N° d'ordre :07/2008-M/IN

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
"HOUARI BOUMEDIENE"
FACULTE D'ÉLECTRONIQUE ET D'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

EN : INFORMATIQUE

Spécialité : Systèmes Intelligents et Ingénierie du Logiciel

Par : OULEFKI SAMIRA

SUJET

Contributions à l'appariement des ontologies: Classification, Processus et Comparaison d'alignements

Soutenu le 15/07/2008, devant le jury composé de :

Mr- M. AHMED-NACER,	**Professeur,**	**USTHB,**	**Président**
Mme- K. AKLI,	**Chargée de cours,**	**USTHB,**	**Directrice de Mémoire**
Mme- M. BOUKALA,	**Professeur,**	**USTHB**	**Examinatrice**
Mme- Z. ALIMAZIGHI,	**Professeur,**	**USTHB,**	**Examinatrice**
Mme- A. AISSANI,	**Professeur,**	**USTHB,**	**Examinatrice**

Université des Sciences et de la Technologie Houari Boumediene

Samira OULEFKI

Contributions à l'appariement des ontologies:

Classification, Processus et Comparaison d'alignements

Encadreur de recherche :

M^me^ Karima Akli-Astouati, chargé de cours à l'USTHB

Résumé

Le web qu'on essaye de construire aujourd'hui, se veut un web ayant un sens, de façon que les ressources soient accessibles à la fois par des humains aussi bien que par des agents logiciels. Les ontologies permettant une représentation explicite du sens sont au centre des travaux de ce nouveau web que l'on qualifie de web sémantique. La diversité des formats de représentation de la connaissance et des modélisations pour un même domaine conduit à l'apparition de problèmes d'hétérogénéité entre différentes ontologies. Cette hétérogénéité posera des problèmes dans l'échange, le traitement, l'intégration, et la recherche d'information. Afin de cohabiter avec cette hétérogénéité, des correspondances sémantiques entre les éléments des ontologies doivent être spécifiées. Ces correspondances constituent une "colle" qui maintient les ontologies ensemble afin de garantir leur interopérabilité. Le processus permettant de découvrir ces correspondances est appelé appariement.

L'appariement d'ontologies est généralement défini comme un processus qui prend en entrée deux ontologies, calcule la similarité de leurs entités et retourne un alignement qui identifie celles ayant une signification identique ou proche.

Dans la littérature, plusieurs techniques pour mesurer la similarité entre les entités des ontologies ont été développées. Ce mémoire propose premièrement de les classifier dans le but de guider l'utilisateur dans le choix de la technique appropriée selon le langage de représentation des ontologies utilisé. Cependant, bien que le calcul de la similarité entre les entités des ontologies soit la base sur laquelle repose tout système d'appariement, il ne constitue qu'une étape parmi d'autres de leur processus.

L'explicitation du processus d'appariement d'ontologies a fait l'objet de plusieurs travaux qui varient selon leur degré de granularité et de détails. Le deuxième objectif de ce mémoire consiste à délivrer un processus d'appariement permettant de prendre en considération le maximum d'éléments pouvant composer un système d'appariement.

De nombreuses approches et de nombreux outils sont proposés pour apparier des ontologies. L'utilisateur se trouve donc devant une multitude d'outils à partir desquels il doit choisir l'outil qui lui convient. Afin de l'aider dans ce choix, certains travaux se sont orientés vers l'évaluation des outils d'appariement. Bien que les outils existants fournissent parfois des alignements complexes, i.e., contenant des liens sémantiques entre plus de deux entités, ces travaux ne considèrent que des alignements simples. Les outils produisant des alignements complexes sont donc lésés dans l'appréciation. Motivée par ce constat, le troisième but de ce mémoire est de proposer une approche qui permet de résoudre ce problème.

Mot clés : Web sémantique, ontologie, appariement, alignement, mesure de similarité, correspondance sémantique, évaluation.

Remerciements

*T*ous mes remerciements à **...**

... Mon encadreur **Mme Karima Akli-Astouati** pour la confiance qu'elle a placé en moi, pour le temps qu'elle m'a consacré et pour ses précieux conseils.

... Mme Aicha Mokhtari, professeur à l'USTHB, pour m'avoir accueilli dans son équipe de recherche et pour ses conseils.

... Mr Mohamed Ahmed-Nacer, professeur à l'USTHB, qui m'a fait l'honneur d'accepter de présider le jury de ce travail.

... Mme Zahia Alimazighi, professeur à l'USTHB, qui a bien voulu accepter d'être membre du jury.

... Mme Malika Ioualalen-Boukala, professeur à l'USTHB, qui a aimablement accepté de participer au jury de ce travail. Elle a pris du temps pour m'écouter et pour discuter avec moi, je la remercie également pour cela.

... Mes parents qui m'ont donné l'amour, la confiance, le soutien et le courage. Ils sont la plus grande grâce que dieu m'a donné.

... Ma grand-mère qui a toujours su me soutenir et me comprendre.

... Mon oncle paternel qui m'a donné beaucoup d'amour et de confiance. Il a été toujours disponible malgré son emploi du temps très chargé.

... Mes oncles maternels, tous **mes cousins et cousines** ainsi que leurs époux qui m'ont énormément soutenu et ont cru en moi.

... Mes amis et mes collègues avec lesquels j'ai partagé les plus beaux moments de la vie.

Dédicaces

À mes chers parents

À ma grand-mère Yamina

À la mémoire de ma grand-mère Zineb

À la mémoire de mes grands-pères

À mon oncle paternel

À mes oncles maternels

À tous mes cousins et cousines

À tous mes amis

Table des matières

INTRODUCTION GENERALE

La quantité d'information disponible sur le Web est importante et elle ne cesse de croître. Les catalogues et les moteurs de recherche en ligne (Yahoo, Google, Lycos…etc.) permettent d'effectuer des requêtes par mot clé en les affinant à l'aide d'opérateurs booléens. Cependant, la tâche la plus lourde revient à l'utilisateur qui doit fouiller dans cette masse d'information pour sélectionner les documents qui lui seront les plus utiles. Les résultats ne sont pas tous pertinents. Cette limite n'est pas d'ordre structurel ou syntaxique, mais concerne la description du sens de l'information représentée (Chamoun, 2006).

L'enjeu fondamental de l'avenir est donc d'assigner une sémantique aux ressources utilisées, de façon à permettre à des agents logiciels de les interpréter de manière autonome. C'est dans cette perspective que s'inscrit le Web sémantique, comme extension du Web actuel, (Berners-Lee, Hendler, & Lassila, 2001), dans la mesure où il vient structurer son sens et permettre la gestion automatique et intelligente de son contenu. Les informations ne seront plus juste stockées, mais aussi comprises (Luong, 2007).

Annoncée « technologie du futur », par son créateur, Tim Berners-Lee, cette nouvelle vision du Web va permettre l'exploitation et la compréhension des informations disponibles dans les différentes ressources non seulement par les hommes mais également par les machines, les programmes et les agents informatiques offrant ainsi la possibilité de s'ouvrir à de nouvelles possibilités d'automatisation dans le Web et d'assurer une meilleure collaboration entre les humains et les machines (Berners-Lee, 2002).

Ainsi, pour rendre possible et réelle cette perspective, la description du contenu des ressources doit être à la fois formelle et signifiante à l'aide d'une *ontologie*. Les ontologies sont donc des éléments principaux pour la construction de l'infrastructure du Web sémantique (Laublet, Reynaud, & Charlet, 2002).

Cadre du mémoire

Au sein de l'équipe « Représentation des connaissances » dirigée par Mme Mokhtari-Aissani Aicha, plusieurs travaux concernant le Web sémantique ont été réalisé. Le travail proposé dans le cadre de ce mémoire de Magister est une suite du travail proposé dans un mémoire de Magister encadré par Mme Akli-Astouati Karima et réalisé par Mme Guebaili Ratiba. Ce dernier a proposé d'introduire l'incertitude dans le langage RDF(S).

Problématiques

Plusieurs ontologies peuvent être accessibles sur le Web et on peut être amené à en utiliser plusieurs simultanément pour une même application. Cependant, des ontologies développées séparément par des organisations et/ou des personnes différentes, même si elles se rapportent au même domaine, ou a des domaines ayant une intersection non vide, sont souvent hétérogènes, à la fois au niveau de leur structure et au niveau de la sémantique des données qu'elles contiennent. Toute la question est donc de pouvoir exploiter simultanément des ontologies différentes.

Afin de garantir une utilisation simultanée et cohérente de plusieurs ontologies, il est nécessaire de déterminer les parties communes aux différentes ontologies. Ceci est réalisé dans le cadre de travaux de recherche relatifs à l'appariement des ontologies qui vise non pas à éliminer l'hétérogénéité dans le Web sémantique mais plutôt à cohabiter avec, en établissant des liens sémantiques entre les entités similaires des différentes ontologies. L'ensemble de ces liens sémantiques est dit « alignement ». Pour ce faire, plusieurs travaux proposant des techniques pour mesurer la similarité des entités des ontologies sont recensés dans la littérature.

Cependant, dans le cadre du Web sémantique, les langages de représentation des ontologies n'ont pas tous la même puissance d'expression. Ils ne permettent pas de représenter le même ensemble d'entités. Par conséquent, le choix des techniques appropriées dépendra du langage de représentation des ontologies utilisé. La question est donc de savoir comment guider le choix d'une technique pour un langage donné ?

D'autre part, la profusion de travaux se rapportant à l'appariement d'ontologies a suscité plusieurs points de vue sur l'explicitation du processus permettant sa réalisation. Ces travaux décrivent ce processus d'appariement avec des niveaux de détails différents. Une vue plus globale et plus détaillée est alors à construire.

De nombreuses approches et de nombreux outils sont proposés pour générer un alignement entre deux (ou plusieurs) ontologies. L'évaluation de l'alignement généré est de rigueur. Bien que les outils d'appariement existant fournissent parfois des alignements complexes, i.e., contenant des liens sémantiques entre plus de deux entités, les travaux traitant de l'évaluation de ces outils ne prennent en compte que les alignements simples. Les outils produisant des alignements complexes sont donc lésés dans l'appréciation.

Apport du mémoire

Ce mémoire présente quelques contributions dans le domaine de l'appariement des ontologies. Il propose :

- **Une classification des techniques utilisées pour mesurer la similarité des entités de différentes ontologies**. Dans le cadre du Web sémantique, nous avons proposé une classification de techniques d'appariement, sous forme d'un modèle en couche, qui s'appuie sur les différentes caractéristiques supportées par les langages du Web sémantique.
- **Un processus d'appariement d'ontologies.** Nous avons proposé une explicitation détaillée du processus d'appariement d'ontologies permettant de prendre en considération le maximum d'ingrédients pouvant composer un système d'appariement.
- **Le développement de l'approche Align-Match.** Motivée par le risque que les outils d'appariement fournissant des alignements complexes soient lésés dans l'évaluation, le troisième but de ce mémoire est de proposer une approche, nommée Align-Match, qui permette de comparer aussi bien des alignements complexes que des alignements simples dans le but de les évaluer.

Plan du mémoire

Ce mémoire est composé de cinq chapitres. Sa structuration n'est pas conforme à l'usage qui consacre les premiers chapitres à l'état de l'art et les suivants aux contributions du travail effectué. Chaque chapitre comporte un état de l'art qui lui est propre et les contributions qui lui sont associées.

- Le premier chapitre se concentre sur la présentation du Web sémantique. Ses principes, son architecture, ainsi que les ontologies, composante majeure du Web sémantique, sont introduits. Les langages ontologiques auxquels on associe les différents niveaux de complexité sémantique des ontologies, hypothèse importante dans la suite de notre travail, sont présentés.

- Le deuxième chapitre porte sur l'appariement des ontologies. Les problèmes d'hétérogénéité qu'un processus d'appariement doit résoudre sont abordés. Une clarification des notions d'appariement, d'alignement, et de similarité est apportée en présentant les différents types d'alignement que l'opération d'appariement peut fournir.

- Le troisième chapitre présente la classification des techniques de base proposée qui sert à guider le choix de la technique appropriée pour comparer l'ensemble des caractéristiques supportées par le langage ontologique utilisé. Avant, une présentation et une étude des différentes classifications des techniques d'appariement de base recensées dans la littérature y sont fournies.

- Dans le quatrième chapitre, nous avons étudié les différents travaux existants traitant des processus d'appariement d'ontologies. De cette étude a découlé un processus d'appariement permettant de prendre en considérations le maximum d'ingrédients pouvant composer un système d'appariement. Une application du processus proposé sur quelques systèmes d'appariement existants a été réalisée.

- Le cinquième chapitre, en rapport avec l'appariement des alignements, dévoile l'approche proposée dans un but d'évaluation des outils d'appariement. Les travaux existants traitant de l'évaluation des outils d'appariement sont présentés afin de motiver notre approche.

Nous terminons par une conclusion en précisant les perspectives à notre travail.

CHAPITRE 1 : WEB SEMANTIQUE ET ONTOLOGIE

Lors de cette dernière décennie, le Web a connu une évolution importante qui se traduit par la croissance permanente des données et des ressources exploitées à travers cette toile, ce qui rend très difficiles leurs localisations et leurs gestions, d'autant plus que le Web actuel ne peut interpréter leurs sémantiques. Cette difficulté peut être constatée clairement à travers un exemple de recherche d'information avec un moteur de recherche actuel basé principalement sur les mots-clés (ex. Yahoo, Google…). Ainsi, si nous effectuons une recherche sur les adresses de peintres industriels, nous obtiendrons peut-être des résultats qui n'ont aucun rapport avec notre requête comme par exemple des documents sur Vincent Van Gogh. Cette limite n'est pas d'ordre structurel ou syntaxique, mais concerne la description du sens de l'information représentée (Chamoun, 2006).

Pour pallier à ces limites, le Web sémantique est apparu comme une nouvelle technologie qui permettra d'assigner une sémantique aux ressources utilisées, de façon à permettre à des agents logiciels de les interpréter et d'effectuer les raisonnements nécessaires, de manière autonome, en se basant sur une ontologie (Chamoun, 2006).

Ainsi, ce chapitre décrit la vision du Web sémantique à travers son architecture et ses fondements (i.e., les ontologies). Il aborde également les langages ontologiques auxquels les différents niveaux de complexité sémantique des ontologies sont associés

1 Le Web Sémantique

1.1 Définition du Web sémantique

L'usage de l'expression de « Web Sémantique » a été originellement proposé par Tim Berners-Lee (Berners-Lee, Hendler, & Lassila, 2001). Cette proposition fait référence à la vision du Web du futur comme une extension du Web actuel supportant des fonctions avancées pour la collaboration (homme-homme, homme-machine, machine-machine) en vue de partager et de raisonner sur le contenu de grands volumes d'informations (Benayache, 2005).

Le principe du Web sémantique consiste à attacher des annotations à toutes les ressources disponibles (documents, pages Web, services…) sur le Web en vue de rendre explicite leur contenu sémantique (Bach, 2006). L'interprétation des annotations, donc la sémantique, est précisée entre des agents logiciels, des machines ou voire des gens grâce à un des éléments les plus importants du Web sémantique : l'ontologie.

Les ontologies sont donc utilisées pour permettre aux machines de raisonner, d'interpréter les informations et d'améliorer la pertinence des recherches (Luong, 2007). Les agents auxquels les utilisateurs délègueront des tâches, doivent communiquer entre eux et interpréter le contenu échangé de la même manière. D'où l'intérêt des ontologies. Le principe consiste alors à définir une interprétation commune d'une partie du monde réel, et de modéliser les concepts et les relations entre ces concepts.

1.2 Architecture du Web sémantique

La vision courante du Web sémantique proposée par (Berners-Lee, Hendler, & Lassila, 2001) est souvent représentée dans une architecture comportant trois niveaux (cf. figure 1): Le niveau de l'adressage, le niveau syntaxique et le niveau sémantique.

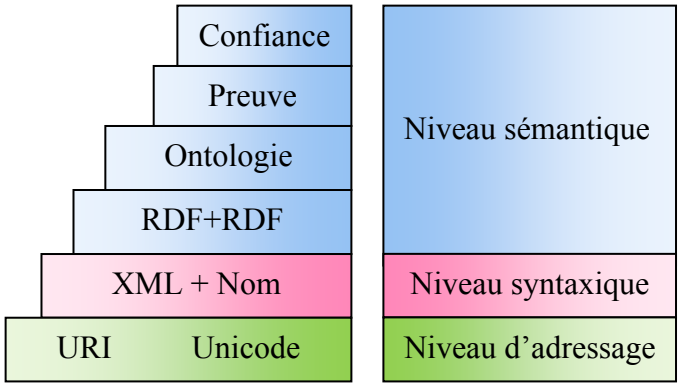


Figure 1. Architecture du Web Sémantique (Berners-Lee, Hendler, & Lassila, 2001)

1.2.1 Niveau «Nommage/Adressage »

Au niveau de la couche la plus basse se trouvent (Luong, 2007) :

- **L'URI (*Uniforme Ressource Identifier*).** C'est l'un des concepts importants sur lesquels repose le World Wide Web. Il fournit un adressage standard universel permettant d'identifier de manière unique et non ambiguë les ressources du Web telles que les pages Web, les adresses email et les images. Un exemple d'URI est l'URL (Uniform Resource Locator) traditionnelle qui identifie les ressources via une représentation de leur mécanisme d'accès (par exemple: http://www.usthb.dz);
- **L'Unicode.** C'est un encodage textuel universel pour échanger des symboles.

1.2.2 Niveau Syntaxique

Le niveau syntaxique est établi par l'utilisation de **XML** (eXtensible Markup Language). Ce dernier fournit un ensemble de règles pour la création de vocabulaires qui structurent à la fois les documents et les données sur le Web (Baneyx, 2007). Il utilise **l'espace de nommage** (namespace) afin d'identifier les noms des balises (tags) utilisées dans les documents XML. Les schémas XML sont ensuite utilisés comme une méthode autorisant la composition de vocabulaires XML.

Bien que XML soit une syntaxe puissante et flexible pour les documents structurés, il n'impose aucune contrainte sémantique à la signification de ces documents.

1.2.3 Niveau Sémantique

Afin de donner une organisation plus structurée des informations présentes sur le Web à travers une description sémantique des données fournies par XML, Tim Berners-Lee propose d'utiliser **RDF** (*Resource Description Framework*).

RDF est un standard permettant la mise en place des annotations (Benayache, 2005). Il permet d'affirmer des relations entre les ressources, et représenter l'information sous forme de triplets< sujet, prédicat, objet >, dont les éléments peuvent être des URIs, des variables ou des littéraux. L'information représentée par RDF est conservée principalement sous forme de déclarations RDF. **RDF Schéma** permet ensuite de décrire les hiérarchies de concepts et les relations entre les concepts. Par exemple, dans un document RDF Schéma décrivant un site dédié aux animaux, on peut préciser que le concept souris est un sous-concept du concept mammifère. On peut également créer une relation entre chat et souris pour indiquer que les chats chassent les souris.

Cependant, la signification sémantique des données XML représentée par RDF ou RDF Schéma, est largement insuffisante pour assurer une bonne distinction entre les différents concepts (Luong, 2007). Par exemple, dans des bases de données distinctes, on peut trouver des identifiants différents, tels que *souris* et *mouse*, qui représentent en fait le même concept. De plus, on peut avoir des concepts différents ayant un même identifiant tel que le terme

souris qui peut désigner un animal ou un périphérique d'ordinateur. Ce problème peut être résolu grâce à l'utilisation des ontologies.

La couche **Ontologie** décrit des sources d'information hétérogènes et distribuées en définissant le consensus du domaine commun et partagé par plusieurs personnes et communautés (Luong, 2007). La section 2 donne une description plus précise de la notion d'ontologie.

La couche **Preuve** a pour but de prouver la pertinence de l'information retournée par les couches de plus bas niveaux. Une des façons de le faire est de garder trace des sources d'information et des raisonnements effectués.

Le Web est un environnement très ouvert et dynamique. De ce fait, toute personne est en mesure d'éditer et de publier des informations de façon très simple. La couche **Confiance**, dans l'architecture proposée par Tim Bernes-Lee, a pour objectif d'évaluer la fiabilité de l'information et des raisonnements (Luong, 2007). Cette couche repose sur les signatures numériques, le cryptage des données et sur la fiabilité des sources d'information (agents de confiances, certifications, etc.)

2 Ontologies : Fondements du Web sémantique

Nous examinons dans cette section la composante essentielle du Web sémantique : l'ontologie. Les ontologies sont la technologie dorsale pour le Web sémantique et – plus généralement - pour le management des connaissances formalisées décrivant les ressources du Web (Luong, 2007). Elles permettent de représenter la sémantique des documents dans l'objectif de garantir aux ordinateurs et aux humains de travailler en plus étroite collaboration.

2.1 La notion de l'ontologie

À l'origine, les ontologies sont issues d'une branche de la philosophie qui s'intéresse à la nature et à l'organisation de la réalité. Elles correspondent à ce qu'Aristote appelait la Philosophie première, *protè philosopha*, c'est-à-dire *la partie de la métaphysique qui s'intéresse à l'être en tant qu'être*, par opposition aux philosophies secondes qui s'intéressent à l'étude des manifestations de l'être (les *étants*) (Garf, 1996).

Le terme «ontologie» a été emprunté par de multiples disciplines, chacune utilisant une signification différente. En IA par exemple, et plus précisément en Ingénierie des Connaissances, le terme est apparu pour modéliser les connaissances du domaine d'un Système à Base de Connaissances (SBC).

2.1.1 Les ontologies en Ingénierie des Connaissances (IC)

En 1980, John McCarthy fut le premier à avoir proposé l'usage du terme « ontologie » en informatique, plus précisément en Ingénierie des Connaissances. L'objectif de cette proposition était de modéliser les connaissances du domaine d'un Système à Base de Connaissances (SBC). McCarthy affirmait que la construction d'un tel artefact (SBC) doit avant tout commencer par modéliser les aspects essentiels des objets du domaine d'études, en d'autres termes de construire une ontologie de leur domaine, et ensuite seulement baser leurs systèmes sur cette ontologie (Psyché & Mendes, 2003).

Au début des années 1990, l'usage du terme était déjà bien répandu dans le domaine de l'intelligence artificielle.

En 1993, T. Gruber (Gruber, 1993) a proposé une définition qui reste jusqu'à présent la définition la plus citée dans les écrits en intelligence artificielle : «*une ontologie est une spécification explicite d'une conceptualisation*». Pour mieux la comprendre, une explication plus détaillée est donnée dans (Guarino & Giaretta, 1995). Il explicite la notion de conceptualisation en la définissant comme l'identification par des termes et/ou des symboles, des concepts du domaine et des relations existantes entre ces concepts. Mais cette conceptualisation est forcément basée sur une vision partielle des connaissances d'un domaine. Par exemple, on peut dégager des concepts et des relations permettant de décrire un triangle. Mais cette conceptualisation ne sera peut être plus valable pour décrire un carré. Il convient donc de bien définir le domaine de connaissances que l'on veut conceptualiser.

Depuis, plusieurs définitions de l'ontologie ont été proposées. En 1997, W. N. Borst (Borst, 1997) modifie légèrement la définition proposée par T. Gruber en énonçant : « *une ontologie est définie comme étant une spécification explicite et formelle d'une conceptualisation partagée* ». Dans cette définition, le terme *conceptualisation* signifie un modèle abstrait d'un phénomène basé sur l'identification de concepts significatifs. Le terme *explicite* signifie que l'ensemble des concepts utilisés et leurs contraintes d'utilisation sont définis d'une façon explicite. L'adjectif *formel* précise que l'ontologie construite doit être lisible par un ordinateur. Enfin, le terme *partagée* montre qu'une ontologie fournit un vocabulaire conceptuel commun et une compréhension partagée par la communauté visée.

2.1.2 Comparaison de la notion d'ontologie et les notions voisines

Afin de préciser la notion d'ontologie, le travail dans (Mizuguchi, 2003) propose une distinction entre cette notion et d'autres qui peuvent être rapprochées.

A. Ontologie versus liste de termes

Les similitudes et les différences existantes entre le concept de l'ontologie et celui de la liste de termes sont distinguées comme suit.

- **Similitudes.** A l'instar d'une liste de termes, une ontologie fournit un vocabulaire commun pour une activité donnée.
- **Différences.** Ce qui distingue le concept de l'ontologie de celui de la liste de termes est la présence d'une structure (en particulier par des liens *is-a*) dans l'ontologie.

B. Ontologie versus hiérarchie de concepts

L'ensemble de similitudes et de différences entre une ontologie et une hiérarchie de concepts est recensé comme suit.

- **Similitudes.** Comme une hiérarchie de concepts, une ontologie comporte une représentation arborescente des concepts (graphe *is-a*).
- **Différences.** Une hiérarchie de concepts n'est pas suffisante en tant qu'ontologie. Dans l'exemple ci-dessous (cf. figure 2), la classification proposée à gauche donne une hiérarchie de la notion de véhicule, mais ne permet pas de définir plusieurs aspects de ce qu'est un véhicule : ses fonctions, les parties qui le composent, etc.

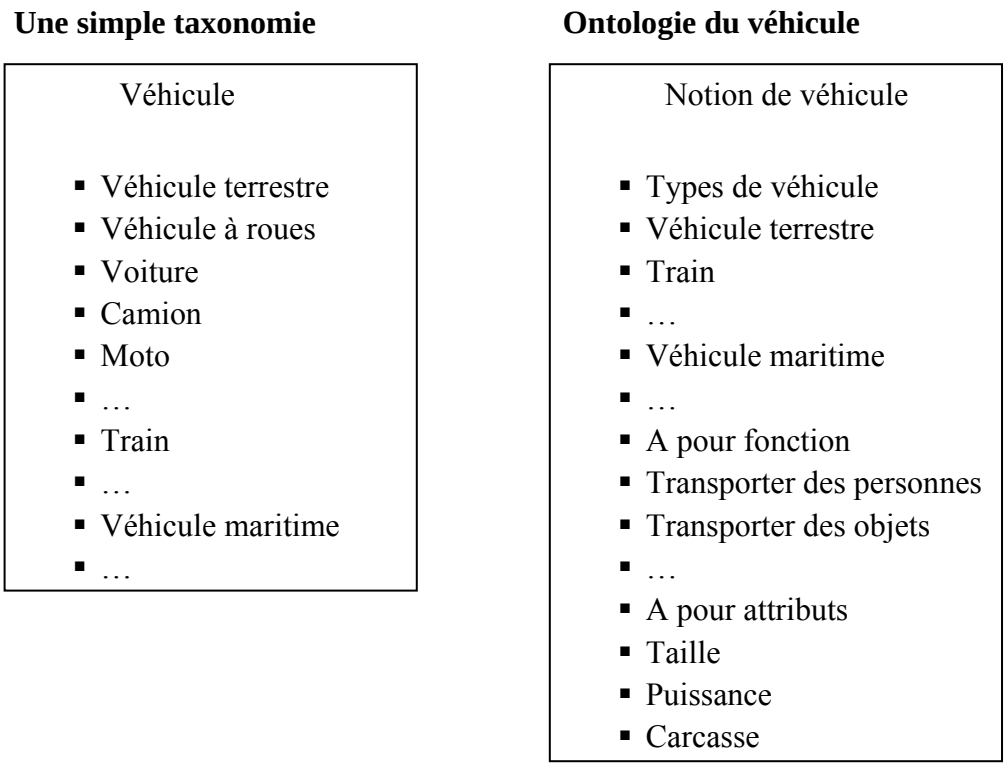


Figure 2. Taxonomie (à gauche) et ontologie (à droite) des véhicules (Mizuguchi, 2003)

2.2 Structuration des ontologies

Les ontologies sont basées sur l'utilisation d'un certain nombre de composantes auxquelles des propriétés spécifiques peuvent être associées.

2.2.1 Constituants de base

Les connaissances traduites par une ontologie sont véhiculées à l'aide d'un ensemble de cinq sortes de composants principaux (Fürst, 2004):

- Les concepts ;
- Les relations ;
- Les fonctions ;
- Les axiomes ;
- Les règles.

A. Les concepts

Un concept représente un objet ou une notion du domaine. C'est une représentation de l'esprit qui abrège et résume une multiplicité d'objets empiriques ou mentaux par abstraction et généralisation des traits communs identifiables (Mellal, 2007). Il peut être divisé en trois parties (Fürst, 2004) : Un terme (ou plusieurs) représentant son nom, des attributs et des instances.

Les attributs, également appelés l'intension du concept, correspondent à des caractéristiques, des spécificités particulières, attachées à un concept et permettent de le définir de manière unique dans le domaine (Luong, 2007). Leurs valeurs sont littérales, i.e., de type primitif, comme une chaîne de caractères ou un nombre entier. Par exemple, le concept « Auteur » peut avoir les attributs tels que : avoir un « nom, » une « date de naissance », une « adresse », etc.

D'autre part, les instances, aussi appelées extension du concept, représentent les objets manipulés à travers le concept (Fürst, 2004). Par exemple, l'instance "Mouloud Feraoun" est du type concept « Auteur ».

Exemple. Une bibliothèque scolaire est caractérisée par un bibliothécaire, des livres de différents domaines, des magazines, etc. On peut définir un concept « Livre » pour désigner les livres. L'intension de ce concept peut être l'ISBN, l'auteur et le Titre du livre. Tandis que son extension pourra être : Les misérables, ontology matching, etc.

Dans (Dieng, Corby, Giboin, & Ribière, 1998), les auteurs affirment l'existence d'une dualité entre l'intension et l'extension : à des intensions incluses $I_1 \subset I_2$ correspondent des extensions incluses $E_1 \supset E_2$. Par exemple, la notion de "voiture" (I_2) inclut la notion de "véhicule" (I_1) et l'extension de I_2 E_2= {la voiture d'Ali, la voiture de Souad} est incluse dans l'extension de I1 E1={le camion de Mohamed, la voiture d'Ali, la voiture de Souad, le vélo de Rahim}.

Par ailleurs, il est remarqué que deux concepts peuvent partager la même extension sans pour autant avoir la même intension (Fürst, 2004). C'est le cas des concepts d'« étoile du matin » et d'« étoile du soir », qui désignent tous les deux Vénus. De plus, des concepts

partageant la même extension mais pas la même intension peuvent être désignés par le même terme. Ceci correspond à des points de vue différents sur un même objet. Par exemple, les chiens peuvent être considérés comme des animaux de compagnie dans le domaine de la botanique, ou comme des ressources culinaires dans le domaine de la gastronomie chinoise.

Les concepts manipulés dans un domaine de connaissance sont organisés au sein d'un réseau de concepts (Fürst, 2004). L'exemple du concept « Table » montre que cette notion ne peut se définir qu'en utilisant d'autres concepts comme « Meuble », « Plateau » et « Pied ».

B. Les relations

Une relation représente un type d'interaction entre les concepts d'un domaine. Elle est caractérisée par un terme (voir plusieurs) représentant son nom et une signature qui précise son domaine et son co-domaine (Luong, 2007). Le domaine spécifie les concepts susceptibles d'initialiser une relation tandis que le co-domaine sert à contraindre les domaines de ses valeurs. Par exemple, une relation désignée par le terme « Ecrire » peut indiquer une relation entre les concepts « Auteur » et « Livre » dans laquelle « Auteur » est le domaine et « Livre » est le co-domaine.

Lors du développement d'une ontologie, décider entre la représentation d'une notion sous forme d'un concept ou d'une relation peut parfois s'avérer nécessaire (Fürst, 2004). Par exemple, l'écriture d'un texte peut être vue comme un concept et le fait qu'une personne écrive sera exprimé en disant que c'est en relation avec le concept « Ecriture », concept lui-même en relation avec un « Texte ». Le choix dépend alors essentiellement de l'usage des termes dans le domaine, et des liens qu'ils entretiennent avec d'autres concepts ou relations dans l'ontologie.

La relation de subsomption « est-un » (is-a) a un statut particulier car elle structure la hiérarchie ontologique (Baneyx, 2007). Un concept C1 subsume un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2, c'est-à-dire si C2 est plus spécifique que C1 (Fürst, 2002). Ainsi, l'extension d'un concept subsumé est forcément plus réduite que celle du concept qui le subsume. Son intension est par contre plus riche. Par exemple, *Homme* subsume *Humain*.

Cependant, la relation de subsomption n'est pas la seule relation qui permet de structurer la hiérarchie ontologique (Baneyx, 2007). Certains domaines, tels que le domaine des connaissances anatomiques en médecine, utilisent plutôt la relation de méronymie, « partie-tout » (part-of).

Enfin, il est à noter que tout comme les concepts, les relations sont organisées de manière hiérarchisée à l'aide de la propriété de subsomption.

C. Les fonctions

Une fonction est un type particulier de relations dans lesquelles le nième élément de la relation est défini de manière unique à partir des n-1 éléments précédents (Chamoun, 2006). Formellement, les fonctions sont définies ainsi : $F : C_1 \times C_2 \ldots \times C_{n-1} \rightarrow C_n$. Comme exemple de fonctions binaires, il y a la fonction mère-de ou carré-de, comme fonction ternaire, le prix d'une voiture usagée sur lequel on peut se baser pour calculer le prix d'une voiture d'occasion en fonction de son modèle, de sa date de construction et de son kilométrage.

D. Les axiomes

Les axiomes représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique (Staab & Maedche, 2000). Ils sont des expressions qui sont toujours vraies. Leur inclusion dans une ontologie peut avoir plusieurs objectifs (Hernandez, 2006): définir la signification des composants, définir des restrictions sur la valeur des attributs ou définir les arguments d'une relation.

Exemple. Dans le domaine de la géométrie, l'axiome de Hilbert « Sur *une droite, il y a au moins deux points* » est une propriété de cardinalité : la relation d'appartenance d'un point à une droite porte une propriété de cardinalité minimum de 2 par rapport à la droite.

E. Les règles

Les règles sont un mécanisme d'inférence de connaissances (Carrillo Ramos, 2007). Dans le domaine des ontologies, les règles s'appliquent sur des faits qui représentent les concepts et les relations d'un domaine. Une règle est une contrainte explicite sur des comportements, fournissant un support à ces faits et à la conduite des activités. Par exemple Si Leïla est la fille de Naïma et que Nassima est la sœur de Naïma, alors on peut déduire que Leïla est la nièce de Nassima sans avoir besoin de le spécifier explicitement, grâce à la règle générale :

(X= fille de Y) et (Y= sœur de Z) $\Rightarrow$ (X= nièce de Z), qui définit, en fait, le terme « nièce ».

2.2.2 Les propriétés portant sur les concepts

Il est possible d'associer aux concepts un certain nombre de propriétés qui peuvent porter aussi bien sur leur extension que sur leur intention. Nous citons ici d'une manière non exhaustive certaines d'entres elles telles qu'elles sont présentées dans (Fürst, 2002) :

- **La généricité.** Un concept est générique s'il n'admet pas d'extension. Par exemple, la vérité est un concept générique;

- **L'abstraction.** Un concept est abstrait si toute instance de ce concept est aussi instance d'un de ses concepts fils. Par exemple, dans une hiérarchie comportant les concepts

« Homme » et « Femme » qui sont fils du concept « Humain ». Le concept « Humain » est abstrait ;

- **La rigidité.** Un concept est rigide si toute instance de ce concept en reste instance dans tous les mondes possibles. Par exemple, « Humain » est un concept rigide, « Etudiant » est un concept non rigide;
- **L'anti-rigidité.** Un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept. Par exemple, « Etudiant » est un concept anti-rigide car l'étudiant est avant tout un humain

En plus de ces propriétés intrinsèques, nous pouvons aussi citer les propriétés inter-concepts qui portent sur des propriétés extrinsèques aux concepts. Il s'agit principalement de (Guarino & Giaretta, 1995; Fürst, 2002):

- **L'équivalence.** Deux concepts sont équivalents s'ils ont la même extension. Par exemple, « Etoile du matin » et « Etoile du soir » ;
- **La disjonction.** (on parle aussi d'incompatibilité) Deux concepts sont disjoints si leurs extensions sont disjointes. Par exemple, « Homme » et « Femme » sont disjoints dans la mesure où aucune instance de l'un ne peut être à la fois un homme et une femme.

2.2.3 Les propriétés portant sur les relations

Tout comme pour les concepts, il existe aussi pour les relations un certain nombre de propriétés. On peut distinguer des propriétés intrinsèques, des propriétés inter-relations, et des propriétés liant une relation et des concepts (Izza, 2006):

- **Les propriétés intrinsèques.** Elles permettent de définir la relation. Il est principalement distingué entre (Fürst, 2004):

  - *Les propriétés algébriques.* Telles que la symétrie, la réflexivité, la transitivité et l'antisymétrie ;
  - *La cardinalité.* Il s'agit du nombre possible de relations de ce type pouvant exister entre les mêmes concepts (ou instances de concept). Par exemple, une « Pièce » a au moins une « Porte ».

- **Les propriétés inter-relations.** Elles portant sur plusieurs relations et peuvent être (Fürst, 2004):

  - *L'incompatibilité.* Deux relations sont incompatibles si elles ne peuvent pas lier les mêmes instances de concepts. Par exemple, les relations "être rouge" et "être vert" sont incompatibles;

- *L'inverse.* Deux relations binaires sont inverses l'une de l'autre si, quand l'une lie deux instances I_1 et I_2, l'autre lie I_2 et I_1. Par exemple, les relations *fils-de* et *père-de* sont inverses ;
- *L'exclusivité.* Deux relations sont exclusives si, quand l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité. Par exemple, l'appartenance et la non appartenance sont exclusives.

▪ **Les propriétés liant une relation et des concepts.** Elles sont définies dans (Kassel, 2002) comme suit :

- *Le lien relationnel.* Il existe un lien relationnel entre une relation R et deux concepts C1 et C2 si, pour tout couple d'instances des concepts C1 et C2, il existe une relation de type R qui lie les deux instances de C1 et C2. Un lien relationnel peut en outre être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept. Par exemple, il existe un lien relationnel entre les concepts « Texte » et « Auteur » d'une part et la relation « a pour auteur » d'autre part;
- *La restriction de relation.* Pour tout concept de type C1, et toute relation de type R liant C1, les autres concepts liés par la relation sont d'un type imposé. Par exemple, si la relation « mange » portant sur une « Personne » et un « Aliment » lie une instance de « Végétarien », concept subsumé par « Personne », l'instance de « Aliment » est forcément instance de « Végétaux ».

