MASTER EN SCIENCES INFORMATIQUES

Recherche de documents à partir d'ontologies de domaines

Lawarrée, François

Award date:
2010

Awarding institution:
Universite de Namur

Faculté Universitaire Notre-Dame de la Paix
Institut d'Informatique
Année Académique 2009-2010

Recherche de documents à partir d'ontologies de domaines

François Lawarrée

Promoteur : **Jean-Luc Hainaut**
Co-promoteur : **Rokia Bendaoud**

Mémoire présenté en vue de l'obtention du grade de master en sciences informatiques

---

Faculté d'Informatique - rue Grandgagnage, 21
B-5000 Namur, Belgique
Tél : +32 (0)81 72 50 02 - Fax : +32 (0)81 72 49 67

Résumé

L'indexation de la plupart des moteurs de recherche consiste généralement à lier une donnée avec sa ressource. Cependant ce mode d'indexation n'est pas sans risques et peut amener à négliger certaines ressources qui auraient pu s'avérer pertinentes pour l'utilisateur. Ceci est dû principalement au fait que la plupart des systèmes de recherche d'informations ne tiennent pas compte du sens sémantique des termes qu'ils indexent. L'utilisation d'ontologies en Recherche d'Informations nous permet de résoudre ce problème et de mettre en place un nouveau type d'indexation considérant le sens des termes indexés.

Dans ce mémoire, nous présentons la conception d'un moteur de recherche fondé uniquement sur les ontologies d'un domaine. Le moteur de recherche proposé se compose de deux ontologies construites à partir des termes extraits d'un domaine. L'une vise à indexer les ressources avec leurs termes pour la recherche d'informations. L'autre, structure l'ensemble des termes issus du domaine afin de proposer trois enrichissements de requête.

Mots clés

ontologie, recherche d'informations, analyse formelle de concepts (AFC), moteur sémantique, enrichissement de requête

Abstract

The indexing of most search engines simply binds a given word with its resource. However, this indexing method is not without risks and can lead to neglect some resources that might be relevant to the user. This is due mainly to the fact that most information retrieval systems do not take into account the semantic meaning of terms they index. The use of ontologies in Information Research allows us to solve this problem and establish a new type of indexing considering the meaning of index terms.

In this work, we present a methodology to build a semantic search engine based only on domain's ontologies. The proposed search engine is based on two ontologies built from terms extracted for a specific domain. One is used to index resources with their terms for the search of information. The other structures the terms from the domain under one ontology to propose three query expansions.

Keywords

ontology, retrieval information, formal concept analysis (FCA), semantic search engine, expansion request

Avant-propos

Je tiens à remercier toutes les personnes qui ont participé directement ou indirectement à la réalisation de ce mémoire :

- En premier lieu Mr Hainaut, professeur aux FUNDP, pour son suivi et son écoute accordés tout au long de ce travail.

- Je remercie mon maître de stage Joseph Roumier, appartenant au CETIC (Centre d'Excellence en Technologies de l'Information et de la Communication) et responsable de mon stage, pour son aide et son soutien.

- Je remercie particulièrement Madame Rokia Bendaoud, post-doctorante aux FUNDP pour son aide tout au long de la rédaction de ce mémoire ainsi que pour ses conseils avisés et son expérience.

- Le personnel du CETIC pour leur accueil et les professeurs des Facultés Universitaires Notre-Dame de la Paix de Namur pour le savoir qu'ils m'ont transmis durant ces trois années.

- Ma fiancée Ariane, qui m'épaula et m'encouragea dans les moments difficiles.

- A tous mes amis, qui m'ont soutenu durant ses années difficiles et qui m'ont permis de ne jamais perdre espoir.

- Enfin, je tenais à remercier toute ma famille et tout particulièrement mes parents pour leur confiance, leur soutien et leur aide donnée tout au long de mes études.

