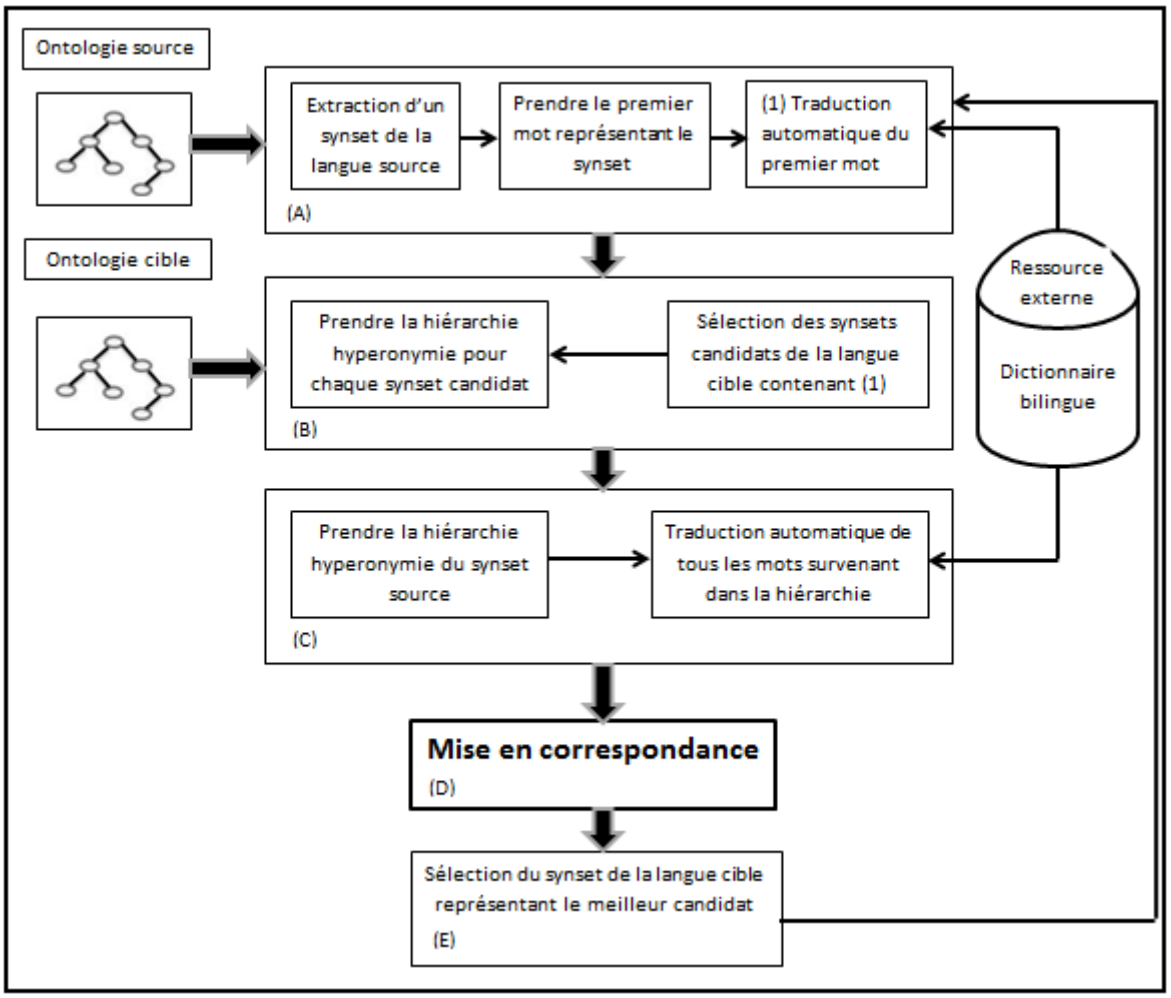


Figure 3.1 L'architecture de l'approche proposée

L'algorithme que nous avons développé se base sur [62], il consiste à faire correspondre chaque concept de l'ontologie source vers le meilleur concept de l'ontologie cible, en se basent sur des heuristiques. L'algorithme utilise les propriétés structurelles de deux WordNets de langues différentes tel que l'anglais et l'espagnol ; une description complète de ces deux WordNets est présentée dans l'annexe A. L'algorithme consiste à faire un alignement automatique pour les synsets des deux WordNets. Les propriétés structurelles définissent la hiérarchie hyperonymie des synsets qui est utilisée dans de nombreux efforts du WSD (Word Sense Disambiguation). Par conséquent, il était le choix naturel pour l'algorithme utilisé.

III.2. Description de l'algorithme

L'algorithme prend en entrée un synset de l'ontologie source et produit en sortie le meilleur synset correspondant de l'ontologie cible. Les différentes étapes de l'algorithme montré dans la figure 3.1 se présentent comme suit.

- (A) Sélection et traduction du premier mot du synset source : le point d'entrée de l'ontologie source vers l'ontologie cible sera de sélectionner le premier mot du synset source parce ce dernier représente le mieux le synset et possède une fréquence d'apparition plus importante par rapport aux autres mots synonymes. Une ressource externe sous forme d'un dictionnaire bilingue sera nécessaire afin de comparer et sélectionner les synsets candidats dans l'ontologie cible.

- (B) Sélection des synsets candidats avec leurs hiérarchies : tous les synsets candidats contenant la traduction du premier mot représentant le synset source seront sélectionnés, une fois les synsets candidats trouvés, la procédure de pondération commence. Le calcul de similarité et la sélection du meilleur candidat est présenté dans les étapes (D) et (E) respectivement.

- (C) Sélection et traduction des mots survenant dans la hiérarchie du synset source :

Nous employons dans notre cas comme dans l'étape (A) des méthodes linguistiques basées sur un dictionnaire bilingue ainsi que des méthodes structurelles afin d'avoir la hiérarchie des deux ontologies dans le même langage naturel.

- (D) Mise en correspondance : cette étape est fondamentale dans le processus de mapping. Elle peut être décomposée en deux étapes décrites comme suite :

(D-1) Calcul de la similarité : les matchers à base de comparaisons syntaxiques et linguistiques sont appliqués sur des couples de mots de deux concepts en prenant en compte toute la hiérarchie hyperonyme afin de mesurer leur similarité terminologique. En raison des limitations des ressources lexicales, nous employons dans notre cas des méthodes syntaxiques tel que la substitution qui vérifie si une chaine de caractères représentant la traduction d'un mot appartenant à un concept dans l'ontologie source est inclut dans la chaine de caractères représentant un mot du concept de l'ontologie cible.

(D-2) Combinaison et génération des mapping candidats (ou des hypothèses de mapping) : les valeurs de similarité retournées dans l'étape précédente sont alors combinées en utilisant la fonction Increment décrite ci-dessous afin de produire une seule valeur de similarité entre chaque couple de concepts, cette valeur de similarité représente le poids du concept candidat par rapport au concept de l'ontologie source. Si une correspondance entre mots de concepts est trouvée, le poids de la hiérarchie des candidats est augmenté. Initialement, les poids de toutes les hiérarchies candidats sont mis à zéro. La fonction suivante est celle utilisé pour calculer la similarité entre deux concepts.

$$Increment = \sum_{i=1}^{n} (Incremet[i]) \ avec \ Increment[i] = \frac{(15 - m[i]) + (15 - n[i])}{2}$$

Avec m[i] et n[i] représente respectivement le niveau hiérarchique pour le mot du concept source et du concept cible. Increment [i] représente la mesure de similarité pour chaque combinaison « i » de couples de mots. De ce fait « n » présente le nombre de combinaison possible pour les quelles, une telle correspondance est trouvé. Cette fonction Increment est calculer entre chaque concept candidat de l'ontologie cible et le concept de l'ontologie source. Le niveau hiérarchique du concept source ainsi que du concept cible est mis a 1, celui des pères de ces concepts est mis a 2 et ainsi de suite.

- (E) Sélection du meilleur candidat: une fois l'ensemble des mappings générés, le concept candidat possédant le poids le plus grand est retourné en sortie. Ce processus de mapping est réalisé tant qu'il existe de synsets sources.

- Le seuil ou bien le niveau de la hiérarchie choisi « N » joue un rôle très important dans la sélection des hyperonymes des deux concepts source et cible, le calcul de la similarité entre les deux concepts se limitent à un niveau inférieur ou égale à N. Pour un concept mis en entrée, si nous avons K candidats dans l'ontologie cible qui correspond à ce concept, le nombre de fois que nous calculerons la similarité présentée par la fonction Increment sera de K fois.
- Comme décrit ci-dessous, une fonction symétrique était nécessaire. Cette fonction diminue en valeur de la similarité entre les concepts qui ont un niveau plus élevé dans la hiérarchie, elle est mentionnée comme étant une heuristique choisie. Le numéro 15 a été sélectionné car la profondeur maximale de la hiérarchie hyperonymie a été trouvé à 15.

III.3. Mécanismes et méthodes du processus de mapping

Dans cette partie, nous décrivons les différentes méthodes d'estimation de similarité tel que les matchers linguistiques, syntaxiques et structurels afin d'augmenter la similarité entre les concepts les plus proches sémantiquement.

III.3.1. Matchers linguistiques et syntaxiques ou matchers terminologiques

Pour les besoins de la phase d'estimation de similarité, on effectue un pré-traitement basé sur les chaines de caractères pour les noms des éléments que l'on souhaite analyser, à savoir les concepts et plus précisément les mots composant le concept (synset). Ce pré-traitement analyse la chaine de caractères des mots contenus dans un synset source et effectue des modifications sur ces mots afin d'avoir une bonne

traduction vers le langage de l'ontologie cible. Ces mots traduits seront pris en considération dans le processus du mapping et aucun mot non signifiant ne sera ignoré afin de garder la sémantique des synsets.