2.2.4 Exemple d'une ontologie du domaine de la recherche scientifique

La Figure suivante présente une ontologie décrivant les connaissances du domaine des activités d'un institut de recherche (Luong, 2007). Dans cette ontologie, nous avons des concepts tels que *Personne*, *Doctorant*, *Chercheur* qui sont classés en ordre hiérarchique pour représenter des employés qui travaillent dans (i.e. la relation) les équipes de recherche (i.e. le concept *Equipe_Recherche*) ou dans le département d'administration (i.e. le concept *Administration*), etc.

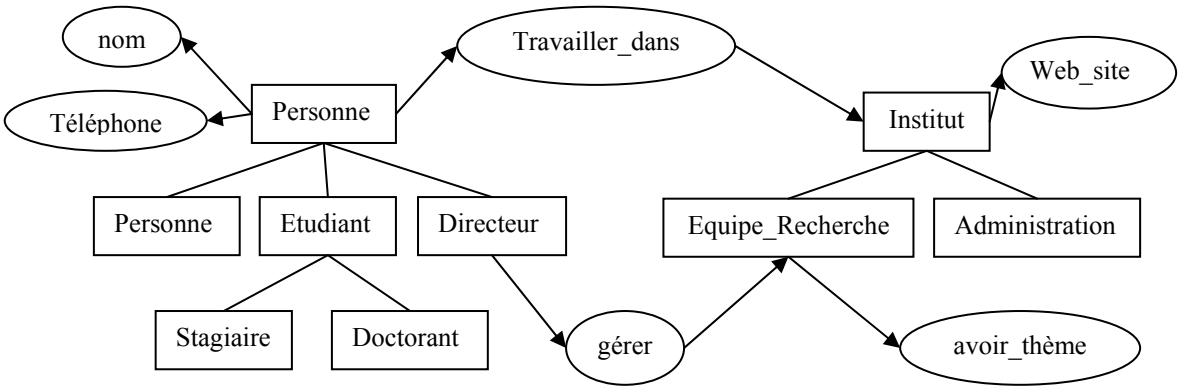


Figure 3. Exemple d'un extrait de l'ontologie

2.3 Classification des ontologies

La construction d'une ontologie doit se faire à partir d'un champ de connaissances bien délimité, et porter sur des connaissances objectives dont la sémantique peut être exprimée rigoureusement et formellement (Fürst, 2004). Partant de ce constat, plusieurs types d'ontologies peuvent être distingués en fonction des différents types de connaissances modélisées.

La classification des ontologies a fait l'objet d'étude de plusieurs travaux (Mizuguchi, 2003; Psyché, Mendes, & Bourdeau, 2003; Uschold & Gruninger, 1996). Cette section n'a pas l'ambition de fournir un état de l'art sur ces classifications. Elle présente néanmoins la classification la plus courante qui est celle définie dans (Psyché, Mendes, & Bourdeau, 2003). Elle distingue six types d'ontologies :

- **Les ontologies supérieures** (aussi appelées ontologies de haut niveau). Ces ontologies modélisent le travail réalisé par les philosophes dans leur travail d'explication de ce qui existe dans le monde. En particulier, les ontologies de haut niveau modélisent les concepts les plus généraux que l'on puisse définir. On peut citer ici par exemple les dix catégories d'Aristote : la matière, la quantité, la qualité, la relation, la position, le temps, etc.
- **Les ontologies génériques.** Elles contiennent des concepts généralistes, mais moins abstraits que ceux contenus dans les ontologies de haut niveau. On pourra réutiliser dans plusieurs domaines les connaissances que l'on y trouve.
- **Les ontologies de tâches.** Ce type d'ontologie est distingué dans (Mizuguchi, 2003). Elles fournissent un lexique systématisé de termes utilisés pour résoudre les problèmes associés à des tâches particulières (faire un diagnostic, planifier une activité) indépendamment d'un domaine particulier. Ces ontologies fournissent un ensemble de termes au moyen desquels on peut décrire au niveau générique comment résoudre un type de problème. Elles incluent des noms génériques (par exemple, plan, objectif, contrainte), des verbes génériques (par exemple, assigner, classer, sélectionner) et des adjectifs génériques (par exemple, assigné).
- **Les ontologies de domaine.** Ces ontologies expriment des conceptualisations spécifiques d'un domaine. Elles sont réutilisables par plusieurs applications de ce domaine. Par exemple, dans le contexte du e-learning, le domaine peut être celui de la formation. Les concepts peuvent être de plusieurs types : personnes (étudiant, tuteur, secrétaire, etc.), documents (livres, supports de présentation, page Web, etc.), numérique (texte, image, audio, vidéo, etc.). La plupart des ontologies existantes sont des ontologies de domaine.
- **Les ontologies de tâches-domaine.** Ce sont des ontologies de tâches spécifiques à un certain domaine. Un exemple d'une telle ontologie est celui d'une ontologie des termes liés à la planification chirurgicale

- **Les ontologies d'application**. Il s'agit du type d'ontologie le plus spécifique. Elles contiennent des connaissances du domaine nécessaires à une application donnée. Elles sont spécifiques et non réutilisables. Ce type d'ontologie décrit des concepts qui dépendent à la fois d'un domaine particulier et d'une tâche particulière. Elles sont souvent des spécialisations à la fois des ontologies de domaine et des ontologies de tâches et correspondent aux rôles joués par les entités de domaine lorsqu'elles effectuent certaines activités. Par exemple, dans le contexte du e-learning, une application peut être : la formation de Statistiques et Probabilités.

3 Complexité sémantique des ontologies et les langages ontologiques du Web sémantique

Lors du développement d'une ontologie, choisir le langage dans lequel elle sera exprimée et utilisée est une décision importante à prendre. Des contraintes sont à prendre en considération. Dans (Baneyx, 2007), la lisibilité et la portabilité sont considérées. Le langage doit être compréhensible pour un utilisateur humain et doit donc avoir une certaine continuité avec le langage naturel pour être lisible. La portabilité est en rapport avec le fait que le langage doit être le plus standard possible afin de pouvoir être réutilisée dans d'autres systèmes. De plus, dans (Gandon, 2002), les auteurs considèrent d'autres critères pour la sélection des langages. Il s'agit de l'expressivité qui est liée au nombre d'éléments pouvant être utilisés pour décrire les composants d'une ontologie et de la performance du langage qui est en rapport avec la capacité et la facilité de représentation des connaissances du domaine. Cependant, d'après (Fensel, 2001), il semblerait que plus un langage est expressif moins il est performant, et vice versa. Il existe aussi un autre dilemme entre l'expressivité et le raisonnement dans le sens où l'expressivité d'un langage doit être parfois limitée afin d'assurer un bon service de raisonnement (Izza, 2006).

Par ailleurs, dans le cadre de ses travaux sur le Web sémantique, le W3C a mis en place un groupe de travail dédié au développement de langages standard pour représenter des ontologies utilisables et échangeables sur le Web assurant ainsi les contraintes de lisibilité et de portabilité. Ce groupe a publié trois langages qui semblent aujourd'hui faire un consensus. Il s'agit de RDF (S), de OWL (Ontology Web Language) et de SWRL. Cependant, ces langages n'ont pas tous la même puissance d'expression.

Afin de faciliter le choix du langage ontologique dans le cadre du Web sémantique, les auteurs dans (Castano, Ferrara, Hess, & Montanelli, 2007) ont identifié des niveaux de complexité sémantique des ontologies. Une association entre ces niveaux et les langages ontologiques a été proposée (cf. figure 4). Chaque langage fournit des constructions pour exprimer les caractéristiques correspondantes à son niveau.

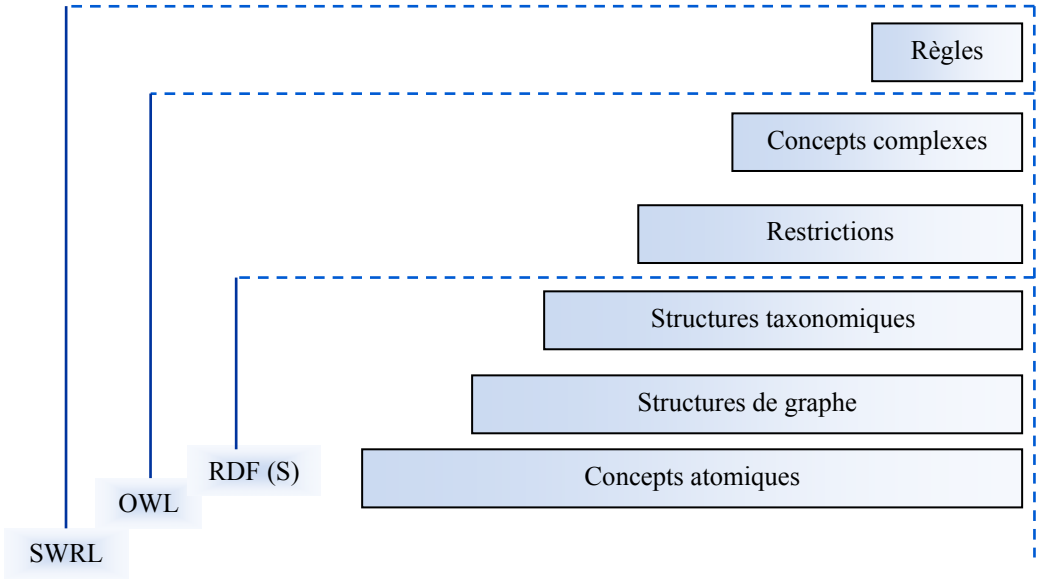


Figure 4. Représentation graphique de la complexité sémantique (Castano, Ferrara, Hess, & Montanelli, 2007)

3.1 RDF (S)

RDF(S)[1] est un langage combinant RDF et RDF Schéma. Il permet d'exprimer les trois premiers niveaux de la complexité sémantique des ontologies. Ces derniers sont présentés comme suit.

A. Les concepts atomiques

Au premier niveau de la complexité sémantique des ontologies, les entités sont désignées seulement par un nom (rdf:id), une étiquette (rdfs:label) et un commentaire (rdfs:comment). Les autres caractéristiques, telles que les relations avec d'autres entités, ne sont pas prises en compte.

Exemple. Le code RDF(S) suivant présente la spécification de cinq concepts atomiques : *Artiste*, *Film* et *Rôle* qui sont des classes, *jouer* qui est une relation et *Rouiched* qui est une instance de la classe *Artiste*.

```
<rdfs:Class rdf:ID="Artiste">
  <rdfs:label xml:lang="en">Artist</rdfs:label>
  <rdfs:label xml:lang="en">performer</rdfs:label>
  <rdfs:label xml:lang="fr">Acteur</rdfs:label>
```

[1] http://www.w3.org/TR/rdf-schema/

```
  <rdfs:comment xml:lang="fr"> a person whose creative work
  shows sensitivity and imagination</rdfs:comment>
</rdfs:Class>
<rdfs:Class rdf:ID="Film">
  <rdfs:comment xml:lang="fr">a form of entertainment that
  enacts a story by a sequence of images giving the illusion of
  continuous movement; "they went to a movie every Saturday
  night"; "the film was shot on location"</rdfs:comment>
</rdfs:Class>
<rdfs:Class rdf:ID="Role"/>
<rdf:Property rdf:ID='Jouer'>
<rdf:Description rdf:ID="Rouichede">
  <rdf:type rdf:resource="#Artiste"/>
</rdf:Description>
```

B. Les structures de graphes

Au deuxième niveau, les ontologies sont vues comme des structures de graphes où les nœuds sont des concepts et les arcs sont des relations entre les concepts. En RDF(S), nous pouvons décrire les relations de l'ontologie en spécifiant leurs domaines (rdfs:domain) et leurs co-domaines (rdfs:range).

Exemple. Soit le code RDF(S) suivant.

```
<rdf:Property rdf:ID="jouer">
  <rdfs:domain rdf:resource="Role"/>
  <rdfs:range rdf:resource="Artiste"/>
</rdf:Property>
<rdf:Property rdf:ID="dirige_par">
  <rdfs:domain rdf:resource="Film"/>
  <rdfs:range rdf:resource="Artiste"/>
</rdf:Property>
<rdf:Property rdf:ID="titre">
  <rdfs:domain rdf:resource="Film"/>
  <rdfs:range rdf:resource="rdfs:Literal"/>
</rdf:Property>
```

Ce code indique la structure de graphe représentée par la figure suivante.

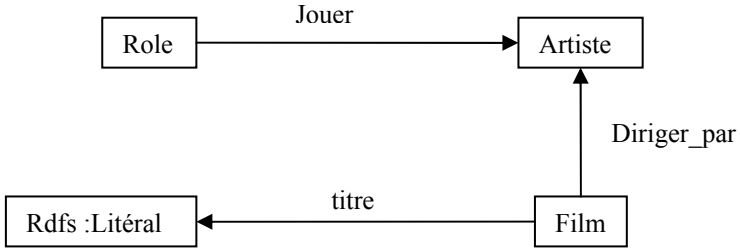


Figure 5. Représentation d'une structure de graphe

C. Les structures taxonomiques

À ce niveau, les ontologies sont vues comme des taxonomies de concepts et de relations qui sont respectivement spécifiées à l'aide des primitives `rdfs:subClassOf` et `rdfs:subPropertyOf`.

Exemple. Le code RDF(S) représente une ontologie en spécifiant la hiérarchie de ses concepts.

```
<rdf:RDF
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd ="http://www.w3.org/2001/XMLSchema#"
 xml:base ="http://www.usthb.dz/lria/exemple#">
   <rdfs:Class rdf:ID="Artiste"/>
   <rdfs:Class rdf:ID="Film"/>
   <rdfs:Class rdf:ID="Comedie">
     <rdfs:subClassOf rdf:resource="#Film"/>
   </rdfs:Class>
   <rdfs:Class rdf:ID="Drame">
     <rdfs:subClassOf rdf:resource="#Film"/>
   </rdfs:Class>
   <rdfs:Class rdf:ID="Action">
     <rdfs:subClassOf rdf:resource="#Film"/>
   </rdfs:Class>
</rdf:RDF>
```

Cependant, les ontologies représentées en RDF(S) ne sont que les ontologies « simples » (Bach, 2006). La sémantique exprimée par ces ontologies est limitée par la puissance d'expression du langage RDF(S). Nous ne pouvons pas exprimer la cardinalité d'une relation, ou le fait qu'une relation soit transitive, symétrique ou anti-symétrique ou réflexive.

De ce fait, si le niveau de complexité sémantique de l'ontologie à développer est supérieur à la puissance d'expression de RDF(S), l'ontologiste doit utiliser OWL.

3.2 OWL

OWL (Ontology Web Langage) (OWL, 2004) est un langage de représentation d'ontologies possédant le statut de recommandation du W3C depuis février 2004. Il reprend la puissance d'expression de RDF(S) en y ajoutant notamment la possibilité de déclarer des restrictions sur les propriétés et de définir des concepts complexes.

A. Les restrictions

L'élément owl:Restriction permet de définir une classe anonyme (Lapique, 2006). La restriction peut s'exprimer sur le co-domaine ou le domaine d'une propriété ou sur la cardinalité d'une propriété.

Exemples. On définit une classe anonyme regroupant des cours assurés uniquement par des enseignants ayant le titre de professeur.

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#estEnseignePar"/>
  <owl:allValuesFrom rdf:resource="#professeur"/>
</owl:Restriction>
```

On peut ensuite définir une classe comme sous-classe d'une classe anonyme définie par une restriction. Par exemple : chaque instance de cours de 1ère année ne peut être enseignée que par un enseignant professeur (contrainte avec quantificateur universel).

```
<owl:Class rdf:about="#cours1ereAnnee">
  <rdfs:subClassOf>
    <owl:Restriction>
     <owl:onProperty rdf:resource="#estEnseignéPar"/>
     <owl:allValuesFrom rdf:resource="#professeur"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

On peut exprimer aussi une restriction de nature existentielle, par exemple : un enseignant doit enseigner au moins un cours de master.

```
<owl:Class rdf:about="#Enseignant">
  <rdfs:subClassOf>
    <owl:Restriction>
     <owl:onProperty rdf:resource="#enseigne"/>
     <owl:someValuesFrom rdf:resource="#coursMaster"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

On peut exprimer une contrainte de cardinalité, par exemple, le fait qu'un cours doit être enseigné par au moins un enseignant.

```
<owl:Class rdf:about="#cours">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#estEnseignePar"/>
         <owl:minCardinality
         rdf:datatype="&xsd;nonNegativeInteger">1
         </owl:minCardinality>
```

```
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

B. Les concepts complexes

Les concepts complexes sont exprimés à l'aide de concepts atomiques reliés par des opérateurs tels que l'intersection, l'union ou le complément.

Exemples. On définit la classe « PersonneUniversite » comme l'union de la classe des enseignants et de celle des étudiants.

```
<owl:Class rdf:about="#PersonneUniversite">
   <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Enseignant"/>
      <owl:Class rdf:about="#Etudiant"/>
   </owl:unionOf>
</owl:Class>
```

Un autre exemple mettant en œuvre trois opérateurs ensemblistes. Le personnel administratif est défini comme le personnel de l'Université n'étant ni personnel enseignant, ni personnel technique.

```
<owl:Class rdf:about="#PersonnelAdministratif">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#PersonnelUniversite"/>
    <owl:Class>
      <owl:complementOf>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Enseignant">
            <owl:Class rdf:about="#Technicien">
          </owl:unionOf>
        </owl:Class>
      </owl:complementOf>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```

Bien que OWL permette d'enrichir la représentation d'une ontologie par la définition des restrictions et des concepts complexes, il lui manque des possibilités pour encoder des connaissances plus générales, relatives en particulier à la composition des relations (Chamoun, 2006). Par exemple, la définition du concept Oncle en OWL est comme suit:

```
intersectionOf(SubClassOf(Homme), estfrereDe(Pere)).
```

Par cette déclaration, nous savons qu'une personne est un Oncle, mais nous ne savons pas de qui**.** Par conséquent, OWL ne permet pas de définir une relation qui représente le fait d'*être oncle d'une personne*. Le langage du Web sémantique recommandé pour résoudre ce problème est SWRL.

3.3 SWRL et les règles

SWRL (Semantic Web Rule Language) (Horrocks, Patel-Schneider, Boley, Tabet, Grosof, & Dean, 2004), est un langage qui enrichit la sémantique d'une ontologie définie en OWL avec des règles.

Les règles SWRL sont construites suivant ce schéma: antécédent -> conséquent tel que l'antécédent et le conséquent sont des conjonctions d'atomes. Un atome est une instance de concept, une relation OWL ou une des deux relations SWRL `same-as(?x,?y)` ou `different-from (?x,?y)` avec x et y des variables.

Le fonctionnement d'une règle est basé sur le principe de satisfiabilité de l'antécédent ou du conséquent. Pour une règle, il existe trois cas de figure:

- l'antécédent et le conséquent sont définis. Si l'antécédent est satisfait alors le conséquent doit l'être;
- l'antécédent est vide. Cela équivaut à un antécédent satisfait ce qui permet de définir des faits;
- le conséquent est vide. Cela équivaut à un conséquent insatisfait, l'antécédent ne doit pas être satisfiable.

Exemple. La relation *être oncle d'une personne* est spécifiée en SWRL comme suit.

```
<swrl:Imp rdf:ID="Def-hasUncle">
  <swrl:head>
   <swrl:AtomList>
    <rdf:first>
     <swrl:IndividualPropertyAtom>
      <swrl:argument1 rdf:resource="#x"/>
      <swrl:argument2 rdf:resource="#z"/>
      <swrl:propertyPredicate rdf:resource="#hasUncle"/>
     </swrl:IndividualPropertyAtom>
    </rdf:first>
    <rdf:rest   rdf:resource="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#nil"/>
   </swrl:AtomList>
  </swrl:head>
  <swrl:body>
   <swrl:AtomList>
    <rdf:first>
```