Je dédie ce mémoire à mon papy André DURIEUX

Table des matières

Introduction

Les moteurs de recherche sont, sans aucun doute, le principal outil mis à disposition des utilisateurs pour la recherche d'informations. Figurant parmi les services les plus utilisés sur le web, ceux-ci jouent un rôle essentiel dans les usages d'internet en guidant les internautes à travers les ressources qu'ils référencent. Ces ressources s'identifient par un URI [1] (Uniform Resource Identifier) et se rapportent à des documents électroniques, des images, du son, des blogs, des wikis, des services, etc.

Chaque jour, le Web croît d'environ un million de pages électroniques venant s'ajouter aux centaines de millions déjà existantes. En raison de cette croissance exponentielle et chaotique, la majorité des moteurs de recherche ont pour rôle de suivre le rythme en indexant au fur et à mesure l'ensemble des nouvelles ressources ajoutées sur la toile. Or, ceux-ci travaillent à partir de bases d'index de grande taille, généralement construites à l'aide de robots accédant aux documents publiés sur le Web. Malgré une amélioration constante des processus d'indexation et de recherche, les résultats fournis sont parfois difficiles à utiliser et ne correspondent pas toujours aux attentes de l'utilisateur. Certains mots-clés retenus pour indexer ces ressources peuvent parfois ne pas être adéquats et pertinents. De plus, l'indexation purement syntaxique utilisée par la plupart des systèmes de recherche d'informations a pour conséquence d'ignorer certaines ressources qui auraient pu s'avérer pertinentes aux yeux de l'utilisateur. Par exemple, lors d'une recherche effectuée à partir d'un terme comme « *chien* », le moteur renverra comme résultat l'ensemble des ressources ayant été indexées avec ce terme, mais négligera toutes les autres même si celles-ci avaient pour sujet « *Labrador* », « *Jack Russel* » ou encore « *Chiwawa* ».

Pour résoudre ce problème, il est nécessaire de tenir compte du sens des termes indexés. En considérant la sémantique des termes, il nous est alors possible de structurer ceux-ci et de les mettre en relation afin de définir la connaissance du domaine auxquels ils appartiennent. Pour tenir compte du sens des mots et représenter la connaissance d'un domaine, les ontologies semblent être la solution la plus

1. URI : http ://websemantique.org/URI. Date : 2/08/2010

adaptée à cette problématique. Elles représentent la connaissance d'un domaine spécifique et par conséquent la sémantique du vocabulaire qu'elles contiennent.

L'utilisation d'une ontologie par un moteur de recherche permet d'ajouter une couche sémantique pouvant être comprise et interprétée par l'ordinateur. Il est alors possible de décrire la connaissance d'un domaine spécifique et de l'utiliser lors du processus de recherche d'informations. Pour illustrer ces notions, nous reprenons l'exemple précédant en le formalisant en une ontologie présentée par la figure 1.11. Celle-ci formalise les connaissances que « *Labrador* », « *Jack Russel* » et « *Chiwawa* » sont des « *Chien* », qu'un chien est un « *Canidé* » et que le chien communique en « *Aboyant* ».