Les *matchers* linguistiques et syntaxiques, appelés *matchers* terminologiques, consistent à comparer les termes. Les méthodes terminologiques comparent des chaines de caractères. Dans notre cas, elles sont appliquées seulement sur les mots composant les synsets et non pas sur les gloss comme décrit dans l'algorithme de Lesk [64] pour trouver les synsets qui sont semblables. Les matchers terminologiques sont utilisés pour vérifier si une chaine de caractères S est la sous-chaine d'une autre chaine T, cela est vraie s'il existe deux chaines de caractères S0 et S00 telles que S0+S+S00 = T. Les deux chaines S et T sont égales (S = T) si et seulement si S est sous-chaine de T et T est sous-chaine de S. dans notre approche, on s'intéresse seulement au cas où S+S00=T, avec S00 représente la terminaison de la chaine de caractères T. Cette technique est intéressante dans le cas du passage de l'anglais vers l'espagnol afin de sélectionner tous les synsets espagnols tels qu'ils soient présentés en pluriel ou bien en singulier pour les noms et à l'infinitif ou bien conjugués pour les verbes.

Dans l'implémentation de notre architecture, nous utilisons comme ressource externe Google Translate API (Application Programming Interface). Le pré-traitement basé sur les chaines de caractères est nécessaire dans l'utilisation de l'API Google parce que cette dernière ne reconnait pas les mots composés séparés par des soulignés (underscores) et la traduction ne sera pas faite correctement.

Puisque les deux ontologies utilisées sont de type WordNet et ce dernier utilise les soulignés pour les mots composés, le pré-traitement basé sur les chaines de caractères sera nécessaire. Le remplacement des soulignés par des espaces simples ainsi que les majuscules par des minuscules lors de la comparaison des chaines de caractères sont les seuls types de normalisation utilisés afin de garder le sens des concepts. La suppression des chiffres, l'élimination des mots de liaison (par exemple : « and, or … »), La suppression des signes accentués (par exemple : « é, à, ù …») pour les langues occidentales ne sera pas pris en compte parce que habituellement ce type de normalisation est utilisé pour comparer des textes longs et non pas deux chaines de caractères. Une fois la traduction est faite, une autre normalisation est nécessaire, les mots traduits composés sont séparés par des espaces, un pré-traitement se fait dans ce cas en replacent les espaces simples par des underscores afin de trouver des équivalences dans l'ontologie cible.

- **Matchers syntaxiques**
  - ✓ **Matcher à base d'égalité de chaines de caractères**

La méthode d'égalité des chaines de caractères est une mesure de similarité entre deux chaines. Elle est définie par f : S x S → {0,1}.
Cette mesure retourne 1 si les deux chaines sont égales et elle retourne 0 sinon. Formellement : f(x, y)=1 et f(x, y)=0 ∀ x#y.

Cette méthode comme toute autre méthode syntaxique est exécuté après une certaine normalisation syntaxique des deux chaines de caractères à comparer.

  - ✓ **Matcher à base de la distance de substring**

Ce type de matcher permet de calculer la mesure de dis-similarité entre deux chaines de caractères. Cette mesure peut être utile dans le cas où on considère que deux chaines de caractères sont semblables, ou si l'une des deux est une sous-chaine de l'autre. Formellement, la distance substring est une fonction définie par :

F : S x S → {0,1} tel que :

$$\begin{cases} F(x, y)=1 \text{ si } \forall (x, y) \in S \times S, \exists (a, b) \in (S \times S) \text{ tel que } x=a+y+b \text{ ou } y=a+x+b ; \\ F(x, y)=0 \text{ sinon.} \end{cases}$$

Il est facile de voir que cette mesure est en effet une similarité entre la totalité des chaines. On a pu également considérer une similitude entre des sous parties de ces chaines. Cette définition peut être employée pour mesurer la similitude basée sur le plus grand préfixe commun.

- **Matchers linguistiques à base d'informations auxiliaires**

Les méthodes basées sur un langage se fondent sur des techniques de traitement du langage naturel (NLP : Natural Language Processing) afin de trouver des associations entre les entités ou les classes. Ces méthodes exigent l'utilisation de ressources externes, par exemple les dictionnaires. Elles se basent essentiellement sur les liens sémantiques entre les termes en langage naturel, utilisant un lexique ou un dictionnaire externe. Un dictionnaire bilingue sera nécessaire dans notre algorithme afin d'appliquer les techniques de correspondances d'ontologies monolingues.

III.3.2. Matchers structurels

L'utilisation de ce type des matchers permet d'augmenter la valeur de similarité ou encore à produire de nouveaux *mappings*. L'idée principale est basée sur la notion d'équivalence de concepts de deux ontologies qui se défini par : si les deux concepts c1 et c2 de l'ontologie source « O » sont liés par une relation « R », leurs concepts correspondants c'1 et c'2 de l'ontologie cible « O' » doivent être liés par la même relation « R ». Cette relation « R » qui est une relation structurelle peut être, par exemple sous-concept ou sur-concept. On s'intéresse seulement sur la relation de sur-concept selon un niveau N dans la hiérarchie afin d'exploiter la structure d'ontologies.

III.3.3. Combinaison des *matchers* et génération des hypothèses de *mapping*

Nous utilisons dans notre implémentation la combinaison parallèle des matchers ; le résultat des différents matchers individuels (listés ci-dessus) sera combiné pour identifier un seul mapping candidat (appelé aussi hypothèse de mapping).

Ces hypothèses entre les couples de concepts vont être modifiées et améliorées en utilisant des règles et des méthodes de comparaison exploitant les relations ontologiques, c'est-à-dire les relations hiérarchiques et sémantiques. La figure 3.2 schématise la composition parallèle des matchers syntaxiques et linguistiques au sein du processus. Après une normalisation de deux chaines de caractères comparées S et T.

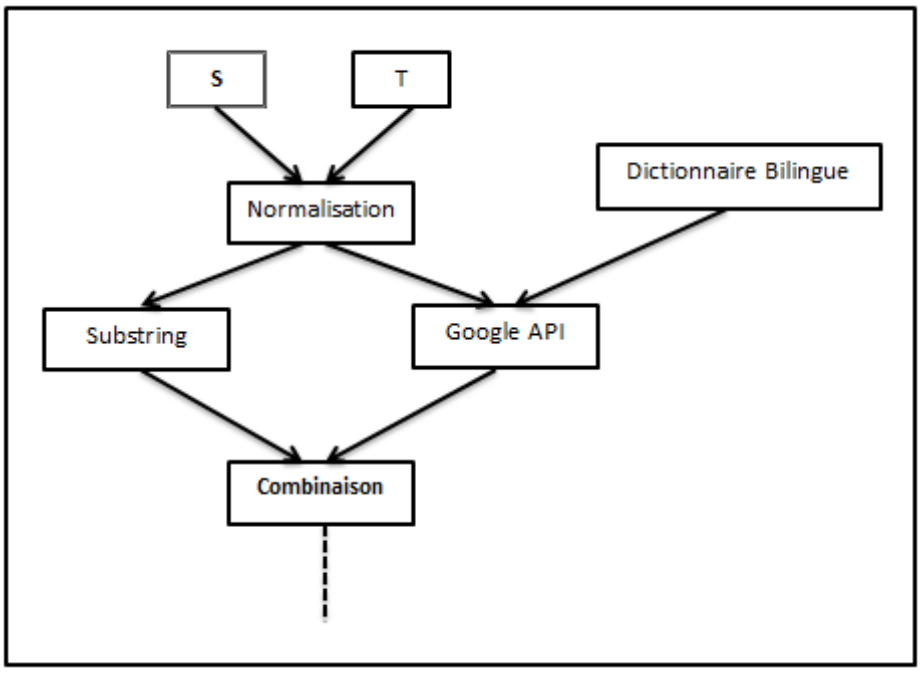


Figure 3.2 Composition des matchers basés sur la comparaison des chaines de caractères

III.4. Extraction des mapping

Lorsque les valeurs de similarité ont été estimées, on doit convertir ces valeurs numériques en termes de correspondances entre les concepts. Dans ce cas, le mapping qui semble être le plus important est retourné en sortie. La paire de concepts choisie passe à une valeur de similarité de 1, alors que toutes les paires impliquant un des deux éléments de celle-ci passent à une similarité de 0.

La façon la plus naturelle d'analyser ces résultats consiste à créer des équivalences entre des paires d'éléments. Pour cela, nous avons extrait pour chaque concept de la première ontologie le concept le plus similaire de la seconde, à condition que leur similarité soit suffisamment élevée, par rapport à toutes les autres similarités des autres paires de concepts.

On souhaite de plus éviter les équivalences multiples, c'est-à-dire les cas où un concept serait estimé comme équivalent à deux ou plusieurs concepts provenant de l'autre ontologie. Enfaite, le choix d'admettre qu'un concept d'une ontologie source possède deux ou plusieurs concepts de l'ontologie cible avec le même degré d'équivalence, le concept candidat qui possède le même offset (identifiant du synset) s'il existe est retourné en sortie sinon le premier concept candidat possédant le poids le plus grand est retourné. Chaque élément d'une ontologie ne peut être automatiquement mis en correspondance qu'avec un seul autre élément.

Notre approche d'extraction de mappings se limite à des équivalences en prenant en compte la structure taxonomique des ontologies présentée par les relations d'hyperonymie entre les concepts. La méthode de mise en correspondance est testée et ses résultats sont discutés dans les sections qui vont suivre.