```
      <swrl:IndividualPropertyAtom>
      <swrl:argument2 rdf:resource="#y"/>
      <swrl:propertyPredicate rdf:resource="#hasParent"/>
      <swrl:argument1 rdf:resource="#x"/>
      </swrl:IndividualPropertyAtom>
    </rdf:first>
    <rdf:rest>
    <swrl:AtomList>
     <rdf:first>
      <swrl:IndividualPropertyAtom>
      <swrl:propertyPredicate rdf:resource="#hasBrother"/>
      <swrl:argument1 rdf:resource="#y"/>
      <swrl:argument2 rdf:resource="#z"/>
      </swrl:IndividualPropertyAtom>
     </rdf:first>
    <rdf:rest   rdf:resource="http://www.w3.org/1999/02/22-rdf-
       syntax-ns#nil"/>
    </swrl:AtomList>
   </rdf:rest>
   </swrl:AtomList>
  </swrl:body>
</swrl:Imp>
```

4 Conclusion

L'apport du Web sémantique est d'une importance capitale pour gérer intelligemment un contenu en croissance permanente à travers sa capacité de manipulation des ressources sur la base de leurs sémantiques en utilisant des ontologies. Ces dernières présentent donc une technologie clé pour ce futur Web. Plusieurs travaux ont portés sur les ontologies. Ceci a engendré de nombreuses ontologies pour le même domaine ou des domaines qui se chevauchent. Leur utilisation simultanée pour une même application peut être envisagée. Mais du fait de la diversité des formats de représentation et de modélisation, des problèmes d'hétérogénéité peuvent apparaître.

Afin de cohabiter avec cette hétérogénéité, il est nécessaire de déterminer les parties que les ontologies ont en commun. Ceci est réalisé dans le cadre de travaux de recherche relatifs à l'appariement des ontologies, objet du chapitre suivant.

CHAPITRE 2 : APPARIEMENT DES ONTOLOGIES

Beaucoup de recherches ayant pour objectifs de modéliser la connaissance d'un domaine et d'améliorer l'échange de données ont abouti à la création d'ontologies. Cependant, plusieurs de ces ontologies concernent le même domaine. Ainsi, il est vite apparu le besoin de disposer d'outils permettant de faire le lien entre toute cette connaissance. Ce qui ne va pas sans engendrer des problèmes, en particulier, ceux liés à l'hétérogénéité.

L'hétérogénéité se situe tant au niveau de la syntaxe qu'au niveau de la sémantique. La première implique la structure du formalisme de représentation de la connaissance, et elle varie en fonction du langage choisi. La deuxième, par contre, est liée à l'étude du contexte dans lequel les concepts sont représentés, et elle est identifiée par une étude sur le contenu des concepts. Cette hétérogénéité posera des problèmes dans l'échange, le traitement, l'intégration, et la recherche d'informations. Aussi et afin d'y faire face, des correspondances entre les entités similaires des ontologies doivent être spécifiées. Ces correspondances constituent une « colle » qui maintient les ontologies ensemble afin de garantir leur interopérabilité. Le processus permettant de découvrir ces correspondances est appelé « appariement ». L'ensemble des correspondances résultant de ce dernier est appelé « alignement ».

Nous nous intéressons dans ce chapitre au problème d'appariement de plusieurs ontologies de domaine. Nous relevons dans la section 1, les problèmes d'hétérogénéité qu'un processus d'appariement doit résoudre. Dans la section 2, nous introduisons les concepts de base liés à notre domaine de recherche, à savoir : L'appariement, l'alignement, les correspondances et la similarité. Puis, afin d'éclaircir ces concepts, nous donnons, dans la section 3, un exemple d'alignement de deux ontologies d'université. Aussi, les différents types d'alignement que l'opération d'appariement peut fournir sont présentés dans la section 4. Dans la section 5, nous présentons les applications possibles pour l'appariement des ontologies. Enfin, la section 6 est consacrée à la conclusion du chapitre.

1 Hétérogénéité des ontologies

Il est évident que la principale raison qui motive la recherche dans le domaine de l'appariement des ontologies est l'hétérogénéité qui caractérise ces ontologies. Cette hétérogénéité reflète la vie dans le monde réel où les informations peuvent être extraites différemment à partir d'une même source par des organisations ou des personnes différentes (Bach, 2006). Quelqu'un peut dire qu'un « livre est écrit par un auteur » alors qu'un autre, affirmera qu'un «ouvrage est soit une monographie dont l'écrivain est X », soit une « revue contenant des articles». Nous remarquons qu'un désaccord peut se produire au niveau de la conceptualisation.

La terminologie choisie pour dénoter les concepts dépend des objectifs fixés et de la subjectivité de la personne qui modélise l'ontologie (Bach, 2006). Les termes utilisés peuvent être identiques, synonymes, différents, etc. L'un peut employer le terme « voiture » dans une ontologie pour dénoter des objets qui sont des véhicules à quatre roues dans le monde réel, quant à l'autre, il peut utiliser le terme « automobile » pour signifier l'objet en question.

La différence peut aussi provenir de la granularité dans la spécialisation des concepts. L'un peut considérer que le concept « Personne » se divise en deux sous-concepts « Homme » et « Femme » tandis que l'autre préfère le diviser en trois sous-concepts différents, à savoir : «Enfant», « Adulte » et « Personne âgée ».

Plusieurs travaux relatifs à la classification des types de disparités entre les ontologies existent (Goh, 1997; Wache, et al., 2001; Klein, 2001). Dans cette section, nous choisissons de présenter la classification proposée dans (Klein, 2001) puisqu'elle est la plus rappelée dans la littérature. Cette classification distingue fondamentalement deux niveaux de disparités qui sont (cf. figure 1):

- les disparités au niveau des langages ;
- les disparités au niveau des ontologies.

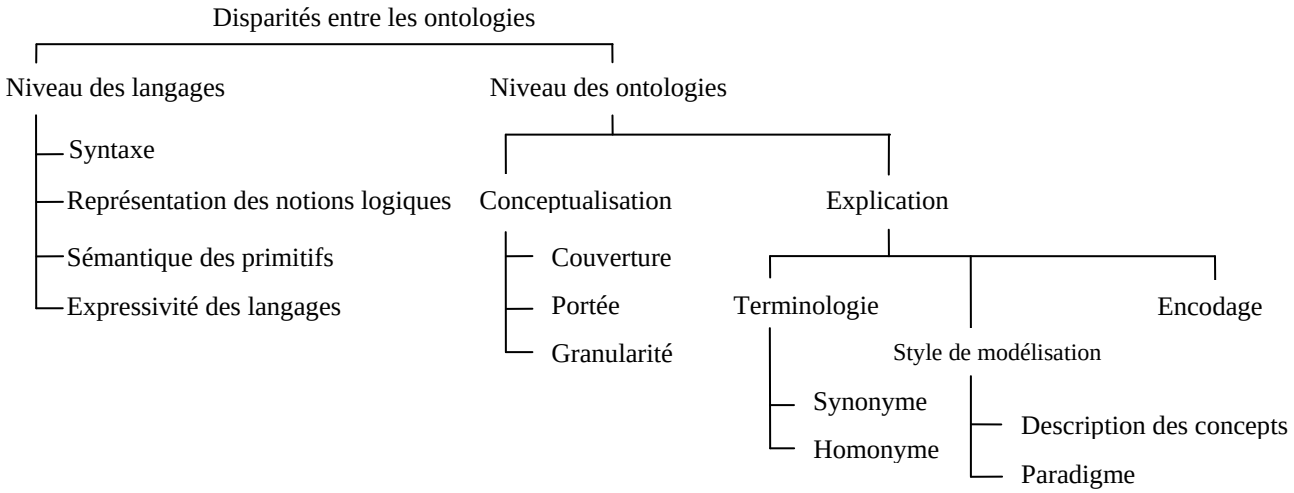


Figure 1. Disparités des ontologies (Klein, 2001)

1.1 Niveau des langages

Les disparités au niveau des langages se produisent lorsqu'on veut combiner des ontologies écrites dans des langages différents. Elles surgissent dans les mécanismes utilisés pour définir les concepts, les relations, etc.

Ces disparités peuvent apparaître dans la syntaxe, dans la représentation des notions logiques, dans la sémantique des primitives ou dans l'expressivité des langages.

1.1.1 Syntaxe

Les différents langages ontologiques emploient souvent des syntaxes différentes. Par exemple, la définition du concept « Chaise » est exprimée dans RDF(S) par la primitive « *<rdfs : Classe ID= "Chaise" >* ». Tandis que dans OIL, le constructeur « *class-def Chaise »* est utilisé pour le même but.

La résolution de ce type de disparité est généralement simple en mettant en œuvre un processus de translation ou de traduction.

1.1.2 Représentation des notions logiques

Ce type de disparité surgit lorsqu'on veut représenter des notions logiques. Par exemple, pour représenter la disjonction de deux classes, certains langages permettent de l'énoncer explicitement par la primitive « disjoint A B »; tandis que d'autres exigent d'utiliser la négation dans des relations de sous-classe : « A subclass-of (NOT B), B subclass-of (NOT A) ».

1.1.3 Sémantique des primitives

Une disparité plus délicate au niveau des langages peut être détectée lorsque deux langages contiennent des primitives nommées identiquement mais avec des sémantiques différentes. On peut citer l'exemple de la déclaration successive de plusieurs primitives <rdfs : domain>. Dans OIL RDF(S) cette déclaration signifie une intersection d'arguments. Tandis que dans RDF(S), elle est interprétée comme une union d'arguments.

1.1.4 Expressivité des langages

La possibilité de translater une ontologie écrite dans un langage donné vers un autre langage est strictement liée à la puissance d'expression des langages source et cible. La divergence dans la puissance d'expression signifie qu'un langage permet d'exprimer des choses qu'un autre langage ne permet pas. Par exemple, le langage KIF (KIF, 2004) permet d'exprimer toute la logique du premier ordre, tandis que OWL DL n'exprime qu'un sous-ensemble de cette logique. Par conséquent, la translation d'une ontologie vers un langage de moindre expressivité risque de provoquer une perte d'information.

1.2 Niveau des ontologies

Ces disparités se produisent lorsque des ontologies modélisées différemment ou décrivant des domaines partiellement recouvrants sont combinées. Elles sont de deux types: les disparités de conceptualisation et les disparités d'explication.

1.2.1 Conceptualisation

Ce type de disparités ne peut pas être résolu automatiquement. Il exige les décisions et la connaissance d'un expert du domaine. Dans (Klein, 2001) trois types de disparités de conceptualisation sont distinguées : la portée, la couverture ainsi que la granularité. Tandis que dans (Bouquet, et al., 2004), en plus de la portée, la couverture et la granularité, la perspective est ajoutée. Ces quatre notions sont détaillées dans ce qui suit :

- **Portée.** Les disparités de la portée surgissent lorsque deux classes semblent représenter le même concept, sans posséder les mêmes instances. Par exemple, toutes les administrations ont la même compréhension du terme « Employé ». Cependant, dans la pratique, un employé aura des droits et des devoirs différents d'une entreprise à une autre.
- **Couverture.** Les ontologies diffèrent aussi souvent par leur couverture d'un domaine particulier. Par exemple, une ontologie o_1 modélise les voitures et les camions, pendant qu'une autre ontologie o_2 ne modélise que les camions.
- **Granularité.** La granularité, avec laquelle les distinctions sont faites, peut être aussi une source de disparité entre les ontologies. Une ontologie de fine granularité est une

ontologie très détaillée, possédant ainsi un vocabulaire riche capable d'assurer une description détaillée des concepts d'un domaine. Une ontologie de large granularité correspond à un vocabulaire moins détaillé. Par exemple, l'ontologie o_1 représente les camions sous des catégories basées sur leur structure physique, poids, but, etc., tandis que l'ontologie o_2 ne prend pas en considération tous ces détails.

- **Perspective.** Ce type de disparités surgit lorsqu'une ontologie représente un point de vue sur un domaine donné différent de celui représenté dans une autre ontologie. Par exemple, on peut définir deux ontologies décrivant les composants microélectroniques avec deux points de vue différents: l'une concerne l'aspect fabrication et l'autre concerne l'aspect marketing.

La figure suivante fournit une représentation graphique des trois derniers types de disparités de conceptualisation.

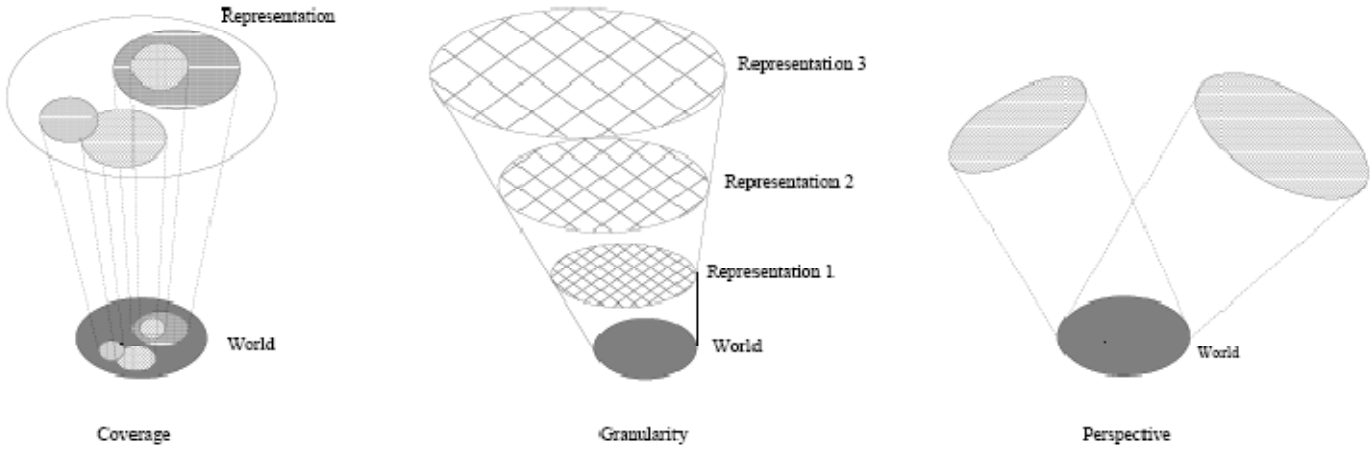


Figure 2. Les disparités de conceptualisation : Couverture, Granularité et Perspective (Bouquet, et al., 2004).

1.2.2 Explication

Les disparités d'explication sont de trois types. Elles concernent les styles de modélisation, les terminologies et l'encodage des données.

1.2.2.1 Style de modélisation

Les ontologies diffèrent dans leurs styles de modélisation sur les paradigmes et les descriptions des concepts qu'elles utilisent.

- **Paradigme.** Des paradigmes différents peuvent être utilisés pour représenter des concepts comme le temps, l'action, etc. Par exemple, on pourra trouver des ontologies qui décrivent un cercle par un point et un rayon, alors que d'autres le décrivent par un ensemble de trois points.

- **Description des concepts.** Plusieurs choix peuvent être faits pour la modélisation des concepts dans une ontologie. Par exemple, le concept « Personne » peut être modélisé soit avec la construction de deux sous-concepts « Mâle » et « Femelle », ou seulement avec la définition d'un attribut « sexe » pour déterminer le sexe de la personne.

1.2.2.2 Terminologie

Les disparités terminologiques sont des disparités liées au processus de nommage qui associe un objet linguistique aux entités décrites dans une ontologie. Les exemples typiques de ces disparités sont l'utilisation de termes synonymes ou homonymes :

- **Termes synonymes.** Deux ontologies peuvent utiliser des termes synonymes pour se référer aux mêmes entités. Un exemple trivial est l'utilisation du terme « Voiture » dans une ontologie et le terme « Automobile » dans une autre ontologie.
- **Termes homonymes.** La signification d'un terme est différente d'un contexte à l'autre. Par exemple, le terme « Conducteur » possède une signification différente dans le domaine de la musique que dans le domaine du génie électrique. La connaissance humaine est exigée pour résoudre cette ambiguïté.

1.2.2.3 Encodage des données

L'encodage des données au sein des ontologies diffère souvent, que ce soit pour les dates, les unités (monnaie, distances, ...), etc. Dans la plupart des cas, une étape de transformation est suffisante pour éliminer ces disparités.

2 Concepts de base liés au domaine de l'appariement des ontologies

Dans le domaine de l'appariement des ontologies, plusieurs terminologies ont été proposées. Malencontreusement, ces terminologies sont très souvent contradictoires. On peut trouver des termes différents se référant aux mêmes concepts. Parfois des concepts différents sont référencés par les mêmes termes. Par exemple, l'opération qui découvre l'ensemble de correspondances entre les entités des ontologies est référencée dans (Ehrig, 2007) par le terme 'alignement'. Ce même terme est également utilisé par l'auteur pour signifier l'ensemble de correspondances résultant de cette opération. Dans (Euzenat & Shvaiko, 2007), une distinction entre l'opération de découverte de correspondances et ses résultats est claire. L'opération est dite « appariement » tandis que l'ensemble des correspondances qui en résulte est dit « alignement ».

Dans le présent mémoire, le choix est d'adhérer à la terminologie la plus récemment proposée qui est celle de (Euzenat & Shvaiko, 2007). Ce choix est dû au fait que cette

terminologie possède une très fine granularité par rapport à d'autres terminologies (Klein, 2001; Noy & Musen, 2001; Ehrig, 2007).

2.1 Appariement, correspondance et alignement

L'appariement des ontologies est le processus de comparaison de deux ontologies et de découverte de relations ou de correspondances entre leurs entités. L'appariement est alors défini comme suit (Euzenat, Mocan, & Scharffe, 2007).

Définition (Appariement d'ontologies). *L'appariement de deux ontologies O_1 et O_2 consiste à trouver les correspondances existantes entre leurs entités.*

Le résultat du processus d'appariement est appelé « alignement ». Ce dernier exprime des correspondances entre les entités appartenant à deux ontologies différentes. Une correspondance doit considérer les entités correspondantes ainsi que la relation supposée exister entre elles.

Les entités d'une ontologie peuvent être les concepts, les relations, les fonctions, les instances, les axiomes, les règles, etc. Les types de relations possibles entre deux entités de deux ontologies peuvent être principalement la relation d'équivalence (=), la subsomption ($\geq$) le recouvrement ($\cap$) ou l'incompatibilité (#). En outre, on peut aussi avoir d'autres types de relations telles que les relations floues et les distributions probabilistes. L'ensemble de ces relations est dénoté par θ.

Pour des raisons pragmatiques, un degré de confiance est assigné à chaque relation liant deux entités. Ce degré de confiance signifie la probabilité d'existence de la relation et il est calculé par des mesures qui estiment la similarité des deux entités en question. Il est défini comme suit (Euzenat & Shvaiko, 2007).

Définition (Structure de confiance). *Une structure de confiance est un ensemble ordonné de degrés $\langle \Xi, \leq \rangle$ pour lesquels il existe un plus grand élément $\top$ et un plus petit élément $\bot$.*

La structure de confiance la plus utilisée dans les systèmes d'appariement d'ontologies est l'intervalle unitaire des nombres réels [0, 1]. Cependant, d'autres structures sont également possibles telles que le treillis booléen et les degrés flous.

Ainsi, les correspondances sont définies à partir des éléments précédemment décrits, i.e., les entités des deux ontologies, la relation d'alignement et le degré de confiance, comme suit (Euzenat J. , et al., 2007).

Définition (Correspondance). *Étant données deux ontologies O_1 et O_2, un ensemble de relations d'alignements θ et une structure de confiance sur Ξ, une correspondance est un quintuple $\langle id, e_1, e_2, r, n \rangle$ tel que :*

– *id est un identifiant unique pour la correspondance en question ;*
– $e_1 \in O_1$ *et* $e_2 \in O_2$;
– $r \in \theta$;
– $n \in \Xi$.

La correspondance $\langle id, e_1, e_2, r, n \rangle$ affirme que la relation *r* existe entre les entités e_1 et e_2 avec une confiance égale à *n*.

Un alignement est donc défini comme suit (Euzenat & Shvaiko, 2007).

Définition (Alignement). *Étant données deux ontologies* O_1 *et* O_2*, un alignement est construit à partir d'un ensemble de correspondances entre les paires d'entités appartenant à* O_1 *et à* O_2 *respectivement.*

Une fois l'appariement de deux ontologies établi, on dit que l'entité e_1 est alignée avec l'entité e_2 s'il existe une correspondance ayant un degré de confiance satisfaisant.

2.2 Similarité, Dissimilarité, Distance et Autres Mesures

Afin de trouver les relations entre les entités exprimées dans des ontologies différentes, il est nécessaire de mesurer la quantité par laquelle ces entités se rapprochent, i.e., leur degré de similarité. De ce fait, la similarité joue un rôle important dans le processus d'appariement des ontologies.

Nous présentons dans cette section la définition de la similarité ainsi que des notions qui lui sont rattachées.

2.2.1 Similarité

La notion de similarité varie selon le contexte dans lequel elle est utilisée. En psychologie, par exemple, elle se rapporte à comment les attitudes, les valeurs, les intérêts et la personnalité présentent des correspondances entre les personnes. Tandis qu'en topologie mathématique, elle est définie par une fonction sur un ensemble de points. La valeur donnée par cette fonction est plus grande quand deux points sont plus proches.

Dans le contexte de l'appariement des ontologies, la notion de similarité est en rapport avec la similarité sémantique. Elle représente une évaluation du lien sémantique entre deux entités dont le but est d'estimer le degré par lequel les entités sont proches dans leur sens (Resnik, 1995).

La similarité sémantique de deux entités telle qu'elle est définie par Lin (Lin, 1998) est liée aux caractéristiques que ces entités possèdent en commun. Elle repose sur les trois suppositions suivantes :

- Plus les entités ont des caractéristiques communes, plus elles sont similaires ;
- Moins les entités possèdent des caractéristiques communes, moins elles sont similaires ;
- La similarité maximale est obtenue lorsque deux entités sont identiques.

Dans la plupart des approches d'appariement d'ontologies, la notion de similarité sémantique est associée à une fonction, appelée *fonction de similarité* (Bach, 2006). La définition formelle suivante de la similarité est donnée dans (Euzenat, et al., 2004).

Définition (Similarité). *Étant donné un ensemble d'entités O, une similarité* $\sigma: O \times O \rightarrow R$ *est une fonction à partir d'une paire d'entités vers un nombre réel exprimant la similarité entre deux entités telle que :*

$$\forall x, y \in O, \sigma(x, y) \geq 0 \qquad (positivité)$$
$$\forall x, y, z \in O, \sigma(x, x) \geq \sigma(y, z) \qquad (maximalité)$$
$$\forall x, y \in O, \sigma(x, y) = \sigma(y, x) \qquad (symétrie)$$

2.2.2 Dissimilarité

Parfois, on utilise plutôt la notion de dissimilarité. Elle représente la fonction inverse de la similarité. La définition suivante est donnée par (Euzenat, et al., 2004).

Définition (Dissimilarité). *Étant donné un ensemble d'entités O, une dissimilarité* $\delta: O \times O \rightarrow R$ *est une fonction à partir d'une paire d'entités vers un nombre réel telle que :*

$$\forall x, y \in O, \delta(x, y) \geq 0 \qquad (positivité)$$
$$\forall x \in O, \delta(x, x) = 0 \qquad (minimalité)$$
$$\forall x, y \in O, \delta(x, y) = \delta(y, x) \qquad (symétrie)$$

2.2.3 Distance

La distance permet de mesurer la dissimilarité entre deux entités. Si la valeur de la fonction de similarité de deux entités est élevée, la distance entre elles est petite et vice-versa (Bach, 2006). Dans (Euzenat, et al., 2004) la définition suivante de la distance est donnée.

Définition (Distance). *Une distance* $\delta: O \times O \rightarrow R$ *est une fonction de dissimilarité satisfaisant la définitivité et l'inégalité triangulaire :*

$$\forall x, y \in O, \delta(x, y) = 0 \;\; SSi \; x = y \qquad (définitivité)$$
$$\forall x, y, z \in O, \delta(x, y) + \delta(y, z) \geq \delta(x, z) \qquad (inégalité\ triangulaire)$$

2.2.4 Normalisation de la similarité

Les mesures précédemment définies sont, très souvent, normalisées pour pouvoir les interpréter d'une manière probabiliste et les combiner dans des formules plus complexes (Bach, 2006). La réduction de chaque valeur à la même échelle en proportion de la taille de l'espace considéré est la manière commune de normaliser (Euzenat, et al., 2004).

Définition ((dis) similarité normalisée). *Une (dis) similarité est dite normalisée si elle s'étend sur l'intervalle unitaire des nombres réels [0 1]. Une version normalisée d'une (dis) similarité* σ *(respectivement,* δ*) est dénotée par* $\overline{\sigma}$ *(respectivement,* $\overline{\delta}$*).*

2.2.5 Représentation de la similarité

Selon les définitions précédentes, la similarité et la dissimilarité sont des fonctions qui font correspondre des paires d'entités à des nombres réels. Une représentation alternative plus commune d'une telle fonction de similarité sur un ensemble fini d'entités est une matrice. La matrice à l'avantage d'être une structure de données finie qui peut être échangée entre les programmes (Euzenat, et al., 2004; Ehrig, 2007).

Exemple. L'application d'une mesure de similarité sur quelques noms de concepts de deux ontologies d'université est donnée par la matrice suivante.

Ontologie 1\ontologie2	People	Student	Faculty	Science	Philosophy	Boxology	Staff	Course	Office
Staff	.56	.65	.33	.64	.12	.11	.63	.22	.13
Professor	.62	.36	.60	.40	.44	.32	.55	.21	.36
Assistant	.40	.44	.58	.62	.46	.33	.43	.32	.22
PhDStudent	.64	.92	.45	.60	.65	.52	.55	.33	.34
Room	.12	.20	.20	.18	.10	.12	.09	.11	.62
Reference	.23	.06	.18	.25	.26	.28	.22	.17	.23
Lecture	.15	.16	.26	.23	.34	.12	.14	.70	.16

Tableau 1. Matrice de similarité

Cette matrice indique par exemple que la valeur de similarité entre « Staff » de l'ontologie 1 et « People » de l'ontologie 2 est égale à 0.56 ; entre « Staff » de l'ontologie1 et « Student » de l'ontologie2 est égale à 0.65, etc.

2.2.6 Niveaux de similarité

Dans (Hakimpour, 2003), quatre niveaux de similarité entre deux entités sont définis : l'égalité (ou l'équivalence), la spécialisation, le recouvrement, et l'incompatibilité (ou la disjointure). Ces niveaux correspondent à l'ensemble de relations θ défini dans la section 2.1 :

- **Incompatibilité.** C'est le niveau ayant le moindre degré de similarité. Deux entités, e_1 et e_2, sont disjointes si elles ne possèdent aucune caractéristique commune, i.e., la conjonction de leurs caractéristiques est vide. Par exemple, « camion » et « employé ».
- **Recouvrement.** Si la conjonction des caractéristiques de deux entités e_1 et e_2 ne peut pas être prouvée comme vide alors elles seront recouvrantes. Cela veut dire qu'il est possible qu'une instance de e_1 soit aussi une instance de e_2.
- **Spécialisation** (respectivement Généralisation). L'entité e_1 est un hyponyme (respectivement hyperonyme) de l'entité e_2 si « e_1 est une sorte de e_2 » (respectivement « e_1 est une super entité de e_2 ») est vrai. Par exemple, "Homme" est un hyponyme de "Personne". De ce fait, toute instance de e_1 (respectivement e_2) est aussi une instance de e_2 (respectivement e_1). La similarité de spécialisation (respectivement généralisation) est une relation partiellement ordonnée.
- **Egalité.** Ce niveau possède le degré de similarité le plus haut. « L'entité e_1 est égale à l'entité e_2 » si e_1 et e_2 possèdent les mêmes caractéristiques. Donc, toute instance de e_1 est aussi une instance de e_2 et vice versa. Par conséquent, si deux entités sont égales, chacune d'entre elles spécialise et généralise l'autre à la fois. Par exemple, les concepts « Véhicule » et « Automobile » sont égaux car elles possèdent les mêmes caractéristiques.

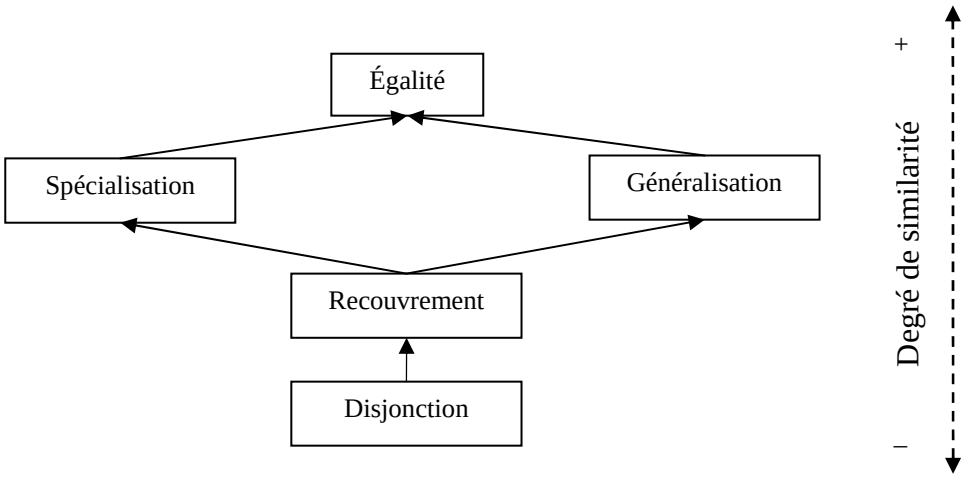


Figure 3. Degrés de similarité entre les entités de différentes ontologies (Hakimpour, 2003)

3 Exemple d'alignement de deux ontologies

Afin d'illustrer la notion d'appariement et d'alignement d'ontologies, nous reproduisons ici l'exemple donné par (Euzenat & Shvaiko, 2007). Il consiste à développer une application pour une université donnée. Cette application vise à fournir une interface unique permettant de gérer deux ontologies différentes. La première ontologie a été développée dans le but de donner une vue globale sur la gestion de l'université, tandis que la deuxième ontologie représente une vue d'un laboratoire de recherche particulier. Les développeurs de cette

application doivent essentiellement apparier ces deux ontologies afin de fournir un alignement permettant une manipulation cohérente des deux ontologies.

La figure suivante montre les deux ontologies à apparier ainsi que les différentes correspondances entre leurs entités.

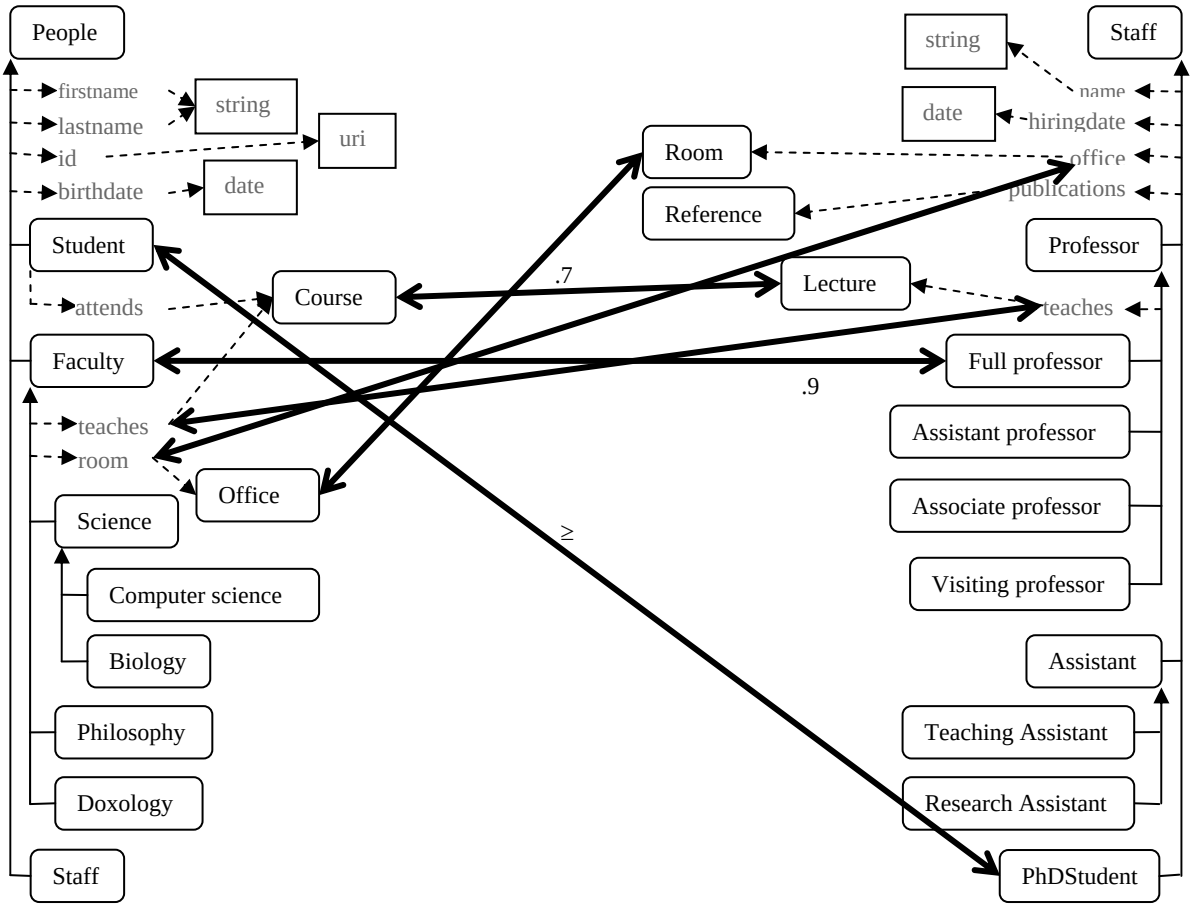


Figure 4. Exemple d'alignement entre deux ontologies (Euzenat & Shvaiko, 2007). Les liens pleins représentent les différentes correspondances entre les entités. La relation et le degré de confiance par défaut sont respectivement l'équivalence (=) et le 1 ; autrement, ils sont mentionnés.

Cet alignement peut être exprimé par les correspondances suivantes :

Student ≥ PhDStudent	Course = $_{0.7}$ Lecture
Faculty = $_{0.9}$ Full professor	teaches = teaches
room = office	Office = Room

4 Typologie des alignements

Dans le domaine de l'appariement des ontologies, quatre types d'alignement ont été distingués (Euzenat & Shvaiko, 2007) :

- **L'alignement simple.** Il se présente comme des ensembles de paires d'entités de deux ontologies, tel que l'alignement présenté dans l'exemple de la figure 4 ;
- **Le multi-alignement.** Cet alignement contient des correspondances reliant plus de deux ontologies;
- **L'alignement complexe.** Défini entre deux ontologies, l'alignement complexe exprime des correspondances entre plus de deux entités. Par exemple, l'alignement de la figure 4 peut être rendu complexe par l'ajout d'une correspondance entre les propriétés « firstname » et « lastname » de la première ontologie et la propriété « name » de la deuxième ontologie. Cependant, l'utilisation des relations binaires pour exprimer une telle correspondance est inappropriée. En général, les entités sont regroupées par des opérateurs tels que la concaténation, les opérateurs arithmétiques ou les connecteurs logiques. Ainsi, la correspondance précédente est exprimée par : « firstname∧lastname=name » ;
- **L'alignement multiple.** La multiplicité de l'alignement concerne le nombre de correspondances reliant les entités. Dans l'exemple d'exécution de la figure 4, l'alignement n'est pas multiple puisque sa multiplicité est ? :? , i.e., les entités des deux ontologies sont impliquées dans au plus une correspondance. Cependant, si on ajoute la correspondance Professor ≥ Faculty, l'alignement devient ? :*. Si on considère que l'alignement relie toute entité de la deuxième ontologie avec une entité de la première ontologie, cet alignement devient ?:+. Les quatre figures suivantes montrent quelques configurations possibles pour l'alignement de deux ontologies composée de trois concepts chacune.

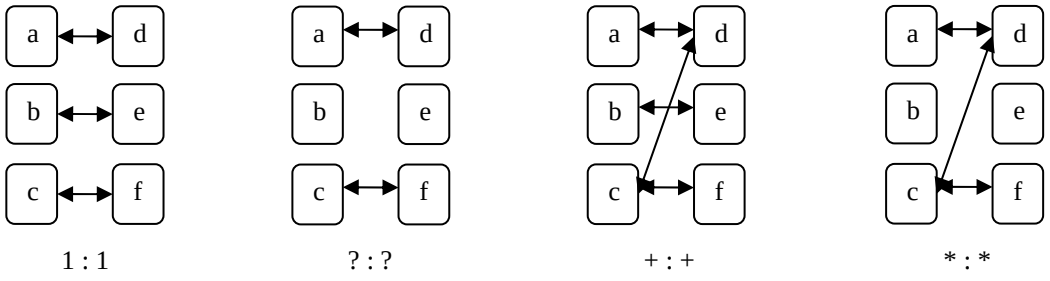


Figure 5. Quelques configurations possibles pour l'alignement de deux ontologies (Euzenat & Shvaiko, 2007)

5 Applications de l'appariement des ontologies

L'appariement des ontologies est supposé non seulement être une théorie de haut niveau, mais aussi destiné à des scénaris pratiques (Ehrig, 2007). Pour cette raison, cette section

examine, à titre non exhaustif, quelques cas d'utilisation concrets d'appariement des ontologies.

5.1 Evolution des ontologies

Quand une ontologie est développée par plusieurs ontologistes d'une manière collaborative et qu'elle devient de plus en plus grande et complexe au niveau de la structure et de la représentation, il est difficile pour un ontologiste de comprendre toutes les parties de l'ontologie (Luong, 2007). Il est aussi difficile pour lui de connaître les parties affectées lorsque des changements sont effectués sur cette ontologie. Par conséquent, nous avons besoin d'un mécanisme qui surveille comment les changements ontologiques ont été effectués, i.e., un mécanisme qui permet de gérer et de maintenir les différentes versions de l'ontologie. Ce mécanisme est assuré à travers l'opération d'appariement (cf. figure 6) dont le rôle est de découvrir les différences entre deux versions de l'ontologie. On y retrouve les entités ontologiques qui ont été ajoutées, supprimées ou renommées.

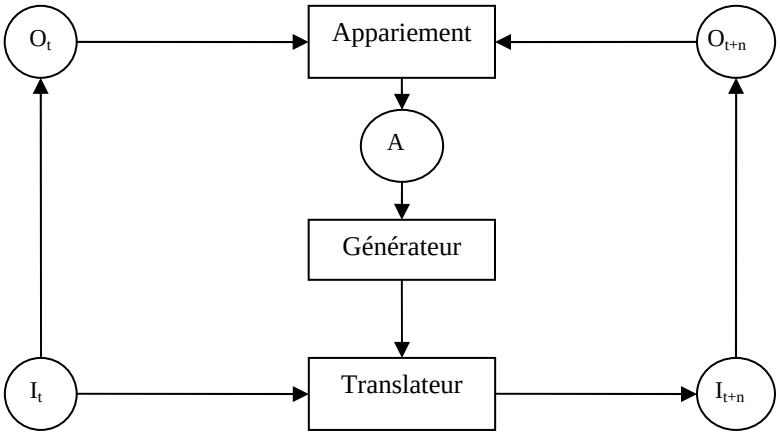
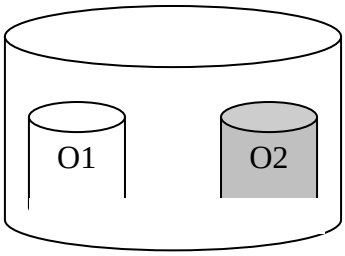


Figure 6. Scénario d'évolution d'une ontologie.

Dans le scénario de la figure 6, il est utile de: (i) trouver les correspondances entre l'ancienne version O_t et la nouvelle version O_{t+n} de l'ontologie, (ii) de générer une transformation par l'utilisation de ces correspondances et (iii) de transformer les instances de données sous-jacentes I_t vers I_{t+n}.

5.2 Fusion d'ontologies

En vue de construire une nouvelle ontologie, les approches adoptant la fusion d'ontologies vise la reprise de conceptualisations spécifiées dans des ontologies déjà existantes liées au domaine à conceptualiser. Pour cela, elles intègrent plusieurs ontologies (Mellal, 2007). Ceci nécessite très souvent une étape d'appariement, qui identifie les concepts et les relations que ces ontologies ont en commun.



ONTOLOGIE O3
=
Une partie de O1 + une partie de O2

Figure 7. Fusion de deux ontologies (Mhiri & Despres, 2005)

5.3 Intégration de schémas

De nos jours, le développement d'une nouvelle application de traitement de données fait le plus souvent appel à des données déjà mémorisées dans plusieurs bases de données indépendantes (Parent & Spaccapietra, 1996). C'est le cas, notamment, des grandes entreprises où l'usage largement répandu de l'informatique se traduit par un développement indépendant de plusieurs bases de données pour les applications spécifiques de chaque service ou filiale. Or, les structures des entreprises évoluent, surtout dans l'environnement économique actuel. Les frontières entre services ou entre sociétés sont alors susceptibles de bouger, créant de nouveaux centres d'intérêts, demandant de nouvelles applications qui devront être construites avec des données prises ici et là, plus qu'avec de nouvelles données spécifiques. Ainsi, le développement de nouveaux systèmes d'informations repose désormais sur leur capacité à réaliser l'interopérabilité entre des bases de données existantes.

Pour ce faire, une solution radicale serait de créer une nouvelle base de données, reprenant toutes les informations provenant des différentes bases de données disponibles. C’est ce que l’on appelle l’intégration des bases de données (Samyn, 2002).

Une autre solution, qui évite de réécrire toutes les applications utilisant ces bases de données, est de créer une base de données virtuelle. Cette base de données n’est qu’une vue, permettant à de nouvelles applications d’accéder à l’ensemble des données disponibles. L’accès à ces données se fait via un logiciel d’intégration qui se charge d’aller récupérer les informations dans les différentes bases de données disponibles. Cette solution est appelée fédération de bases de données.

D’un point de vue conceptuel, ces deux solutions sont identiques (Samyn, 2002). Elles nécessitent toutes les deux la création d’un schéma conceptuel réunissant les points communs des bases de données à unifier, et la traduction de ce schéma vers d’autres niveaux d’abstraction.

Le processus permettant de créer le schéma conceptuel doit comporter les trois étapes suivantes (Parent & Spaccapietra, 1996):

- **Pré-intégration.** C’est une étape dans laquelle les schémas en entrée sont transformés de différentes manières pour les rendre plus homogènes (sur les plans sémantique et syntaxique);
- **Recherche des correspondances.** Elle consiste à apparier les schémas initiaux afin d’identifier les éléments semblables et décrire les liens inter-schémas ;
- **Intégration.** L'étape finale qui unifie les éléments en correspondance en un schéma intégré et produit les règles de traduction associées entre le schéma intégré et les schémas initiaux.

6 Conclusion

Dans ce chapitre, nous avons décrit ce qu’est l’hétérogénéité sémantique et pourquoi elle exige un appariement. Des raisons variées expliquant les disparités qui peuvent apparaître entre les ontologies sont présentées. Nous avons, également défini l’opération d’appariement d’ontologies et son résultat : L’alignement. De plus, certaines applications qui exigent de faire recours à un appariement sont données.

Pour trouver un alignement entre deux ontologies, la base sur laquelle repose tout système d’appariement est le calcul de la similarité entre leurs entités. De ce fait, le chapitre suivant est consacré à la présentation et à la classification des différentes techniques permettant de calculer cette similarité.

CHAPITRE 3 : CLASSIFICATION DES TECHNIQUES D'APPARIEMENT D'ONTOLOGIES ORIENTEE WEB SÉMANTIQUE

Trouver les relations entre les entités exprimées dans des ontologies différentes constitue le but de l'appariement des ontologies. Ces relations sont, généralement, découvertes par le biais de mesures de similarité. Dans la littérature, plusieurs techniques de base permettant de mesurer la similarité entre les entités des ontologies ont été développées. Ces techniques sont appliquées sur un aspect particulier des entités, tel que le nom, les attributs et les relations avec d'autres entités (Bach, 2006; Euzenat & Shvaiko, 2007). Sur chaque aspect, les caractéristiques d'une entité sont comparées avec les caractéristiques correspondantes d'une autre entité. Cela retourne une valeur de similarité dite individuelle.

Cependant, les langages de représentation des ontologies n'ont pas tous la même puissance d'expression. Par exemple, contrairement au langage OWL, le langage RDF(S) ne permet pas d'exprimer la cardinalité d'une relation. Par conséquent, le choix des techniques de base appropriées dépendra du langage de représentation des ontologies utilisé. La question est donc de savoir comment guider le choix d'une technique pour un aspect donné.

Afin de répondre à cette question, ce chapitre propose une classification des techniques de base qui sert à guider l'utilisateur dans le choix des techniques appropriées pour comparer l'ensemble des caractéristiques supporté par le langage de représentation des ontologies utilisé.

Pour cela, le chapitre est organisé comme suit. La section 1 présente les différentes classifications des techniques d'appariement de base recensées dans la littérature. Dans la section 2, nous exposons les principes de notre classification. Les sections 3 et 4 sont consacrées à la présentation de notre approche. La section 5 conclut ce chapitre.

1 Travaux traitant de la classification des techniques d'appariement de base

Dans la littérature, plusieurs classifications des techniques de base ont été proposées (Rahm & Bernstein, 2001; Kalfoglou & Schorlemmer, 2003; Wache, et al., 2001; Euzenat, et al., 2004; Castano, Ferrara, Hess, & Montanelli, 2007; Ehrig, 2007). Ces travaux abordent le problème de l'appariement à partir de plusieurs domaines tels que les systèmes d'information, les bases de données et le Web sémantique. L'objet de cette section est de présenter les classifications relatives au Web sémantique.

1.1 Classification 1 (Euzenat & Shvaiko, 2007)

Cette classification (cf. figure 1) se basent principalement sur celle de (Rahm & Bernstein, 2001) en la complétant par l'introduction de nouvelles classes. Quatre directives sont proposées. Il s'agit de :

- **L'exhaustivité.** L'extension des classes appartenant à une catégorie particulière doit couvrir ses extensions, c.à.d. leur agrégation doit donner l'extension complète de la catégorie en question.
- **La disjointure.** Les sous-catégories composant une catégorie doivent être disjointes.
- **L'homogénéité.** Les critères utilisés pour diviser les catégories sont de même nature.
- **La saturation.** La prise en compte de nouvelles techniques ne doit pas exiger la définition de nouvelles classes.

Afin d'atteindre la couche des techniques de base, la classification globale de la figure 1 peut être lue aussi bien d'une façon descendante que d'une façon ascendante. La lecture descendante considère la manière avec laquelle les techniques interprètent l'information en entrée. Tandis que la lecture ascendante est concernée par le type d'objets manipulés par les techniques de base.

1.1.1 Lecture descendante

La classification adoptée par la couche supérieure est divisée en deux niveaux :

- **La granularité.** La granularité avec laquelle les techniques d'appariement considèrent les entités peut être selon deux degrés:

  - *Degré des éléments.* Les techniques basées sur les éléments considèrent chaque entité isolément, en ignorant ses relations avec d'autres entités.
  - *Degré des structures.* Les techniques basées sur les structures calculent les correspondances en analysant comment les entités apparaissent ensemble dans une structure.

- **L'interprétation des entrées.** Les techniques de base interprètent les entrées selon trois manières différentes :

  - *Interprétation syntaxique.* Les entrées sont interprétées uniquement en fonction de leur structure syntaxique.
  - *Interprétation en utilisant les ressources externes.* Ces techniques exploitent des ressources (externes) auxiliaires telles que les thesaurus[1] de connaissances communes ou de domaine spécifique pour obtenir la signification des termes utilisés dans les ontologies.
  - *Interprétation sémantique.* La caractéristique des techniques sémantiques est qu'elles emploient de la sémantique formelle, telle que l'utilisation de la logique propositionnelle, pour interpréter les entrées.

1.1.2 Lecture ascendante

La classification de la couche inférieure de la figure 1 est divisée en deux niveaux :

- Le premier niveau est classé selon le type de données sur lequel les techniques opèrent. C'est soit des chaînes de caractères (terminologiques), des structures (structurelles), des instances (extensionnelles) ou des modèles (sémantiques).
- Dans le deuxième niveau, on retrouve les techniques terminologiques qui sont décomposées selon la manière avec laquelle elles interprètent les mots, à savoir comme des chaînes de caractères ou comme des objets linguistiques. On y retrouve également les techniques structurelles qui sont divisées en deux types : Celles considérant la structure interne des entités (par exemple, les attributs et leurs types) et celles prenant en compte la relation de l'entité avec d'autres entités.

[1] Les thesaurus sont des listes de termes avec leurs significations prenant en considération la sémantique ressortant des définitions des relations entre les termes (comme la relation de synonymie)

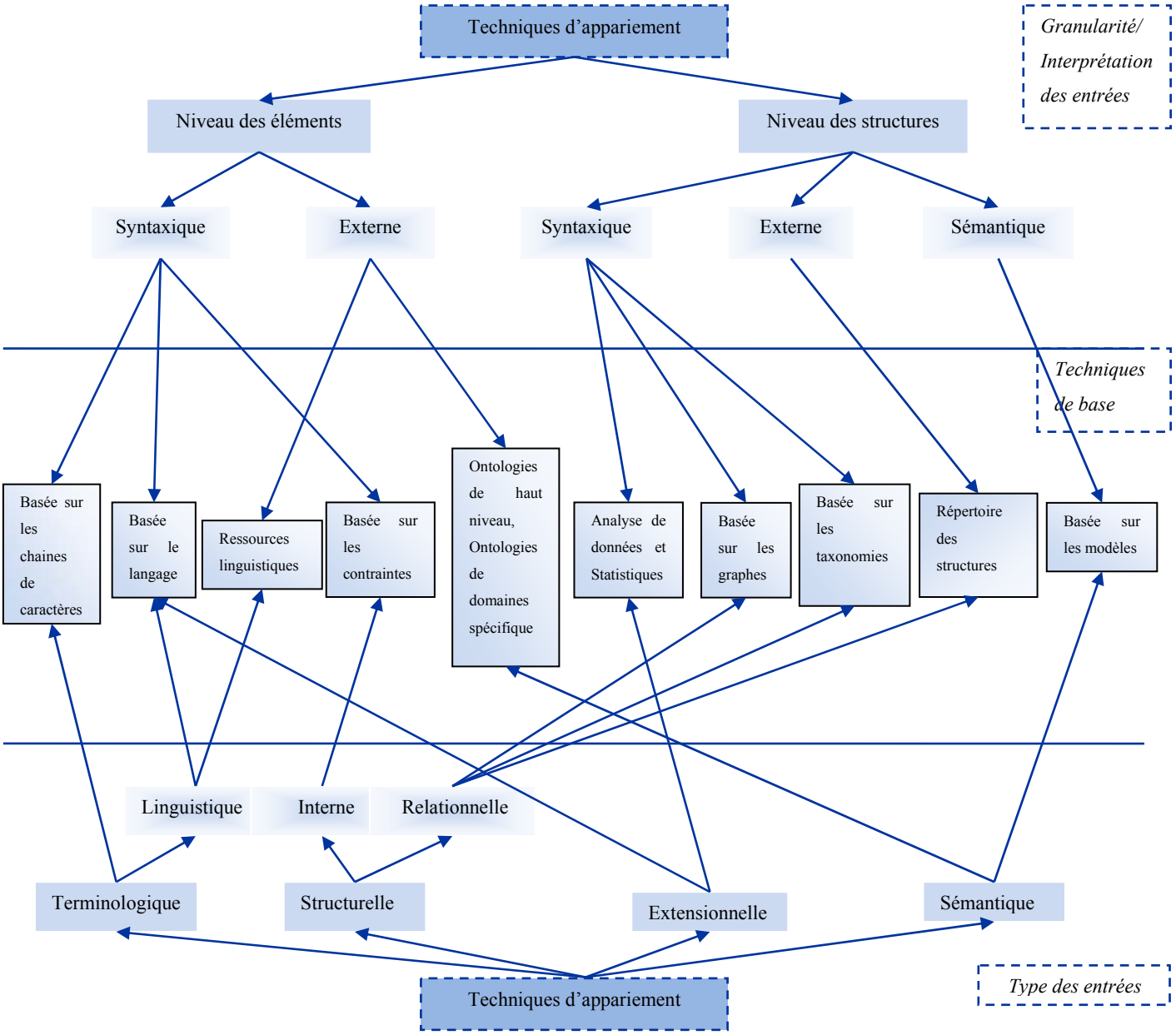


Figure 1. Classification des techniques d'appariement selon (Euzenat & Shvaiko, 2007)

1.1.3 Discussion

La classification présentée ci-dessus couvre toutes les techniques de base existantes dans la littérature. Elle satisfait parfaitement la directive de saturation. Cependant, ceci risque d'être très prochainement rompu puisque le développement de techniques pour apparier des règles ne peut trouver place dans cette classification et exigera donc l'introduction d'une nouvelle classe de techniques basée sur les méthodes de réécriture telles que les raisonnements équationnels classiques, ou les démonstrateurs par induction basés sur la procédure de complétion de Knuth et Bendix (Knuth & Bendix, 1970).

De plus, dans la couche des techniques de base, les auteurs distinguent les techniques utilisant des ressources linguistiques de celles utilisant des ontologies de support. Toutefois, ces deux techniques possèdent exactement le même principe de fonctionnement. Elles peuvent donc être regroupées dans une seule classe qui est celle des techniques basées sur la structure d'une ressource externe. De ce fait, la directive de disjointure de classes est rompue dans cette classification.

1.2 Classification 2 (Castano, Ferrara, Hess, & Montanelli, 2007)

Cette classification comprend deux catégories principales, à savoir : les techniques linguistiques et les techniques contextuelles (cf. figure 2).

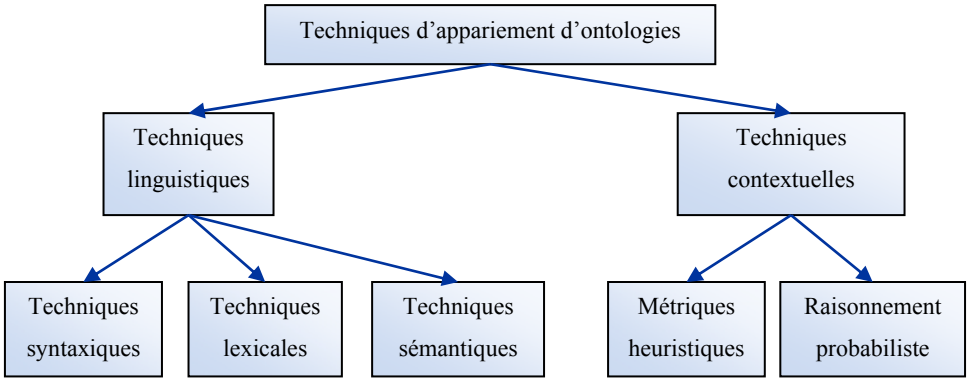


Figure 2. Classification des différentes techniques d'appariement d'ontologies selon (Castano, Ferrara, Hess, & Montanelli, 2007)

1.2.1 Techniques linguistiques

Selon les auteurs, les techniques linguistiques sont celles adoptées pour évaluer la similarité entre les noms et les étiquettes des concepts ainsi qu'entre les contextes de concepts en termes d'attributs et de relations sémantiques. Elles sont divisées en trois classes, à savoir :

- **Les techniques syntaxiques.** Elles sont basées sur l'idée que les éléments ontologiques similaires sont dénotés par des noms ou des étiquettes similaires. D'un point de vue syntaxique, les noms et les étiquettes des éléments ontologiques sont considérés comme des chaînes de caractères.
- **Les techniques lexicales.** Elles considèrent les chaînes de caractères, utilisées pour nommer et étiqueter les éléments des ontologies, comme des mots du langage naturel. Dans ce cas, les techniques de traitement du langage naturel peuvent être adoptées pour analyser ces chaînes, afin d'utiliser ensuite les techniques syntaxiques ou sémantiques.
- **Les techniques sémantiques.** Elles sont basées sur la signification des termes utilisés dans les éléments des ontologies. L'idée est d'apparier les termes aux entrées d'une ressource externe puis de dériver leurs relations terminologiques.

1.2.2 Techniques contextuelles

Les techniques contextuelles sont adoptées pour évaluer la similarité entre les concepts en analysant leurs contextes. Les auteurs considèrent que chaque contexte est représenté par une structure différente dans la description du concept, à la fois en termes d'attributs et en termes des relations sémantiques avec d'autres concepts. L'approche générale est de commencer par la similarité entre les éléments des deux contextes et d'évaluer le nombre d'éléments qui sont semblables. Les auteurs présentent trois catégories de techniques différentes pour l'évaluation de la similarité contextuelle :

- Les techniques basées sur une métrique heuristique. Ces techniques évaluent la similarité au moyen d'analyse de la représentation graphique ou taxonomique du contexte ;
- Les techniques basées sur le raisonnement probabiliste. Elles dépendent de la probabilité que deux concepts auront les mêmes instances ;
- Les techniques basées sur le raisonnement automatique. Elles exploitent la sémantique formelle associée à chaque élément du contexte où un concept est vu comme un ensemble d'attributs logiques.

1.2.3 Discussion

Une première critique peut être faite sur la disjointure non vérifiée entre les définitions des deux classes principales de cette classification. En effet, les techniques linguistiques prennent en compte les contextes de concepts en termes d'attributs et de relations sémantiques, ce qui est également couvert par les techniques contextuelles. De plus, cette classification ne couvre pas toutes les techniques de base existantes. Par exemple, pour déduire la similarité entre les instances, elle ne considère que les techniques basées sur un raisonnement probabiliste malgré l'existence d'autres approches telles que celles basées sur les techniques statistiques (Cox & Cox, 1994).

1.3 Classification 3 (Ehrig, 2007)

Dans ce cas, une classification en trois couches est adoptée (cf. Figure 3) : Données, Ontologie, et Contexte. À travers toutes les couches, l'auteur désigne un champ orthogonal supplémentaire qui représente la connaissance du domaine.

1.3.1 La couche Données

Dans la première couche, la comparaison des entités se fait en considérant seulement les valeurs de données des types de données simples ou complexes, tel que les nombres entiers et les chaînes de caractères. Pour cela, l'auteur considère la distance d'édition pour les chaînes de caractères et la similarité basée sur la distance pour les nombres entiers.

1.3.2 La couche Ontologie

La couche Ontologie est divisée en quatre niveaux, à savoir :

- Les réseaux sémantiques. Dans le niveau le plus bas, les ontologies sont vues comme des graphes avec des concepts et des relations. Cependant, l'auteur ne spécifie aucune technique permettant d'exploiter ce niveau ;
- Les logiques de description. A ce niveau, la structure taxonomique, basée sur la relation « is-a », est considérée. Selon l'auteur, les techniques utilisées ici sont celles basées sur le nombre d'arcs séparant deux concepts ;
- Les restrictions. Pour les restrictions, l'auteur ne spécifie aucune technique qui peut les exploiter ;
- Les règles. L'auteur affirme que les règles peuvent aussi devenir intéressantes pour des considérations de similarité. Toutefois, il n'y a pas eu de recherche suffisante dans cette direction.

Il est à noter que les mesures de similarité de la couche Ontologie peuvent inclure les mesures de similarité de la couche Données.

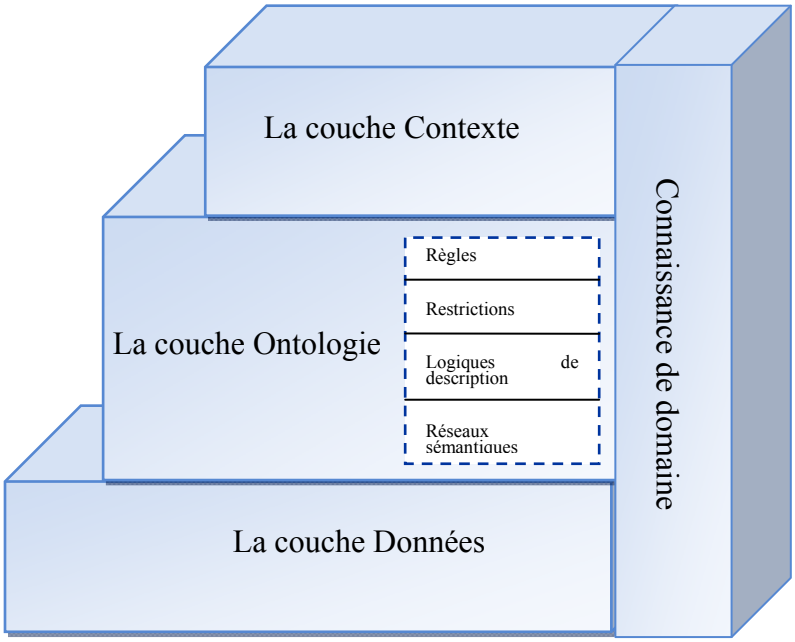


Figure 3. Modèle en couche selon (Ehrig, 2007)

1.3.3 La couche Contexte

Cette couche est concernée par l'usage concret des entités dans le contexte d'une application donnée. De ce fait, l'appariement est effectué en comparant les usages des entités dans les applications basées sur les ontologies. Le principe est que les entités similaires sont souvent utilisées dans un contexte similaire. Par exemple, dans le portail amazon.com, étant donnée une information sur quelles personnes achètent quels livres, on peut décider si deux

livres sont semblables ou pas dans un contexte donné. L'auteur utilise les deux directions de l'implication dans la découverte de la similarité : si deux entités sont utilisées dans le même contexte alors elles sont similaires et vice versa : si les mêmes entités sont utilisées dans deux contextes alors ces derniers sont similaires.

1.3.4 Connaissances de domaine

La connaissance du domaine peut être située à tous les niveaux de la classification. Elle est donc présentée comme une case verticale à travers toutes les couches.

1.3.5 Discussion

Bien que la classification décrite dans cette section suit les niveaux de complexité sémantique, elle est très générale et ne permet pas d'avoir une classification détaillée de chaque niveau.

2 Principes de la classification proposée des techniques d'appariement

L'objectif de la classification proposée (cf. figure 4) est à la fois de guider le choix, dans le cadre du Web sémantique, de la technique appropriée selon le langage ontologique utilisé et de pallier aux insuffisances constatées dans les différentes classifications existantes (Oulefki & Akli-Astouati, 2008 b).

Pour cela, les techniques de base sont regroupées dans un premier lieu en trois couches, où chaque couche présente les techniques pouvant être utilisées pour apparier les niveaux de complexité sémantique supportés par un langage ontologique donné. De ce fait, puisqu'il existe trois langages ontologiques recommandés dans le Web sémantique, la classification proposée est présentée en trois couches.

A l'intérieur de chaque couche, les techniques de base sont classées selon le niveau de complexité sémantique qu'elles considèrent. Pour chaque niveau de complexité sémantique, une classification plus détaillée est également proposée.

La classification ainsi proposée assure les quatre directives de la classification 1 :

- **L'homogénéité.** Cette directive est satisfaite puisque le critère de classification utilisé au niveau des couches ainsi qu'à l'intérieur de ces couches est le même, i.e., les techniques sont distinguées selon leur applicabilité sur des ontologies écrites dans un langage donné puis selon le niveau de complexité sémantique qu'elles considèrent.
- **La saturation.** La prise en compte de nouvelles techniques n'exige pas la définition de nouvelles classes puisque la classification proposée couvre tous les niveaux de la complexité sémantique.

- **L'exhaustivité.** Puisque le modèle de classification proposé est en couche, donc l'extension d'une classe appartenant à une couche particulière étend cette couche.
- **La disjointure.** Puisque les techniques de base sont différemment conçues pour chaque niveau de la complexité sémantique, une classification basée sur la complexité sémantique comme un critère de classification assure la disjointure.

.

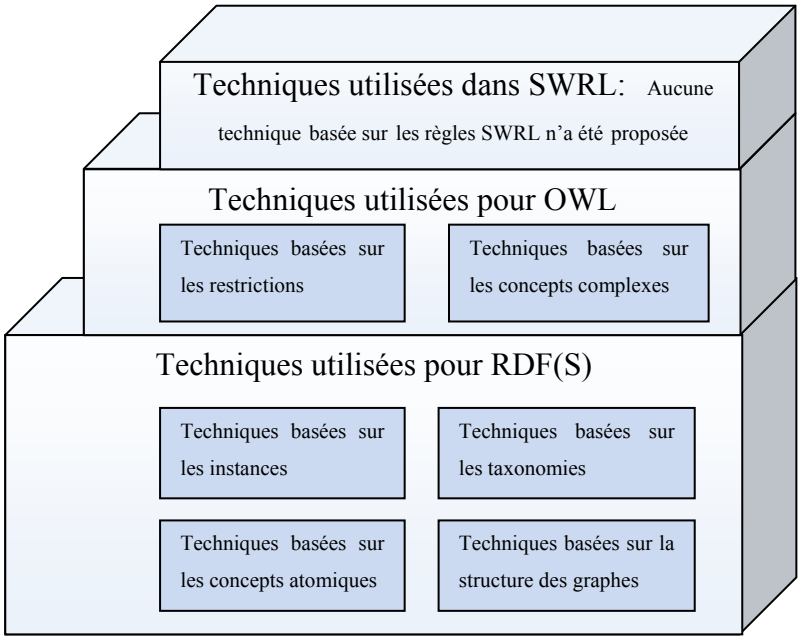


Figure 4. Modèle en couche pour classifier les techniques d'appariement des ontologies

Dans les sections suivantes, nous examinons, pour chaque couche du modèle proposé, les techniques de base qui lui correspondent. Nous présentons également, pour chaque niveau de complexité sémantique, une classification des techniques de base qui l'exploitent.

3 Techniques utilisées pour apparier les caractéristiques supportées par RDF(S)

Les techniques utilisées pour apparier les caractéristiques pouvant être exprimées par des primitives RDF(S) sont celles exploitant les concepts atomiques, les structures de graphes, les instances et les structures taxonomiques.

3.1 Techniques utilisées au niveau des concepts atomiques

Rappelons ici qu'au niveau des concepts atomiques, une entité est dénotée seulement par son nom, son étiquette et ses commentaires. Le nom et l'étiquette sont habituellement un mot, un terme, ou au maximum une expression de quelques mots. Ils sont alors représentés par des chaînes courtes de caractères. Cependant, les commentaires sont généralement une expression, une phrase, ou voir un paragraphe. Ils sont alors représentés par des chaînes

longues de caractères. Les mesures de similarité proposées sont différemment conçues pour chacune de ces deux catégories.

Cependant, avant de comparer les entités au niveau des concepts atomiques, il est souvent nécessaire de procéder à des traitements linguistiques afin d'améliorer les résultats. Ces traitements linguistiques sont, généralement, empruntées au domaine du traitement du langage naturel (TALN). Ils visent à réduire l'hétérogénéité syntaxique produite lorsque les mêmes termes sont exprimés selon différentes manières sans modifier intrinsèquement leurs sens. Ceci est généralement référencé comme la variation de termes dont les principaux types sont (Maynard & Ananiadou, 1999) :

- **Morphologique.** C'est la variation dans la forme d'un mot en se basant sur la même racine ;
- **Syntaxique.** C'est la variation dans la structure grammaticale d'un terme ;
- **Sémantique.** C'est la variation du terme en utilisant, généralement, un hyperonyme ou un hyponyme.

Par ailleurs, (Euzenat, et al., 2004) introduit les deux types de variations de termes suivants :

- **Multi–lingue.** Elle se présente lorsque le variant du terme est exprimé dans une langue différente ;
- **Morphosyntaxique.** C'est la combinaison entre la variation morphologique et la variation syntaxique.

Des sous types de ces types de variations de termes ont été également définis (Euzenat, et al., 2004). Le tableau suivant rapporte un extrait des différents types de variations de termes ainsi que certains exemples.

Type	Sous-type	Exemple
Morphologique	Inflexion Dérivation Flexionnel-Dérivationnel	enzyme activities enzymatic activity enzymatic activities
Syntaxique	Insertion Permutation Coordination	enzyme amidolytic activity activity of enzyme enzyme and bactericidal activity
Morpho-syntaxique	Dérivation-Coordination Inflexion- Permutation	enzymatic and bactericidal activity activity of enzymes
Sémantique		Fermentation
Multilingue	French	activité d'enzyme

Tableau 1. Variantes du terme enzyme activity (extrait de (Euzenat, et al., 2004))

Une fois les techniques linguistiques appliquées, les entités des ontologies peuvent être comparées par des techniques classées selon que les chaînes de caractères soient courtes ou longues (cf. figure 5) :

- Les chaînes courtes de caractères. Elles sont généralement comparées en utilisant des techniques syntaxiques et/ou basées sur la structure d'une ressource externe :
- *Les techniques syntaxiques.* Elles considèrent les entités comme similaires si leurs noms et/ou leurs étiquettes sont syntaxiquement similaires ;
- *Les techniques basées sur la structure d'une ressource externe.* Elles sont fondées sur l'idée d'apparier les termes aux entrées des ressources externes et de dériver leurs relations.
- Les chaînes longues de caractères. Les techniques utilisées pour comparer des chaînes longues de caractères, dites « *techniques basées sur les tokens* », découpent premièrement les chaînes de caractères en plusieurs morceaux appelés « tokens » puis calculent la similarité entre ces ensembles de tokens.

L'intégration des techniques basées sur la structure d'une ressource externe dans le calcul de la similarité des chaînes longues de caractères n'est pas utile puisqu'elle est coûteuse en temps de calcul et donne plus de bruit que d'informations utiles (Bach, 2006). Les commentaires et les descriptions contiennent eux-mêmes relativement assez d'information descriptive pour pouvoir les comparer aux autres en employant seulement les techniques basées sur les tokens.

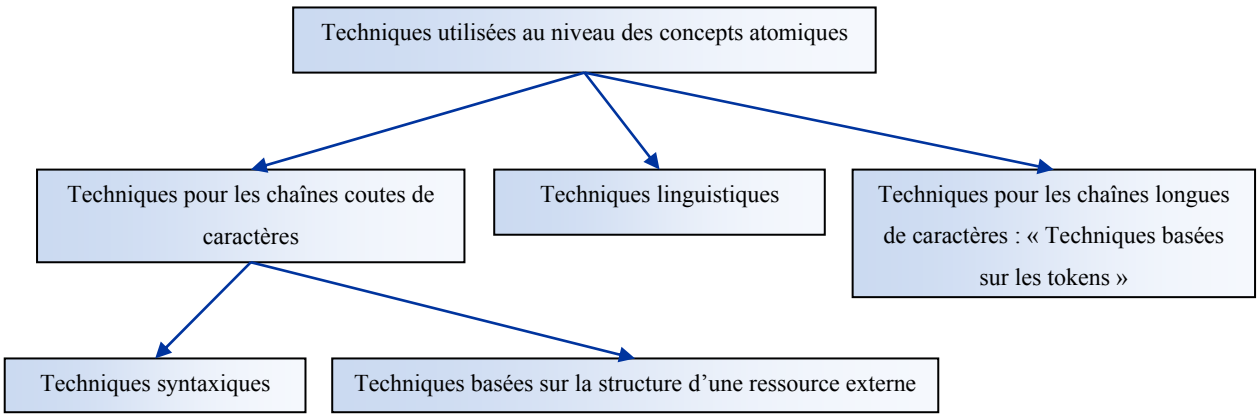


Figure 5. Classification des techniques d'appariement de base utilisées au niveau des concepts atomiques.

3.1.1 Techniques linguistiques

Les méthodes linguistiques fonctionnent avec le principe de chercher la forme canonique d'un terme (lemme) à partir de ses variantes linguistiques (lexème). La similarité entre deux termes est donc décidée en comparant seulement leurs lemmes (Bach, 2006). Par exemple, le résultat de la mesure de similarité de l'égalité de chaînes de caractères (cf. section *3.1.2.1*) appliquée aux deux termes « match » et «matching» sera égal à 0 (c'est-à-dire qu'ils sont

différents), alors que le résultat de la même mesure appliquée aux lemmes de ces deux termes « match » et « match » sera égal à 1, ce qui indique que « match » et « matching » sont similaires (Bach, 2006).

Afin d'obtenir rapidement le lemme d'un mot, des suites logicielles linguistiques ont été développées (Euzenat & Shvaiko, 2007). Elles exécutent généralement les fonctions suivantes (Euzenat, et al., 2004) :

- **Tokenization.** L'idée est de diviser une chaîne de caractères en un ensemble de tokens par un tokenizer. Habituellement, un ensemble de caractères spéciaux est adopté pour diviser la chaîne de caractères, tels qu'une ponctuation ou des espaces. Par exemple, la tokenization du terme « ComestibleFruit » utilise les espaces pour le diviser en ses composants de base, i.e., « Comestible » et « Fruit » ;
- **Lemmatisation.** Les chaînes de caractères sous-jacentes aux tokens sont analysées morphologiquement afin de les réduire en des formes basiques normalisées. L'analyse morphologique comprend deux étapes : une étape de détermination de la catégorie lexicographique d'un token et une étape d'application des règles de normalisation qui invoque la suppression du temps, des genres, des signes de numérotation, des majuscules, des diacritiques, des accents, de la ponctuation et des blancs. Ce processus de découverte de la racine d'un terme est appelée « *lemmatisation* ». Celle-ci exige la connaissance de la grammaire d'une langue et des différentes règles de normalisation. Elle est donc lourde, compliquée et difficile à implémenter (Bach, 2006). Alternativement, les systèmes peuvent utiliser une technique de lemmatisation approximative plus efficace et plus légère appelée stemmatisation ;
- **Stemmatisation.** Un stemmer est un algorithme qui détermine la forme radicale à partir d'une forme infléchie ou dérivée d'un mot donné. Les radicaux (stems) trouvés par les stemmers n'ont pas besoin d'être identiques à la racine morphologique du mot. Il suffit que les mots similaires soient associés à un même radical, même si ce radical n'est pas une racine de mot valide (Bach, 2006). Par exemple, un stemmer pour le français devrait identifier les chaînes de caractères « maintenaient », « maintenait », « maintenant » et « maintenir » comme basées sur la racine "mainten" (Bach, 2006) ;
- **Elimination des mots vides.** Les tokens qui sont reconnus comme des articles, des prépositions, des conjonctions, etc., sont considérés comme des mots non significatifs (vides) et seront donc éliminés. Par exemple, la chaîne « Brand Of Computer » où le token « Of » peut être éliminé devient « Brand Computer ».

Bien que les techniques ainsi présentées améliorent considérablement les résultats de la comparaison des chaines de caractères, elles doivent être utilisées avec vigilance pour différentes raisons (Euzenat & Shvaiko, 2007):

- Elles peuvent aboutir à la perte de certaines informations significatives. Par exemple, « *carbone*-14 » devient « *carbone* », « *là* » (adverbe de lieu) devient «*la*» (article) ;
- Elles peuvent réduire les variations, mais accroître les synonymes. Par exemple, « *livre* » et « *livré* ».

3.1.2 Techniques utilisées pour évaluer la similarité des chaînes courtes de caractères

Ces techniques sont divisées en deux catégories : les techniques syntaxiques, qui sont aussi référencées dans (Bach, 2006) sous l'appellation « *techniques basées sur les chaînes de caractères* », et « *les techniques basées sur la structure d'une ressource complémentaire ou externe*».

3.1.2.1 Les techniques syntaxiques

Ces techniques examinent la structure de la chaîne de caractères en la considérant comme une séquence de lettres. Elles sont classées selon la manière avec laquelle elles considèrent une chaîne de caractères : comme une séquence exacte de lettres ou comme une séquence fictive de lettres. À partir de là, deux types de techniques sont distinguées.

A. Les techniques considérant les chaînes de caractères comme des séquences exactes de lettres

A.1 L'égalité des chaînes de caractères

L'égalité des chaînes de caractères est une mesure de similarité qui renvoie 1 si les chaînes de caractères à comparer sont identiques, sinon elle renvoie 0. La définition formelle suivante est donnée par (Euzenat, et al., 2004).

Définition (Egalité des chaînes de caractères). *L'égalité des chaînes de caractères est une similarité* $\sigma_{egch} : S \times S \rightarrow [0\ 1]$ *telle que* $\forall x, y \in S$, $\sigma_{egch}(x, x)=1$ *et si* $x \neq y$, $\sigma_{egch}(x, y)=0$.

Exemple. Soient trois chaînes de caractères s_1, s_2 et s_3 telles que s_1 = *'véhicule'*, s_2 = *'véhicule' et* s_3 = *'véhicules'*. L'application de l'égalité des chaînes de caractères sur s_1, s_2 et s_3 donne les résultats suivants :

$\sigma_{\text{egch}}(s_1, s_2) = 1$; Puisque s_1 et s_2 sont exactement les mêmes chaînes de caractères.

$\sigma_{\text{egch}}(s_1, s_3) = 0$; Ceci est dû au caractère « s » à la fin de s_3.

$\sigma_{\text{egch}}(s_2, s_3) = 0$; Ceci est dû également au caractère « s » à la fin de s_3.

Ce résultat illustre l'utilité des techniques linguistiques pour améliorer davantage les résultats des comparaisons en normalisant les chaînes de caractères (dans ce cas, en supprimant les multiples).

Comme il peut être remarqué, la mesure de l'égalité des chaînes de caractères ne donne pas d'informations sur comment les chaînes de caractères sont différentes (Euzenat, et al., 2004).

A.2 La distance de Hamming

Pour pallier aux limites de la mesure de l'égalité des chaînes de caractères, la distance de Hamming (Hamming, 1950) propose de calculer le nombre de positions dans lesquelles les deux chaînes de caractères diffèrent. La version normalisée de cette distance est obtenue en la divisant par la longueur de la chaîne la plus longue des deux chaînes de caractères. Elle est définie dans (Euzenat, et al., 2004) comme suit.

Définition (Distance de Hamming). *La distance de Hamming est une dissimilarité δ, tel que $\forall s, t \in S$:*

$$\bar{\delta}_{Hamming}(s,t) = \frac{\left(\sum_{i=1}^{min\,(|s|,|t|)} s[i] \neq t[i]\right) + \big||s| - |t|\big|}{max\,(|s|,|t|)}$$

Puisque la métrique de Hamming est une métrique normalisée, donc la mesure de similarité de Hamming est obtenue par : $\bar{\sigma}_{Hamming} = 1 - \bar{\delta}_{Hamming}$.

Exemple. En reprenant l'exemple précédent, l'application de la distance de Hamming sur s_1, s_2 et s_3 donne les résultats suivants :

$\bar{\delta}_{Hamming}(s_1, s_2) = 0$; Donc, s_1 et s_2 sont exactement les mêmes chaînes de caractères.

$\bar{\delta}_{Hamming}(s_1, s_3) = 0.1$; Ce qui veut dire que s_1 et s_3 diffèrent au niveau d'un seul caractère qui est le « s » à la fin de s_3.

$\bar{\delta}_{Hamming}(s_2, s_3) = 0.1$; Ceci est dû au caractère « s » à la fin de s_3.

A.3 Test de sous-chaînes

Le test de sous-chaînes est une variante obtenue à partir de l'égalité des chaînes de caractères. Il est basé sur l'idée que les chaînes de caractères sont similaires si l'une est sous-chaîne de l'autre (Euzenat, et al., 2004) :

Définition (Test de sous-chaînes). *Le test de sous-chaînes est une similarité $\sigma_{tsc} : S \times S \rightarrow [0\ 1]$ telle que $\forall x, y \in S$, s'il existe p, s $\in S$ où $x = p + y + s$ ou $y = p + x + s$, alors $\sigma_{tsc}(x, y) = 1$, sinon $\sigma_{tsc}(x, y) = 0$.*

En se basant sur le modèle de sous-chaînes, plusieurs variations peuvent être définies, à savoir : la similarité de sous-chaînes, le test de préfixe, le test de suffixe, la similarité de *n*-gramme et la similarité de Jaccard.

La similarité de sous-chaînes mesure le ratio de la partie commune entre deux chaînes de caractères. Elle est définie dans (Euzenat, et al., 2004) comme suit.

Définition (Similarité de sous-chaînes). *La similarité de sous-chaînes est une similarité $\sigma_{sous\text{-}ch} : S \times S \rightarrow [0\ 1]$ telle que $\forall x, y \in S$, et t la plus longue sous-chaîne commune de x et y :*

$$\sigma_{sous-ch}(x, y) = \frac{2 \times |t|}{|x| + |y|}$$

Exemple. La similarité de sous-chaînes entre « Student » et « PhDStudent » renvoie 7 /10=0.7. Tandis qu'entre « Staff » et « Staff », elle donne 5/5=1.

Le test de préfixe (de suffixe, respectivement) prend en entrée deux chaînes de caractères et vérifie si la première commence (se termine, respectivement) par la deuxième. Ceci peut-être utile pour les chaînes de caractères dénotant un concept plus général qu'un autre puisque dans certains langages, les termes composés dénotent souvent une spécialisation de la signification du composant le plus à droite du terme. Par exemple, « PhDStudent » est considéré plus spécifique que « Student ».

La similarité de *n*-gramme, quant à elle, calcule le nombre de *n*-grammes, i.e., les séquences de *n* caractères, communs entre deux chaînes de caractères. Par exemple, les trigrammes de la chaîne de caractères « *véhicule* » sont : *véh, éhi, hic, icu, ule*. La définition suivante est fournie par (Euzenat, et al., 2004).

Définition (Similarité de n-gramme). *Soit ngram (s, n) l'ensemble des chaînes de s de longueur n. La similarité de n-gramme est une similarité $\sigma_{ngram} : S \times S \rightarrow R$ tels que : σ_{ngram} (s, t) = $|$ ngram (s, n) $\cap$ ngram (t, n) $|$*

La version normalisée de cette fonction est comme suit :

$$\overline{\sigma}_{ngram}(s, t) = \frac{|\, ngram\,(s, n) \cap ngram\,(t, n)\,|}{min(|s|, |t|) - n + 1}$$

Cette fonction est parfaitement efficace quand seulement quelques caractères sont manquants (Euzenat, et al., 2004).

Exemple. La similarité de 3-gramme entre « Student » et « PhDStudent » est 4/5=0.8. Tandis que entre « Staff » et « Staff », elle donne 3/3=1.

La similarité de Jaccard est une variante de la similarité de *n*-gramme où *n*=1. La définition suivante est donnée par (Bach, 2006).

Définition (Similarité de Jaccard). *Soient s et t deux chaînes de caractères. Soient S et T les ensembles des caractères de s et t respectivement. La similarité de Jaccard est une similarité $\sigma_{Jaccard} : S \times S \rightarrow R$ telle que : $\sigma_{Jaccard}$ (s, t) = $|\, S \cap T \,|$.*

La version normalisée de cette fonction est comme suit :

$$\overline{\sigma}_{Jaccard}\ (s,t)\ = \frac{|\,S \cap T\,|}{|\,S \cup T\,|}$$

Exemple. L'application de la similarité de Jaccard sur les chaînes de caractères « Student » et « PhDStudent » renvoie 7/10=0.7.

B. Les techniques considérant les chaînes de caractères comme des séquences fictives de lettres

Elles évaluent davantage comment une chaîne de caractères peut-être une version fictive d'une autre chaîne de caractères (Euzenat & Shvaiko, 2007). Les mesures classées dans cette catégorie sont : la distance d'édition, la mesure de Jaro et celle de Jaro-Winkler.

B.1 La distance d'édition

La distance d'édition est définie par le coût minimal d'opérations devant être appliquées pour convertir une chaîne de caractères *s* en une autre chaîne *t*. Les opérations d'édition incluent l'insertion, la substitution et la suppression d'un caractère telles qu'à chacune d'entre elles est assignée un coût. La définition formelle suivante est donnée par (Bach, 2006).

Définition (Distance d'édition). *Soient un ensemble Op d'opérations d'édition, op∈Op (op : S→ S), et une fonction de coût d'édition w : op → R, tels que pour toutes paires de chaînes de caractères, il existe une séquence d'opérations qui transforme la première en la deuxième (et vice versa), la distance d'édition est une dissimilarité* $\delta_{édition}$ *: S × S → [0 1] où* $\delta_{édition}$ *(s, t) est le coût de la séquence d'opérations la moins coûteuse qui transforme s en t.*

$$\delta_{édition}\,(s,t) = \min_{(op_i)_I;op_n(\ldots op_1(s))=t} \left(\sum_{i\in I} w_{op_i}\right)$$

Les variantes de la distance d'édition diffèrent au niveau des coûts assignés aux opérations d'édition. On peut trouver :

- La distance de Levenshtein (Levenshtein, 1965) où tous les coûts sont égaux à 1.
- La distance de Needleman-Wunsch (Needleman & Wunsch, 1970) avec des grands coûts assignés aux opérations d'insertion et de suppression.

Exemple. La distance d'édition de Levenshtein entre « *chane* » et « *chaîne* » est 1. Ceci peut être lu comme suit : pour transformer « *chane* » en « *chaîne* » une seule opération, qui est l'insertion, est requise ; et pour transformer « *chaîne* » en « *chane* » on a, également, besoin d'une seule opération, qui est dans ce cas la suppression.

B.2 La mesure de Jaro

La mesure de Jaro (Jaro, 1989) est basée sur le nombre et l'ordre des caractères communs entre deux chaînes de caractères. Elle est définie comme suit (Euzenat & Shvaiko, 2007).

Définition (Mesure de Jaro). *Soient s et r deux chaînes de caractères. La mesure de Jaro est une fonction de la similarité* σ*:* $S \times S \rightarrow [0\ 1]$ *tels que :*

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

où m est le nombre de caractères correspondants et t est le nombre de transpositions (voir ci-dessous).

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

$$\left\lfloor \frac{\max(|s|,|r|)}{2} \right\rfloor - 1$$

Le nombre de transpositions est obtenu en comparant l'i-ème caractère correspondant de s avec l'i-ème caractère correspondant de r. Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de transpositions.

Exemple. Soient deux chaînes *s = « MARTHA » et r = « MARHTA », i.e.,* | s | = 6; | r| = 6. *La table de correspondance de s et r est comme suit :*

	M	A	R	T	H	A
M	1	0	0	0	0	0
A	0	1	0	0	0	0
R	0	0	1	0	0	0
H	0	0	0	0	**1**	0
T	0	0	0	**1**	0	0
A	0	0	0	0	0	1

Ce qui donne m = 6 (nombre de 1 dans la table);

Les caractères correspondants sont {M,A,R,T,H,A} pour s et {M,A,R,H,T,A} pour r. En considérant ces ensembles ordonnés, on a donc 2 couples (T/H et H/T) de caractères correspondants différents, soit deux demi-transpositions. D'où $t = \frac{2}{2} = 1$

La similarité de Jaro est donc :

$$\sigma_{Jaro}(s,r) = \frac{1}{3}\left(\frac{6}{6} + \frac{6}{6} + \frac{6-1}{6}\right) = 0.944$$

B.3 La mesure de Jaro-Winkler

La mesure de Jaro-Winkler (Winkler, 1999) est une variante de la mesure de Jaro qui utilise un coefficient de préfixe *P* pour favoriser les chaînes ayant le plus long préfixe commun. Elle est définie comme suit (Bach, 2006).

Définition (Mesure de Jaro-Winkler). *Soient s et r deux chaînes de caractères. La mesure de Jaro-Winkler* $\sigma_{Jaro\text{-}Winkler} : S \times S \rightarrow [0\ 1]$ *est comme suit :*

$$\sigma_{\text{Jaro-Winkler}}(s,r) = \sigma_{Jaro}(s,r) + (P \times Q \times (1 - \sigma_{Jaro}(s,r)))$$

où σ_{Jaro} *est la distance de Jaro entre s et r ; Q est la longueur du préfixe commun ; P est un coefficient qui permet de favoriser les chaînes avec un préfixe commun. Winkler propose que P soit égal à 0.1.*

Exemple. En reprenant l'exemple précédent, la mesure de Jaro-Winkler avec p = 0.1 et Q= 3 est : $\sigma_{\text{Jaro-Winkler}}(s,r) = 0.944 + (0.1 \times 3 \times (1 - 0.944)) = 0.961$

C. Synthèse et discussion

Une étude comparative concernant les techniques syntaxiques a été faite (Cohen, Ravikumar, & Fienberg, 2003). Elle montre les points forts de chaque technique pour une tâche particulière. Chaque mesure de distance ou de similarité s'adapte mieux dans certains domaines d'application. Une synthèse de ce travail a été réalisée par (Bach, 2006) (cf. tableau 2).

Mesure de similarité	Domaine d'application
Distance de Hamming	Utilisée principalement pour les entités numériques ayant des tailles fixes, comme les codes postaux ou les numéros de sécurité sociale.
N-gramme	Bigramme (n = 2) est efficace avec des erreurs typographiques mineures.
Distance d'édition	Peut être appliquée aux entités ayant une longueur variable. Pour atteindre une exactitude raisonnable, les coûts des opérations de modification dépendent de chaque domaine.
Distance de Jaro/Jaro-Winkler	La meilleure performance au niveau des résultats dans plusieurs expériences. Peut être employée dans plusieurs domaines.

Tableau 2. Critères principaux d'utilisation des mesures de la similarité appliquées aux chaînes courtes de caractères (extrait de (Bach, 2006))

Les techniques présentées jusqu'ici sont utiles si les chaînes de caractères utilisées pour dénoter les noms et les étiquettes des entités sont très similaires. Dans certains cas, ces techniques deviennent inapplicables, en particulier, si des synonymes ou des hyponymes sont utilisés. Par conséquent, le processus d'appariement doit utiliser des sources d'information plus fiables, tels que les lexiques et les thesaurus. Les techniques permettant d'exploiter ces ressources sont dites « techniques basées sur la structure d'une ressource externe » et feront l'objet de la sous-section suivante.

3.1.2.2 Les techniques basées sur la structure d'une ressource externe

L'appariement de deux ontologies nécessite qu'elles soient recouvrantes, c.-à-d. qu'elles aient un terrain commun sur lequel la comparaison peut être faite. L'existence de ce terrain commun mène à supposer l'existence d'au moins une ressource, qui est représentée le plus souvent sous la forme d'une ontologie, dite « de background » ou « de support » pouvant le définir (Safar, Reynaud, & Calvier, 2007).

De nombreux travaux récents portent sur l'utilisation de ces connaissances de support. Le schéma général suivi est donné comme suit (Safar, Reynaud, & Calvier, 2007).

A. Schéma général

Soient O_1 et O_2 deux ontologies. L'ensemble des concepts d'O_1 (O_2, respectivement) est dénoté par C_{o1} (C_{o2}, respectivement). Soit R l'ensemble des relations exprimables entre deux concepts appartenant respectivement à l'une et à l'autre des deux ontologies. L'approche générale suivie par les techniques basées sur la structure d'une ressource externe afin d'identifier l'existence d'une correspondance de la forme (X_{o1} relation Y_{o2}) où $X_{o1} \in C_{o1}$, $Y_{o2} \in C_{o2}$, et relation $\in$ R, se décompose en 2 phases : l'ancrage et la dérivation (cf. Figure 6).

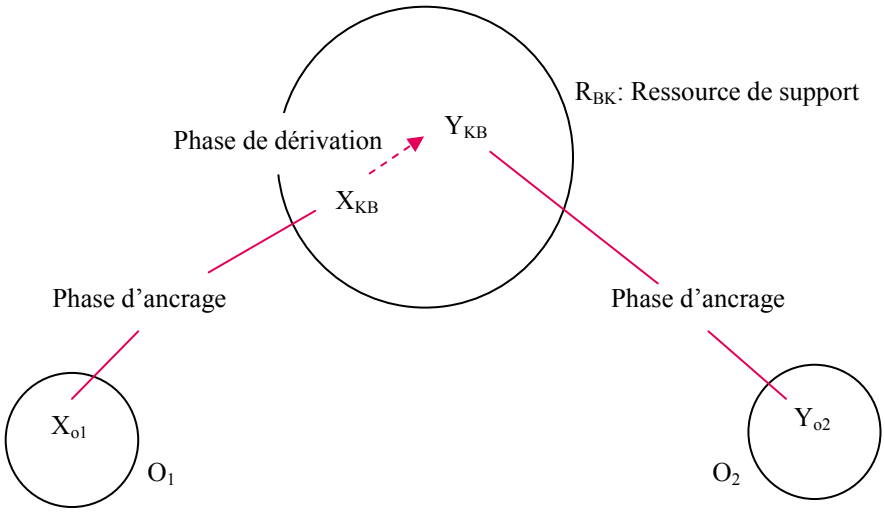


Figure 6. Schéma général suivi par les techniques basées sur la structure d'une ressource externe afin de trouver « X_{o1} relation Y_{o2} » (Safar, Reynaud, & Calvier, 2007)

A.1 Ancrage

Aussi référencé par la contextualisation, l'ancrage consiste à apparier chacun des concepts X_{o1} et Y_{o2}, pris indépendamment l'un de l'autre, avec un ou des concepts de la ressource externe (R_{BK}), c'est-à-dire, à identifier les correspondances de la forme (X_{o1} *relation* X_{BK}) et (Y_{o2} *relation* Y_{BK}) où X_{BK} et $Y_{BK} \in C_{BK}$ (l'ensemble des concept de R_{BK}) et sont appelés des *ancres* ou *points d'ancrage*. Ceci est généralement réalisé par l'utilisation des méthodes syntaxiques.

A.2 Dérivation de relations

C'est l'appariement indirect des ontologies O_1 et O_2 en utilisant les correspondances découvertes durant l'étape d'ancrage. Ici, les travaux existants se différencient selon la stratégie mise en œuvre comme suit :

- L'utilisation d'un ensemble de règles afin d'essayer de dériver des relations entre les différents points d'ancrage X_{BK}, Y_{BK} identifiés. Les relations recherchées appartiennent à l'ensemble $\{\leq, \geq, \equiv\}$ où $X \leq Y$ peut se lire, suivant les cas, *«X is-a Y» ou «X part-of Y».* Des exemples de règles utilisées pour chercher ces relations sont de la forme : Si ($X_{o1} \leq X_{BK}$) et ($X_{BK} \leq Y_{BK}$) et ($Y_{BK} \leq Y_{o2}$) alors ($X_{o1} \leq Y_{o2}$), ou encore Si ($X_{o1} \geq X_{BK}$) et ($X_{BK} \geq Y_{BK}$) et ($Y_{BK} \geq Y_{o2}$) alors ($X_{o1} \geq Y_{o2}$). Dans cette stratégie, les travaux (Aleksovski, Klein, Kate, & Harmelen, 2006) supposent que la recherche de dérivation peut s'effectuer sur une seule ressource de support, préalablement identifiée et qui couvre a priori tous les concepts des ontologies à apparier. A l'inverse, d'autres travaux (Sabou, D'Aquin, & Motta, 2006) font l'hypothèse opposée : la recherche de dérivation ne peut s'effectuer qu'au sein de multiples ressources de support sélectionnées dynamiquement. Une étude comparative de ces travaux peut être trouvée dans (Safar, Reynaud, & Calvier, 2007) ;
- L'utilisation d'une mesure de similarité entre les nœuds d'un même graphe, pour identifier pour chaque ancre X_{BK} d'un concept de l'ontologie O_1, l'ancre Y_{BK} du concept de l'ontologie O_2 qui lui est le plus similaire (Safar, Reynaud, & Calvier, 2007).

Dans la littérature, les techniques basées sur une ressource de support utilisent généralement WordNet (Euzenat & Shvaiko, 2007) en appliquant des mesures de similarité sur sa structure graphique. Par conséquent, dans la suite de cette section, nous présentons ce type particulier de techniques après avoir donné une définition de WordNet.

B. Utilisation de la ressource WordNet

B.1 Définition de WordNet

WordNet (Miller, 1995) est une base de données électronique lexicale pour l'anglais. Sa conception est conforme aux théories courantes de la psycholinguistique portant sur l'organisation de la mémoire lexicale humaine. Il se démarque un peu du concept de

dictionnaire en ce sens que ce qui importe n'est pas de donner une définition du mot mais de situer celui-ci dans un réseau (hiérarchique) sémantique ou lexical de mots.

WordNet peut donc être considéré comme un réseau sémantique où les nœuds représentent les concepts du monde réel. Chaque nœud est composé d'un ensemble de synonymes qui représentent le même concept. Cet ensemble s'appelle « *synset* ». Les synsets sont reliés par des arcs qui décrivent les relations sémantiques entre les différents concepts : hyponymie, hyperonymie, méronymie, métonymie, synonymie, antonymie, implication et causalité (Zargayouna & Salotti, 2004).

WordNet offre aussi des descriptions textuelles des concepts, appelées « *gloses* », contenant des définitions et des exemples.

Un système similaire, inspiré de la structure de WordNet, nommé EuroWordNet, est construit pour des langues européennes telles que le hollandais, l'italien, l'espagnol, l'allemand, le français, le tchèque et l'estonien.

Dans (Euzenat & Shvaiko, 2007), WordNet est dénoté comme une ressource de synonymes partiellement ordonnée en donnant la définition suivante.

Définition (Ressource de synonymes partiellement ordonnée). *Une ressource de synonymes partiellement ordonnée Σ sur un ensemble de mots W, est un triplet $\langle E, \leq, \lambda \rangle$, tel que E est un ensemble de synsets, $\leq$ est la relation d'hyperonymie entre les synsets et λ est une fonction à partir des synsets à leurs gloses. Pour un terme t, $\Sigma(t)$ dénote l'ensemble des synsets associé à t.*

Exemple. Nous reproduisons ici l'entrée WordNet pour le mot « Student » où chaque sens est numéroté.

The noun student has 2 senses

Ensemble de synonymes *Définition (Glose)*

2 sens

1. student, pupil, educate -- (a learner who is enrolled in an educational institution)

2. scholar, scholarly person, bookman, student -- (a learned person (especially in the humanities); someone who by long study has gained mastery in one or more disciplines)

B.2 Méthodes exploitant WordNet

Afin de mesurer la similarité entre les deux points d'ancrage en utilisant WordNet, trois catégories de méthodes sont distinguées dans (Euzenat & Shvaiko, 2007) :

- Celles qui considèrent que deux points d'ancrage sont similaires s'ils appartiennent au même synset ;
- Celles qui prennent avantage de la structure d'hyperonymie en mesurant les distances entre les synsets correspondants aux deux points d'ancrage ;
- Celles qui prennent avantage des gloses afin d'évaluer la distance entre les synsets associés aux deux points d'ancrage.

Dans chacune de ces trois catégories, plusieurs mesures de similarité ont été développées. Nous les présentons comme suit.

B.2.1 Méthodes basées sur l'hypothèse que les deux points d'ancrage doivent appartenir au même synset

Afin de calculer la similarité entre deux points d'ancrage sous l'hypothèse qu'ils ne sont similaires que s'ils appartiennent au même synset, deux principales mesures ont été recensées (Euzenat, et al., 2004): la similarité de synonyme et la similarité de co-synonymie.

La similarité de synonyme

Cette mesure calcule le rapport de synonymie entre deux points d'ancrage. Elle est définie dans (Euzenat & Shvaiko, 2007) comme suit:

Définition (Similarité de synonyme). *Étant donnés deux termes s et t et une ressource de synonyme Σ, la synonymie est une similarité* $\sigma_{syn}: S \times S \rightarrow [0\ \ 1]$ *telle que :*

$$\sigma_{syn}(s,t) = \begin{cases} 1 & si\ \Sigma(s) \cap \Sigma(t) \neq \emptyset \\ 0 & sinon \end{cases}$$

Exemple. La similarité de synonymie entre les concepts « Student » et « Pupil » renvoie 1 puisqu'il existe un synset contenant ces deux termes à la fois. Tandis qu'entre « Student » et « People », cette similarité donne 0 car il n'existe aucun synset qui comporte ces deux termes à la fois.

La similarité de co-synonyme

Une autre mesure calculant la similarité de co–synonymie a été proposée. La définition donnée par (Euzenat J. , et al., 2007) est comme suit.

Définition (Similarité de co–synonymie). *Soient deux termes s et t. Étant donnée une ressource de synonyme Σ, la co–synonymie est une similarité* $\sigma: S \times S \rightarrow [0\ \ 1]$ *telle que :*

$$\sigma(s,t) = \frac{|\Sigma(s) \cap \Sigma(t)|}{|\Sigma(s) \cup \Sigma(t)|}$$

Cependant, cette exploitation stricte des synonymes ne permet pas de savoir si deux termes sont proches lorsqu'ils ne sont pas synonymes.

B.2.2 Méthodes prenant en compte que les deux points d'ancrage peuvent appartenir à plusieurs synsets.

Pour remédier aux insuffisances des méthodes basées sur l'hypothèse que les deux points d'ancrage doivent appartenir au même synset pour qu'ils soient similaires, d'autres mesures exploitant plutôt la hiérarchie d'hyponymie/hyperonymie entre les synsets ont été proposées (Euzenat & Shvaiko, 2007). Nous donnons ici quelques exemples de telles mesures.

La mesure edge-count

La mesure edge-count a été proposée par (Rada, Mili, Bicknell, & Blettner, 1989). Elle compte le nombre d'arcs séparant deux synsets.

Exemple. La mesure edge-count appliquée aux termes « Staff » et « Student » donne une valeur de similarité égale à 0.1.

Cependant, l'utilisation du chemin le plus court ne prend pas en considération la position des concepts dans la ressource externe (ici, WordNet). Intuitivement, deux concepts classés en bas de cette ressource sont très spécifiques et sont donc à un degré de granularité plus fin que deux concepts classés en haut. Ainsi, les synsets « *plant* » et « *animal* » ont la même distance, i.e., 0.333, entre eux que « *egyptian-cat* » et « *siamese-cat* », alors qu'intuitivement les deux derniers sont plus proches. La mesure de Wu-Palmer apporte une réponse à ce problème en comptant la position des concepts par rapport à la racine de WordNet (Zargayouna H. , 2005).

Mesure de similarité de Wu-Palmer

La mesure de Wu-Palmer (Wu & Palmer, 1994) pondère le compte des arcs avec la position des synsets dans la hiérarchie (Euzenat & Shvaiko, 2007). Elle est présentée dans la section 4.1 car la hiérarchie, à cet égard, est similaire à une hiérarchie de concepts. Ainsi, toutes les mesures définies dans la section 4.1 peuvent être utilisées sur le graphe d'hyperonymie de WordNet. Une étude expérimentale de cette mesure sur plusieurs paires d'ontologies peut être trouvée dans (Safar, Reynaud, & Calvier, 2007). Les auteurs ont remarqué qu'elle donnait des résultats pertinents quand les domaines d'application des ontologies à comparer étaient proches et très focalisés. En revanche, l'expérience a montré également que si les domaines d'application étaient plus larges et ne se recoupaient pas, les résultats étaient beaucoup moins satisfaisants. Le problème est dû aux contresens et aux rapprochements erronés qui peuvent en découler.

Mesure de similarité de Resnik

La similarité de Resnik (Resnik, 1995) repose sur la notion du contenu informationnel. Le contenu informatif, dénoté par CI, d'un concept traduit la pertinence d'un concept dans le

corpus en tenant compte de la fréquence de l'apparition des mots auxquels il se réfère ainsi que de la fréquence d'apparition des concepts qu'il généralise (Zargayouna H. , 2005). Plus précisément, il est calculé par la formule suivante :

$$CI(C) = -log(\pi(C))$$

Où $\pi(C)$ est la probabilité de retrouver qu'un mot du corpus soit une instance du concept *C* (un des mots référés par le concept *C* ou par un de ses descendants). Usuellement, π(c) est calculéé par la somme des occurrences du terme divisée par le nombre totale de concepts. Il est tel que le concept le plus spécifique possède la moindre probabilité car plus un concept est général, plus son contenu informatif est faible (Zargayouna H. , 2005).

L'intuition de la notion de contenu informatif est que la similarité entre deux concepts est la portion d'information qu'ils ont en commun qui, dans le cadre de WordNet, peut être déterminée par le concept le plus spécifique qui les subsume (*ppac*) (Zargayouna H. , 2005). De ce fait, Resnik définit la similarité entre deux concepts par la quantité d'information qu'ils partagent. Cette information partagée est évaluée numériquement par le contenu informatif du plus petit ancêtre commun (*ppac*).

Définition (Similarité sémantique de Resnik). *Étant donnés deux termes s et t et une ressource de synonymes partiellement ordonnée $\Sigma = \langle E, \leq, \lambda \rangle$ fournie avec une mesure de probabilité π, la similarité sémantique de Resnik est une similarité $\sigma : S \times S \rightarrow [0\ \ 1]$ telle que :*

$$\sigma(s,t) = CI(ppac(s,t))$$

Ainsi, si deux points d'ancrage sont très éloignés et ont comme *ppac* la racine, leur similarité est égale à 0. L'approche de Resnik essaye d'éviter le problème de granularité, cité au dessus, en diminuant le rôle des arcs dans le calcul de similarité (Zargayouna H. , 2005). En effet, les arcs ne sont utilisés que pour retrouver le *ppac*. Elle est de ce fait un peu sommaire car nous pouvons avoir *ppac*(*hill, shore*) = *ppac*(*shore, natural-elevation*) même si *shore* et *natural-elevation* sont plus proches de leur *ppac* (*geological-formation*) que *shore* et *hill*.

Mesure de Jiang et Conrath

Pour pallier aux limites de la similarité de Resnik, Jiang et Conrath (Jiang & Conrath, 1997) proposent une similarité hybride qui combine le contenu informatif du *ppac* à ceux des concepts. Cette similarité prend, également, en compte le nombre d'arcs en calculant le contenu informatif de chaque concept (Zargayouna H. , 2005).

Définition (Similarité de Jiang et Conrath). *Étant donnés deux termes s et t et une ressource de synonyme partiellement ordonnée $\Sigma = \langle E, \leq, \lambda \rangle$ fournie avec une mesure de*

probabilité π, la similarité de Jiang et Conrath est une similarité $\sigma : S \times S \rightarrow [0\ \ 1]$ *telle que :*

$$\sigma(s,t) = \frac{1}{CI(s) + CI(t) - (2 \times CI(ppac(s,t)))}$$

B.2.3 Méthodes basées sur l'utilisation des gloses

Les gloses fournies par WordNet peuvent être également exploitées pour les considérations de similarité entre deux points d'ancrage. Dans ce cas, chaque entrée de WordNet $s \in \Sigma$ est identifiée par l'ensemble des mots correspondants à $\lambda(s)$. Ainsi, toutes les mesures syntaxiques peuvent être utilisées (Euzenat & Shvaiko, 2007).

Définition (Le recouvrement de gloses). *Étant donné une ressource de synonyme partiellement ordonnée* $\Sigma = \langle R, \leq, \lambda \rangle$*, le recouvrement de gloses fourni entre deux chaînes de caractères s et t est défini par la similarité de Jaccard entre leurs gloses :*

$$\sigma(s,t) = \frac{|\lambda(s) \cap \lambda(t)|}{|\lambda(s) \cup \lambda(t)|}$$

B.3 Discussion