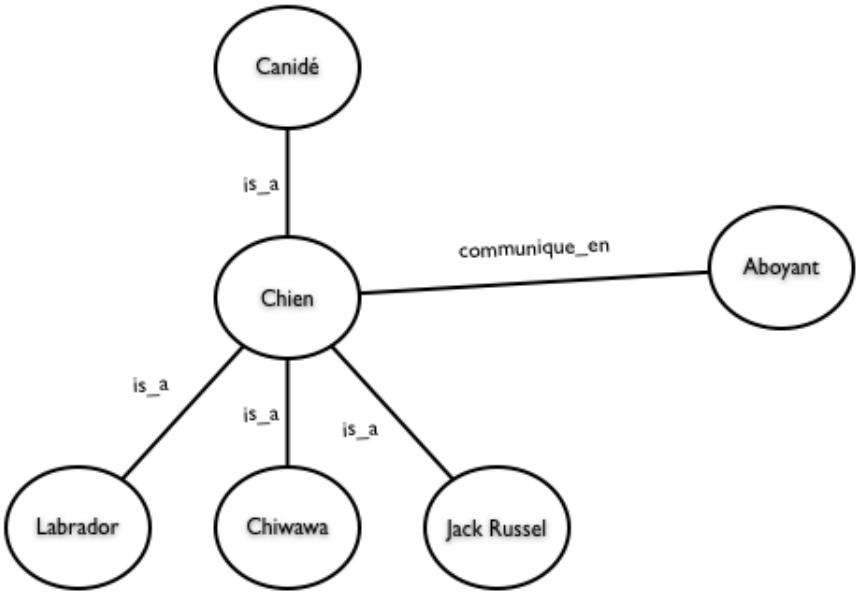


FIGURE 1 – **Exemple d'ontologie**

Par l'utilisation de ce modèle de connaissance, il nous est alors possible d'enrichir la requête de l'utilisateur. La pertinence des résultats retournés est améliorée et donne accès aux ressources ayant été négligées. Nous utilisons trois enrichissements de requête : par généralisation, par spécialisation et par relation. Ainsi, à partir d'une recherche sur le terme « *chien* » ;

- l'enrichissement par généralisation nous renverra « *Canidé* ».
- l'enrichissement par spécialisation nous renverra « *Chiwawa* », « *Labrador* » et « *Jack Russel* ».
- l'enrichissement par relation avec la relation « *communique_ en* » nous renverra « *Aboyant* ».

Le contexte de ce mémoire s'inscrit dans le projet « e-Health » [2] financé par l'UE et la région Wallonne. Il fait intervenir le CETIC [3] (Centre d'Excellence en Technologies de l'Information et de la Communication) en tant que chef de file du projet et responsable de la coordination et de la gestion de celui-ci. Le centre de recherche PReCISE [4] des FUNDP participe également au projet et est responsable de la gestion de services de santé distribués ainsi que de la gestion des données médicales via une communauté de prestataires « e-Health ». Dans ce domaine, il apparaît que les ressources échangées sont extrêmement variées et donc difficiles à intégrer au coeur d'un même système de recherche. Dès lors, une interprétation sémantique de leur contenu au travers d'ontologies permettrait de faire face à cette diversité. Des ontologies du domaine permetteraient également de guider et d'améliorer la recherche d'informations des utilisateurs à partir d'enrichissements de requêtes. Ainsi, la sémantique des ontologies donnerait à un médecin ou à tout autre utilisateur la possibilité d'interroger une base de connaissance d'un domaine particulier lors d'une recherche d'informations. L'accès à ce savoir susciterait la découverte de nouvelles connaissances pour l'utilisateur mais aussi permettrait d'améliorer la pertinence des résultats retournés au travers d'un enrichissement de termes appropriés.

Ce mémoire propose une méthodologie pour la conception d'un moteur de recherche intelligent fondé sur l'utilisation conjointe de deux ontologies d'un domaine.

La première ontologie reprend l'idée précédemment illustrée, qui consiste à structurer un ensemble terminologique extrait d'un domaine spécifique en une ontologie et tenant compte de diverses relations sémantiques. La structuration s'appuie sur les concepts et relations sémantiques contenues dans un thesaurus du domaine. L'objectif de cette ontologie est d'améliorer la recherche de l'utilisateur au moyen de trois types d'enrichissements. Parmi ceux-ci, les enrichissements par « généralisation » et par « spécialisation » tirent parti de l'idée utilisée par [Carpineto and Romano, 1996]. Le troisième type d'enrichissement, par « relation », offre à l'utilisateur la possibilité d'interroger la connaissance du domaine sur base d'un terme et d'une relation.

La deuxième ontologie n'est pas la représentation de connaissances tout comme pour la première ontologie mais est utilisée comme moyen afin d'indexer les docu-

2. e-Health : http ://www.cetic.be/article962.html. Date : 2/08/2010.
3. CETIC : http ://www.cetic.be/. Date : 12/08/2010.
4. PReCISE : http ://www.fundp.ac.be/en/precise/. Date : 2/08/2010.

ments du domaine avec leurs termes pour permettre la recherche d'informations. Pour la conception de cette dernière, nous utilisons une méthode de classification conceptuelle appelée l'Analyse Formelle de Concepts (AFC) [Ganter et al., 1999]. Cette technique de recherche d'informations permet de structurer un ensemble d'objets en fonction des propriétés communes qu'ils partagent. Nous appliquons cette méthode formelle sur des textes et leurs termes propres afin de classer les textes d'après les termes qu'ils contiennent. Nous représentons ensuite le treillis résultant en une ontologie en nous appuyant sur un processus de transformation s'inspirant de [Bendaoud et al., 2008].