III.5. Exemple illustratif

Dans cette partie, afin de mieux comprendre le fonctionnement de l'algorithme, nous proposons un exemple applicatif qui met en correspondance un synset anglais vers son équivalent en espagnol, le traitement du mapping dans l'autre sens (mise en correspondance entre l'espagnol et l'anglais) se fait de la même façon. La figure 3.3 illustre la représentation hiérarchique du synset anglais mis en entrée {express, state} et deux synsets candidats qui sont présentés par {expresar, indicar} et {expresar, mostrar}. Ces deux synsets sont les plus pertinents parmi un ensemble de synsets candidats.

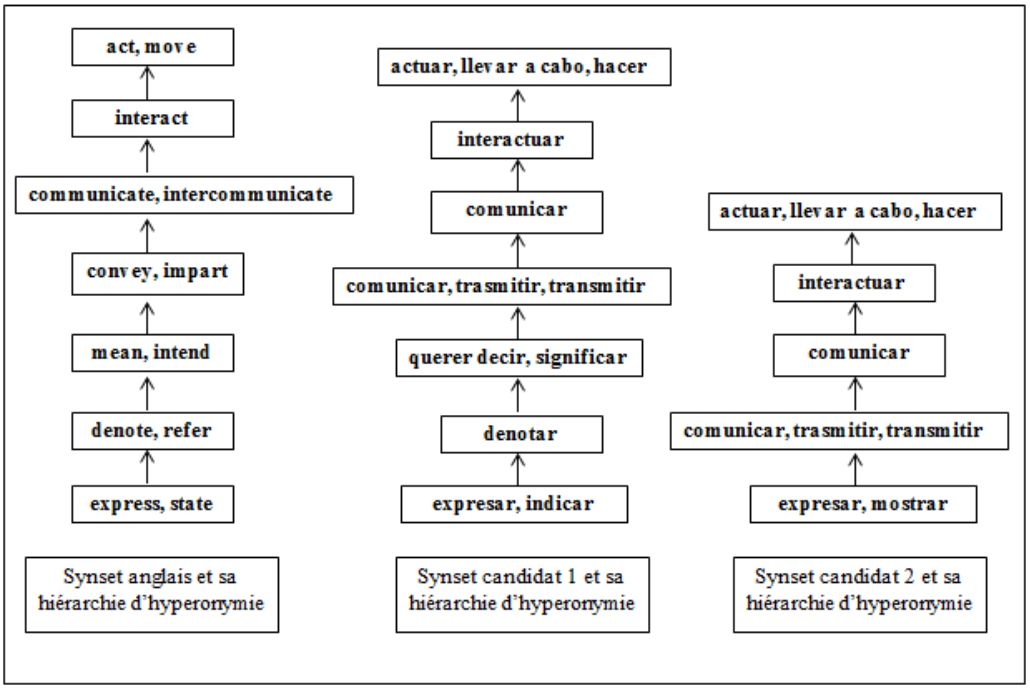


Figure 3.3 Exemple illustratif de l'approche proposée

- Le synset mis en entrée a été extrait des synsets verbes qui appartiennent au mot « express » et plus exactement le 4ème synset contenant ce dernier parmi cinq synsets.
- La traduction des mots appartenant à la hiérarchie du synset anglais est présentée comme suit :
  - express : expresa ;
  - state : estado ;
  - denote : denotar ;
  - refer : se refieren ;
  - mean : significa ;
  - intend : la intencián de ;
  - convey : transmitir ;
  - impart : impartir ;
  - communicate : comunicar ;
  - intercommunicate : intercomunicar ;
  - interact : interactuar ;
  - act : acto ;
  - move : mueven.

- <u>Premier Candidat :</u>
  - ✓ Le mot « expresa » qui correspond à la traduction du mot anglais « express » contenu dans le synset de base est équivalent au mot «expresar » qui se trouve dans le synset espagnol {expresar, indicar} d'où :

    Increment [1]= [(15-1) + (15-1)]/2=14 ;

- ✓ Le mot « denotar » qui correspond à la traduction du mot anglais « denote » contenu dans le synset anglais {denote, refer} est équivalent au mot «denotar » qui se trouve dans le synset espagnol {denotar} d'où :
  Increment [2]= [(15-2) + (15-2)]/2=13 ;
- ✓ Le mot « significa » qui correspond à la traduction du mot anglais « mean » contenu dans le synset anglais {mean, intend} est équivalent au mot « significar » qui se trouve dans le synset espagnol {querer decir, significar} d'où : Increment [3]= [(15-3) + (15-3)]/2=12 ;
- ✓ Le mot « transmitir » qui correspond à la traduction du mot anglais « convey » contenu dans le synset anglais {convey, impart} est équivalent au mot « transmitir » qui se trouve dans le synset espagnol {comunicar, trasmitir, transmitir} d'où : Increment [4]= [(15-4) + (15-4)]/2=11 ;
- ✓ Le mot « comunicar » qui correspond à la traduction du mot anglais « communicate » contenu dans le synset anglais {communicate, intercommunicate} est équivalent au mot « comunicar » qui se trouve à la fois dans le synset espagnol {comunicar, trasmitir, transmitir} et dans le synset {comunicar} d'où :

Increment [5]= [(15-5) + (15-4)]/2=10,5 et Increment [6]= [(15-5) + (15-5)]/2=10;

- ✓ Le mot « interactuar » qui correspond à la traduction du mot anglais «interact» contenu dans le synset anglais {interact} est équivalent au mot « interactuar » qui se trouve dans le synset espagnol {interactuar} d'où :
  Increment [7]= [(15-6) + (15-6)]/2=9 ;

- Par conséquent, le poids net de la hiérarchie du premier candidat est de 79,5.
- <u>Deuxième Candidat :</u>
  - ✓ Le mot « expresa » qui correspond à la traduction du mot anglais « express » contenu dans le synset de base est équivalent au mot «expresar » qui se trouve dans le synset espagnol {expresar, mostrar} d'où :
    Increment [1]= [(15-1) + (15-1)]/2=14 ;
  - ✓ Le mot « transmitir » qui correspond à la traduction du mot anglais « convey » contenu dans le synset anglais {convey, impart} est équivalent au mot «transmitir» qui se trouve dans le synset espagnol {comunicar, trasmitir, transmitir} d'où : Increment [2]= [(15-4) + (15-2)]/2=12 ;
  - ✓ Le mot « comunicar » qui correspond à la traduction du mot anglais «communicate» contenu dans le synset anglais {communicate,

intercommunicate} est équivalent au mot « comunicar » qui se trouve à la fois dans le synset espagnol {comunicar, trasmitir, transmitir} et dans le synset {comunicar} d'où : Increment [3]= [(15-5) + (15-2)]/2=11,5 et
Increment [4]= [(15-5) + (15-3)]/2=11;

- ✓ Le mot « interactuar » qui correspond à la traduction du mot anglais «interact» contenu dans le synset anglais {interact} est équivalent au mot «interactuar» qui se trouve dans le synset espagnol {interactuar} d'où :
Increment [5]= [(15-6) + (15-4)]/2=10 ;

- Par conséquent, le poids net de la hiérarchie du deuxième candidat est de 58,5.

- La hiérarchie du candidat correspondant au synset {expresar, indicar} à plus de nombre de correspondance par rapport au nombre de correspondances de la hiérarchie de l'autre candidat {expresar, mostrar}. Le poids finale de la première hiérarchie est de 79,5 par contre le poids de la deuxième hiérarchie est de 58,5. De ce fait le synset {expresar, indicar} est mappé avec le synset {express, state} sachant que ces deux derniers possèdent le même offset, d'où l'obtention d'un mapping correcte.

- Pour certains synsets concernant leurs structures hiérarchiques, il peut y avoir qu'un synset dans un certain niveau possède deux ou plusieurs ancêtres, chacun de ses ancêtres possède sa propre hiérarchie d'hyperonymie dont chacun des ancêtres de ces derniers peuvent de même avoir d'autres ancêtres et ainsi de suite. De ce fait, le travail de mis en correspondance sera itératif sur certains concepts et long à la fois. Nous avons préféré éliminer les doublons (les synsets) qui apparaissent plus qu'une fois afin de garder qu'une seule instance de chaque synset et parcourir toute la hiérarchie du synset source et des synsets candidats. Par exemple pour le synset anglais {student, pupil, educatee}, la structure hiérarchique de ce dernier est présentée comme suite :

1 => {student, pupil, educatee}
2 => {enrollee}
3 => {learner, scholar}
4 => {person, individual, someone, somebody, mortal, human, soul}
5 => {life form, organism, being, living thing}
6 => {entity, something}
5 => {causal agent, cause, causal agency}
6 => {entity, something}

- Nous remarquons que le quatrième synset possède deux ancêtres, chacun a sa propre hiérarchie. Le sixième synset apparait deux fois, dans ce cas-là, la deuxième instance de ce dernier sera supprimée. Ce processus s'applique à la fois sur le WordNet anglais ainsi que le WordNet espagnol.

IV. Expérimentation et évaluation de l'approche

Afin d'évaluer et valider la contribution présentée dans ce mémoire, une phase d'expérimentation s'avère indispensable. Cette phase a pour objectif d'étudier les performances de notre approche de mapping et de montrer ses avantages sur les deux WordNets utilisés. En outre, ceci nous permettra aussi d'identifier les contraintes et les insuffisances de notre approche et d'en proposer des solutions.

Une présentation de l'environnement de développement qui va supporter notre application ainsi que les différentes ressources utilisées sont décrites dans un premier lieu dans cette section. La suite sera dédiée aux résultats des similarités entre concepts. L'étude de ces résultats nous permettra une évaluation plus précise de notre approche et l'examen de son intérêt du point de vue pratique.