Le travail de Budanitsky et Hirst (Budanitsky & Hirst, 2006) évalue plusieurs mesures de similarité basées sur WordNet, entre autres les mesures présentées dans la section B.2.2 de la section 3.1.2.2. Les expérimentations menées ont dévoilé que la mesure de Jiang & Conrath donne les meilleurs résultats.

Cependant, dans (Zargayouna H. , 2005), il est remarqué que les incohérences présentes dans WordNet risquent de diminuer les performances même avec les mesures les plus pertinentes. Pour illustrer ces incohérences, l'auteur vérifie la place des synsets *européen, asiatique, africain* et *américain*, qui sont au même niveau de granularité (habitants d'un continent). Ceci a montré que *européen*, *asiatique* et *américain* sont classés en dessous d'*habitant* par contre *africain* est classé en dessous de *personne*. Rien dans la glose d'*africain* n'explique sa position en dessous de personne au lieu d'habitant.

Par ailleurs, les ressources externes linguistiques, telles que les lexiques et les thésaurus, permettent l'interprétation des termes utilisés dans les expressions des ontologies. De ce fait, elles fournissent l'avantage d'ouvrir de nouveaux appariements entre les entités puisqu'elles reconnaissant que deux termes peuvent dénoter le même concept (Euzenat & Shvaiko, 2007). Malencontreusement, elles reconnaissent également que le même terme peut dénoter plusieurs concepts à la fois. Par conséquent, ces techniques fournissent un grand nombre d'appariements possibles à partir des quels il faut choisir.

La prise en compte des niveaux supérieurs de la complexité sémantique constitue une manière de choisir les appariements les plus cohérents.

3.1.3 Techniques utilisées pour évaluer la similarité des chaînes longues de caractères

Les techniques suivantes opèrent usuellement sur les textes longs (comparant plusieurs mots) en les divisant en plusieurs morceaux appelés tokens. Par conséquent, les chaînes de caractères deviennent des ensembles de tokens. La similarité entre de tels ensembles de tokens est produite grâce à des mesures de similarité dites « mesures basées sur les tokens » (Bach, 2006).

Dans la littérature, plusieurs mesures basées sur les tokens ont été proposées (Bach, 2006). Nous présentons ici les mesures les plus utilisées dans les approches d'appariement d'ontologies.

A. Le coefficient d'appariement

La distance de Hamming appliquée aux ensembles d'entités a été adaptée aux ensembles de tokens. Elle est appelée le coefficient d'appariement (Euzenat, et al., 2004). Ce dernier correspond au nombre de tokens différents entre deux ensembles normalisé par la taille de leur union.

B. La similarité de Jaccard

La similarité de Jaccard peut être étendue pour comparer des ensembles de tokens en définissant la similarité comme le rapport entre la cardinalité de l'intersection des ensembles sur la cardinalité de leur union (Bach, 2006).

C. La similarité hybride

Dans (Monge & Elkan, 1997), les auteurs proposent une méthode hybride pour comparer des chaînes longues de caractères, qui découpe ces deux chaînes en plusieurs chaînes plus courtes. Ensuite, ces dernières sont comparées par une technique quelconque basée sur les chaînes courtes de caractères. Enfin, les résultats obtenus sont combinés.

Définition (Similarité hybride). *Soient $s = a_1...a_K$ et $t = b_1...b_L$ deux chaînes de caractères, où a_i et b_j sont des sous-chaînes de s et t respectivement. Soit S une mesure de la similarité entre deux chaînes courtes de caractères. La similarité hybride est une fonction de la similarité $\sigma_{Hybride} : S \times S \rightarrow [0, 1]$ telle que :*

$$\bar{\sigma}_{Hybride}(s, t) = \frac{1}{K} \sum_{i=1}^{K} \max_{j=1...L} S(a_i, b_j)$$

D. Les méthodes basées sur l'utilisation de la technique TF/IDF

Ces méthodes sont empruntées au domaine de la recherche d'information. Afin de les appliquer dans le contexte de l'appariement des ontologies, certaines adaptations ont été réalisées (Bach, 2006):

- Les concepts des deux ontologies à apparier sont considérés comme des documents ;
- Dans le cas où un concept est défini avec plusieurs commentaires, ces commentaires sont concaténés en un seul commentaire ;
- Les mots dans les commentaires d'un concept sont des mots du document ;
- L'univers des documents est l'univers de concepts qui est construit avec tous les concepts des deux ontologies ;
- Chaque concept est associé à un vecteur ; Les dimensions de ces vecteurs sont calculées par la technique TF/IDF avec l'univers de concepts et les mots des commentaires.

Dans sa conception originale, la technique TF/IDF est utilisée pour mesurer la pertinence d'un terme dans un ensemble de documents. La fréquence du terme, TF, dans un document donné montre l'importance de ce terme dans le document en question. La fréquence inverse de document, IDF, est une mesure de l'importance générale du terme dans l'ensemble des documents de l'univers (Bach, 2006). Dans le contexte de l'appariement d'ontologies, cette mesure est définie comme suit.

Définition (Term frequency/ Inverse document frequency adaptée au contexte de l'appariement des ontologies). *Soient O_1 et O_2 deux ontologies à apparier. Soit U l'univers des mots construit des mots contenus dans les commentaires de tous les concepts de O_1 et de O_2. Soit S = |U| le nombre des mots distincts dans l'univers U. Soit v = (w_1, w_2, … w_S) un vecteur représentant un certain concept c. La valeur de la i-ème dimension w_i du vecteur est calculée par la technique TF/IDF comme suit :*

$$w_{\mathrm{i}} = TF_{\mathrm{i}} \times IDF_{\mathrm{i}}$$

$$IDF_{\mathrm{i}} = log_2\left(\frac{N}{n_{\mathrm{i}}}\right)$$

où tf_i (fréquence de terme) est le nombre de fois où le i-ème mot dans l'univers U apparaît dans le commentaire du concept c, idf_i (fréquence inverse de document) est l'inverse du pourcentage des concepts qui contiennent le i-ème mot de l'univers U, N est le nombre de concepts dans les deux ontologies O_1 et O_2, n_i est le nombre de concepts qui contiennent le i-ème mot dans l'univers U au moins une fois.

Ainsi, la similarité entre deux commentaires est obtenue à partir de la distance entre les deux vecteurs qui les représentent. Plus deux vecteurs sont proches (la distance petite), plus les deux commentaires correspondants sont similaires (leur valeur de similarité est élevée). Les distances utilisées sont celles empruntées à la géométrie telles que la distance euclidienne, la distance de Manhattan et toute instance de la distance de Minkowski (Euzenat & Shvaiko,

2007). Nous présentons ici la similarité de cosinus qui mesure le cosinus des angles construit par deux vecteurs. La définition suivante est donnée par (Euzenat, et al., 2004).

Définition (Similarité de cosinus). *Étant donnés $\vec{s}$ et $\vec{t}$, les vecteurs correspondants aux deux chaînes de caractères s et t dans un espace de vecteur V, la similarité de cosinus est la fonction $\sigma_V : V \times V \rightarrow [0\ \ 1]$ telle que :*

$$\sigma_V(s,t) = \frac{\sum_{i \in |V|} \vec{s_i} \times \vec{t_i}}{\sqrt{\sum_{i \in |V|} \vec{s_i}^2 \times \sum_{i \in |V|} \vec{t_i}^2}}$$

E. Discussion

En plus de comparer les techniques syntaxiques, l'étude menée dans (Cohen, Ravikumar, & Fienberg, 2003) a également considéré les techniques utilisées pour évaluer la similarité des chaînes longues de caractères. Elle a montré que les méthodes basées sur la technique TF/IDF donnent les meilleurs résultats.

3.2 Techniques utilisées au niveau des structures de graphes

L'idée de ces techniques repose sur l'exploitation des relations existantes entre les concepts dans la déduction de la similarité entre les entités. Selon le type des relations considérées, ces techniques sont de deux catégories : celles exploitant la structure méréologique et celles considérant les autres types de relations. Elles sont présentées comme suit.

3.2.1 Les techniques exploitant la relation de méréologie

Dans les considérations de la similarité, la relation la plus importante est la méréologie, i.e., la relation « part-of ». Le principe est que les concepts seront plus similaires s'ils partagent des parties similaires (Euzenat & Shvaiko, 2007).

Cependant, l'exploitation de la structure méréologique présente la difficulté qu'il n'est pas facile de trouver les relations qui la portent puisqu'elles peuvent porter n'importe quel nom (Euzenat & Shvaiko, 2007). Par exemple, le concept « Proceedings » peut avoir une certaine relation « part-of » avec le concept « Inproceedings », mais elle sera exprimée à travers la relation « communications ».

3.2.2 Les techniques exploitant les autres relations

Le principe adopté par ces techniques est que si l'on a deux entités similaires *A* et *A*', et si elles sont connectées par un même type de relation *R* avec deux autres entités *B* et *B′*, alors on peut déduire que *B* et *B′* sont similaires (Bach, 2006). De même, si on sait que *A* et *A′* sont similaires et que *B* et *B′* sont aussi similaires, alors les relations *A*-*B* et *A′*-*B′* peuvent être similaires. Par exemple, si le concept « Faculty » est lié au concept « Course » par la relation

« teaches » dans une ontologie O_1, et si le concept « Professor » est lié au concept « Lecture » par la relation « teaches » dans une autre ontologie O_2, alors sachant que les concept « Lecture » et « Course » sont similaires, et que les relations « teaches » et « teaches » sont similaires, on peut inférer que « Faculty » et « Professor » sont aussi similaires.

L'idée a été étendue pour un ensemble d'entités et de relations (Bach, 2006). C'est à dire que si on possède un ensemble de relations $R_1....R_2$ dans la première ontologie qui sont similaires à un autre ensemble de relations $R'_1....R'_2$ dans la deuxième ontologie, alors les entités qui sont les domaines ou les co-domaines de ces relations sont considérées comme similaires.

Cependant, le problème causé par cette approche est qu'elle est basée sur l'utilisation de la similarité des relations pour inférer la similarité de leurs concepts de domaine ou de co-domaine. Ceci introduit une circularité dans le calcul de la similarité. Afin de pallier à cette circularité, plusieurs solutions ont été proposées (Euzenat & Shvaiko, 2007):

- La similarité des relations est calculée en se basant sur leurs étiquettes en utilisant les techniques basées sur les chaînes de caractères courtes.
- Si les relations sont organisées dans une taxonomie, les méthodes exploitant la structure taxonomique (cf. section 4) peuvent être utilisées.

3.3 Techniques utilisées au niveau extensionnel

La similarité entre deux concepts est décidée par les méthodes extensionnelles en analysant leurs extensions, c.-à-d., leurs ensembles d'instances (Bach, 2006). Généralement, ces méthodes sont divisées en deux catégories (Euzenat J. , et al., 2007) : celles qui s'appliquent aux ontologies ayant un ensemble d'instances commun, et celles qui opèrent sur des ontologies avec des ensembles d'instances disjoints.

3.3.1 Les techniques basées sur l'exploitation de l'ensemble d'instances commun

Dans le domaine des bases de données, les auteurs des travaux (Larson, Navathe, & Elmasri, 1989; Sheth, Larson, Cornelio, & Navathe, 1988) comparent deux classes C_1 et C_2 en utilisant le test de l'intersection de leur ensemble d'instances A et B respectivement. Ils considèrent que C_1 et C_2 sont très similaires quand $A \cap B = A = B$, plus générale si $A \cap B = B$ ou $A \cap B = A$, disjoint si $A \cap B = \phi$. Cependant, si aucun de ces cas ne s'applique (par exemple, quand les classes ont certaines instances en commun mais pas toutes), la dissimilarité ne peut être qu'égale à 1 (Euzenat J. , et al., 2007).

Par conséquent, l'utilisation de la distance de Hamming entre deux extensions peut résoudre ce problème (Euzenat & Shvaiko, 2007). Ceci correspond au nombre d'éléments différents entre deux ensembles normalisé par la taille de leur union.

Définition (Distance de Hamming adaptée pour les ensembles d'instances). *La distance de Hamming entre deux ensembles est une fonction de dissimilarité* $\delta_{Hamming\_Instance} : 2^E \times 2^E \rightarrow R$ *tel que* $\forall x, y \subseteq E$ *:*

$$\delta_{Hamming\_Instance}(x, y) = \frac{|x \cup y - x \cap y|}{|x \cup y|}$$

L'utilisation d'une telle distance dans la comparaison des ensembles d'instances est plus robuste que l'utilisation de l'égalité (Euzenat & Shvaiko, 2007) car elle tolère que certaines instances n'appartiennent pas à l'intersection des deux ensembles d'instances tout en produisant une courte distance.

L'adaptation de la similarité de Jaccard pour les ensembles des instances est également possible : ça convient au rapport entre l'intersection des ensembles et leur union (Bach, 2006).

Définition (Similarité de Jaccard). *Étant donnés deux ensembles A et B, soit P(X) la probabilité qu'une instance aléatoire soit dans l'ensemble X. La similarité de Jaccard est définie comme suit :*

$$\sigma_{Jaccard\_Instance}(A, B) = \frac{P(A \cap B)}{P(A \cup B)}$$

3.3.2 Comparaison des ensembles d'extensions disjoints

Dans le cas où les ensembles d'instances des ontologies ne partagent aucune partie commune, les mesures présentées dans la section précédente ne sont plus applicables (la valeur de similarité retournée sera toujours égal à 0, c.-à-d. les entités à comparer sont toujours différentes). Par conséquent, l'utilisation des méthodes approximatives pour comparer les instances se révèle une bonne solution. Dans ce contexte, deux types d'approches sont identifiées, à savoir (Euzenat & Shvaiko, 2007) : celles basées sur des techniques statistiques et celles qui se fondent sur des correspondances entre les ensembles.

3.3.2.1 Les approches statistiques

Les approches statistiques emploient des techniques de l'analyse multidimensionnelle (Cox & Cox, 1994). Elles reposent sur l'idée que si deux instances ont des distances très similaires à toutes autres instances, elles doivent être très similaires (Bach, 2006).

Les instances dans les ensembles sont représentées par des vecteurs, dont la valeur d'une dimension est la similarité de l'instance en question avec une autre instance dans les deux ensembles. La similarité entre deux ensembles est donc la similarité (la valeur du cosinus) des deux vecteurs moyens de ces deux ensembles (Bach, 2006).

Définition (Similarité des ensembles d'instances). *Soient $S = \{s_1, s_2, ...\}$ et $T = \{t_1, t_2, ...\}$ deux ensembles d'instances. Soit $\sigma_q(s_i, t_j)$ une mesure de similarité quelconque. Soit $\vec{s}_i = (sim(e_i,e_1), sim(e_i,e_2), ... sim(e_i,f_1), sim(e_i,f_2), ...)$ le vecteur représentant de l'instance e_i. La similarité des ensembles d'instances entre S et T est une fonction de la similarité $\sigma_{Set} : 2^E \times 2^E \rightarrow [0, 1]$ telle que :*

$$\bar{\sigma}_{set}(S,T) = \frac{\sum_{s \in S} \vec{s}}{\left|\sum_{s \in S} \vec{S}\right|} \otimes \frac{\sum_{t \in T} \vec{t}}{\left|\sum_{t \in T} \vec{t}\right|}$$

3.3.2.2 Techniques basées sur des correspondances

En partageant une idée similaire avec la mesure hybride (Monge & Elkan, 1997), une mesure de similarité basée sur les correspondances (match-based similarity) est définie dans (Valtchev, 1999) comme la similarité moyenne des paires des éléments dans Pairing, où Pairing est l'ensemble des correspondances entre les deux ensembles d'instances ayant la somme maximale des valeurs de similarité de ces correspondances (Bach, 2006).

Définition (Similarité basée sur des correspondances). *Soient S et T deux ensembles d'instances. Soit $\sigma_q(s, t)$ une mesure de similarité. Soit Pairing(S, T) l'ensemble des correspondances entre S et T ayant la somme maximale des valeurs de similarité de ces correspondances. La similarité basée sur les correspondances entre deux ensembles S et T est une fonction de la similarité $\sigma_{Corr} : 2^E \times 2^E \rightarrow [0, 1]$ telle que :*

$$\bar{\sigma}_{Corr}(S,T) = \frac{1}{max(|S|,|T|)} \sum_{(s,t) \in Pairings(S,T)} \sigma_q(n,n')$$

3.3.2.3 Discussion

Les ontologies développées indépendamment, même si elles portent sur le même sujet, sont très souvent hétérogènes aussi bien sur le plan conceptuel que représentatif (le langage de représentation utilisé) . Cependant, puisque les informations contenues dans les extensions des entités sont indépendantes de la partie conceptuelle de l'ontologie, elles sont supposées être moins sujettes à l'hétérogénéité (Euzenat & Shvaiko, 2007). Par conséquent, elles peuvent être utilisées pour apparier correctement les concepts.

Toutefois, il existe des situations dans lesquelles l'information sur les instances de données n'est pas disponible. Ceci peut être causé par l'indisponibilité des données ou par des besoins de confidentialité. Dans une telle situation, les techniques présentées dans cette section ne deviennent plus applicables.

3.4 Techniques utilisées au niveau des structures taxonomiques

En plus des techniques décrites dans la section précédente, le processus d'appariement des ontologies écrites en RDF(S) peut utiliser les techniques basées sur la structure taxonomique,

i.e., le graphe construit avec la relation « subClassOf » qui constitue la base sur laquelle reposent les ontologies. De nombreuses recherches ont porté sur ce type de structure et plusieurs mesures ont été proposées. Ces dernières sont divisées en deux catégories, à savoir : les mesures taxonomiques globales qui prennent en compte l'ensemble de la taxonomie pour évaluer la similarité entre les concepts, et les mesures taxonomiques locales, qui prennent seulement avantage des primitives rdfs: subClassOf et rdfs: subPropertyOf.

3.4.1 Les mesures taxonomiques globales

Les mesures taxonomiques globales les plus communes sont basées sur le calcul du nombre d'arcs, dans la taxonomie, entre deux concepts. Elles sont présentées comme suit.

A. La dissimilarité topologique structurelle

Proposée par (Valtchev & Euzenat, 1997), cette mesure se base sur la distance du chemin le plus court dans un graphe.

Définition (Dissimilarité topologique structurelle sur les hiérarchies). *La dissimilarité topologique structurelle $\delta_{Top} : o \times o \rightarrow R$ est une dissimilarité sur une hiérarchie $H=\langle o, \leq \rangle$, tel que :*

$$\forall\, e, e' \in o, \delta_{Top}(e, e') = \min_{c \in o}\left[\delta_{Top}(e, c) + \delta_{Top}(e', c)\right]$$

Où $\delta_{Top}(e, c)$ est le nombre d'arcs intermédiaires entre un élément e et un autre c.

Cependant, dans le cas où les concepts sont proches de la racine d'une hiérarchie, ils seront considérés proches l'un de l'autre en terme d'arcs bien qu'ils soient très différents conceptuellement (Euzenat & Shvaiko, 2007).

B. Similarité de Wu-Palmer

Afin d'apporter une réponse au problème constaté lors de l'utilisation de la dissimilarité topologique structurelle, les auteurs dans (Wu & Palmer, 1994) suggèrent de compter la position des concepts par rapport à la racine de l'ontologie. Ils définissent donc une similarité, la similarité de Wu-Palmer, comme suit.

Définition (Similarité de Wu-Palmer). *La similarité de Wu- Palmer $\sigma_{WP} : o \times o \rightarrow R$ est une similarité sur une hiérarchie $H=\langle o, \leq \rangle$, telle que :*

$$\sigma_{WP}(c, c') = \frac{2 \times \delta(c \wedge c', \rho)}{\delta(c, c \wedge c') + \delta(c', c \wedge c') + 2 \times \delta(c \wedge c', \rho)}$$

où ρ est la racine de la hiérarchie, $\delta(c, c')$ est le nombre d'arcs intermédiaires entre un concept c et un autre concept c' et $c \wedge c' = \{c'' \in o; c \leq c'' \wedge c' \leq c''\}$

C. Similarité cotopique d'upward

Dans (Mädche & Staab, 2002), les auteurs s'inspirent de la distance de Jaccard et proposent une mesure de similarité basée sur la notion de « upward cotopy » qui est définie par $UC(c, H) = \{c' \in H ; c \leq c'\}$, i.e., l'ensemble des super-concepts du concept c dans la hiérarchie H. Elle est définie comme suit.

Définition (Similarité cotopique d'upward). *La similarité cotopique de upward* σ_{CU} *:* $o \times o \rightarrow R$ *est une similarité sur une hiérarchie* $H = \langle o, \leq \rangle$*, tel que :*

$$\sigma_{CU}(c, c') = \frac{|UC(c, H) \cap UC(c', H)|}{|UC(c, H) \cup UC(c', H)|}$$

D. Discussion

Les mesures taxonomiques globales, comme présentées, opèrent en supposant que les deux schémas à apparier partagent la même taxonomie. Cependant, dans le contexte de l'appariement des ontologies, puisque les ontologies ne sont pas supposées partager la même taxonomie *H,* ces mesures ne peuvent pas être appliquées telles qu'elles sont (Euzenat & Shvaiko, 2007). Par conséquent, il est nécessaire que ces mesures soient adaptées pour pouvoir s'appliquer sur une paire d'ontologies.

La solution proposée dans (Ziegler, Kiefer, Sturm, Dittrich, & Bernstein, 2006) consiste à incorporer les ontologies dans un seul arbre. Ainsi, les concepts racines des ontologies sont des sous-concepts directs d'un concept racine appelé « Super–Thing ».

3.4.2 Les mesures taxonomiques locales

Les mesures taxonomiques locales considèrent, pour chaque paire de concepts à apparier, les sous-concepts directs et les super-concepts directs en exploitant les primitives rdfs:subClassOf et rdfs:subPropertyOf. Nous présentons, ci-dessous, certaines mesures rapportées dans (Euzenat & Shvaiko, 2007).

A. Les règles de sous-concepts et super-concepts

Ces règles capturant l'intuition que les concepts sont similaires si leurs super ou sous-concepts sont similaires. Si les concepts sont les mêmes, ils auront les mêmes super-concepts, et si les super-concepts sont les mêmes, les concepts sont alors semblables. Les sous-concepts de deux concepts semblables seront aussi semblables. Si les sous-concepts sont les mêmes, les concepts comparés sont semblables. Aussi, si les concepts ont des frères et sœurs semblables, ils sont aussi semblables (Dieng & Hug, 1998; Ehrig & Sure, 2004).

B. L'appariement de chemin limite

Cette technique a été introduite par Anchor-PROMPT (Noy & Musen, 2001). Le principe ici est de prendre deux chemins, définis par les relations hiérarchiques, entre les concepts, de comparer les concepts et leurs positions le long de ces chemins, et d'identifier ceux qui sont similaires. Une explication plus détaillée sur le principe de fonctionnement d'Anchor-PROMPT est donnée dans le chapitre 4.

C. Discussion

Les techniques taxonomiques les plus utilisées reposent sur les règles de sous-concepts et super-concepts. Cependant, ces dernières présentes quelques inconvénients (Euzenat & Shvaiko, 2007) :

- Quand il y a plusieurs sous ou super-concepts, alors ils seront tous appariés avec le même concept ;
- La similarité entre les sous-concepts ou les super-concepts dépendra, à son tour, sur celle de leurs super ou sous-concepts. Ceci conduit à une circularité dans le calcul de la similarité.

4 Techniques utilisées pour apparier les caractéristiques supportées par OWL

On retrouve deux niveaux associés à OWL. Le premier est consacré aux restrictions. Tandis que le deuxième niveau soulève l'aspect de concepts complexes.

4.1 Techniques utilisées au niveau des restrictions

Ces techniques sont aussi référencées dans (Bach, 2006; Euzenat, et al., 2004) par « *les méthodes basées sur la structure interne* » et dans (Rahm & Bernstein, 2001) comme « *les méthodes basées sur les contraintes* ». Afin de mesurer la similarité entre les entités, ces méthodes utilisent les informations concernant les attributs des entités telles que les types de données acceptés pour instancier les co-domaines ou la multiplicité (Bach, 2006; Euzenat & Shvaiko, 2007). Par exemple, les informations pouvant être exploitées concernant les attributs du concept « car » sont : l'intervalle des valeurs de données pour l'attribut « hasSpeed », à savoir [200, 350] ou bien la cardinalité de l'attribut « belongsTo » qui est égale à 1.

4.1.1 Comparaison de types de données

La comparaison des types de données est généralement basée sur l'idée de les interpréter comme des ensembles de valeurs (Valtchev, 1999; Valtchev & Euzenat, 1997). Ces derniers ne sont pas complètement disjoints puisqu'il existe des règles par lesquelles une valeur d'un certain type peut être convertie dans la représentation mémoire d'un autre type. Par conséquent, la similarité entre deux types de données doit être maximale quand ils sont du

même type, inférieur quand ils sont compatibles (par exemple, integer et float sont compatibles puisqu'ils peuvent être convertis l'un dans l'autre) et minimale quand ils ne sont pas compatibles (par exemple, booléen et integer) (Euzenat & Shvaiko, 2007).

Exemple. Soient deux ontologies O_1 et O_2. L'ontologie O_1 comporte un concept « Staff » avec trois attributs « firstname », « lastname » et « birthdate » tels que « firstname » et « lastname » sont de type « string » tandis que « birthdate » est de type « date ». Parmi les éléments de l'ontologie O_2, on trouve le concept « Person » qui est caractérisé par l'attribut « name » de type « string » et l'attribut « hiringdate » de type « date ». La comparaison des types de données des concepts « Staff » et « Person » doit nous permettre d'apparier « firstname » et « lastname » avec « name », et « birthdate » avec « hiringdate ».

Cette comparaison donne des résultats intéressants puisqu'elle trouve les correspondances attendues. Cependant, elle trouve aussi des correspondances incorrectes (birthdate-hiringdate). Par conséquent, elle ne peut pas être utilisée isolément.

4.1.2 Comparaison de multiplicités

Les attributs sont, généralement, contraints par des multiplicités qui représentent les cardinalités acceptables de l'ensemble des valeurs d'un attribut. Ces multiplicités sont, très souvent, exprimées à travers l'intervalle des entiers positifs $[0 + \infty [$.

Le principe sur lequel repose la comparaison de deux multiplicités est qu'elles soient considérées compatibles si l'intersection de leurs intervalles correspondants n'est pas vide (Euzenat, et al., 2004).

Dans (Euzenat & Valtchev, 2004), les auteurs s'inspirent de la similarité de Jaccard et proposent la mesure de similarité suivante.

Définition (Similarité de multiplicité). *Étant données deux expressions de multiplicité [b e] et [b′ e′], la similarité de multiplicité est une similarité entre les intervalles "integer" non négatifs* $\sigma_{Mul} : 2^{\tau} \times 2^{\tau} \rightarrow [0\ \ 1]$*, telle que :*

$$\sigma_{Mul}([b\ e],[b'e']) = \begin{cases} 0, & si\ b' > e\ ou\ b\ > e' \\ \dfrac{\min(e,e') - \max(b,b')}{\max(e,e') - \min(b,b')}, & sinon \end{cases}$$

4.1.3 Discussion

Les restrictions ne fournissent pas beaucoup d'informations sur les entités à comparer puisque des concepts très différents peuvent avoir des attributs avec les mêmes types de données (Euzenat & Shvaiko, 2007). Par conséquent, les méthodes présentées dans cette section sont généralement combinées avec d'autres techniques, telles que les méthodes basées sur la structure d'une ressource externe, afin de réduire le nombre de correspondances candidates en éliminant les correspondances incompatibles.

4.2 Techniques utilisées au niveau des concepts complexes

Afin de déduire la similarité entre deux entités, les méthodes utilisées à ce niveau, dites « *techniques sémantiques* », se basent sur des modèles de logique tels que la satisfiabilité propositionnelle (SAT), la SAT modale ou les logiques de description (Bach, 2006).

Pour une tâche inductive comme l'appariement d'ontologies, les méthodes déductives pures ne fonctionnent pas très bien toutes seules (Castano, Ferrara, Hess, & Montanelli, 2007). Elles ont besoin d'une phase de prétraitement qui fournit des 'ancres', i.e., les entités qui sont déclarées comme étant équivalentes en utilisant, par exemple, les techniques basées sur les chaînes courtes de caractères ou les entrées de l'utilisateur.

4.2.1 Techniques propositionnelles

Plusieurs approches appliquant des techniques de la satisfiabilité propositionnelles (SAT) à l'appariement des ontologies ont été proposées (Giunchiglia & Shvaiko, 2003; Giunchiglia, Shvaiko, & Yatskevich, 2004; Shvaiko, 2006). Elles incluent, généralement, les étapes suivantes (Euzenat & Shvaiko, 2007):

(i) Construire une théorie ou une connaissance de domaine (Axiomes) pour les deux ontologies O_1 et O_2 comme une conjonction des axiomes disponibles. La théorie est construite en utilisant les techniques des niveaux de la complexité sémantique discutées dans les sections précédentes, par exemple, celles basées sur la structure d'une ressource externe.

(ii) Pour chaque paire de concepts c_1 et c_2 appartenant aux deux ontologies O_1 et O_2 respectivement, construire une formule d'appariement. Le critère qui détermine l'existence d'une relation entre deux concepts est le fait qu'elle est entrainée par les prémisses. Donc, une requête d'appariement est créée par la formule suivante :

$$Axioms \rightarrow r(c_1, c_2)$$

pour chaque paire de concepts c_1 et c_2 pour lesquelles on veut tester la relation r tel que $r=\{=, \sqsupseteq, \sqsubseteq, \perp\}$.

(iii) Vérifier que la formule est valide, à savoir que c'est vrai pour tous les assignements de vérité de toutes les variables proportionnelles occurrentes dans la formule. Une formule proportionnelle est valide si et seulement si sa négation est insatisfaisable. Ce qui est vérifié en utilisant un solveur SAT.

Exemple. Nous reproduisons ici l'exemple donné dans (Euzenat & Shvaiko, 2007).

Etape 1 : Supposons qu'une ontologie 1 contient les concepts « images » et « Europe », tandis qu'une autre ontologie 2 contient les concepts « pictures » et « Europe ». En utilisant la similarité de synonyme, on peut déterminer que images = pictures. Aussi, l'utilisation de la

mesure d'égalité des chaînes de caractères permet de trouver que les concepts Europe dans les deux ontologies sont identiques, i.e., Europe=Europe. Puis, la translation des relations entre les concepts en question en des formules proportionnelles donne les axiomes suivants :

(images↔pictures)∧ (Europe↔Europe)

Etape 2 : Supposons que *c* est identifié comme « Europe ∏ images » qui signifie le concept « European images », tandis que *c′* est défini comme « pictures ∏ Europe » qui signifie le concept « pictures of Europe ». Supposons aussi qu'on veut savoir si *c* est équivalant (↔) à *c′*. Ainsi, cette tâche d'appariement exige la construction de la formule suivante :

((images ↔ pictures) ∧ (Europe ↔ Europe))→ ((Europe∧images) ↔(Europe ∧ pictures))

Etape 3 : La négation de cette formule est invalide. Donc, la relation d'équivalence existe.

Il est à noter que ces approches, en plus de réduire les correspondances incorrectes, découvrent aussi de nouvelles correspondances entre les concepts complexes (Euzenat & Shvaiko, 2007). Cependant, elles n'acceptent que des prédicats unaires, i.e., des concepts, et ne prennent pas en compte les relations.

Afin de pallier aux limites liées au SAT propositionnel, le travail dans (Giunchiglia & Shvaiko, 2003) propose d'utiliser le SAT modal. Ce dernier permet de faire des calculs avec des prédicats binaires et d'employer des opérateurs de la logique modale (Bach, 2006). La validité de l'ensemble de formules exprimées en logique modale est aussi vérifiée en utilisant des procédures de recherche de la satisfiabilité (SAT). Si la validité est satisfaite, les relations hypothétiques entre des entités, qui sont des traductions de la requête sur la relation entre ces entités en logique modale, sont confirmées (Bach, 2006).

4.2.2 Les techniques de la logique de description

Les techniques des logiques de description (telles que le test de subsomption) peuvent être employées pour vérifier des relations sémantiques entre des entités telles que l'équivalence, la subsomption ou l'exclusion (Bach, 2006).

Exemple. Nous reproduisons ici l'exemple donné dans (Euzenat & Shvaiko, 2007).

Considérons deux ontologies:

- Micro-company = Company ⊓ $\leq_5$ employee : Signifiant qu'une Micro-compagnie est une compagnie avec au plus 5 employés ;
- SME = Firm ⊓ $\leq_{10}$ associate : Signifiant qu'une SME est une Firme avec au plus 10 associés.

L'alignement initial fourni inclut:

- Company = Firm : Signifiant que « Company » est équivalent à « Firm » ;
- Associate ⊑ employee : Signifiant que « associate » est un sous-concept de « employee ».

Ceci évidemment entraine :

Micro-company ⊑ SME : Signifiant que « Micro-company » est un sous concept de « SME ».

4.2.3 Discussion

Bien qu'elles exigent un ensemble d'ancres préalable pour mener à bien leur tâche, les techniques utilisées au niveau des concepts complexes sont d'une grande importance puisqu'elles permettent d'assurer la complétude, i.e., trouver toutes les correspondances qui doivent exister, et la consistance, i.e., trouver les correspondances qui mènent à l'inconsistance de l'alignement (Euzenat & Shvaiko, 2007).

Un challenge important de ces techniques est le fait de choisir des correspondances alternatives au lieu de celles qui sont inconsistances (Euzenat & Shvaiko, 2007).

5 Conclusion

Dans ce chapitre, nous avons présenté une classification des techniques de base en se basant sur les arguments utilisées dans les différents niveaux de la complexité sémantique associés aux différents langages du Web sémantique. Cette classification est naturelle puisqu'elle fournit un guide d'utilisation selon le langage ontologique utilisé.

Il est à noter que les techniques de base sont, en général, inspirées de domaines différents comme les Statistiques, les Bases de Données, les Mathématiques ou encore l'Intelligence Artificielle. L'appariement d'ontologies tire ses sources donc de plusieurs domaines.

Cependant, la tâche d'appariement d'ontologies est censée trouver des rapports entre toutes les entités des ontologies qui peuvent être des concepts, des relations, des instances, des axiomes et des règles (Bach, 2006). Les techniques proposées jusqu'à présent n'exploitent pas le niveau de complexité sémantique correspondant aux règles. A notre connaissance, aucune recherche n'a été menée dans ce sens.

Bien que le calcul de la similarité entre les entités des ontologies soit la base sur laquelle repose tout système d'appariement, il ne constitue qu'une étape parmi d'autres de leur processus. Subséquemment, une autre partie de l'état de l'art de l'appariement d'ontologies comptent sur l'explicitation du processus d'appariement. Cette dernière est le sujet du chapitre suivant.

CHAPITRE 4 : PROCESSUS D'APPARIEMENT D'ONTOLOGIES

L'appariement d'ontologies est généralement défini comme un processus qui prend en entrée deux ontologies et retourne un alignement qui identifie les entités ayant une signification identique ou proche.

Plusieurs explicitations du processus d'appariement d'ontologies ont été proposées dans la littérature (Castano, Ferrara, Hess, & Montanelli, 2007; Ehrig, 2007; Euzenat & Shvaiko, 2007). Ces processus diffèrent souvent dans leurs degrés de granularité et de détails. L'objet de ce chapitre est de les étudier et de tenter de proposer un processus d'appariement qui permette de prendre en considération le maximum d'ingrédients pouvant composer un système d'appariement.

Pour cela, le chapitre est organisé comme suit. La section 1 est consacrée aux travaux existants traitant des processus d'appariement d'ontologies. Dans la section 2, une explicitation du processus d'appariement proposé est présentée. Une conclusion terminera ce chapitre.

1 Processus existants

Dans la littérature, plusieurs processus d'appariement d'ontologies ont été proposés. Cette section a pour objectif de les présenter selon leur degré de granularité et de détail progressif.

1.1 Processus 1 (Euzenat J. , et al., 2007)

Dans (Euzenat J. , et al., 2007), les auteurs définissent le processus d'appariement comme suit

Définition (Processus d'appariement) *Le processus d'appariement peut être considéré comme une fonction f qui prend en entrée une paire d'ontologies o et o', un alignement en entrée A, un ensemble de paramètres p et un ensemble de ressources r et retourne en sortie un alignement A' entre ces ontologies :*

$$A' = f(o, o', A, p, r)$$

Cela peut être schématisé comme suit.

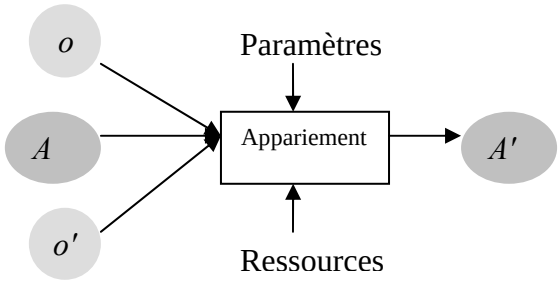


Figure 1. Processus 1 (Euzenat J. , et al., 2007)

1.2 Processus 2 (Castano, Ferrara, Hess, & Montanelli, 2007)

Dans (Castano, Ferrara, Hess, & Montanelli, 2007), les auteurs présentent le processus d'appariement d'ontologies en trois principales étapes (cf. figure 2).

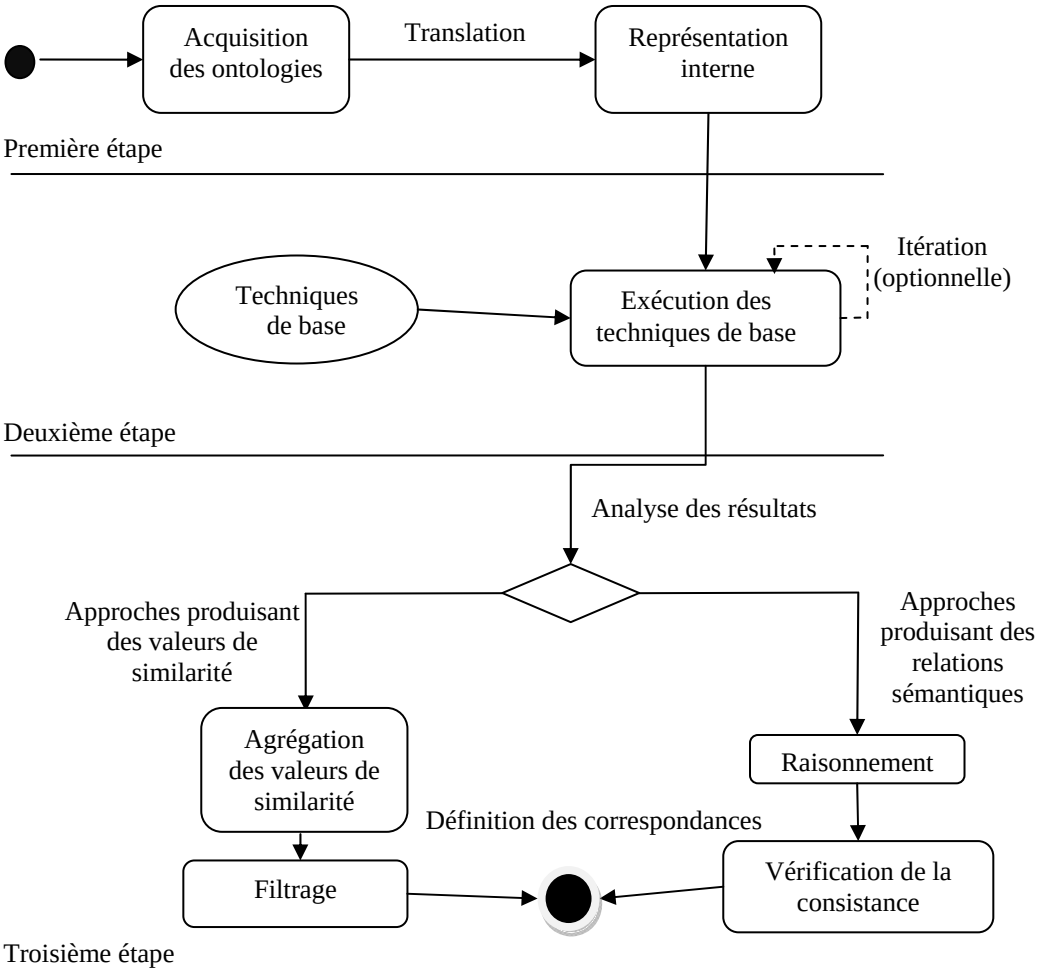


Figure 2. Processus 2 (Castano, Ferrara, Hess, & Montanelli, 2007)

A. La première étape

Cette étape consiste à acquérir les ontologies devant être appariées et à les représenter dans un modèle interne.

B. La deuxième étape

C'est l'étape d'exécution des techniques d'appariement de base. Elle est souvent répétée plusieurs fois pour raffiner les résultats obtenus dans les exécutions antérieures.

C. La troisième étape

Dans cette étape, les correspondances entre les éléments des ontologies seront déterminées. Ici, on peut avoir deux tâches différentes selon les résultats des techniques d'appariement de base exécutées.

- **Agrégation des valeurs de similarité.** Dans le cas des techniques d'appariement de base produisant une valeur de similarité entre les éléments des ontologies, les

différentes valeurs de similarité individuelles sont combinées dans une valeur de similarité représentative. Les résultats qui ne sont pas considérés pertinents sont supprimés.

- **Raisonnement et vérification de la consistance.** Dans le cas des techniques d'appariement de base qui produisent une relation sémantique entre les éléments des ontologies, une tâche de raisonnement est exécutée afin d'inférer de nouvelles relations et de vérifier la consistance des correspondances.

Enfin, un alignement est déterminé entre les éléments des ontologies en entrée.

1.3 Processus 3 (Ehrig, 2007)

Dans (Ehrig, 2007), les auteurs considèrent que la plupart des approches d'appariement d'ontologies sont subsumées par le processus de la figure suivante.

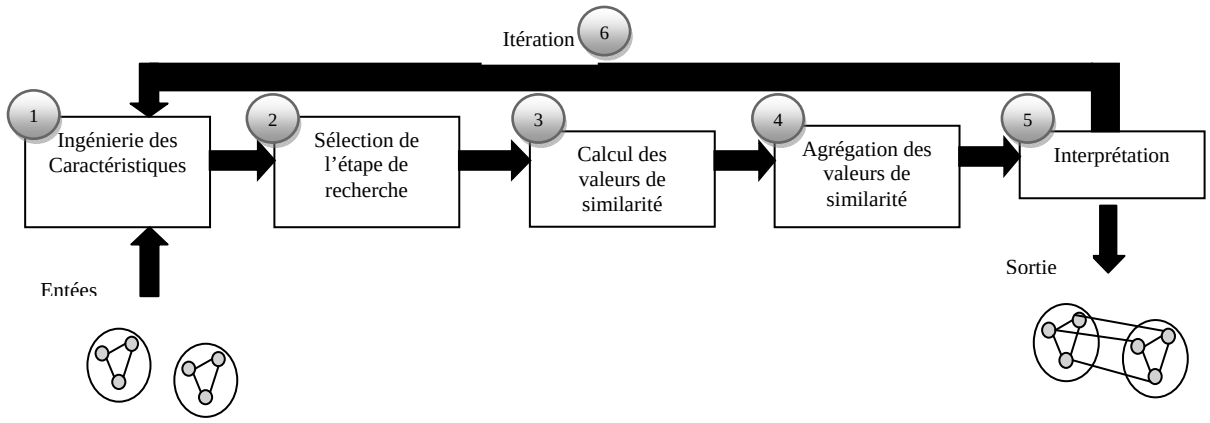


Figure 3. Processus d'appariement d'ontologies selon (Ehrig, 2007)

A. Les entrées

Les entrées d'un processus d'appariement d'ontologies sont deux ontologies qui doivent être appariées l'une à l'autre. Si plus de deux ontologies sont prises, elles seront comparées deux à deux. En plus, il est souvent possible de faire entrer des alignements pré-connus (qui peuvent être manuels) qui peuvent aider à améliorer la recherche d'alignements.

B. Ingénierie des caractéristiques

Afin de comparer deux entités de deux ontologies différentes O_1 et O_2, un sous ensemble spécifique de leurs caractéristiques est extrait. Ces extraits ne sont pas des constructions arbitraires, mais possèdent plutôt une signification spécifique dans l'ontologie. Par exemple, un algorithme d'appariement peut utiliser seulement un sous-ensemble de primitives OWL tel que la taxonomie, ou les descriptions linguistiques (par exemple, l'étiquette « Voiture » pour décrire le concept « Voiture »).

C. Sélection de l'étape de recherche

Avant de procéder à la comparaison des entités, il est nécessaire de choisir les paires d'entités à considérer. Cette étape a pour objectif de spécifier les paires d'entités candidates.

D. Calcul de la similarité

Le calcul de la similarité entre deux entités e et f de deux ontologies différentes fournit des valeurs de similarité individuelles.

E. Agrégation des valeurs de similarité

Dans cette étape, les différentes valeurs de similarité des différentes caractéristiques d'une paire d'entités candidate sont agrégées afin d'avoir une seule valeur de similarité représentative de la paire d'entités en question.

F. Interprétation

L'interprétation utilise les valeurs de similarité représentatives ou individuelles pour dériver un alignement entre les entités. Le mécanisme le plus commun est celui des seuils (Do & Rahm, 2002; Noy & Musen, 2001).

G. Itération

Puisque la similarité d'une paire d'entité influence la similarité des paires d'entités voisines, des itérations sur l'ensemble du processus sont exécutées. Dans chaque itération, les similarités d'un alignement candidat sont recalculées en se basant sur les similarités des paires d'entités voisines. L'itération termine lorsqu'aucune nouvelle correspondance n'est proposée ou après un nombre fixe d'itération.

H. Les sorties

La sortie est une représentation des alignements en utilisant une table d'alignement avec des degrés de confiance.

1.4 Discussion

Le processus 1 se présente comme une boîte noire où aucune spécification des étapes devant être suivies n'est indiquée. D'autre part, les processus 2 et 3 sont à un niveau de détails comparables. Les étapes définissant ces deux processus sont bien étoffées par moment et le sont moins par d'autres. Par exemple, dans le processus 2, l'étape de sélection des entités sur lesquelles vont être appliquées les techniques de base n'est pas indiquée. De plus, ce processus anticipe l'étape de vérification de la consistance en la considérant propre aux approches fournissant des relations sémantiques. Cependant, il existe des outils, tels que

Prompt, qui utilisent la vérification de consistance après avoir produit des valeurs de similarité entre les entités comparées.

D'autre part, le processus 3 ne fait pas la distinction entre les approches utilisant un raisonnement logique et celles qui opèrent sur des valeurs de similarité.

Aussi, on ne voit à aucun moment au niveau de quelle étape l'intervention humaine est située. Cette remarque est très importante puisque, à l'heure actuelle, la plupart des approches d'appariement sont semi-automatique. L'implication humaine dans les différentes étapes du processus est donc de rigueur.

Par conséquent, nous proposons dans la section suivante une combinaison des trois processus présentés afin d'en fournir un plus détaillé qui prend en compte le maximum d'ingrédients pouvant composer un système d'appariement.

2 Processus proposé

Le processus que nous proposons est résumé dans la figure suivante (Oulefki & Akli-Astouati, 2008 a).

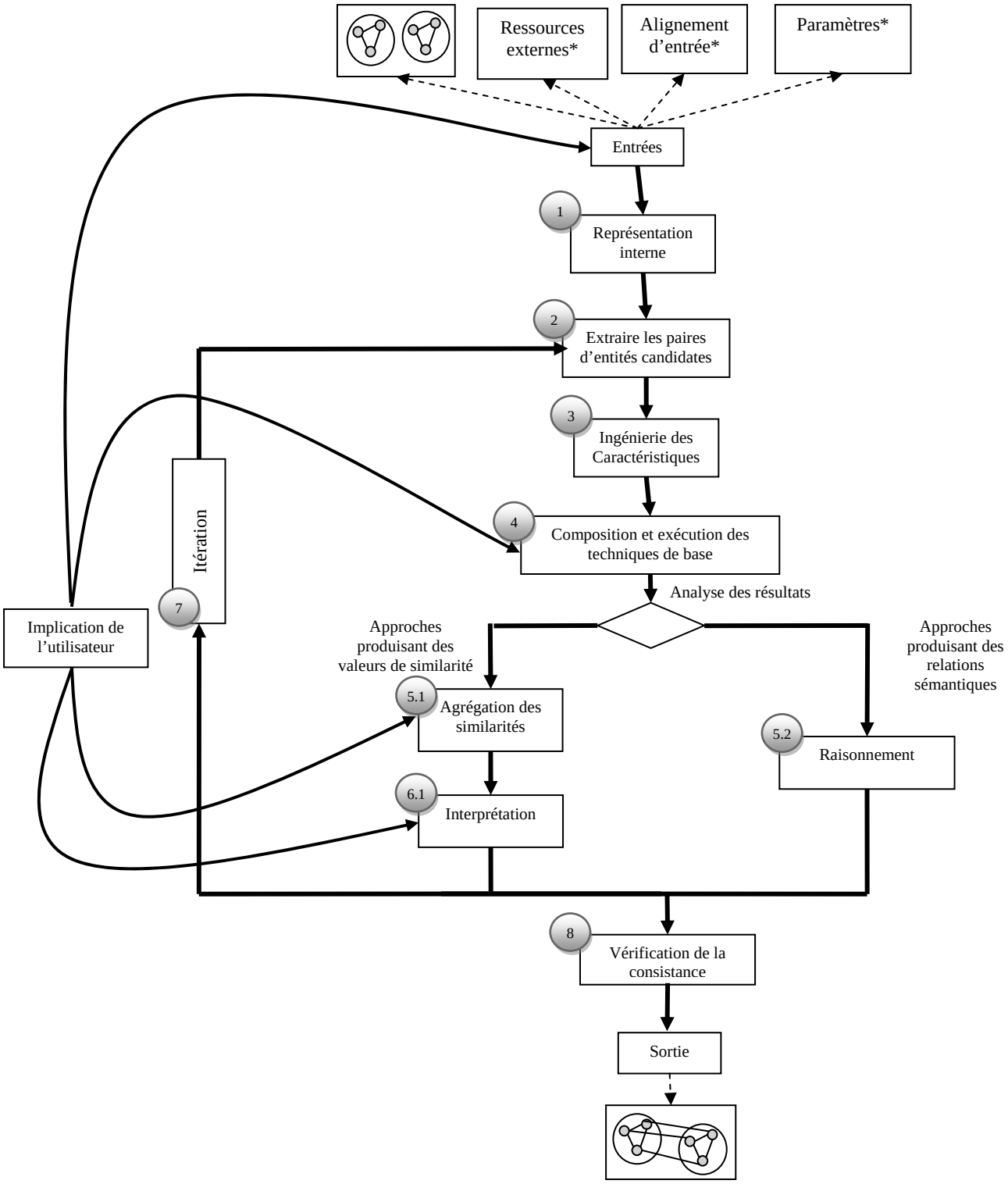


Figure 4. Processus d'appariement d'ontologies proposé.

Les éléments étiquetés par * dans la figure 4 sont facultatifs.

2.1 Les entrées

Les entrées à considérer dans notre processus sont :

- Deux ontologies (ou plus) O_1 et O_2 qui doivent être appariées l'une à l'autre. Celles-ci sont indispensables pour tout système d'appariement;
- Les paramètres du système, tels que le seuil de filtrage des correspondances utilisé dans l'étape d'interprétation. Dans certains systèmes d'appariement, ces paramètres ne sont pas requis puisqu'ils sont définis automatiquement, par exemple, par l'utilisation des méthodes d'apprentissage (Euzenat & Shvaiko, 2007);
- L'alignement initial qui contraindra le système pour fournir un alignement. Cet alignement pour la plupart des systèmes d'appariement est facultatif ;
- Les ressources externes qui servent de support pour le calcul de la similarité.

2.2 Représentation dans un modèle interne

Dans cette étape, les ontologies à apparier sont représentées dans un modèle interne qui peut avoir différentes formes selon l'approche d'appariement utilisée. Par exemple, afin d'éviter les conflits provoqués par l'hétérogénéité syntaxique, les ontologies sont écrites à l'aide d'un langage ontologique commun (Eklöf & Martenson, 2006).

2.3 Extraire les paires d'entités candidates

Cette étape est définie de la même manière que l'étape « Sélection de l'étape de recherche » du processus 3 décrit précédemment. Elle consiste donc à extraire les paires d'entités candidates. Les méthodes les plus communes pour faire le choix des entités candidates sont (Ehrig, 2007) :

- Choisir toutes les entités d'une première ontologie avec toutes les entités de l'autre ontologie;
- Choisir seulement les entités de même type (concepts, relations, instances).

2.4 Ingénierie des caractéristiques

Les algorithmes d'appariement d'ontologies extraient un sous ensemble spécifique de caractéristiques des ontologies afin de leur appliquer un calcul de similarité. Ces caractéristiques sont extraites conformément aux niveaux de complexité sémantique. Dans une étape ultérieure, ces caractéristiques sont utilisées pour être comparées.

2.5 Composition et exécution des techniques de base

La tâche fondamentale de tous les systèmes d'appariement est de trouver les relations entre les entités exprimées dans des ontologies différentes en calculant leur degré de similarité. Ceci est réalisé grâce aux techniques de base qui s'appliquent sur une caractéristique particulière des entités, tel que le nom, les attributs et les relations. Ainsi, les caractéristiques d'une entité sont comparées avec les caractéristiques correspondantes d'une autre entité (Bach, 2006).

Afin de couvrir toutes les caractéristiques des ontologies, les techniques de base doivent être composées. Cette composition peut être soit séquentielle ou parallèle (Euzenat & Shvaiko, 2007).

2.5.1 Composition séquentielle

La composition séquentielle consiste par exemple, à utiliser une technique de base pour comparer les noms des entités avant d'exécuter une autre qui est basée sur les relations sémantiques (Euzenat & Shvaiko, 2007). La figure suivante présente graphiquement la composition séquentielle.

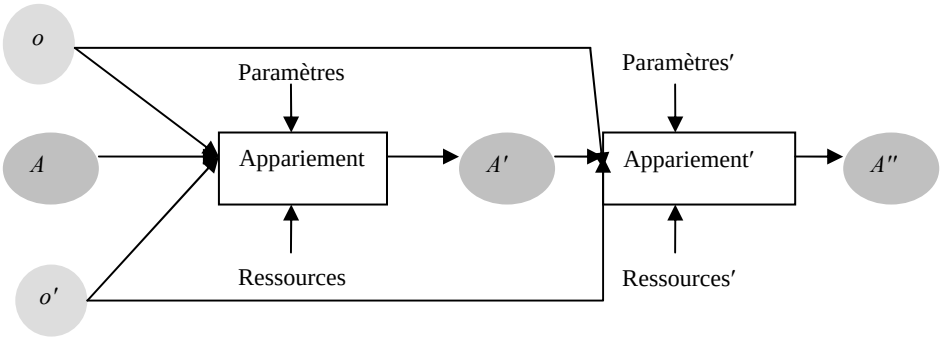


Figure 5. La composition séquentielle des techniques de base

2.5.2 Composition parallèle

La composition parallèle consiste à exécuter indépendamment plusieurs techniques de base où chacune est exécutée sur une caractéristique particulière des entités (cf. figure 6).

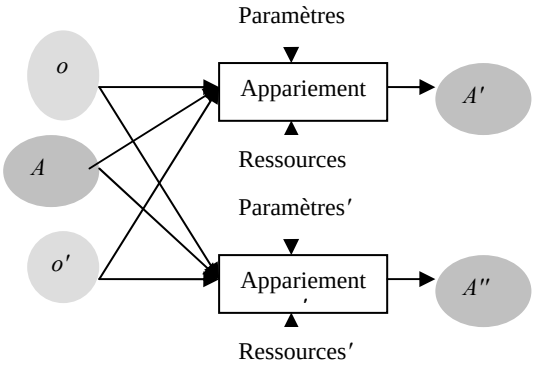


Figure 6. Composition parallèle de techniques de base

Les résultats fournis par l'exécution des techniques de base sont soit des valeurs de similarité individuelles, tels que les résultats retournés par les techniques syntaxiques, ou des relations sémantiques, tels que les résultats renvoyés par les techniques appliquées au niveau des concepts complexes.

2.6 Agrégation de Similarité

Dans le cas des techniques de base qui produisent des valeurs de similarité individuelles entre les éléments des ontologies, ces différentes valeurs doivent être agrégées afin de fournir une seule valeur de similarité représentative pour deux entités à comparer. Par exemple, le calcul de la similarité entre deux concepts exige l'agrégation, dans une seule valeur de similarité représentative, de la similarité obtenue à partir de leurs noms, de leurs superclasses et celle de leurs instances.

Nous présentons dans ce qui suit les méthodes d'agrégation les plus citées.

2.6.1 Distance de Minkowski et Somme pondérée

Cette distance est définie comme suit (Bach T. L., 2006).

Définition (Distance de Minkowski). *Étant donné B un ensemble d'objets qui peuvent être analysés dans n dimensions, la distance de Minkowski entre deux objets appartenant à B est comme suit :*

$$\forall\, x, x' \in B, \delta(x, x') = \left(\sum_{i=1}^{n} \delta(x_i, x_i')^p\right)^{\frac{1}{p}}$$

Où $\delta(x_i, x_i')$ est la dissimilarité de la paire d'objets le long de la i-ème dimension, i.e., la i-ème caractéristique.

Cette distance est une mesure généralisée avec différentes valeurs de p, p ≥1. Quelques exemples des distances de Minkowski sont la distance Euclidienne (où $p = 2$), la distance de Manhattan (où $p = 1$) et la distance de Chebichev (où $p = \infty$).

Cependant, il est généralement constaté que les valeurs de similarité devant être agrégées n'ont pas la même importance. Par exemple, la similarité des noms est plus importante que celle des commentaires. Il faut donc un moyen pour contrôler l'influence (ou l'importance) de chaque dimension sur la valeur finale de la distance. Ceci est réalisé en associant des poids à chacune des dimensions. Plus les dimensions sont importantes plus les poids sont élevés (Euzenat & Shvaiko, 2007).

Dans la plupart des cas, les poids doivent être déterminés manuellement. Cependant, l'utilisation d'un apprentissage automatique est également possible.

La distance de Minkowski ainsi que ses variantes peuvent alors bénéficier de la technique des poids associés aux dimensions afin de donner plus d'importance à des dimensions particulières. L'exemple typique et le plus utilisé dans les approches d'appariement est la somme pondérée qui associe des poids à la distance de Manhattan. Elle est définie comme suit (Bach T. L., 2006).

Définition (Somme pondérée). *Étant donné B un ensemble d'objets qui peut être analysé en n dimensions, la somme pondérée entre deux objets appartenant à B est comme suit :*

$$\forall x, x' \in B, \delta(x, x') = \sum_{i=1}^{n} w_i \times \delta(x_i, x_i')$$

Tel que $\delta(x_i, x_i')$ est la dissimilarité de la paire d'objets le long de la i-ème dimension et w_i est le poids de la dimension i.

Si toutes les valeurs sont normalisées, i.e., $\sum^{n}{}_{i=1}\, w_i=1$, alors ce type de mesures peut-être normalisé (Euzenat J. , et al., 2007).

2.6.2 Produit pondéré

Le produit pondéré est défini comme suit (Euzenat & Shvaiko, 2007):

Définition (Produit pondéré). *Soit B un ensemble d'objets qui peut être analysé dans n dimensions. Le produit pondéré entre deux objets appartenant à B est donné par :*

$$\forall x, x' \in B, \delta(x, x') = \prod_{i=1}^{n} \delta(x_i, x_i')^{w_i}$$

Tel que $\delta(x_i, x_i')$ est la dissimilarité de la paire d'objets le long de la i-ème dimension et w_i est le poids de la dimension i.

Le produit pondéré présente l'inconvénient que si l'une de ses dimensions possède une mesure égale à 0, le résultat sera aussi égal à 0 (Euzenat & Shvaiko, 2007).

2.6.3 Moyenne pondérée

Dans (Gal, et al. 2005), la moyenne pondérée est définie comme suit.

Définition (Moyenne pondérée). *Étant donné B un ensemble d'objets qui peut être analysé en n dimensions. La moyenne pondérée entre deux objets appartenant à B est donnée par :*

$$\forall\, x, x' \in B, \delta(x, x') = \frac{\sum_{i=1}^{n} w_i \times \delta(x_i, x_i')}{\sum_{i=1}^{n} w_i}$$

Tel que $\delta(x_i, x_i')$ est la dissimilarité de la paire d'objets le long de la i-ème dimension et w_i est le poids de la dimension i.

Si les valeurs sont normalisées, la moyenne pondérée est normalisée. En fait, la somme pondérée normalisée est aussi une moyenne (Euzenat & Shvaiko, 2007).

2.7 Interprétation

Les techniques de base fournissent un grand ensemble de correspondances à partir desquelles l'alignement doit être extrait (Euzenat & Shvaiko, 2007). Ceci est accompli par une méthode d'extraction spécialisée qui agit sur la matrice de similarité.

Les méthodes d'extraction d'alignements sont soient manuelles, semi-automatiques ou automatiques :

- **Extraction manuelle.** Les systèmes utilisant l'extraction manuelle affichent les paires d'entités avec leurs valeurs de similarité, et laissent aux utilisateurs le choix des paires appropriées (Euzenat & Shvaiko, 2007).
- **Extraction semi-automatique.** Les systèmes adoptant une extraction semi-automatique offrent une ou plusieurs méthodes d'extraction à partir desquelles les utilisateurs choisissent la méthode convenable ainsi que ses paramètres.
- **Extraction automatique.** Les systèmes optant pour une extraction automatique d'un alignement définissent des algorithmes qui automatisent cette extraction à partir des valeurs de similarité.

Le mécanisme le plus commun pour l'extraction des correspondances est l'utilisation d'un filtrage basé sur un seuil en sélectionnant les correspondances ayant une valeur de similarité supérieur à ce seuil (Euzenat & Shvaiko, 2007). L'application des seuils exige donc que l'alignement extrait soit d'une qualité suffisante.

Plusieurs méthodes pour spécifier la valeur du seuil de filtrage peuvent être trouvées dans la littérature (Do & Rahm, 2002; Ehrig & Sure, 2004). Nous présentons dans ce qui suit celles rapportées dans (Ehrig, 2007)

A. La méthode Delta où le seuil est défini en prenant la plus grande valeur de similarité et lui soustraire une constante :

$$\theta = \max_{e \in o, f \in o'} (\sigma(e, f)) - const$$

B. La méthode N. pourcent qui est très liée à la précédente. Ici, on choisit la plus grande valeur de similarité trouvée et on lui soustrait un pourcentage p :

$$\theta = \max_{e \in o, f \in o'} \big(\sigma(e, f)\big) - (1 - p)$$

C. La méthode proportionnelle qui consiste à utiliser, comme seuil, le pourcentage de la plus grande valeur de similarité.

2.8 Raisonnement

Dans le cas des techniques d'appariement de base qui produisent une relation sémantique entre les éléments des ontologies, une tâche de raisonnement est exécutée afin d'inférer de nouvelles relations comme dans le processus de (Castano, Ferrara, Hess, & Montanelli, 2007).

2.9 Itération

Le calcul de la similarité mené jusqu'ici reste encore local puisque les techniques de base prennent en compte que les voisins d'un nœud (Euzenat & Shvaiko, 2007). Cependant, le calcul de la similarité doit impliquer l'ensemble des ontologies car les valeurs de similarité finales peuvent dépendre de toutes les entités des ontologies. D'autre part, quand l'ontologie n'est pas réduite à un graphe acyclique dirigé, la distance définie par les techniques de base peut-être définie d'une manière circulaire (Euzenat & Shvaiko, 2007). Cette circularité peut être illustré, par exemple, quand la distance entre deux concepts dépend des distances entre leurs instances qui elles-mêmes dépendent de la distance entre leurs concepts.

La manière classique de palier à ces problèmes implique le calcul itératif de la distance ou de la similarité en raffinant à chaque étape les dernières valeurs calculées. L'itération termine lorsque aucune nouvelle correspondance n'est proposée ou après un nombre fixe d'itération (Ehrig, 2007).

2.10 Vérification de la consistance

Cette étape consiste à trouver les correspondances qui mènent à l'inconsistance de l'alignement. Le challenge ici est le fait de trouver des correspondances alternatives à celles qui sont inconsistances (Euzenat & Shvaiko, 2007).

2.11 Implication des utilisateurs

L'implication des utilisateurs est un autre élément qui doit être pris en compte lors de la conception d'un système d'appariement. Cette implication peut être faite dans les champs suivants:

- **Fournir les entrées.** Les utilisateurs peuvent fournir plusieurs types d'entrées aux systèmes d'appariement tels que : les ontologies devant être appariées, les paramètres du système et l'alignement initial ;
- **Composer les techniques de base.** Certains environnements offrent aux utilisateurs des techniques permettant de composer les techniques de base (Euzenat & Shvaiko, 2007) ;
- **Agrégation des valeurs de similarité.** Dans certains systèmes d'appariement, les utilisateurs spécifient la méthode d'agrégation appropriée ainsi que ses paramètres (le poids) parmi un ensemble de méthodes d'agrégation proposées ;
- **Interprétation des correspondances.** Les utilisateurs sont impliqués dans l'étape d'interprétation dans les systèmes adoptant des méthodes d'extraction manuelles ou semi-automatiques.

2.12 Les sorties

La sortie est un alignement composé d'un ensemble de correspondances entre les entités des ontologies O_1 et O_2.

3 Présentation de quelques systèmes d'appariement

Afin d'illustrer la généralité du processus proposé, nous présentons dans cette section son application sur quelques systèmes existants d'appariement d'ontologies. Un panorama plus étendu peut être trouvé dans (Kalfoglou & Schorlemmer, 2003; Rahm & Bernstein, 2001).

3.1 PROMPT (Stanford SMI)

PROMPT (Noy & Musen, 2000) est un plug-in intégré dans Protégé qui effectue la tâche d'alignement et de fusion d'ontologies. L'ensemble des phases associées à son processus est comme suit.

- **Entrées.** Les entrées qu'admet PROMPT sont deux ontologies, O_1 et O_2, écrites en OWL ou en RDF(S). L'une de ces deux ontologies (soit O_2) doit être définie par l'utilisateur comme étant moins générale que l'autre (soit O_1). L'utilisateur doit également spécifier le type du processus qu'il veut exécuter, i.e., la génération d'un alignement ou d'une fusion.
- **Extraire les paires d'entités candidates.** L'outil compare tous les concepts de O_1 avec tous ceux de O_2.
- **Ingénierie des caractéristiques.** PROMPT compte seulement sur l'utilisation des noms de concepts.

- **Composition et exécution des techniques de base.** L'outil utilise uniquement l'égalité des chaînes de caractères pour déterminer la similarité entre deux noms de classes. Par conséquent, la composition de techniques de base n'est pas nécessaire.
- **Agrégation de Similarité.** Puisque PROMPT n'utilise qu'une seule technique de base, aucune méthode d'agrégation ne doit être adoptée.
- **Interprétation.** Basé sur les correspondances trouvées dans l'étape précédente, PROMPT crée une liste de suggestions, dite "`ToDo`". Un item sur la liste `ToDo` est une opération simple ou une disjonction d'opérations. Dans le dernier cas, la sémantique de base est que seulement une des opérations de la disjonction doit être effectuée.

  La création de la liste `ToDo` se fait selon que le processus correspond à la génération d'une fusion ou d'un alignement.

  - *Si le processus est la génération d'une fusion,* PROMPT suggère, pour chaque paire de concepts avec des noms identiques, de les fusionner ou de supprimer l'un d'entre eux.

  - *Si le processus est la génération d'un alignement,* les suggestions que fait PROMPT sont fondées sur l'hypothèse que les classes de l'ontologie la moins générale, O_2, nécessitent d'être liées par des relations de sous-classe/super-classe avec les classes de l'ontologie la plus générale, O_1. Pour chaque classe C de O_2, les suggestions suivantes sont données : S'il y a une classe dans O_1 avec le même nom que C, fusionner les deux classes, Autrement, trouver un parent pour la classe C.

  A partir de la liste `ToDo`, l'utilisateur choisit une des opérations suggérées. PROMPT exécute l'opération demandée et réalise automatiquement les changements supplémentaires basés sur le type d'opération choisi.

- **Vérification de la consistance.** En se basant sur la nouvelle structure de l'ontologie, PROMPT met à jour la liste ToDo et crée une nouvelle liste dite "`conflits`".

  Les items sur la liste `conflits` représentent les inconsistances dans l'état actuel de la base de connaissances. Un item de cette liste se compose d'une description d'un conflit et d'une action suggérée qui remédiera à ce conflit. Il contient également une solution par défaut qui sera appelée si l'utilisateur demande à PROMPT de résoudre le conflit automatiquement. L'utilisateur est ensuite invité à valider tout ou une partie des suggestions selon ses besoins.

- **Itération.** Les deux étapes précédentes sont répétées jusqu'à ce que les ontologies soient entièrement fusionnées ou alignées.
- **Sortie.** L'alignement retourné est simple et il est de multiplicité 1 :1.

3.2 Anchor-PROMPT (Stanford SMI)

Anchor-PROMPT (Noy & Musen, 2001) est une extension de PROMPT qui permet d'aligner et de fusionner des ontologies en utilisant un mécanisme prompt pour découvrir automatiquement les concepts sémantiquement similaires.

- **Entrées.** L'outil reçoit en entrée deux ontologies écrites en OWL ou en RDF(S). Le système prend, également, en entrée un ensemble d'ancres, i.e., un alignement en entrée, qui sont des couples de concepts liés. Cette liste de paires peut être fournie par l'utilisateur ou identifiées automatiquement.
- **Représentation interne.** Chaque ontologie est représentée par un graphe étiqueté orienté à partir de la hiérarchie des concepts (appelés classes dans l'algorithme) et de la hiérarchie des relations (appelées slots dans l'algorithme), où les nœuds dans le graphe sont des concepts et les arcs sont des relations dénotant des liens entre les concepts (les étiquettes des arcs sont les noms des relations).
- **Extraire les paires d'entités candidates.** L'outil compare toutes les entités des deux ontologies quelque soit leur type.
- **Ingénierie des caractéristiques.** Anchor-PROMPT compte aussi bien sur l'utilisation des noms des concepts que sur leur hiérarchie.
- **Composition et exécution des techniques de base.** Les auteurs proposent une méthode séquentielle pour calculer la similarité des concepts de deux ontologies. Dans une première passe, si l'ensemble d'ancres n'est pas spécifié manuellement, Anchor-PROMPT utilise l'égalité des chaînes de caractères pour le déterminer.

  Ensuite, les auteurs supposent que les concepteurs d'ontologies relient les concepts similaires par des relations similaires même s'ils ne les nomment pas avec les mêmes noms.

  Pour chaque paire d'ancres, Anchor-PROMPT considère deux chemins de même longueur telle que la longueur d'un chemin est le nombre d'arcs qui le composent. Il sélectionne tous les concepts intermédiaires deux à deux qui occupent les mêmes positions. Ceci permet de juger si ces paires de concepts sont équivalentes. Deux concepts se trouvant dans la même position entre deux ancres, sont également équivalents. Ils assignent donc un degré de confiance représentant la similarité de chaque paire de concepts considérés comme étant équivalents.

  Cette étape est répétée pour chaque paire de chemins de longueur égale entre les deux ancres considérées. Les auteurs augmentent le degré de confiance pour chaque paire de concepts équivalents rencontrés. Par conséquent, la paire de concepts qui apparaît le plus souvent dans les mêmes positions sur les chemins délimités par la paire d'ancres considérée possèdera le plus grand degré de confiance.

Chaque paire de concepts considérés similaires aura un nombre de valeur de similarité égal au nombre de paires d'ancres reliées par des chemins traversant les paires de concepts en question.

Par ce traitement, Anchor-PROMPT suppose que les deux ontologies sont construites de la même façon. Les résultats retournés seront donc limités si les structures des ontologies sont différentes (par exemple, l'une est profonde avec beaucoup de concepts au milieu, et l'autre est peu profonde).

- **Agrégation de Similarité.** La moyenne des valeurs de similarité fournies par l'étape précédente est calculée afin de n'avoir qu'une seule valeur de similarité représentative pour chaque paire de concepts.
- **Interprétation.** Anchor-PROMPT applique un seuil pour filtrer l'ensemble des correspondances produites dans les étapes précédentes. Les concepts qui sont considérés fortement similaires sont présentés à l'utilisateur. Ce dernier sélectionne manuellement les paires de concepts qu'il estime correctes afin qu'Anchor-PROMPT procède à leur fusion.
- **Itération.** Dans Anchor-PROMPT, l'itération a pour objectif de permettre un perfectionnement manuel. Une fois que l'utilisateur a validé une proposition de correspondance, le système recalcule les similarités et présente de nouvelles propositions de fusion.
- **Sorties.** L'alignement résultant est sauvegardé dans un fichier RDF.

3.3 ASCO

ASCO (Bach, Dieng-Kuntz, & Gandon, 2004) est un outil développé à INRIA Sophia Antipolis qui permet de chercher des entités correspondantes entre deux ontologies représentées en RDF(S). Le processus suivi par cet outil est détaillé comme suit.

- **Entrées.** ASCO prend en entrée deux ontologies.
- **Représentation interne.** Cette étape se charge de formaliser les ontologies en entrée en RDF(S) (si elles ne le sont pas déjà).
- **Extraire les paires d'entités candidates.** L'alignement candidat est construit des paires d'entités de deux ontologies ayant le même type.
- **Ingénierie des caractéristiques.** ASCO exploite les points forts du formalisme RDF(S) tels que la capacité de représenter le nom, les étiquettes et les commentaires d'une classe, d'une relation ou d'une instance via ses primitives rdf:id, rdfs:label, rdfs:comment, respectivement, ainsi que les liens de subsomptions (spécialisations) entre des classes et entre des relations, en utilisant les primitives rdfs:subClassOf et rdfs:subPropertyOf respectivement.
- **Composition et exécution des techniques de base.** ASCO calcule la similarité des entités des ontologies en deux phases séquentielles :

- *La phase linguistique.* Dans cette phase, la mesure de Jaro-Winkler est utilisée pour comparer les noms et les étiquettes des entités, tandis que la technique TF/IDF est adaptée pour comparer leurs commentaires. Afin d'améliorer la précision du calcul, ASCO a intégré WordNet (Miller, 1995), pour exploiter les relations de synonymie ou hyperonymie entre les termes utilisés pour dénoter les entités.
- *La phase structurelle.* Le calcul de la similarité basé sur les informations structurelles dépend des valeurs de similarité linguistiques. Les voisines de deux entités sont considérées similaires si celles-ci sont linguistiquement similaires.

- **Agrégation de Similarité.** Cette étape consiste à agréger premièrement, pour chaque paire d'entités, les valeurs de similarité calculées dans la phase linguistique puis celles calculées dans la phase structurelle. Les deux valeurs de similarité résultantes de ces agrégations sont à leur tour agrégées. Dans ces trois cas d'agrégation, la méthode adoptée est la somme pondérée avec des poids choisis au début de l'algorithme, suivant la nature des ontologies à aligner.
- **Interprétation.** Pour chaque entité de la première ontologie, ASCO cherche l'entité de la deuxième ontologie qui est la plus similaire (la valeur de similarité finale entre elles est la plus élevé). Si cette valeur de similarité finale dépasse un seuil de similarité prédéfini, ces deux entités sont ajoutées dans la liste des correspondances à fournir.
- **Itération.** Le calcul de la similarité ne compte sur aucune correspondance précédemment calculée. Par conséquent, un cycle est suffisant dans ASCO.
- **Sorties.** Les correspondances retournées sont sous forme de triplets (e_1, e_2, sim) où e_1 est une entité (concept ou relation) de la première ontologie, e_2 est une entité de même type que e_1 de la deuxième ontologie, et sim est la valeur de similarité entre ces deux entités. Ces correspondances peuvent être celles de type 1-1 : une entité d'une ontologie possède une et seulement une entité correspondante dans l'autre ontologie ; ou 1-n : une entité d'une ontologie peut correspondre à une ou plusieurs entités de l'autre ontologie.

4 Conclusion

Le processus d'appariement d'ontologies proposé dans ce chapitre est le résultat d'une étude théorique sur les différentes explicitations recensées dans la littérature ainsi que sur quelques systèmes d'appariement existants. Cette explicitation apparaît plus complète que toutes celles déjà proposées. Afin de l'illustrer nous avons présenté un survol de quelques systèmes d'appariement existants en déroulant leur fonctionnement selon les étapes du processus proposé.

Le nombre sans cesse croissant d'outils d'appariement laisse l'utilisateur indécis sur le choix de l'outil approprié. Ceci nécessite de procéder à l'évaluation des outils d'appariement afin d'identifier les points forts et faibles de chacun et savoir leur domaine d'application. Cette évaluation fera l'objet de développement du chapitre suivant.

CHAPITRE 5 : COMPARAISON D'ALIGNEMENTS

Le nombre d'outils disponibles pour apparier des ontologies ne cesse de croître. Il est vite apparu le besoin de les évaluer afin de guider le choix de l'utilisateur pour la méthode appropriée. Cependant, très peu de travaux existent dans ce sens et ne prennent en compte, dans leur processus d'évaluation que des alignements simples. Les systèmes produisant des alignements complexes ne sont appréciés à leurs juste valeur.

L'objectif de ce chapitre est de proposer une approche permettant d'évaluer aussi bien les systèmes fournissant des alignements simples que ceux procurant des alignements complexes.

Pour cela, nous présentons en premier les travaux existant traitant de l'évaluation des systèmes d'appariement pour dévoiler ensuite notre approche Align-Match permettant de comparer des alignements.

1 Travaux existants traitant de d'évaluation des systèmes d'appariement

Dans le domaine de l'évaluation des méthodes d'appariement d'ontologies, peu de travaux peuvent être dénombrés. Nous avons recensé dans la littérature seulement deux travaux : I^{3}CON et OAEI.

I^{3}CON (Information Interpretation and Integration Conference)[1] est le premier effort s'intéressant à l'évaluation des systèmes d'appariement. Il se présente comme un framework d'évaluation systématique et fournit pour cela 10 paires d'ontologies. Le Tableau suivant résume les informations concernant les paires d'ontologies de test, avec leur nom, leur langage de représentation, le nombre de concepts, de relations et d'instances dans les deux ontologies.

Paire d'ontologie	Langage	Nombre de concepts	Nombre de relations	Nombre d'instances
Animals	OWL	13-13	15-14	11-0
Sports	DAML	-	-	-
Computer Science	DAML	-	-	-
Hotels	OWL	10-8	3-6	7-10
Computer Networks	OWL	27-27	5-6	0-2
Pets	OWL	121-113	15-15	21-20
Pets (pas d'instances)	OWL	120-112	15-15	0-0
Russia	RDF	162-151	80-75	214-158
Wine	OWL	33-33	0-0	0-0
Weapons	OWL	79-81	0-0	0-0

Tableau 1. Les paires d'ontologies dans la campagne de tests I^{3}CON

Le deuxième effort dans ce contexte est OAEI (Ontology Alignment Evaluation Initiative)[2]. Celui-ci se compose de 51 paires d'ontologies de test. Ces dernières sont systématiquement générées à partir d'une ontologie de référence dans le domaine de la bibliographie en modifiant ou supprimant un certain nombre d'informations afin d'évaluer comment les algorithmes se comportent quand ces informations sont modifiées ou supprimées. Par exemple, les noms des entités peuvent être remplacés par des chaînes de caractères aléatoires ou par des synonymes ; les commentaires peuvent être supprimés ou changés ; les structures des hiérarchies de concepts ou de relations peuvent être modifiées…

[1] http://www.atl.lmco.com/projects/ontology/i3con.html
[2] http://oaei.ontologymatching.org

1.1 Processus d'évaluation adopté

Les deux compagnes I^{3}CON et OAEI adoptent exactement le même processus d'évaluation. La première phase consiste à apparier manuellement les paires d'ontologies de tests. Le résultat obtenu est considéré comme l'alignement de référence, dénoté par A_1. L'ensemble des correspondances contenues dans ce dernier est dénoté par « Ntrue ».

Dans la deuxième phase, le système d'appariement à évaluer est exécuté sur les paires d'ontologies de tests. Cela produit un alignement, dénoté par A_2, qu'il faut évaluer dans les phases ultérieures. L'ensemble des correspondances contenues dans ce dernier est dénoté par « Nfound»

La troisième phase est la comparaison de l'alignement de référence avec celui obtenu par le système d'appariement à évaluer. Cette comparaison a pour objectif de trouver l'intersection des deux ensembles Nfound et Ntrue, i.e., l'ensemble des paires appartenant à la fois à A_1 et A_2. Cette intersection est dénotée par « Ncorrect ».

La dernière phase consiste à évaluer la qualité de A_2 en calculant la valeur des mesures « précision », « rappel », « f-mesure » et « mesure globale » (overall measure) (Bach, 2006). Ce sont des métriques qui permettent une analyse fine des performances des systèmes d'appariement.

La précision est la proportion des correspondances correctes, i.e., Ncorrect, parmi l'ensemble de celles contenues dans A_2, i.e., Nfound. Cette mesure reflète la précision d'un outil. Plus la valeur de précision est élevée, plus le bruit dans le résultat de l'outil est réduit, et donc plus la qualité du résultat est imposante. Ainsi, la fonction précision est définie par :

$$\text{Précision} = \frac{|\text{Ncorrect}|}{|\text{Nfound}|}$$

Le rappel est la proportion de correspondances correctes contenue dans A_2, i.e., Ncorrect, parmi toutes celles qui sont correctes, i.e., Ntrue. Le rappel mesure l'efficacité d'un outil. Plus la valeur de rappel est élevée, plus A_2 couvre toutes les correspondances correctes. La fonction *rappel* est donc définie par :

$$\text{rappel} = \frac{|\text{Ncorrect}|}{|\text{Ntrue}|}$$

La F-mesure est un compromis entre le rappel et la précision. Elle permet de comparer les performances des algorithmes par une seule mesure. Plus F-mesure augmente, plus l'outil à évaluer est performant. La f-mesure est définie par :

$$\text{f} - \text{mesure} = \frac{2 \times \text{rappel} \times \text{précision}}{\text{rappel} + \text{précision}}$$

La mesure globale, définie dans (Melnik, Garcia-Molina, & Rahm, 2001), correspond à l'effort requis pour corriger le résultat renvoyé par l'outil d'appariement afin d'obtenir le résultat correct. Cette mesure est toujours inférieure à la f-mesure. Elle n'a de sens que dans le cas où la précision n'est pas inférieure à 0,5, c'est-à-dire si au moins la moitié des correspondances renvoyées par l'algorithme sont correctes (Bach, 2006). En effet, si la majorité des correspondances est erronée, l'utilisateur doit fournir plus d'effort pour supprimer les correspondances incorrectes et ajouter les correspondances correctes mais absentes, que pour mettre en correspondance manuellement les deux ontologies à partir de zéro.

$$\text{overall} = \text{rappel} \times \left(2 - \frac{1}{\text{précision}}\right)$$

1.2 Discussion

Les compagnes d'évaluation présentées dans cette section fournissent un grand nombre de tests afin de permettre d'évaluer les systèmes d'appariement dans plusieurs conditions d'exécution et de ce fait de dégager leurs points forts et leurs points faibles.

Toutefois, les références bibliographiques (Euzenat, Ehrig, & Castro, 2005) concernées par la description de ces compagnes ne précisent à aucun moment comment l'alignement de références est-il comparé avec l'alignement fourni par les systèmes à évaluer. Il parait que le fait d'appliquer simplement l'égalité des chaînes de caractères sur chaque paire de correspondances appartenant respectivement à l'alignement de référence et à l'alignement fourni par le système à évaluer est suffisant car ces deux derniers concernent la même paire d'ontologies et l'alignement de référence est simple. Afin de renforcer cette hypothèse, dans le cadre de l'équipe, un projet de fin d'étude soutenu en décembre 2007 a implémenté un prototype d'évaluation basé sur l'égalité des chaînes de caractères. Nous avons pour cela comparé quelques alignements de référence fournis par ces compagnes avec des alignements qu'on a construits manuellement. Les résultats obtenus étaient très raisonnables.

La raison pour laquelle les compagnes d'évaluation utilisent un simple algorithme de comparaison d'alignements est qu'elles ont principalement pour objectif de tester les systèmes d'appariement sur un maximum de conditions diverses. Leurs efforts vont donc essentiellement dans ce sens.

Par ailleurs, la comparaison d'une correspondance complexe avec une autre qui est simple en utilisant l'égalité des chaînes de caractère donnera toujours zéro même si la correspondance simple est incluse dans la correspondance complexe. Par exemple, si nous comparons la correspondance représentant l'équivalence des attributs fisrtname et lastname d'une ontologie O1 et l'attribut name d'une autre ontologie O2 avec la correspondance de référence représentant l'équivalence des attributs fisrtname de O1 et l'attribut name de O2,

alors le résultat retourné sera égal à 0. Par conséquent, les systèmes qui fournissent des alignements complexes seront lésés dans l'appréciation.

Afin de comparer automatiquement deux alignements qu'ils soient simples ou complexes, nous proposons dans la section suivante l'approche Align-Match.

2 Approche proposée pour apparier des alignements

2.1 Architecture de haut niveau d'Align-Match

L'algorithme d'Align-Match prend en entrée deux alignements A_1 et A_2, calcule la similarité des correspondances de A_1 et de A_2 en utilisant un ensemble de mesures de similarité et renvoie en sortie une liste de paires de correspondances similaires entre A_1 et A_2, i.e., Ncorrect. Dans la suite de ce chapitre, nous appelons les correspondances contenues dans les alignements en entrée, A_1 et A_2, « correspondances» et celles retournées par Align-Match « correspondances en sortie». La figure suivante montre le processus adopté par Align-Match.

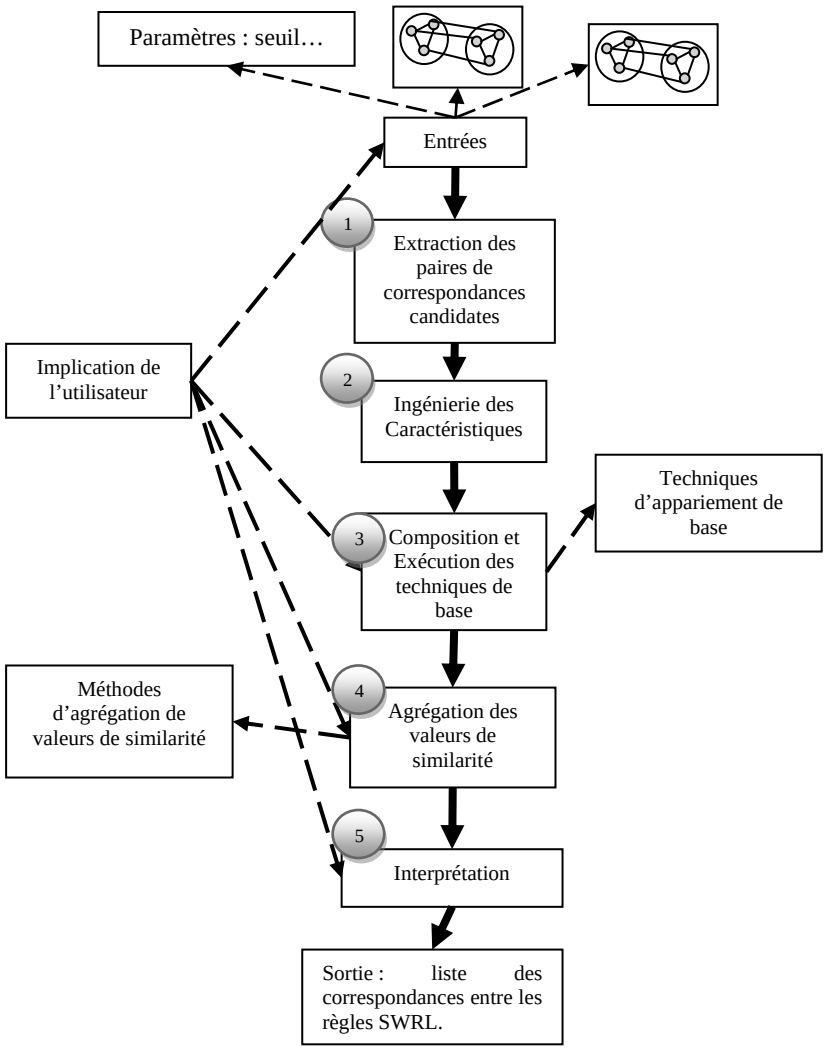


Figure 1. Processus d'appariement d'alignements d'Align-Match

Les différentes composantes du processus présenté dans la figure 1 sont détaillées comme suit.

2.2 Align-Match en détail

2.2.1 Les entrées

Les entrées d'Align-Match sont :

Deux alignements A_1 et A_2 qui doivent être appariés l'un à l'autre. Puisque Align-Match prend en compte aussi bien les alignements simples que les alignements complexes, chaque correspondance appartenant à A_1 ou A_2 est considérée comme un quintuple : <{name, label, comment}, {$e_{11}, e_{12},\ldots, e_{1m}$}, {$e_{21}, e_{22},\ldots, e_{2k}$}, r, n> tel que :

- {name, label, comment} est l'ensemble représentant le nom, l'étiquette et le commentaire de la correspondance en question : Le nom est un identifiant unique pour la correspondance dans un alignement ; l'étiquette est employée pour fournir une version lisible du nom de la correspondance pour les utilisateurs humains tandis que le commentaire donne une description servant à fournir des informations descriptives de la correspondance. Il est à noter que le label et le comment peuvent être vides car il existe des systèmes d'appariement qui ne les fournissent pas.
- {$e_{11,} e_{12},\ldots, e_{1m}$} et {$e_{21,} e_{22},\ldots, e_{2k}$} sont les ensembles alignés d'entités tels que $e_i \in \varphi$ et φ={concept, relation, instance}. Lorsque l'alignement est simple ces deux ensembles ne contiendront qu'un seul élément ;
- r $\in \theta$ tel que θ = {=, ≥, ≤} où « = » signifie l'équivalence et « ≥ », »≤ » signifie la subsumption ;
- n est le degré de confiance exprimé dans l'intervalle [0 1].

Le seuil de filtrage des correspondances en sortie qui sert dans l'étape d'interprétation.

Il est à noter que ces entrées sont fournies manuellement par l'utilisateur d'Align-Match.

Exemple. Soit l'exemple d'alignement d'ontologies d'université du chapitre 2. L'alignement de référence « A_1» est donné comme suit.

<{name= "Student_PHDStudent", label= "Correspondence between Student and PHDStudent", Comment=" this correspondence asserts that the class "Student" from the second ontology is super-class of the class "PHDStudent" from the first ontology"}, {Student}, {PHDStudent}, ≥, 1.0>

<{name= "Course_Lecture", label= "Correspondence between Course and Lecture", Comment=" this correspondence asserts that the class "Course" from the second ontology is equivalent to the class "Lecture" from the first ontology"}, {Course}, {Lecture}, =, 0.7>

<{name= “Faculty_Full Professor”, label= “Correspondence between Faculty and Full Professor”, Comment=” this correspondence asserts that the class "Faculty" from the second ontology is equivalent to the class "Full Professor" from the first ontology”}, {Faculty}, {Full Professor}, =, 0.9>

<{name= “teaches_teaches”, label= “Correspondence between teaches and teaches”, Comment=” this correspondence asserts that the attribute "teaches" from the second ontology is equivalent to the attribute "teaches" from the first ontology”}, {teaches}, {teaches}, =, 1.0>

<{name= “room_ office”, label= “Correspondence between room and office”, Comment=” this correspondence asserts that the attribute "room" from the second ontology is equivalent to the attribute "office" from the first ontology”}, {room}, {office}, =, 1.0>

<{name= “Office_Room”, label= “Correspondence between Office and Room”, Comment=” this correspondence asserts that the class "Office" from the second ontology is equivalent to the class "Room" from the first ontology”}, {Office}, {Room}, =, 1.0>

<{name= “firstname+lastname_name”, label= “Correspondence between firstname lastname and name”, Comment=” this correspondence asserts that the attributes "firstname" and "lastname" from the second ontology are equivalent to the attribute "name" from the first ontology”}, {firstname, lastname}, {name}, =, 1.0>

<{name= “street+city_address”, label= “Correspondence between street city and address”, Comment=” this correspondence asserts that the attributes "street" and "city" from the second ontology are equivalent to the attribute "address" from the first ontology”}, {street, city}, {address}, =, 1.0>

L’alignement à évaluer « A_2» est comme suit.

<{name= “Student_PHDStudent”, label= “Correspondence between Student and PHDStudent”, Comment=””}, {Student}, {PHDStudent}, ≥, 1.0>

<{name= “teaches_teaches”, label= “”, Comment=””}, {teaches}, {teaches}, =, 1.0>

<{name= “room_office”, label= “Correspondence between room and office”, Comment=” this correspondence asserts that the attribute "room" from the second ontology is equivalent to the attribute "office" from the first ontology”}, {room}, {office}, =, 1.0>

<{name= “Office_Room”, label= “Correspondence between Office and Room”, Comment=” this correspondence asserts that the class "Office" from the second ontology is equivalent to the class "Room" from the first ontology”}, {Office}, {Room}, =, 1.0>

<{name= "firstname_name", label= "Correspondence between firstname and name", Comment=" this correspondence asserts that the attribute "firstname" from the second ontology is equivalent to the attribute "name" from the first ontology"}, {firstname}, {name}, =, 0.8>

<{name= "street+city_address", label= "Correspondence between street city and address", Comment=" this correspondence asserts that the attributes "street" and "city" from the second ontology are equivalent to the attribute "address" from the first ontology"}, {street, city}, {address}, =, 1.0>

2.2.2 Extraction des paires de correspondances candidates

Align-Match compare toutes les correspondances de A_1 avec toutes celles de A_2. Par conséquent, l'alignement candidat qu'il construit est composé de toutes les paires possibles de correspondances de A_1 et de A_2, respectivement. Ces paires de correspondances sont dites « correspondances en sortie candidate ». Chaque correspondance en sortie candidate est de la forme {<{$name_1$, $label_1$, $comment_1$}, ant_1, $cons_1$, r_1, n_1>, <{$name_2$, $label_2$, $comment_2$}, ant_2, $cons_2$, r_2, n_2>} tel que :

- Ant= {$e_{11}, e_{12},\ldots, e_{1m}$};
- Cons={$e_{21}, e_{22},\ldots, e_{2k}$};
- <{$name_1$, $label_1$, $comment_1$}, ant_1, $cons_1$, r_1, n_1> $\in A_1$;
- <{$name_2$, $label_2$, $comment_2$}, ant_2, $cons_2$, r_2, n_2> $\in A_2$.

Exemple. Si nous considérons l'exemple précédent, l'alignement candidat sera composé de 8×6 soit 48 correspondances puisque A1 est défini avec 8 correspondances et A2 comporte 6 correspondances.

2.2.3 Ingénierie des caractéristiques

Dans ce cas, les types de caractéristiques, composant les correspondances des alignements en entrée, suivants sont exploités:

- **Les caractéristiques d'identification**. L'ensemble de caractéristiques d'identification considéré correspond à l'ensemble {name, label, comment}.
- **Les caractéristiques de composition.** Les caractéristiques de composition d'une correspondance sont ses ensembles Ant et Cons.

2.2.4 Composition et exécution des techniques de base

Afin de déterminer la similarité entre les différentes caractéristiques, nous avons utilisé plusieurs techniques de base. Chaque technique est appliquée sur une caractéristique particulière des correspondances, à savoir : leurs noms, leurs étiquettes, leurs commentaires,

leurs ensembles Ant et leurs ensembles Cons. Afin de couvrir toutes ces caractéristiques, ces techniques de base doivent donc être composées.

Dans l'approche proposée, les calculs de la similarité sont indépendants c'est-à-dire que la similarité de chaque caractéristique n'influence la similarité d'aucune autre caractéristique. Par conséquent, la technique de composition adoptée est la composition parallèle comme définie dans (Euzenat & Shvaiko, 2007) telle que chaque technique de base est exécutée, indépendamment, sur une caractéristique particulière.

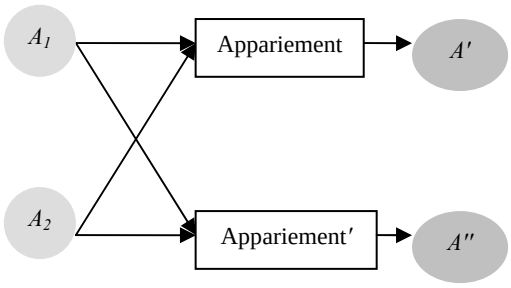


Figure 2. Composition parallèle de techniques de base d'appariement d'alignements

Les techniques de base utilisées sont divisées en deux catégories : la première catégorie regroupe les techniques utilisées pour mesurer la similarité des caractéristiques d'identification tandis que la deuxième concerne les techniques définies pour estimer la similarité des caractéristiques de composition.

2.2.5.1 Techniques de base pour mesurer la similarité des caractéristiques d'identification

A l'instar de la définition d'un concept dans une ontologie, le nom et les étiquettes d'une correspondance sont des chaînes de caractères courtes tandis que ses commentaires sont considérés comme des chaînes de caractères longues.

Dans (Bach, 2006), l'auteur définit un ensemble d'algorithmes permettant de calculer la similarité entre les concepts de deux ontologies en exploitant leurs noms, leurs étiquettes ainsi que leurs commentaires. Par conséquent, nous nous sommes inspiré principalement de cet ensemble d'algorithmes pour calculer la similarité des caractéristiques d'identification de deux correspondances composant une correspondance en sortie candidate.

Afin d'exploiter le maximum d'informations contenues dans les composantes d'identification, l'évaluation de la similarité est définie différemment selon la nature de chaque composante, i.e., une chaîne de caractères courte ou longue (Bach, 2006). Cependant, elle suit généralement un processus en deux étapes :

- La première étape consiste en l'application des techniques linguistiques sur les différents composants des caractéristiques d'identification des deux correspondances ;

- La deuxième étape applique des mesures de similarité sur les différentes composantes des caractéristiques d'identification.

A. Similarité des noms de deux correspondances

Le nom d'une correspondance est généralement une chaîne de caractères sans espace qui peut être un mot, un terme, ou une expression (une combinaison de mots).

A.1 Normalisation des noms de deux correspondances

L'algorithme adopté pour normaliser le nom d'une correspondance consiste premièrement à appliquer un tokenizer qui reconnaît la ponctuation, la casse (majuscule) et les chiffres. Le résultat de cette tokenization est un ensemble de tokens sauvegardé dans un vecteur. Ensuite, la normalisation procède à minuscullariser (rendre minuscule) tous les tokens contenus dans les vecteurs de tokens de noms.

Algorithme 1 Normalisation (nom)