Ce mémoire se répartit en cinq chapitres :

- **Le premier chapitre** constitue l'état de l'art reprenant les principes fondamentaux utilisés et se subdivise en trois sections. La première section concerne la Recherche d'Informations dans laquelle nous présentons différentes techniques de classification de ressources ainsi qu'un tour d'horizon des différentes méthodes d'enrichissement de requête. Dans la deuxième section, nous nous attardons sur ce qu'est la connaissance, d'où provient-elle et au travers de quelles technologies pouvons nous la représenter ? La dernière section porte sur la conception d'une base de connaissances à partir de ressources textuelles.

- **Le deuxième chapitre** présente la méthodologie et le processus de construction défini pour la conception d'un moteur de recherche sémantique. Dans cette partie, nous présentons chacune des étapes de notre approche au travers d'un exemple détaillé pour un domaine défini.

- **Le troisième chapitre** introduit un ensemble de prototypes conçus pour automatiser certaines étapes de la méthodologie détaillée dans le chapitre deux. Développés lors d'un stage effectué au CETIC, nous présentons ces outils à travers leur architecture, leur conception et leur l'utilisation.

- **Le quatrième chapitre** aborde une expérimentation de la méthodologie développée dans le cadre d'un stage en entreprise pour le domaine de la maladie de Parkinson. Elle fait intervenir également les trois prototypes exposés dans le troisième chapitre.

- **Le cinquième et dernier chapitre** conclura ce mémoire et énoncera différentes perspectives d'extension ce travail.

Chapitre 1

Etat de l'art

1.1 La Recherche d'Informations

1.1.1 Introduction

La Recherche d'Informations (RI) est une science qui comprend un ensemble de méthodes, de procédures et de techniques permettant, en fonction de critères de recherche propres à l'usager, de sélectionner l'information dans un ou plusieurs fonds documentaires plus ou moins structurés. Cette discipline a toujours eu le souci d'établir des représentations de documents dans le but d'en récupérer des informations, à travers la construction d'index. De nos jours, la recherche d'informations est un champ transdisciplinaire qui œuvre dans le but de trouver des solutions pour améliorer son efficacité. D'une manière générale, la RI intervient essentiellement dans deux aspects : « *l'indexation des corpus* » et « *l'interrogation du fonds documentaire constitué* ».

L'indexation des corpus consiste à classifier des textes ou documents à partir d'un ensemble terminologique représentant au mieux leur contenu. Cette classification permet ainsi de regrouper des textes partageant les mêmes termes et de les distinguer à partir des termes qui leur sont propres. Pour parvenir à cette classification, des méthodes numériques ou symboliques existent afin de représenter au mieux les documents issus d'un corpus. Les méthodes les plus souvent employées dans la littérature sont fondées sur des critères statistiques ou numériques [Salton and Buckley, 1988, Curran and Moens, 2002, Crouch, 1988]. Celles-ci se portent sur la fréquence et la cooccurrence des mots d'un texte. Les méthodes symboliques quant à elles, ne tiennent pas compte de mesures statistiques et se fondent sur une analyse syntaxique des textes. Certaines utilisent une classification conceptuelle qui construit une hiérarchie de concepts de façon incrémentale en regroupant un ensemble d'instances (textes) d'après leurs descriptions (termes)

[Ganter et al., 1999, Fisher, 1987, Feigenbaum, 1961, Gennari et al., 1989].