IV.1. Technologies et outils de développement

Le prototype a été développé sur une machine possédant les caractéristiques suivantes : un processeur Intel® Core ™ 2 Duo CPU T6670, 2.20 GHz et une mémoire de 4 GO, l'ensemble est piloté par le système d'exploitation Windows SEVEN Professionnel 64 bits, la dernière version du système d'exploitation Microsoft, avec des outils Open source développés en JAVA. Les outils et langages utilisés pour la manipulation des données ainsi que l'implémentation de l'interface utilisateur sont décrits comme suite.

- ✓ **Le langage JAVA**

Pour le langage de programmation, notre choix s'est porté sur le langage JAVA, et cela parce que :

- JAVA est un langage orienté objet simple ce qui réduit les risques d'incohérence ;
- JAVA est portable. Il peut être utilisé sous Windows, sous Linux, sous Macintosh et sur d'autres plateformes sans aucune modification. JAVA est donc un langage multiplateformes, ce qui permet aux développeurs d'écrire un code qu'ils peuvent exécuter dans tous les environnements ;
- JAVA possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers ainsi que beaucoup de fonctionnalités qui peuvent être utilisées pour développer des applications diverses ;

- Il existe une multitude de bibliothèques développées et fournies pour être utilisées en JAVA, les API des autres langages autres que JAVA ne sont pas finalisées et doivent encore être mises à jour.

✓ **Environnement de développement**

Pour le choix de l'environnement de développement, on a opté pour NetBeans 6.9.1 (La version actuelle, NetBeans 7.0 (sortie le 20 avril 2011), est disponible en 23 langues) car il possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web) ;
- Une plateforme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plugins dont le but est de fournir un environnement robuste et convivial pour les développeurs des logiciels, outils et systèmes informatiques ;
- Support de plusieurs plates-formes d'exécution : Windows, Linux, Mac OS;
- Les principaux modules de base pour NetBeans fournis concernent le langage de programmation Java. Les modules agissent sur des fichiers qui sont inclus dans l'espace de travail (appelé workspace). L'espace de travail regroupe les projets qui contiennent une ou plusieurs arborescences de fichiers ;
- Bien que développé en Java, NetBeans présente de très bonnes performances d'exécution, car il n'utilise pas Swing pour l'interface homme-machine, mais il utilise un toolkit particulier nommé SWT associe à la bibliothèque JFace. SWT (Standard Widget Toolkit) est développé en Java, en utilisant au maximum les composants natifs fournis par le système d'exploitation sous-jacent. JFace utilise SWT et propose une API pour faciliter le développement des applications nécessitant des interfaces graphiques.
- La construction incrémentale des projets Java grâce à son propre compilateur, qui permet en plus de compiler le code même avec des erreurs, de générer des messages d'erreurs personnalisés, de sélectionner la cible, …etc.

- ✓ **JWNL API**

JWNL est utilisée pour accéder à WordNet dans de multiples formats, cela nous a permis d'accéder à l'ontologie WordNet de Princeton à partir de notre application. JWNL permet la découverte des relations hiérarchiques et de transformation morphologique. Elle est compatible avec les versions WordNet 1.6 à 3.0, et est implémenter complétement en Java. Le courant et (surtout) la version la plus stable est la JWNL 1.3 (qui peut étre téléchargé directement depuis le site http://sourceforge.net/projects/jwordnet/). JWNL 1.4 est en cours d'élaboration. Elle est publiée sous la licence BSD (Berkeley Software Distribution licence), et est utilisé par de nombreuses applications commerciales, y compris un plug-in de GATE (General Architecture for Text Engineering).

- ✓ **WordNet 1.6**

Pour la mise en correspondance entre les ontologies, on a choisi d'utiliser deux WordNets de langues différentes tel que l'anglais et l'espagnol (première et troisième langue respectivement parlée dans le monde), pour comparer les termes et donner la mesure de similarité entre les concepts. Le choix de WordNet 1.6 été cause de diverses raisons :

- C'est la base la plus riche et la plus générale qui contient tous les domaines ;
- WordNet utilise la langue anglaise, et c'est la langue la plus utilisé dans le monde. Des versions de ce dernier existent pour d'autres langues, la raison pour laquelle on à utiliser le WordNet espagnol qui est moins riche par rapport à la version anglaise qui reste la plus complète à ce jour ;
- WordNet est open source et disponible sur internet ;
- Il possède plusieurs API pour une exploitation en utilisant le langage JAVA.

- ✓ **MySQL Connector/ODBC**

MySQL Connector permet de se connecter à un serveur de base de données MySQL grâce à l'utilisation de l'API ODBC (Open DataBase Connectivity). On peut ainsi exploiter des bases de données MySQL depuis notre application développée en JAVA. Les mises à jour régulières dont profite MySQL Connector/ODBC prennent en compte les évolutions de MySQL. Cette API était nécessaire d'être utilisé afin d'accéder aux données du WordNet espagnol qui est décrit sous forme d'une base de données. Une description complète de ce dernier est présentée dans l'annexe A.

✓ **Dictionnaire Bilingue**

Nous avons utilisé le dictionnaire gratuit Google API [65] qui permet de faire la traduction des termes anglais en espagnol et vice versa, en exploitant ces données via internet. Google API permet seulement de donner une seule traduction pour un terme mis en entré, une seule traduction sera généré que ce soit pour les noms ou bien pour les verbes, cette traduction est la plus représentatif du mot mis en entré. Puisque les termes des deux ontologies sont limités, on a voulu faire une traduction automatique de tous les mots et faire une sauvegarde des données afin de travailler en local et accélérer ainsi le fonctionnement de l'algorithme.

IV.2. Les métriques utilisées pour l'évaluation

Dans le domaine du *mapping* d'ontologies, les mesures de *Précision*, *Rappel* et *Fmesure* [66] sont des métriques largement employées pour évaluer la qualité des alignements obtenus. L'OAEI (Ontology Alignement Evaluation Initiative) [57] retient ces mesures pour l'évaluation de la qualité de l'alignement.

L'objectif principal de ces mesures est l'automatisation du processus de comparaison des méthodes d'alignement ainsi que l'évaluation de la qualité des alignements produits.

- **Précision :**

La précision représente une mesure de l'efficacité du système par rapport au nombre de cas traités. Pourtant, elle n'est pas suffisante pour caractériser le comportement global du système parce qu'une précision de 100% n'indique pas toujours un fonctionnement parfait. Par exemple, un système qui ne traite que seulement 2 cas d'un total de 10, même pour une précision de 100% (2 réponses correctes de 2 cas traités), ne représente pas un système satisfaisant. Formellement, la précision peut être présentée comme-suit.

$$\text{Précision} = 100\ x\ \frac{nb.correspondances\ correctes}{nombre\ de\ cas\ traités}$$

- **Rappel:**

En revanche, le rappel tient compte de cet aspect, en indiquant pour l'exemple considéré une performance de 20% pour des traitements corrects par rapport aux nombre total des cas à traiter. De ce fait, ces deux mesures sont complémentaires. Formellement, le rappel est présenté comme suit.

$$\text{Rappel} = 100\ x\ \frac{nb.correspondances\ correctes}{nombre\ de\ cas\ \text{à}\ traiter}$$

- **Fmesure**

La métrique *Fmesure* est une moyenne harmonique. Elle combine la *précision* et le *rappel* et se définit par la formule suivante :

$$\text{Fmesure} = 2\ x\ \frac{Précision\ x\ Rappel}{Précision +\ Rappel}$$

Etant donné que nous nous focalisons d'une part, sur la précision des résultats de *mapping*, c'est-à-dire connaitre l'exactitude des correspondances fournies par le système, et d'autre part, sur le nombre des correspondances correctes non détectées par le système, nous utilisons ainsi principalement dans nos expérimentations les deux métriques précision et rappel ainsi que la moyenne harmonique Fmesure.

IV.3. Evaluation

WordNet de Princeton consiste à identifier de nouveaux synsets et la création des liens appropriés entre les synsets, qui reste une tâche permanente, un nouveau synset sera au moins lié a son parent hyperonyme. Un synset dans une base de données plus mature est succeptible d'avoir une plus grande « richesse » en étant liée à plus de synsets, alors que dans un nouveau WordNet, une plus grande proportion de synsets n'aura que le lien parental, c'est la raison pour laquelle, on s'intéresse beaucoup plus sur la relation taxonomique d'hyperonymie parce que les relations taxonomiques (hyperonymie/hyponymie) dominent habituellement l'appariement entre les ontologies.

Dans le cadre de notre prototype, nous nous sommes focalisés sur la structure taxonomique de deux ontologies utilisant le même langage après avoir fait une traduction de l'une des deux ontologies dans le langage naturel de l'autre. Cependant, on peut repérer des paires de concepts qui sont similaires en se basent seulement sur des relations de synonymies dont la similarité ne prend pas en compte les informations concernant la structure des ontologies. Celle-ci peut permettre d'augmenter ou diminuer la précision des résultats.

Les deux tableaux III.1 et III.2 montrent respectivement les valeurs de précision, de rappel et Fmesure obtenus tout au long du processus de mise en correspondance pour les deux mapping traités, à savoir, le mapping du WordNet anglais vers le WordNet espagnol et inversement en fonction du niveau de la hiérarchie.