```
résultat <- créer un vecteur vide
tokens <- Tokenization (nom)
n <- nombre de tokens dans tokens
Pour i de 1 à n
ex <- Minusculisation(tokens[i])
résultat <- Ajouter(ex, résultat)
Fin Pour
Retourner résultat
```

A.2. Calcul de la similarité des noms de deux correspondances

En adhérant à l'étude comparative réalisée par (Cohen, Ravikumar, & Fienberg, 2003) sur les mesures de similarités basées sur les chaînes de caractères, la meilleure mesure de similarité qui peut être appliquée sur deux chaînes de caractères courtes semble être celle de Jaro-Winkler. En utilisant cette mesure, la similarité entre deux noms (σ_{Nom}) est définie dans (Bach, 2006) comme la moyenne des valeurs de similarité entre chaque token d'un vecteur et le token le plus similaire dans l'autre vecteur.

Définition (Similarité des noms). *Soient n_1 et n_2 deux noms de deux correspondances. Soient N_1 et N_2 deux ensembles des tokens obtenus après la normalisation de n_1 et de n_2, respectivement. La similarité des noms est une fonction de la similarité $\sigma_{Nom} : S \times S \rightarrow [0, 1]$ telle que :*

$$\sigma_{Nom}(n_1, n_2) = \frac{\sum_{m_1 \in N_1} MJW(m_1, N_2) + \sum_{m_2 \in N_2} MJW(m_2, N_1)}{|N_1| + |N_2|}$$

Où m_i est le i-ème token de N, $MJW(m_i, N) = max_{m_j \in N}\, \sigma_{Jaro-Winker}(m_i, m_j)$ et $|N_i|$, i = 1, 2, est la cardinalité de l'ensemble N_i

L'algorithme permettant de calculer $MJW(token, tokens)$est donné par (Bach, 2006) comme suit :

Algorithme 2 Sim_Max_Dans_Ensemble(token, tokens)