Tableau III.1 Résultats d'évaluation du mapping de l'anglais vers l'espagnol

Niveau de hiérarchie	Noms			Verbes			Total		
	Présicion(%)	Rappel(%)	Fmesure(%)	Présicion(%)	Rappel(%)	Fmesure(%)	Présicion(%)	Rappel(%)	Fmesure(%)
1	62.25	26.06	39.10	43.12	17.80	25.20	59.32	24.79	34.97
2	59.47	24.90	35.10	37.03	15.29	21.64	56.03	23.41	33.02
3	60.04	25.14	35.44	37.03	15.29	21.64	56.51	23.61	33.31
4	60.47	25.32	35.69	37.09	15.31	21.67	56.88	23.76	33.52
5	60.68	25.41	35.82	37.05	15.30	21.66	57.05	23.84	33.63
6	60.63	25.38	35.78	37.05	15.30	21.66	57.01	23.82	33.60
7	60.51	25.33	35.71	37.09	15.31	21.67	56.92	23.78	33.55
8	60.44	25.30	35.69	37.07	15.30	21.66	56.85	23.75	33.50
9	60.40	25.29	35.65	37.09	15.31	21.67	56.83	23.74	33.49
10	60.41	25.29	35.65	37.11	15.32	21.69	56.84	23.75	33.50
11	60.41	25.29	35.65	37.11	15.32	21.69	56.84	23.75	33.50
12	60.41	25.29	35.65	37.11	15.32	21.69	56.84	23.75	33.50
13	60.37	25.28	35.64	37.11	15.32	21.69	56.81	23.73	33.48
14	60.40	25.29	35.65	37.11	15.32	21.69	56.83	23.74	33.49
15	60.38	25.28	35.64	37.11	15.32	21.69	56.81	23.73	33.48
Moyenne	60.48	25.32	35.86	37.49	15.48	21.91	56.96	23.80	33.57

Tableau III.2 Résultats d'évaluation du mapping de l'espagnol vers l'anglais

Niveau de hiérarchie	Noms			Verbes			Total		
	Présicion(%)	Rappel(%)	Fmesure(%)	Présicion(%)	Rappel(%)	Fmesure(%)	Présicion(%)	Rappel(%)	Fmesure(%)
1	71.14	53.53	61.09	46.31	40.75	43.35	66.80	51.57	58.21
2	71.06	53.47	61.02	42.36	37.28	39.66	66.04	50.98	57.54
3	71.00	53.43	60.97	42.45	37.35	39.74	66.00	50.96	57.51
4	71.03	53.45	61.00	42.29	37.21	39.59	66.00	50.96	57.51
5	71.12	53.51	61.07	42.26	37.19	39.56	66.07	51.01	57.57
6	71.13	53.52	61.08	42.24	37.17	39.54	66.08	51.01	57.57
7	71.13	53.52	61.08	42.19	37.12	39.49	66.07	51.00	57.57
8	71.10	53.50	61.06	42.20	37.13	39.50	66.05	51.00	57.56
9	71.09	53.49	61.05	42.20	37.13	39.50	66.04	50.98	57.54
10	71.08	53.49	61.04	42.20	37.13	39.50	66.03	50.98	57.54
11	71.07	53.48	61.03	42.20	37.13	39.50	66.02	50.97	57.53
12	71.07	53.48	61.03	42.20	37.13	39.50	66.02	50.97	57.53
13	71.07	53.48	61.03	42.20	37.13	39.50	66.02	50.97	57.53
14	71.07	53.48	61.03	42.20	37.13	39.50	66.02	50.97	57.53
15	71.07	53.48	61.03	42.20	37.13	39.50	66.02	50.97	57.53
Moyenne	71.08	53.49	61.04	42.51	37.41	39.80	66.09	51.02	57.59

On remarque que le premier mapping (voir Tableau III.1) produit un faible rappel (beaucoup de *mappings* ignorés) et une faible précision par rapport au deuxième (voir Tableau III.2). Le rappel du premier mapping reste beaucoup plus faible par rapport au deuxième, cela est dû à la richesse du WordNet anglais qui contient bien beaucoup plus de concepts par rapport au WordNet espagnol.

Nouns avons éxpérimenté notre approche en fonction du niveau hiérarchique compris entre 1 (niveau du synset source et cible) et 15 (niveau maximum de la hiérarchie du synset source et cible) afin d'evaluer les résultats de notre étude, mais nous avons remarqué que malgré l'augmentation du niveau de la hiérrachie, le nombre de correspondances ne change pas a grande échelle. Les résultats pour un mapping a des niveaux inférieurs est plus que celle accordé a des niveaux supérieurs. En effet, les synsets ayant des niveaux plus élevés sont plus loin – en termes de sens qui les portent – du synset originale. D'une autre part, suivant l'approche étudiée, l'augmentation du poids hiérarchique des synsets candidats depend du niveau des synsets anglais et espagnols plus que d'une maniére symétrique.

En raison des limitations des ressources lexicales utilisées, nous avons employé des techniques de substitution afin de diminuer le nombre de mapping qui peuvent être ignorés mais cela reste seulement valable dans le cas du mapping de l'anglais vers l'espagnol. Puisque le WordNet espagnol est présentée sous forme d'une base de données, on utilise des requêtes SQL pour interroger ce dernier.

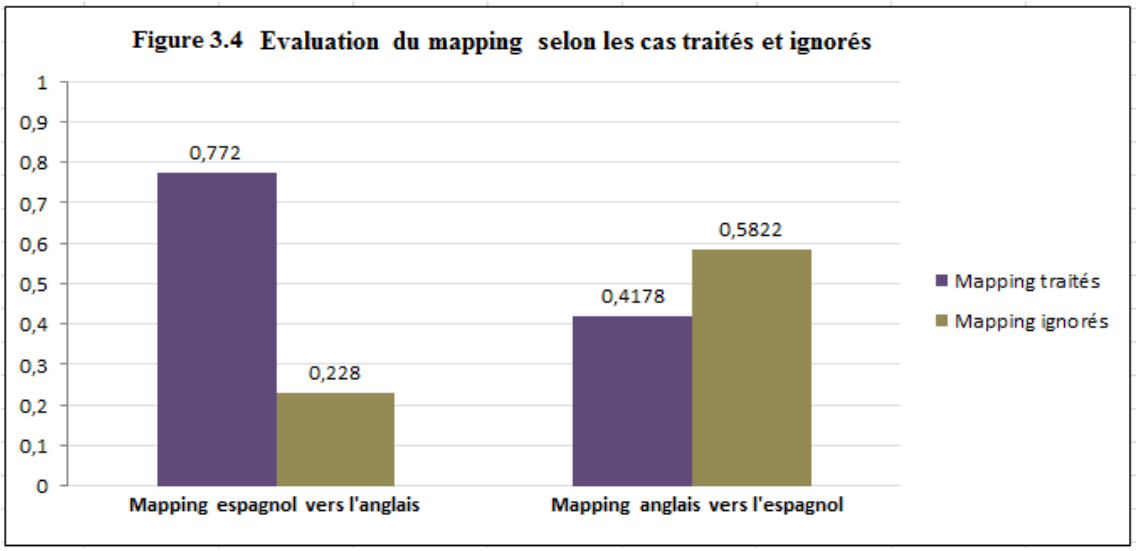
VI.4. Résultats et discusions

Nous comparons et évaluons les résultats de notre approche suivant deux scénarios. La figure 3.4 illustre les résultats de l'étude suivant le nombre de cas traités et ignorés dans les deux mappings effectués (le mapping de l'anglais vers l'espagnol et de l'espagnol vers l'anglais). La figure 3.5 illustre les résultats finaux des deux mappings, la comparaison dans le deuxième scénario est faite en se basant sur trois métriques qui sont présentées comme suit.

(i) le pourcentage de véritables *mappings* positifs par rapport au nombre de mapping traités, c'est-à-dire les correspondances correctes présentées par la précision;

(ii) le pourcentage de véritables *mappings* positifs par rapport au nombre total de mapping, autrement dit, les cas traités avec correspondances correctes par rapport au nombre de cas à traiter présentés par le rappel ;

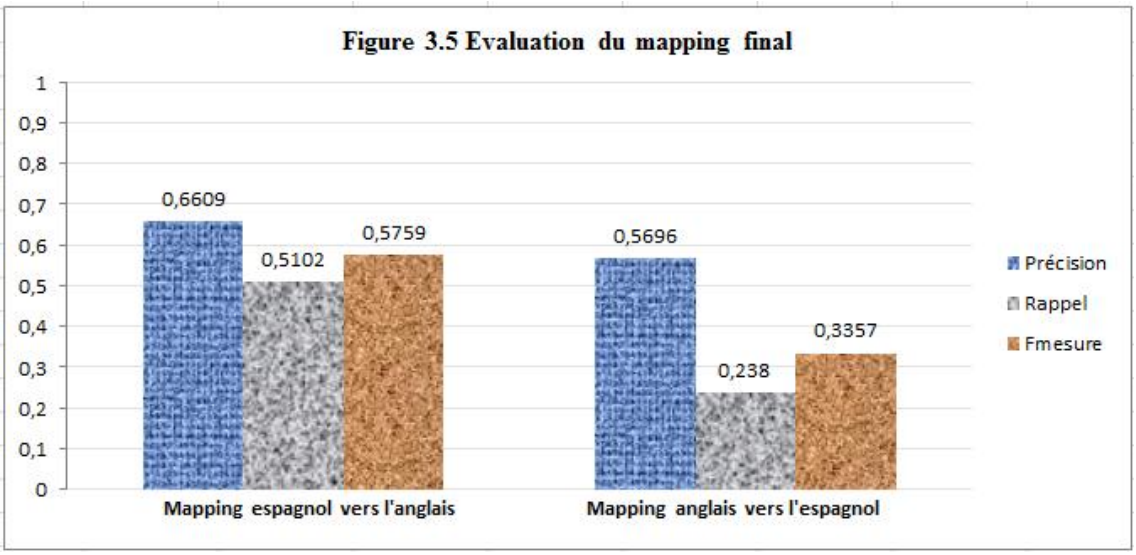
(iii) le pourcentage de mapping présenté par la moyenne harmonique Fmesure.