```
valeur_max <- 0
n <- nombre de tokens dans tokens
Pour i de 1 à n
valeur_sim <- Mesure_Jaro_Winkler(token, tokens[i])
Si valeur_sim > valeur_max
   valeur_max <- valeur_sim
Fin Si
Fin Pour
Retourner valeur_max
```

L'algorithme qui calcule la similarité entre deux noms est aussi donné comme suit :

Algorithme 3 Similarité_des_Noms (nom1, nom2)

```
tokens1 <- Normalisation(nom1)
tokens2 <- Normalisation(nom2)
n1 <- nombre de tokens dans tokens1
n2 <- nombre de tokens dans tokens2
somme1 <- 0
Pour i de 1 à n1
sim <- Sim_Max_Dans_Ensemble(tokens1[i], tokens2)
somme1 <- somme1 + sim
Fin Pour
somme2 <- 0
Pour j de 1 à n2
sim <- Sim_Max_Dans_Ensemble(tokens2[j], tokens1)
somme2 <- somme2 + sim
Fin Pour
Si n1 = n2 = 0
résultat <- 1
Sinon
résultat <- (somme1 + somme2) / (n1 + n2)
Fin Si
Retourner résultat
```

B. Similarité des étiquettes de deux correspondances

Puisque le nom d'une correspondance est unique dans un alignement, on peut associer à la même correspondance zéro, une ou plusieurs étiquettes. Cela permet, par exemple, d'avoir plusieurs versions lisibles dans plusieurs langues naturelles pour la correspondance.

B.1. Normalisation des étiquettes

L'algorithme adopté pour normaliser les étiquettes d'une correspondance est similaire à celui utilisé pour normaliser son nom. Il consiste à appliquer le même tokenizer utilisé dans la normalisation des noms. A la différence de la tokenization du nom, le résultat sera un ensemble de vecteurs (probablement vide) où chaque vecteur représente l'ensemble des tokens d'une étiquette particulière. Ensuite, comme dans la normalisation des noms, l'algorithme procède à minuscullariser (rendre minuscule) tous les tokens contenus dans les vecteurs de tokens des étiquettes.

Algorithme 4 Normalisation (étiquettes)