Suivant la figure 3.4, si nous insistons sur le nombre de cas traités parmi ceux qui ont été ignorés, seulement 32651 des synsets anglais ont été traités parmi un ensemble de 78152 (dont les quels 27644 synsets pour noms ont été traités parmi 66025 et 5007 synsets pour verbes ont été traités parmi 12127). Le plus grand nombre de synsets ignorés est dû à plusieurs raisons : (i) le nombre de synsets anglais est considérablement plus important par rapport aux nombre de synsets espagnols, (ii) l'immaturité dans la structure du WordNet espagnol et (iii) les lacunes qui se trouvent dans le dictionnaire bilingue utilisé dont la plus importante réside dans le fait que ce dernier ne permet de générer qu'une seule traduction.

Dans l'autre sens (le mapping de l'espagnol vers l'anglais), seulement 39209 des synsets espagnols ont été traités parmi un ensemble de 50788 (dont les quels 32349 synsets pour noms ont été traités parmi 42992 et 6860 synsets pour verbes ont été traités parmi 7796). Les résultats de ce mapping sont beaucoup plus intéressants par rapport au premier, parce que d'une part les synsets espagnols ont tous leur équivalent dans le WordNet anglais et d'une autre part, une lemmatisation (stemming) a été effectuée sur les traductions des mots survenant dans la hiérarchie des synsets espagnols afin de diminuer le nombre de correspondances qui peuvent être ignorés mais malgré cela, ce dernier reste beaucoup plus élevé.

Nous constatons durant notre analyse des deux tableaux précédents que l'augmentation du niveau de la hiérarchie n'a pas influencé beaucoup sur les résultats. Cependant, plus de 40% des résultats étaient erronés dans le mapping de l'anglais vers l'espagnol et plus de 30% des résultats dans le mapping de l'espagnol vers l'anglais.



Le problème d'un mauvais alignement entre deux synsets des deux WordNets peut apparaître dans plusieurs cas et cela est dû en particulier aux lacunes qui se trouvent dans le WordNet espagnol. Par exemple dans le synset {egress, egression, emergence}, l'entrée vers les synsets candidats se fait par la traduction du mot « egress » donnée par « salida » dont 11 synsets espagnols sont candidats, mais aucun d'eux représente l'équivalent de ce dernier. L'équivalent du synset anglais est donné par le synset espagnol {emergencia}. Un autre exemple est donné par le synset espagnol {aterrizaje} dont le mot «aterrizaje » représente à lui seul trois synsets. La traduction du mot contenu dans le synset est donnée par « landing », seulement un seul synset parmi les trois synsets a été bien aligné, les deux autres non pas d'équivalents même parmi les synsets candidats. D'une autre part, pour ce qui concerne le premier mapping, la faible précision dans certains cas est due à l'absence des synsets espagnols qui n'ont pas été validés et vérifiés, ce dernier était l'un des causes de la faible précision du premier mapping par rapport au deuxième.

Le traitement effectué dans le premier mapping était de faire une lemmatisation sur tous les mots qui apparaissent dans la hiérarchie du synset anglais afin d'avoir que des lemmes de ces derniers pour ensuite appliquer une traduction automatique. Cette dernière retourne des lemmes pour des termes espagnols dont la recherche dans le WordNet espagnol consiste à trouver des chaines de caractères équivalentes à ces derniers ou bien même avec des terminaisons espagnoles pour se rapprocher des correspondances qui peuvent être ignorés. Par contre dans le cas du mapping de l'espagnol vers l'anglais, la tâche est plus facile puisque le WordNet de Princeton nous offre des techniques de lemmatisation. De ce fait, la similarité sera calculée simplement

par des relations d'équivalence entre chaînes de caractères présentées par la lemmatisation des traductions en anglais des mots espagnols appartenant à la hiérarchie du synset source et les mots anglais survenant des hiérarchies des synsets candidats.

Les résultats de nos expérimentations affirment que notre approche est intéressante vue les limites des ressources utilisées. Nous pensons qu'avec l'enrichissement du WordNet espagnol et l'utilisation d'un autre dictionnaire bilingue plus riche par rapport à Google Translator API [65] (qui donnera plus de propositions pour une traduction et non pas seulement un mot qui restera le plus signifiant) donnera bien de meilleurs résultats.

V. Conclusion

Ce chapitre a été consacré à la description et la mise en œuvre de notre approche divisé en trois sections. La première section été dédié à l'algorithme de Lesk. Dans la deuxième section, nous avons fait une présentation de l'architecture de notre prototype, ainsi qu'à définir les stratégies utilisées pour le calcul de similarité entre concepts. Un exemple illustratif été proposé afin de bien comprendre le fonctionnement de l'algorithme. Dans la troisième section, nous avons présentés les technologies et les outils de développement utilisés ainsi que les métriques de comparaison utilisées pour évaluer les résultats de mapping. Ensuite nous avons fourni une évaluation expérimentale pour le *mapping* des ontologies réelles, à savoir : WordNet de Princeton 1.6 et WordNet espagnol mappé sur la version de ce dernier.

Au terme des objectifs listés dans le chapitre d'introduction, l'évaluation et l'expérimentation de notre méthode ont permis de résoudre le problème d'hétérogénéité afin de ne pas limiter les possibilités d'interopérabilité entre les ontologies. L'accès vers des ontologies multilingues peut enfin être possible en se basent sur des dictionnaires bilingues afin d'appliquer des techniques d'appariement monolingues. Nous avons fait un mapping total entre les deux ontologies, de l'anglais vers l'espagnol et de l'espagnol vers l'anglais. Nous avons constaté que les résultats obtenus garantissent favorablement l'efficacité du deuxième mapping (espagnol vers l'anglais) par rapport au premier à cause de la richesse du WordNet anglais dont beaucoup de concepts anglais n'ont pas leurs équivalentes dans le WordNet espagnol. Nous avons développé une application IHM afin de réaliser un mapping sélectif en se basent sur le mot recherché par l'utilisateur qui est exprimé sous forme d'une requête à traiter par le système de mapping.

Conclusion et perspectives

Les travaux menés dans ce mémoire se situent dans le domaine de l'ingénierie des connaissances et du Web sémantique. Notre objectif a été de tirer profit des travaux menés dans le domaine de l'interopérabilité sémantique des connaissances, dans le but d'aligner des ontologies et également d'accéder à une ressource distante de façon transparente telle qu'un dictionnaire bilingue. Une implémentation de la méthode de mise en correspondance des ontologies proposée a été développée. Cette implémentation est décrite en java et s'appuie sur l'utilisation de deux ontologies de type WordNet de langue différente, l'une est décrite sous forme d'une base de données et l'autre ressource est appelée en utilisant l'API JWNL. Cette mise en correspondance est appliquée pour en déduire une série de mappings.

Notre méthode réside dans la combinaison des caractéristiques suivantes :

- **Flexibilité**

Plusieurs méthodes d'appariement (syntaxique, linguistique et structurel) sont utilisées et combinées.

- **Evolutivité**

Les mappings sont génères en une seule fois et d'une manière automatique suivant un certain niveau hiérarchique, mais à la demande et de façon transparente pour l'application IHM créée pour explorer le mapping.

- **Mapping sélectif ou total**

Nous avons développé un système de *mapping* capable d'identifier les correspondances de *tout* ou *partie* des éléments de l'ontologie, c'est-à-dire l'intégration ou le *mapping partiel*. Nous proposons un processus de mapping à deux modes de fonctionnement : mapping sur deux ontologies complètes ou bien mapping à partir d'une requête sur une ontologie source vers une ontologie cible.

- **Améliorer le nombre de mapping trouvés**

Les mécanismes utilisés pour améliorer le nombre des mappings trouvés se basent sur le niveau hiérarchique de généralisation des concepts des deux ontologies qui vont être capable de prendre en compte les spécificités des deux ontologies. Les matchers syntaxiques, linguistique et structurelles ont été combinées afin d'améliorer les correspondances et d'augmenter le nombre de concepts correctement alignés.

Les expérimentations menées montrent que le rappel est beaucoup plus faible par rapport à la précision. Cela est dû au nombre important des cas non traités. Le rappel lors du mapping de l'espagnol vers l'anglais est beaucoup plus grand par rapport au mapping de l'anglais vers l'espagnol parce que le WordNet de Princeton est beaucoup plus riche et la majorité des concepts espagnols possèdent leur équivalent dans le WordNet anglais. Les combinaisons de *matchers* de différents types utilisés sont efficaces pour trouver un maximum de bons *mappings,* mais en génèrent également beaucoup de faux mappings.

Nos idées ont été appliqué et porté donc sur des ontologies de taille relativement grandes contenant des centaines de concepts mais l'une des deux ontologies est beaucoup plus riche par rapport à l'autre. Comme suite de ce travail, il est important d'évaluer nos propositions sur des ontologies de tailles plus équivalentes (plus près au nombre de concepts vérifiés et validés) afin d'augmenter le rappel et du coût le nombre de correspondances entre concepts. La validation de notre approche sur d'autres ontologies de langues différentes autre que l'anglais et l'espagnol fera partie de notre futur travail.

ANNEXE A

WORDNET ET SON ECOSYSTEME

Un ensemble de ressources linguistiques de large couverture

I. Introduction

Dans notre travail, nous avons utilisés différentes ontologies génériques de type WordNet, pour faire le mapping et de sortir avec les meilleurs correspondances, nous proposons dans cet annexe une présentation de WordNet et son écosystème qui définit un ensemble de ressource linguistiques de large couverture.