```
n ← nombre d'étiquettes
résultat ← créer un vecteur vide
Pour i de 1 à n
résultat [i] ← créer un vecteur vide
tokens ← Tokenization (étiquette)
m ← nombre de tokens dans tokens
Pour j de 1 à m
ex ← Minusculisation (tokens[j])
résultat[i] ← Ajouter(ex, résultat[i])
Fin Pour
Fin Pour
Retourner résultat
```

B.2. Calcul de la similarité de deux ensembles d'étiquettes de deux correspondances

Le calcul de la similarité entre deux ensembles d'étiquettes est obtenu par extension du calcul de la similarité des noms (Bach, 2006). Pour chaque étiquette d'une correspondance, l'étiquette la plus similaire (ayant la valeur de similarité calculée par σ_{Nom} la plus élevée de l'autre correspondance) est recherchée. La valeur de la similarité des étiquettes est la valeur moyenne de toutes les valeurs de similarité des noms de paires d'étiquettes trouvées. L'adaptation de la définition formelle donnée par (Bach, 2006) de cette similarité pour les correspondances est comme suit.

Définition (Similarité des étiquettes). *Soit $\mathcal{L}_1$ et $\mathcal{L}_2$ deux ensembles d'étiquettes de deux correspondances de deux alignements. La similarité des étiquettes est une fonction de la similarité $\sigma_{Étiquette} : 2^E \times 2^E \rightarrow [0, 1]$ telle que :*

$$\sigma_{étiquette}(\mathcal{L}_1, \mathcal{L}_2) = \frac{\sum_{L_1 \in \mathcal{L}_1} MNS(L_1, \mathcal{L}_2) + \sum_{L_2 \in \mathcal{L}_2} MNS(L_2, \mathcal{L}_1)}{|\mathcal{L}_1| + |\mathcal{L}_2|}$$

Où L_i est la i-ème étiquette de $\mathcal{L}$;

$MNS(L_i, \mathcal{L}) = max_{L_j \in \mathcal{L}} \sigma_{Nom}(L_i, L_j)$ et $|\mathcal{L}_i|$, i = 1, 2, est la cardinalité de l'ensemble $\mathcal{L}_i$.

L'algorithme permettant de calculer $MSN(étiq, étiqs)$ est donné par (Bach, 2006) comme suit :

Algorithme 5 Sim_Max_Dans_Ensemble_Etiq(etiq, etiqs)

```
valeur_max <- 0
n <- nombre d'étiquettes dans etiqs
Pour i de 1 à n
valeur_sim <- Similarité_des_Noms(etiq, etiqs[i])
Si valeur_sim > valeur_max
valeur_max <- valeur_sim
Fin Si
Fin Pour
Retourner valeur_max
```

Le calcul de la similarité entre deux ensembles d'étiquettes de deux correspondances est aussi donné par l'algorithme suivant :

Algorithme 6 Similarité_des_Etiquettes (Ens_etiqs1, Ens_etiqs2)

```
etiqs1 <- Normalisation(Ens_etiqs1)
etiqs2 <- Normalisation(Ens_etiqs2)
n1 <- nombre d'étiquettes dans etiqs1
n2 <- nombre d'étiquettes dans etiqs2
somme1 <- 0
Pour i de 1 à n1
sim <- Sim_Max_Dans_Ensemble_Etiq(etiqs1[i], etiqs2)
somme1 <- somme1 + sim
Fin Pour
somme2 <- 0
Pour j de 1 à n2
sim <- Sim_Max_Dans_Ensemble_Etiq(etiqs2[j], etiqs1)
somme2 <- somme2 + sim
Fin Pour
Si n1 = n2 = 0
```

```
résultat <- 1
Sinon
résultat <- (somme1 + somme2) / (n1 + n2)
Fin Si
Retourner résultat
```

C. Similarité des commentaires de deux correspondances

Les commentaires ont pour objectif de fournir des informations descriptives d'une correspondance. Similairement aux étiquettes, on peut associer à la même correspondance zéro, une ou plusieurs commentaires.

C.1. Normalisation des commentaires

La normalisation des commentaires d'une correspondance est effectuée en quatre étapes. Premièrement, dans le cas où la correspondance est définie avec plusieurs commentaires, Align-Match concatène ces derniers en un seul commentaire. Puis, il procède à la suppression des mots vides. Ensuite, il applique le même tokenizer utilisé pour les noms. Le résultat de cette tokenization est un ensemble de tokens sauvegardé dans un vecteur. Enfin, une minusculisation est appliquée sur tous les tokens contenus dans le vecteur de tokens.

Algorithme 7 Normalisation (commentaires)

```
n ← nombre de commentaires
comt<- créer une chaîne de caractères vide
Pour i de 1 à n
comt_intermédiaire <- chaîne de caractères vide
comt_intermédiaire <- extraire le i-ème commentaire
comt <- comt.comt_intermédiaire
Fin Pour
Supprimer_Mots_Vides(cmt)
tokens <- Tokenisation(cmt)
m ← nombre de tokens dans tokens
Pour j de 1 à m
ex ← Minusculisation (tokens[j])
résultat[j] ← Ajouter(ex, résultat[j])
Fin Pour
Retourner résultat
```

C.2. Calcul de la similarité

Ici, nous adaptons la procédure de calcul de la similarité des commentaires de deux concepts définie dans (Bach, 2006) comme suit:

- Les correspondances sont considérées comme des documents ;
- Les mots dans les commentaires d'une correspondance sont des mots du document ;
- L'univers des documents est l'univers des correspondances qui est construit avec toutes les correspondances des deux alignements ;
- Chaque correspondance est associée à un vecteur ; Les dimensions de ces vecteurs sont calculées par la technique TF/IDF avec l'univers des correspondances et les mots des commentaires.

Les algorithmes permettant de construire l'univers de correspondances est comme suit.

Algorithme 8 Construire_Univers (align1, align2, U)

```
n1 <- nombre de correspondances dans l'alignement align1
n2 <- nombre de correspondances dans l'alignement align2
Pour i de 1 à n1
une_correspondance <- prendre la i-ème correspondance de
l'alignement align1
tokens1<-normalisation(commentaires de une_ correspondance)
n_tokens1 <- nombre de tokens dans tokens1
Pour j de 1 à n_tokens
Si l'univers U ne contient pas le token tokens1[j]
U <- Ajouter(tokens1[j], U)
Fin Si
Fin Pour
Fin Pour
Pour k de 1 à n2
   une_ correspondance <- prendre la k-ème correspondance de
   l'alignement align2
tokens2<-normalisation(commentaires de une_ correspondance)
n_tokens2 <- nombre de tokens dans tokens2
Pour l de 1 à n_tokens2
Si l'univers U ne contient pas le token tokens2[l]
U <- Ajouter(tokens2[l], U)
Fin Si
Fin Pour
Fin Pour
Retourner U
```

Pour chaque correspondance, l'algorithme de construction d'un vecteur qui lui correspond est donné comme suit.

Algorithme 9 Construire_Vecteur(correspondance, U)

```
S <- nombre de mots distincts dans l'univers U
vecteur <- créer une matrice de S éléments
N <- nombre de correspondances dans les deux alignements
```

```
Pour i de 1 à S
n <- nombre de correspondances qui contiennent le mot U[i] au moins une fois
idf <- log2(N/n)
tf <- nombre de fois où le i-ème mot dans l'univers U apparaît dans le commentaire de la correspondance
vecteur[i] <- tf * idf
Fin Pour
Retourner vecteur
```

Par conséquent, le calcul de la similarité entre deux commentaires de deux correspondances est effectué par une distance entre les deux vecteurs qui les représentent. L'adaptation de la similarité des commentaires de deux concepts de (Bach, 2006) pour s'appliquer à deux correspondances est la suivante.

Définition (Similarité des commentaires de deux correspondances). *Soit* $v_i = (w_{i1}, w_{i2}, \ldots w_{iS})$ *et* $v_j = (w_{j1}, w_{j2}, \ldots w_{jS})$ *deux vecteurs représentant deux commentaires de deux correspondances appartenant à deux alignements. La similarité des commentaires est une fonction de la similarité (coefficient de cosinus)* $\sigma_{Commentaire}$ *:* $\mathcal{V} \times \mathcal{V} \rightarrow [0, 1]$ *telle que :*

$$\sigma_{Commentaire} = \frac{\sum_{k=1}^{S} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{S} (w_{ik})^2 \times \sum_{k=1}^{S} (w_{jk})^2}}$$

L'algorithme qui permet le calcul de la similarité des commentaires de deux correspondances est le suivant.

Algorithme 10 Similarité_des_Commentaires(correspondance1, correspondance2)

```
U <- créer un ensemble vide
U <- Construire_Univers(align1, align2, U)
vecteur1 <- Construire_Vecteur(correspondance1, U)
vecteur2 <- Construire_Vecteur(correspondance2, U)
S <- nombre de mots distincts dans l'univers U
somme1 <- 0
somme2 <- 0
somme3 <- 0
Pour i de 1 à S
somme1 <- somme1 + vecteur1[i] * vecteur2[i]
somme2 <- somme2 + vecteur1[i] * vecteur1[i]
somme3 <- somme3 + vecteur2[i] * vecteur2[i]
```

```
Fin Pour
résultat <- somme1 / racine(somme2 + somme3)
Retourner résultat
```

2.2.5.2 Techniques de base pour mesurer la similarité des caractéristiques de composition

L'idée utilisée pour calculer la similarité entre deux caractéristiques de composition de même type, i.e., ant_1 et ant_2 ou $cons_1$ et $cons_2$, pour deux correspondances est de contempler l'ensemble des entités appartenant à chacune de ces caractéristiques comme une concaténation de chaînes de caractères.

Par conséquent, nous proposons, pour mesurer la similarité entre deux caractéristiques de composition, d'adapter les mesures de similarité syntaxiques. Cependant, l'adaptation des mesures de similarité syntaxiques doit respecter les deux contraintes suivantes :

- La comparaison doit être faite sur des entités de même type ;
- Dans une caractéristique de composition d'une correspondance, l'ordre des entités n'est pas important. Par exemple, l'ensemble *Ant* d'une correspondance *c* est {lastname, firstname} et celui de la correspondance *c′* est {firstname, lastname}. La mesure de similarité appliquée sur ces deux ensembles doit retourner la valeur 1.

Par conséquent, certaines mesures de similarité syntaxiques ne peuvent être adaptées pour mesurer la similarité des caractéristiques de composition. En particulier, ce sont celles qui ne prennent pas en compte la position des caractères dans les chaînes telles que l'égalité des chaînes de caractères et la similarité de sous-chaînes.

Des exemples de mesures qui peuvent être adaptées pour calculer la similarité entre deux caractéristiques de composition sont la distance de Hamming et la similarité de Jaccard.

A. Distance de Hamming adaptée pour comparer deux caractéristiques de composition

La première mesure que nous proposons d'adapter pour mesurer la similarité entre deux caractéristiques de composition est la distance de Hamming. Dans sa conception originale, cette distance calcule le nombre de positions dans lesquelles deux chaînes de caractères diffèrent. Elle a été également adaptée pour mesurer la distance entre deux ensembles d'instances de deux ontologies.

L'adaptation consiste à calculer le nombre de concepts, de relations et d'instances différents entre les deux caractéristiques normalisé par l'union des ensembles de concepts additionnée avec l'union des ensembles de relations et l'union des ensembles d'instances. La définition formelle que nous donnons est comme suit.

Définition (Distance de Hamming adaptée pour les caractéristiques de composition). *Soit E l'ensemble des caractéristiques de composition. La distance de Hamming entre deux*

caractéristiques de composition est une fonction de dissimilarité $\delta_{Hamming\_Composition} : 2^E \times 2^E \rightarrow R$ *tel que* $\forall x, y \subseteq E$ *:*

$$\delta_{Hamming\_Composition}(x, y) = \frac{\sum_{E_x, E_y \in \varphi} |E_x \cup E_y - E_x \cap E_y|}{\sum_{E_x, E_y \in \varphi} |E_x \cup E_y|}$$

où E_x *est l'ensemble des entités de même type, i.e., tous les éléments de* E_x *sont soit des concepts, des relations ou des instances, apparues dans x et* E_y *est l'ensemble des entités de même type apparues dans y.*

Puisque la distance de Hamming est une mesure normalisée, donc la mesure de similarité de Hamming $\sigma_{Hamming\_Composition}$ est obtenue par: $\sigma_{Hamming\_Composition} = 1-\delta_{Hamming\_Composition}$.

B. Similarité de Jaccard adaptée pour comparer deux caractéristiques de composition

La similarité de Jaccard peut être également étendue pour comparer deux caractéristiques de composition. Nous donnons la définition suivante.

Définition (Similarité de Jaccard adaptée pour les caractéristiques de composition). *Soit E l'ensemble des caractéristiques de composition. La similarité de Jaccard entre deux caractéristiques de composition est une fonction de similarité* $\sigma : 2^E \times 2^E \rightarrow R$ *tel que* $\forall x, y \subseteq E$ *:*

$$\sigma(x, y) = \frac{\sum_{E_x, E_y \in \varphi} |E_x \cap E_y|}{\sum_{E_x, E_y \in \varphi} |E_x \cup E_y|}$$

où E_x *est l'ensemble des entités de même type apparues dans x et* E_y *est l'ensemble des entités de même type apparues dans y.*

La mesure utilisée dans Align-Match est la similarité de Jaccard adaptée pour les caractéristiques de composition puisqu'elle est plus rapide à calculée que celle de Hamming.

Maintenant, il reste à savoir comment calculer les ensembles $E_x \cap E_y$, $E_x \cup E_y$?

C. Calcul de $E_x \cap E_y$, $E_x \cup E_y$

L'intersection de deux ensembles E_x and E_y est l'ensemble qui contient tous les éléments de E_x qui appartiennent également à E_y (ou d'une manière équivalente, tous les éléments de E_y qui appartiennent également à E_x). Ainsi, si un concept c_1 appartenant à E_x (ou E_y, respectivement) est similaire à un concept c_2 appartenant à E_y (ou E_x, respectivement) alors $E_x \cap E_y$ doit contenir soit c_1 ou c_2. De plus, si un concept c_3 appartenant à E_x (E_y, respectivement) n'est similaire à aucun concept de E_y alors $E_x \cap E_y$ ne doit pas contenir c_3.

L'union des ensembles E_x et E_y est l'ensemble qui contient tous les éléments de E_x et tous les éléments de E_y.

Deux éléments de deux ensembles E_x et E_y sont similaires si leurs noms sont similaires. La similarité des noms de ces deux éléments est calculée de la même manière que dans la similarité des noms de correspondances.

2.2.5 Agrégation des valeurs de similarité

Dans l'étape précédente, le calcul de la similarité des caractéristiques d'identification a produit trois valeurs de similarité $\{\sigma_{\text{Nom}}, \sigma_{\text{étiquette}}, \sigma_{\text{commentaire}}\}$. Tandis que celui des caractéristiques de composition a fourni deux valeurs de similarité $\{\sigma_{\text{Ant}}, \sigma_{\text{Cons}}\}$. L'objectif de cette étape est d'agréger chacun de ces deux ensembles de valeurs similarité afin de fournir une seule valeur de similarité représentative pour chaque type de caractéristiques d'une paire de correspondances.:

Afin d'agréger $\{\sigma_{\text{Nom}}, \sigma_{\text{étiquette}}, \sigma_{\text{commentaire}}\}$ et $\{\sigma_{\text{Ant}}, \sigma_{\text{Cons}}\}$ de deux correspondances c_1 et c_2, Align-Match adopte une technique d'agrégation semi-automatique qui est basée sur les méthodes suivantes:

La somme pondérée :

$$\sigma_{\text{identification}}(c_1, c_2) = w_1 \times \sigma_{\text{Nom}}(c_1, c_2) + w_2 \times \sigma_{\text{étiquette}}(c_1, c_2) + w_3 \times \sigma_{\text{commentaire}}(c_1, c_2)$$

$$\sigma_{\text{composition}}(c_1, c_2) = w_4 \times \sigma_{\text{Ant}}(c_1, c_2) + w_5 \times \sigma_{\text{Cons}}(c_1, c_2)$$

Le produit pondéré :

$$\sigma_{\text{identification}}(c_1, c_2) = \sigma_{\text{Nom}}(c_1, c_2)^{w_1} + \sigma_{\text{étiquette}}{}^{w_2} + \sigma_{\text{commentaire}}(c_1, c_2)^{w_3}$$

$$\sigma_{\text{composition}}(c_1, c_2) = \sigma_{\text{Ant}}(c_1, c_2)^{w_4} + \sigma_{\text{Cons}}(c_1, c_2)^{w_5}$$

La moyenne pondérée :

$$\sigma_{\text{identification}}(c_1, c_2) = \frac{w_1 \times \sigma_{\text{Nom}}(c_1, c_2) + w_2 \times \sigma_{\text{étiquette}}(c_1, c_2) + w_3 \times \sigma_{\text{commentaire}}(c_1, c_2)}{\sum_{i=1}^{3} w_i}$$

$$\sigma_{\text{composition}}(c_1, c_2) = \frac{w_4 \times \sigma_{\text{Ant}}(c_1, c_2) + w_5 \times \sigma_{\text{Cons}}(c_1, c_2)}{\sum_{i=4}^{5} w_i}$$

Tel que w_i sont des poids spécifiés par l'utilisateur afin de décider l'importance d'une caractéristique par rapport aux autres.

2.2.6 Interprétation

Le calcul de similarité ainsi mené produit un ensemble important de « correspondances en sortie ». Afin d'en extraire un alignement de qualité satisfaisante, on adopte une méthode d'extraction d'alignements semi-automatique basé sur le seuil. L'utilisateur choisit le seuil approprié, puis toutes les correspondances ayant une valeur de similarité au dessous de ce seuil seront supprimées.

2.2.7 Les sorties

La sortie est un alignement simple constitué d'une liste de correspondances entre A_1 et A_2. Chaque correspondance en sortie est un 9-uple $\langle id, idA_1, idA_2, \sigma_{identification}, \sigma_{composition}, r_1, r_2, n_1, n2\rangle$ tel que :

- Id est l'identifiant de la correspondance en sortie. Cet identifiant est unique dans l'alignement en sortie;
- idA_1 (idA_2, respectivement) est l'identifiant de la correspondance alignée de A_1 (de A_2, respectivement)
- $\sigma_{identification}$ ($\sigma_{composition}$, respectivement) est la valeur de similarité des caractéristiques d'identification (de composition, respectivement) des correspondances idA_1 et idA_2 ;
- r_1 (r_2, respectivement) est la relation contenue dans la correspondance idA_1(idA_2, respectivement) ;
- n_1 (n_2, respectivement) est le degré de confiance de la relation contenue dans la correspondance idA_1(idA_2, respectivement).

L'alignement produit est de multiplicité * : *, i.e., une correspondance de A_1 peut avoir aucune, une ou plusieurs correspondances de A_2, et réciproquement.

3 Conclusion

Dans ce chapitre, nous avons développé une approche permettant de comparer aussi bien des alignements simples que complexes. La motivation principale de cette approche était de pallier au problème d'évaluation de systèmes d'appariement produisant des alignements complexes. De plus, la même approche peut être aussi utilisée dans un autre contexte qui est celui de la déduction d'alignements à partir d'alignements existants. En effet, Align-Match peut intervenir pour trouver l'alignement M entre deux alignements existants, M1 et M2 lesquels sont obtenus respectivement à partir des ontologies O1 et O2 et P1 et P2. Selon (InterOp, 2008), il doit être possible de créer automatiquement un alignement entre O1 et P1, O2 et P2, etc., à partir de l'alignement M.

Toutefois, la validation permettant de prouver le bon fonctionnement d’Align-Match n’a pas été encore réalisée, nous envisageons, donc, de le faire.

CONCLUSION ET PERSPECTIVES

Dans le cadre du Web sémantique, ce mémoire a présenté une classification des techniques d'appariement de base en partant des niveaux de complexité sémantique associés aux différents langages du web sémantique. Cette classification fournit donc un guide d'utilisation pour apparier les ontologies du Web.

Nous avons également proposé un nouveau processus d'appariement d'ontologies qui est le résultat d'une étude théorique sur les différentes explicitations recensées dans la littérature. Cette explicitation apparaît plus complète que toutes celles déjà proposées.

Une approche contribuant au processus d'évaluation des outils d'appariement est également proposée. Cette approche permet de découvrir les correspondances entre des alignements contenant des correspondances exprimées entre concepts, relations et instances. Les alignements qui prennent en considération d'autres entités des ontologies, telles que les axiomes et les règles, ne sont pas considérés par cette approche. Par conséquent, la suite de ce travail s'inscrit dans le cadre d'une extension d'Align-Match en prenant en compte les correspondances exprimées entre toutes les entités possibles des ontologies après une validation permettant de prouver le bon fonctionnement de sa version actuelle.

Les techniques qui seront utilisées dans ce but sont celles exploitant le niveau de similarité le plus haut : les règles. Selon (Ehrig, 2007), aucune technique n'a été proposée dans ce sens. Nous envisageons donc d'exploité ce niveau de complexité sémantique à la fois pour étendre l'approche de comparaison d'alignements et pour proposer un algorithme d'appariement d'ontologies prenant en compte le niveau des règles.

Bibliographie

Aleksovski, Z., Klein, M., Kate, W. t., & Harmelen, F. v. (2006). Matching unstructured vocabularies using a background ontology. Dans Praha(CZ) (Éd.), *15th Intenational Conference on Knowledge Engineering and Knowledge Management (EKAW). 4248*, pp. 182-197. Lecture notes in computer science.

Bach, T. L., Dieng-Kuntz, R., & Gandon, F. (2004). On Ontology Matching Problems (for building a corporate Semantic Web in a multi-communities organization). *Proceedings of ICEIS 2004*, pp. 14-17. Porto, Portugal.

Bach, T. L. (2006). *Construction d'un Web sémantique multi-points de vue*. Thèse de doctorat, École des Mines de Nice à Sophia Antipolis.

Baneyx, A. (2007). *Construire une ontologie de la pneumologie: Aspects théoriques, modèles et expérimentations.* Thèse de doctorat, Université Paris 6.

Benayache, A. (2005). *Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : LE PROJET MEMORAE.* Thèse de doctorat, université de technologie de compiegne.

Berners-Lee, T. (2002). *Prepare for Next-Gen Web Now.* Von http://boston.internet.com/news/article.php/2001_1013111.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *In Scientific Americai*, pp. 20-88.

Borst, W. (1997). *Construction of Engineering Ontologies.* In Center for Telematica and Information Technology, University of Tweenty, Enschede, NL.

Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krotzsch, M., et al. (2004). *Specification of a common framework for characterizing alignment.* Deliverable D2.2.1, Knowledge web NoE.

Budanitsky, A., & Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics , 32* (1), pp.13-47.

Carrillo Ramos, A. C. (2007). *Agents ubiquitaires pour un accès adapté aux systèmes d'information : Le Framework PUMAS.* Thèse de doctorat, Université Joseph Fourier.

Castano, S., Ferrara, A., Hess, G. N., & Montanelli, S. (2007). *State of the Art on Ontology Coordination and Matching.* deliverable V.1.0, Bootstrapping Ontology Evolution with Multimedia Information.

Chamoun, M. (2006). *Intégration de l'Internet 3G au sein d'une plateforme active.* Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications de Paris.

Cohen, W., Ravikumar, P., & Fienberg, S. (2003). A comparison of string metrics for matching names and records. *Dans Proc. KDD-2003 Workshop on Data Cleaning and Object Consolidation*, pp. 73-78. Washington (DC US).

Cox, T. F., & Cox, M. A. (1994). Multidimensional Scaling. *Chapman and Hall* .

Dieng, R., & Hug, S. (1998). Comparison of "personal ontologies" represented through conceptual graphs. *In Proc. 13th ECAI,*, pp. 341–345. Brighton (UK).

Dieng, R., Corby, O., Giboin, A., & Ribière, M. (1998). Methods and tools for corporate knowledge management. *In Proceedings of the 11th workshop on Knowledge Acquisition, Modeling and Management (KAW' 98)*, pp. 17–23. Banff, Canada.

Do, H.-H., & Rahm, E. (2002). Coma - a system for flexible combination of schema matching approaches. *Dans Proc. 28ième Internationnal Conference on Very Large Data Bases (VLDB)*, pp. 610–621. Hong Kong (CN).

Do, H.-H., & Rahm, E. (2002). Coma - a system for flexible combination of schema matching approaches. *Dans Proc. 28ième Internationnal Conference on Very Large DaVLDB*, pp. 610–621.

Ehrig, M., & Sure, Y. (2004). Ontology mapping – an integrated approach. In J. D. In Christoph Bussler (Hrsg.), *Proc. 1st ESWS. volume 3053 of Lecture Notes in Computer Science*, p.p. 76–91. Hersounisous (GR): Springer Verlag.

Ehrig, M. (2007). *Ontology alignement: bridging the semantic gap*. New-York (NY US): Semantic web and beyond: computing for human experience. Springer.

Eklöf, M., & Martenson, C. (2006). *Ontology Interoperability*. Rapport de recherche, FOI Swedish Defence Research Agency.

Euzenat, J., Bach, T., Barrasa, J., Bouquet, P., Bo, J., Dieng-Kuntz, R., et al. (2004). *State of the art on ontology alignment.* Knowledge web NoE. Knowledge web NoE.

Euzenat, J., & Valtchev, P. (2004). Similarity-based ontology alignment in OWL-lite. *In Proc. 15th ECAI.* Valencia (ES).

Euzenat, J., Bach, T. L., Dieng, R., Duc, C. L., Napoli, A., Zimmermann, A., et al. (2007). *Knowledge web 2.2: Heterogeneity in the semantic web.* Rapport technique, NoE Knowledge Web project.

Euzenat, J., Mocan, A., & Scharffe, F. (2007). *Ontology alignements: An ontology management perspective.*

Euzenat, J., & Shvaiko, P. (2007). *Ontology matching.* Springer.

Fensel, D. (2001). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce.* Springer.

Fürst, F. (2002). *Ingénierie ontologique.* Rapport de Recherche, Institut de Recherche en Informatique de Nantes.

Fürst, F. (2004). *Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationalisation.* Thèse de doctorat de l'université de Nantes.

Gal, A., Anaby-Tavor, A., Trombetta, A., & Montesi, D. (2005). A framework for modeling and evaluating automatic semantic reconciliation. *VLDB Journal , 14* (1), pp. 50-67.

Gandon, F. (2002). *Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web.* thèse de doctorat, INRIA et université de Nice - Sophia Antipolis.

Garf, B. (1996). *Lexique de philosophie.* Paris: Edition du Seuil.

Giunchiglia, F., & Shvaiko, P. (2003). Semantic matching. *Dans Proc. IJCAI Workshop on ontologies and distributed systems*, pp. 139-146. Acapulco (MX).

Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004). S-Match: an algorithm and an implementation of semantic matching. *Dans Proceedings ESWS*, pp. 61–75. Heraklion (GR).

Goh, C. H. (1997). *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources.* Thèse de doctorat, MIT, Cambridge (MA US).

Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* (5), pp.199-220.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases, towards a terminological clarification. *Towards very large knowledge bases : knowledge building and knowledge sharing*, pp. 25-32. Dans N. Mars, eds.

Hakimpour, F. (2003). *Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata.* Thèse de doctorat, Faculté de mathématique et d'informatique de l'université de Zürich.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal* (29), pp.147–160.

Hernandez, N. (2006). *Ontologies de domaine pour la modélisation du contexte en recherche d'information.* Thèse de doctorat, Université Paul Sabatier de Toulouse.

InterOp, N. (2008). *Ontology Interoperability.* WP8, subtask 3 State of the Art Report .

Izza, S. (2006). *Intégration des systèmes d'information industriels: Une approche flexible basée sur les services sémantiques.* thèse de doctorat, Ecole Nationale Supérieure des Mines de Saint-Etienne.

Jaro, M. (1989). Advances in record-linkage methodology as applied to matching the census of Tampa. *Journal of the American Statistical Association* , 84(4), pp.14-420.

Jiang, J., & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *Dans Proceedings on International Conference on Research in Computational Linguistics.*

Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review , 18* (1), pp.1–31.

Kassel, G. (2002). OntoSpec : une méthode de spécification semi-informelle d'ontologies. *Dans Actes des journées francophones d'Ingénierie des Connaissances (IC'2002)*, (pp. 75-87).

KIF. (2004). *KIF, Knowledge Interchange Format Home Page*. Von http://logic.stanford.edu/kif/dpans.html

Klein, M. (2001). Combining and relating ontologies : an analysis of problems and solutions. *Dans Proceedings de IJCAI Workshop on Ontologies and Information Sharing.* Seattle (WA US).

Knuth, D., & Bendix, P. (1970). Simple word problems in universal algebras. *Computational Problems in Abstract Algebra* , pp. 263-297.

Lapique, F. (2006). Le langage d'ontologie Web OWL. *Ecole polytechnique fédérale de Lausanne, Flash Informatique (FI), FI 8/06* .

Larson, J., Navathe, S., & Elmasri, R. (1989). A theory of attributed equivalence in databases with application to schema integration. *IEEE Transactions on Software Engineering , 15* (4), pp.449-463.

Laublet, P., Reynaud, C., & Charlet, J. (2002). Sur Quelques Aspects du Web Sémantique. *Actes des deuxièmes assises nationales du GdRI3*, (pp. 59-78).

Levenshtein, V. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Doklady akademii nauk SSSR , 163* (4), pp.845-848.

Lin, D. (1998). An information-theoretic definition of similarity. *In Proceedings of the 15th International Conference on Machine Learning*, (pp. 296-304). Madison (WI US).

Luong, P. H. (2007). *Gestion de l'évolution d'un Web Sémantique d'entreprise.* thèse de doctorat, l'Ecole des Mines de Paris.

Mädche, A., & Staab, S. (2002). Measuring similarity between ontologies. *In Proc. Of the 13th Int. Conference on Knowledge Engineering and Management (EKAW-2002).* Siguenza, Spain: Springer-Verlag.

Maynard, D., & Ananiadou, S. (1999). Term extraction using a similarity-based approach. *Dans Recent Advances in Computational Terminology*. John Benjamins.

Mellal, N. (2007). *Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information.* Thèse de doctorat, Polytech'Savoie.

Melnik, S., Garcia-Molina, H., & Rahm, E. (2001). *Similarity Flooding: A Versatile Graph Matching Algorithm.* Extended Technical Report, http://dbpubs.stanford.edu/pub/2001-25.

Mhiri, S., & Despres, S. (2005). Intégration d'ontologies : Recherche exploratoire et expérimentation d'outils d'alignement d'ontologies au format OWL. *7ème Journées Doctorales Informatique et Réseau.* Université de Technologie de Troyes.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM , 38* (11), pp.39- 41.

Mizuguchi, R. (2003). Tutorial on ontological engineering - Part 1: Introduction to Ontological Engineering. *New Generation Computing , 21* (4), pp.365 – 384.

Monge, A., & Elkan, C. (1997). An efficient domain-independent algorithm for detecting approximately duplicate database records. *SIGMOD Workshop on Data Mining and Knowledge Discovery*. Tucson (AZ US).

Needleman, S., & Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology , 48* (3), pp. 443-453.

Noy, N., & Musen, M. (2001). Anchor-PROMPT: Using non-local context for semantic matching. *In Proc. IJCAI 2001 workshop on ontology and information sharing*, (pp. 63–70). Seattle (WA US).

Noy, N., & Musen, M. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. *Dans Proc. 17th National Conference of Artificial Intelligence (AAAI)*, (pp. 450–455). Austin (TX US).

Oulefki, S., & Akli-Astouati, K. (2008, a). Appariement des ontologies : Un nouveau processus. *Journée Jeunes Chercheurs en Informatique (JCI'2008)*. Guelma, Algérie.

Oulefki, S., & Akli-Astouati, K. (2008, b). Classification des techniques d'appariement d'ontologies. *Dans Proc. International Conference on Web and Information Technologies "ICWIT'O8"*. Sidi Bel Abbes, Algérie. .

OWL. (2004). *OWL - Web Ontology Language - (W3C Recommendation 10 February 2004).* Von W3C: http://www.w3.org/TR/2004/REC-owl-guide-20040210/

Parent, C., & Spaccapietra, S. (1996). Intégration de bases de données: Panorama des problèmes et des approches. *Ingénierie des sytèmes d'information , Vol.4* (N°3).

Psyché, J. B., & Mendes, O. (2003). Apport de l'ingénierie ontologique aux environnements de formation à distance. *Revue STE* , S. pp 89–126.

Psyché, V., Mendes, O., & Bourdeau, J. (2003). Apport de l'ingénierie ontologique aux environnements de formations à distance. *revue sticef.org , Volume 10*.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man, and Cybernetics, , 19* ((1)), pp.17–30.

Rahm, E., & Bernstein, P. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal , 10* (4), pp.334-350.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy., *10*, pp. 448– 453. Montréal, Canada.

Sabou, M., D'Aquin, M., & Motta, E. (2006). Using the Semantic Web as Background Knowledge for Ontology Mapping. *In proc. ISWC'06 Workshop on Ontology Matching (OM-2006)*, (S. 1-12). Athens (GA US).

Safar, B., Reynaud, C., & Calvier, F.-E. (2007). Techniques d alignement d ontologies basees sur la structure d une ressource complementaire. *1eres Journees Francophones sur les Ontologies* .

Samyn, O. (2002). *Traduction entre niveaux d'abstraction pour des applications de bases de données.* Mémore d'études approfondies en sciences appliquées, Université Libre de Bruxelles.

Sheth, A., Larson, J., Cornelio, A., & Navathe, S. (1988). A tool for integrating conceptual schemas and user views. *In Proc. 4th International Conference on Data Engineering (ICDE),* pp. 176-183. Los Angeles (CA US).

Shvaiko, P. (2006). *Iterative schema-based semantic matching.* Thèse de doctorat, International Doctorate School in Information and Communication Technology, Université de Trento, Trento (IT).

Staab, S., & Maedche, A. (2000). *Axioms are objects too: Ontology engineering beyong the modeling of concepts and relations.* Research report 399, Institute AIFB, Karlsruhe.

Uschold, M., & Gruninger, M. (1996). Ontologies : Principales, Methodes and Applications Knowledge. *Engineering Review , vol. 11* (n°2).

Valtchev, P. (1999). *Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets.* Thèse d'informatique, Université Grenoble 1.

Valtchev, P., & Euzenat, J. (1997). Dissimilarity measure for collections of objects and values. *In P. Coen X. Liu and M. Berthold, editors, Proc. 2nd Symposium on Intelligent Data Analysis, volume 1280,* pp. 259–272.

Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., et al. (2001). Ontology-based integration of information - a survey of existing approaches. *In Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 108–117.

Winkler, W. (1999). *The state of record linkage and current research problems.* Statistics of Income Division. Internal Revenue Service Publication.

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. *In Proc. 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 133-138. Las Cruces (NM US).

Zargayouna, H. (2005). *Indexation sémantique de documents XML*. Thèse de doctorat, Université Paris XI Orsay.

Zargayouna, H., & Salotti, S. (2004). Mesure de similarité sémantique pour l'indexation de documents semi-structurés. *Dans le 12ème Atelier de Raisonnement à Partir de Cas* .

Ziegler, P., Kiefer, C., Sturm, C., Dittrich, K. R., & Bernstein, A. (2006). Detecting Similarities in Ontologies with the SOQA-SimPack Toolkit. *In Proceedings of EDBT*.