WordNet est une ressource lexicale de large couverture, développée depuis plus de 20 ans pour la langue anglaise. Elle est utilisable librement, y compris pour un usage commercial, ce qui en a favorisé une diffusion très large. Plusieurs autres ressources linguistiques ont été constituées (manuellement ou automatiquement) à partir de, en extension à, ou en complément à WordNet. Des programmes issus du monde de l'Intelligence Artificielle ont également établi des passerelles avec WordNet.

L'ensemble constitue un « écosystème » complet couvrant des aspects lexicaux, syntaxiques et sémantiques. Combinées, ces ressources fournissent un point de départ intéressant pour les développements sémantiques en TAL (Traitement Automatique de la Langue) ou dans le cadre du Web sémantique, tels que la recherche d'information, la représentation des documents dans des applications web afin d'extraire les différentes informations, l'inférence pour la compréhension automatique de textes, la désambiguïsation lexicale, L'étiquetage sémantique de corpus de données qui se base sur les relations hiérarchiques entre concepts utilisé pour la structuration et la catégorisation des documents ainsi que dans les systèmes questions/réponses afin d'enrichir la représentation par des synonymes et des hyperonymes.

II. Description de WordNet

II.1. Description du WordNet de Princeton (EWN)

WordNet (Miller, 1995) (EWN : English WordNet) [66] est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991.

Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Des interfaces de programmation sont disponibles pour de nombreux langages.

S'il n'est pas exempt de critiques (granularité très fine, absence de relations paradigmatiques…), WordNet n'en reste pas moins l'une des ressources de TAL les plus populaires.

II.2. Description du WordNet Espagnol (ESWN)

Espagnol WordNet (ESWN) est un thésaurus qui suit la même structure du WordNet américain - pour l'anglais de Princeton - en termes de synsets ainsi que les différentes relations qui peuvent être portés sur les synsets. Les différentes parties du discours telles que les noms, verbes et adjectifs espagnols sont organisés en ensembles de synonymes, chacun présentant une catégorie dans la base lexicale. Jusqu'à maintenant, les adverbes n'ont pas été intégrés dans la version du WordNet espagnol.

La version utilisée dans ESWN est mappée avec la version du WordNet de Princeton 1.6 afin de faciliter la tâche à l'utilisateur pour accéder à des données dans sa langue préféré que ce soit l'anglais ou bien l'espagnol. Donc, les langues sont reliées entre elles de sorte qu'il est possible de passer de mots dans une langue à des mots similaires dans toute autre langue. La base de données peut être utilisée, entre autres, pour l'information monolingue ainsi que les informations multilingues. La version 2006/11 pour Novembre 2006 du WordNet espagnol est le résultat de nombreux efforts combinés par plusieurs groupes de recherches, tel que : UPC Natural Language Research GROUP at TALP RESEARCH CENTER, UB Computaional Linguistics Group et Grupo UNED [67]. ESWN est présenté sous forme d'une base de données dont les informations peuvent

être extraites depuis MYSQL (un serveur de base de données), les données sont interrogées en utilisant le langage SQL. ESWN est présenté sous forme de trois tables, chacune contient des informations appropriées à la table, décrites comme suit.

- Eswn-variant : <pos>|<offset>|<word>|<sense>|<csco>|<status>

  Dont :

  - pos : La partie du discours ;
  - offset : l'identifiant du synset équivalent à EWN 1.6 ;
  - word : Le mot synonyme espagnol ;
  - sense : le nombre de sens du mot synonyme ;
  - csco : le score de confiance du mot synonyme ;
  - status : toujours « - » ou bien vide dans les nouveaux synsets de la version actuelle.

    Exemple : n|50005900|articulación|6|99|- équivalent à « joint »

    n|50005901|información|4|99| équivalent à « information »

- Eswn-synset : <pos>|<offset>|<sons>|<lexical>|<gloss>

  Dont :

  - sons : le nombre de synsets hyponymes (synsets plus spécifiques) ;
  - lexical : prend deux valeurs, soit :
    - « i-» signifie que le synset a été vérifier et verrouiller par un expert ;
    - « -n » signifie que le synset ne possède aucune forme lexicale ;
  - gloss : présente le label du synset, c'est-à-dire la définition ainsi que les exemples portant sur le synset s'ils existent.

    Exemple: a|00005515|0|i-| Completo en extensión o grado: "un desastre total"; "la verdad absoluta".

  - Les mots composant le synset sont : « completo, absoluto, total, perfecto »

- Eswn-relation :

  <relation>|<sourcePos>|<sourceSynset>|<targetPos>|<targetSynset>|<csco>

  Dont :

  - relation : le type de relation entre deux synsets (sourceSynset et targetSynset) ;
  - sourcePos : la partie du discours du synset source ;
  - targetPos : la partie du discours du synset cible ;

- sourceSynset : l'identifiant qui présente l'offset du synset source ;
- targetSynset : l'identifiant qui présente l'offset du synset cible vers la relation définie dans la colonne relation.
  Exemple: has_hyponym|v|50000203|v|50000204|99

II.3 Caractéristiques des deux WordNets

Les données ainsi que le nombre des éléments composant les synsets contenus dans les WordNets décrits précédemment et selon [68] se présentent dans les tableaux suivants, à savoir, le tableau A.1 présente les caractéristiques sur les données sur EWN et le tableau A.2 présente les caractéristiques des données sur ESWN.

-	**Total**	**Noms**	**Verbes**	**Adjectifs**	**Adverbes**
Sens des mots	173941	116317	22066	29881	5677
Lemmes	129502	94474	10319	20170	4546
Synsets	99642	66025	12127	17915	3575
Noms propres	3876	-	-	-	-

Tableau A.1 Caractéristiques des données de l'EWN version 1.6

- La première version du WordNet espagnol a été alignée sur EWN version 1.5, le nombre de synsets contenu dans la version aligné à EWN 1.6 est présenté dans le tableau A.3.

-	**Total**	**Noms**	**Verbes**	**Adjectifs**	**Adverbes**
Sens des mots	93729	62644	12568	18517	-
Lemmes	62025	47577	5346	9102	-
Synsets	69040	43801	9302	15937	-
Noms propres	3956	-	-	-	-
Nouveaux synsets	5838	5351	213	274	

Tableau A.2 Caractéristiques des données de l'ESWN aligné à EWN version 1.5

-	**Total**	**lexical**	**Noms**	**Verbes**	**Adjectifs**	**Adverbes**
Synsets vérifiés	62720	i-	42992	7796	11932	0
Synsets non vérifiés	42796	-n	28418	4546	6257	3575
Total	105516	-	71410	12342	18189	3575

Tableau A.3 Nombre de synsets de l'ESWN aligné à EWN version 1.6

RÉFÉRENCES

[1]. P. Cimiano.: Ontology Learning and Population from Text Algorithms, Evaluation and Applications. ISBN-10: 0-387-30632-3, ISBN-13: 978-0-387-30632-2, e-ISBN-10: 0-387-39252-1, e-ISBN-13: 978-0-387-39252-3, Springer Science+Business Media, LLC 2006.

[2]. R. Neches, RE. Fikes, T. Finin, TR. Gruber, T. Senator, WR. Swartout.: Enabling technology for knowledge sharing. AI Magazine 12(3) pp 36–56, 1991.

[3]. Tr.Gruber.: A translation approach to portable ontology specification. Knowledge Acquisition 5(2): pp 199–220, 1993.

[4]. WN. Borst.: Construction of Engineering Ontologies. Centre for Telematica and Information Technology, University of Tweenty. Enschede, The Netherlands, 1997.

[5]. R. Studer, VR. Benjamins, D.Fensel.: Knowledge Engineering: Principles and Methods. IEEE Transactions on Data and Knowledge Engineering, 25(1-2): pp 161– 197, 1998.

[6]. B. Bachimont.: Engagement semantique et engagement ontologique: conception et realisation d'ontologies en ingenierie des connaissances, in J. CHARLET, M. Zacklad, G. Kassel, D. Bourigault, eds., Ingenierie des connaissances : evolutions récentes et nouveaux défis, Eyrolles, pp 305-323, 2000.

[7]. R. Mizoguchi, J. Vanwelkenhuysen, M. Ikeda.: Task Ontology for reuse of problem solving knowledge. In: Mars N (ed) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS'95). University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, The Netherlands, pp 46–57,1995.

[8]. N. Guarino.: Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, pp. 3-15, 1998.

[9]. O. Lassila, D. McGuinness. : The Role of Frame-Based Representation on the McGuinness. : The Role of Frame-Based Representation on the Semantic Web. Technical Report KSL-01-02. Knowledge Systems Laboratory. Stanford University. Stanford, California.2001.

[10]. A. Gomez-Perez, M. Fernandez-Lopez, O. Corcho.: Ontological Engineering, ISBN 1- 85233-551-3 Springer-Verlag London Limited 2004.

[11]. G. Van Heijst, Ath. Schreiber, BJ. Wielinga. : Using explicit ontologies in KBS development. International Journal of Human-Computer Studies, pp 45:183–292,1997.

[12]. A. Farquhar, R. Fikes, J. Rice.: Ontolingua server: a tool for collaborative ontology construction, in International journal of Human-Computer studies (46), pp 707-727, 2000.

[13]. M. Blazquez, M. Fernandez, J.M. Garcia-Pinar, A. Gomez-Perez.: Building Ontologies at the Knowledge Level using the Ontology Design Environment, in Proceedings of the Banff Workshop on Knwoledge Acquisition for Knwoledge-based Systems, 1998.

[14]. N. Noy, R.W. Fergerson, M.A. Musen.: The knowledge model of Protege2000: combining interoperability and flexibility, in Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW'00), 2000.
[15]. M. Minsky.: A framework for representing knowledge, in Winston, P. H. The Psychology of Computer vision, New York, McGraw-Hill, pp 211-277, 1975.
[16]. M. Kifer, G. Lausen, J. Wu.: Logical Foundations of Object-Oriented and Frame- Based Languages. Journal of the ACM, 42(4): pp 741–843, 1995.
[17]. J. Sowa.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 1984.
[18]. M.R. Quillian.: Semantic Memory. In Semantic Information Processing. MIT Press, pp 227–270, 1968.
[19]. R.J. Brachman, J.G. Schmolze.: An overview of the KL-ONE knowledge Representation System. Cognitive Science, 9(2): pp 171–216, April 1985.
[20]. D.B. Lenat, R.V.Juha : Building large knowledge-based Systems : representation and Inference in the Cyc Project. Addison-Wesley, Boston, Massachussetts.1990.
[21]. M.R. Genesereth, R.E. Fikes.: Knowledge Interchange Format. Version 3.0. Reference Manual. Technical Report Logic-92-1. Computer Science Department. Stanford University, California.1992.
[22]. R. MacGregor.: Inside the LOOM clasifier. SIGART bulletin, 2(3): pp70–76,1991.
[23]. E. Motta.: Reusable Components for Knowledge Modelling: Principles and Case Studies in Parametric Design. IOS Press, Amsterdam. The Netherlands. 1999.
[24]. Ontology Web Language (OWL) ; http://www.w3.org/OWL/.
[25]. M. Klein.: Combining and relating ontologies: an analysis of problems and solutions. In Proc. IJCAI Workshop on Ontologies and Information Sharing, Seattle (WA US), 2001.
[26]. P. Bouquet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krotzsch, L. Serafini, G. Stamou, Y. Sure, S. Tessaris : Specification of a common framework for characterizing alignment. Deliverable D2.2.1, Knowledge web NoE, 2004.
[27]. J. Euzenat.: Towards a principled approach to semantic interoperability. In *Proc. IJCAIWorkshop on Ontologies and Information Sharing*. Seattle (WA US). pp 19–25, 2001.
[28]. INTEROP.: Deliverables of INTEROP Project. On Line www.interop-ne.org, 2005.
[29]. H.S. Pinto, A. Gomez-Perez, J.P. Martins.: Some Issues on Ontology Integration. In Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods:Lessons Learned and Future Trends. Stockholm, Sweden, 1999.
[30]. N.F Noy, M. Musen.: Smart : automated support for ontology merging and alignment, 1999.
[31]. M. Ehrig.: Ontology alignment: bridging the semantic gap. Semantic web and beyond: computing for human experience. Springer, New-York (NY US), 2007.
[32]. J. Euzenat, P. Shvaiko.: Ontology Matching. ISBN 978-3-540-49611-3, Springer- Verlag Berlin Heidelberg 2007.
[33]. Sekt.: Deliverables of SEKT Project. On Line: http://www.sektproject.com/ Deliverable D.4.2.1 (State-of-the-art survey on Ontology Merging and Aligning V1) 2004.

[34]. G. Hirst, D. St Onge.: Lexical chains as representations of context for the detection and correction of malapropisms, In Christiane Fellbaum (editor), WordNet: An electronic lexical database, Cambridge, MA: The MIT Press, 1998.

[35]. Rahm E. et Bernstein P. A. : A survey of approaches to automatic schema matching. *VLDB Journal The International Journal on Very Large Data Bases*, 10(4):334 – 350, décembre 2001.

[36]. Damerau F. : *Markov Models & Linguistic Theory*. Paris : Mouton, The Hague, 1971.

[37]. Hamming R. : Error Detecting & Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160, 1950.

[38]. Van R¼sbergen C. J. : *Information Retrieval*. Butterworth, 2sd edition, 1979.

[39]. Masolo C., Borgo S., Gangemi A., Guarino N., Oltramari A., Oltramari R., Schneider L. et Horrocks I. : The WonderWeb Library of Foundational Ontologies and the DOLCE ontology. Deliverable D17, WonderWeb, 2002.

[40]. McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). The Chimaera Ontology Environment. (AAAI/IAAI 2000). pp. 1123-1124.

[41]. Chalupsky, H. (2000). OntoMorph: A translation system for symbolic knowledge. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, KR 2000, Principles of Knowledge Representation and Reasoning Proceedings of the Seventh International Conference. Colorado, USA. pp. 471-482.

[42]. Noy, N. F., & Musen, M. A. (2000). Prompt: algorithm and tool for automated ontology merging and alignment. In Proceeding of Seventeenth National Conference on Artificial Intelligence AAAI.

[43]. Stumme, G., & Maedche, A. (2001). Fca-merge: Bottom-up merging of ontologies. In 7th Intl. Conf. On Artificial Intelligence (IJCAI '01), Seattle, WA, USA. pp. 225–230.

[44]. Melnik, S., Garcia-Molina, H., & Rahm, E. (2002, 26 February - 1 March). Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In : Proceedings of the 18th International Conference on Data Engineering, San Jose, CA. pp. 117-128.

[45]. Noy, N. F., & Musen, M. A. (2001, aout). Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In : Proceedings of workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), Seattle, WA.

[46]. Do, H. H., & Rahm, E. (2002, Août). COMA - A System for Flexible Combination of Schema Matching Approaches. In : Proceedings of the 28th Intl. Conference on Very Large Databases (VLDB), Hongkong.

[47]. Doan, A., Madhavan, J., & Domingos, P. (2002, May 7-11). Learning to Map between ontologies on the Semantic Web. In : the eleventh International World Wide Web conference (WWW2002), Honolulu, Hawaii, USA.

[48]. Madhavan, J., Bernstein, P. A., & Rahm, E. (2001, September 11-14). Generic Schema Matching with Cupid. In : Proceedings of Very Large Databases Journal (VLDB01). pp. 49-58.

[49]. Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). MAFRA - A MApping FRAmework for Distributed Ontologies. (EKAW 2002). pp. 235-250.

[50]. Omelayenko, B. (2002). RDFT: A mapping meta-ontology for business integration. In Proceedings of the Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at the 15-th European Conference on Artificial Intelligence, Lyon, France. pp. 76–83.

[51]. Giunchiglia, F., Shvaiko, P., & Yat, M. (2004). S-match: an algorithm and an implementation of semantic matching. In Proceedings of ESWS'04, number 3053 in LNCS, Heraklion, Greece. pp. 61–75.

[52]. Kiryakov, A., Simov, K. I., & Dimitrov, M. (2001). Ontomap: The upper-ontology portal. In Proceedings of "Formal Ontology in Information Systems", Ogunquit, Maine.

[53]. Nodine, H., Fowler, J., Ksiezyk, T., Perry, B., Taylor, M., & Unruh, A. (2000). ,. Active information gathering in infosleuth. International Journal of Cooperative Information Systems, 9(1-2):3–28.

[54]. 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. pp. 223–271.

[55]. Bergamaschi, S., Castano, S., & Vincini, M. (1999). Semantic integration of semistructured and structured data sources. SIGMOD Record, 28(1). pp. 54–59.

[56]. Mitra, P., & Wiederhold, G. (2001). An algebra for semantic interoperability of information sources. In IEEE International Conference on Bioinformatics and Biomedical Egineering. pp. 174–182.

[57]. OAEI. (2009). http://oaei.ontologymatching.org/, consulte en.

[58]. Bo Fu, Rob Brennan, and Declan O'Sullivan. 2009. Crosslingual ontology mapping - an investigation of the impact of machine translation. In Proceedings of the 4th Asian Semantic Web Conference.

[59]. Benjamins R. V., Contreras J., Corcho O., Gomez-Perez A., Six Challenges for the Semantic Web. SIGSEMIS Bulletin, April (2004).

[60]. Carpuat M., Ngai G., Fung P., Church W. K., Creating a Bilingual Ontology: A Corpus-Based Approach for Aligning WordNet and HowNet. In Proceedings of the 1st Global WordNet Conference (2002).

[61]. Pazienza M. T., Stellato A., An Open and Scalable Framework for Enriching Ontologies with Natrual Lanauge Content. In Proceedings of the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (2006).

[62]. J. Ramanand, A. Ukey, B. K. Singh, and P. Bhattacharyya, "Mapping and structural analysis of multilingualwordnets," in Bulletin of the IEEE Computer Society Technical Comittee on DATA Engineering, IEEE Computer Society, 2000.

[63]. The Alignment API project homepage, http://alignapi.gforge.inria.fr, last accessed December 2008.

[64]. M. Lesk. *Automatic sense disambiguation using machine readable dictionaries: How to tell a pinecone from a ice cream cone*. Proceedings of the SIGDOC '86, 1986.

[65]. Google Translation API (http://code.google.com/p/google-api-translate-java).

[66]. Fellbaum C., éditeur. *WordNet : An Electronic Lexical Database*. MIT Press, mai 1998. http://wordnet.princeton.edu/.

[67]. UPC Natural Language Research GROUP at TALP RESEARCH CENTER, UB Computaional Linguistics Group et Grupo UNED ; Spanish WordNet (http://www.lsi.upc.es/~nlp; http://clic.fil.ub.es).

[68]. Luisa Bentivogli (ITC-irst), Inaki Algeria Loinaz (EHU), Jordi Atserias, Battala (UPC), Rob Koeling (Univ. Sussex): Developing Multilingual Web-Scale Language Technologies; IST-2001-34460.