Le deuxième aspect de la RI, vise à interroger l'ensemble documentaire constitué et indexé afin de trouver le ou les documents répondant de manière efficace et pertinente à une requête donnée. Cependant, malgré une bonne classification des textes, il est fréquent de constater que les termes employés par l'utilisateur ne correspondent pas toujours à ceux utilisés pour indexer le texte recherché et ce, malgré une sémantique commune. Pour réduire au maximum cette distance sémantique, de nombreuses études dans le domaine de la RI ont produit des solutions à travers un enrichissement de la requête de l'utilisateur. Cet enrichissement consiste à rajouter un certain nombre de termes à la requête initiale afin de guider l'utilisateur dans sa recherche d'informations, mais aussi d'améliorer la pertinence des documents présentés. Cette technique d'enrichissement est aussi appelée « extension de requête ».

Dans cette partie, nous commencerons par distinguer les deux méthodes de classification documentaire (numérique et symbolique) à partir d'une approche statistique et d'une approche symbolique utilisant l'analyse formelle de concept. Ensuite, nous expliquerons plus en détail l'apport des extensions de requête ainsi que les différentes approches utilisées principalement dans la RI.

1.1.2 Méthode de classification documentaire

Méthode numérique : l'approche statistique

Durant les années 60, les travaux de G. Salton [Salton, 1968, Salton and Buckley, 1988] furent à l'origine de nombreuses approches statistiques. L'approche statistique se fonde sur une extraction des cooccurrences des termes dans un contexte particulier. Les outils statistiques tentent de parvenir à une représentation du sens des textes qui les caractérisent les uns par rapport aux autres. La représentation d'un document est calculée en fonction de l'ensemble des documents contenus dans la base documentaire. L'idée suggère de s'appuyer sur l'analyse de la fréquence des mots pour indexer les documents (ou textes). Pour ce faire, les outils statistiques sont bien adaptés à la détection des récurrences contextuelles contrairement aux analyseurs linguistiques qui n'observent pas les régularités sur des grosses masses de données. Cette information est exploitée en utilisant différentes mesures de similarité [Salton and Buckley, 1988]. Ces mesures se fondent sur l'hypothèse suivante : l'emploi de deux termes en cooccurrence est l'expression d'une relation sémantique entre ces termes (par exemple, « élève et professeur », « voiture et garage » ou encore « aéroport et avion »). Néanmoins, ces outils ne tiennent pas compte de la réelle sémantique des mots et

seul le calcul de différents indices mathématiques et statistiques est pris en compte.

L'article [Yu and Salton, 1977] apporta l'idée de « discrimination », dans le cadre des recherches sur les bases de données documentaires. Celle-ci consiste à calculer le poids informationnel de chaque « item » par une fonction destinée à faire émerger une distinction entre les documents.

Cette fonction est à la base de la théorie de l'information :

- Plus un mot est rare sur l'ensemble des documents et fréquent dans un document particulier, plus il est caractéristique de ce document.
- Le pouvoir d'un terme i pour représenter un document j est basé sur sa fréquence dans le document « tf_{ij} » (term frequency) et sur l'inverse du nombre de documents de la base « idf_i » dans lesquels il apparaît (inverse document frequency).

Parmi les nombreux indices de similarité entre documents proposés et validés par G. Salton et son équipe [Salton and Buckley, 1988], le plus répandu et apprécié est nommé par ses auteurs « *best fully weighted system* » (ou encore TFIDF : term frequency, inverse document frequency). Cet indice permet de donner une importance à un terme en fonction de sa fréquence dans le document (TF) pondéré par la fréquence de ce terme parmi tout le corpus (IDF) :

$$W_{ij} = tf_{ij}.idf_i \tag{1.1}$$

$$W_{ij} = tf_{ij}.((log_2 \frac{N}{DF_i}) + 1) \tag{1.2}$$

où :
- i : le terme
- j : le document
- tf_{ij} : la fréquence d'apparition du terme i dans le document j
- N : le nombre de documents du corpus
- DF_i : nombre de documents où le terme i apparaît

Le terme idéal (W_{ij}) est celui qui apparaît fréquemment dans un document tout en étant particulièrement rare dans les autres.

Néanmoins, le problème majeur de ces modèles et approches statistiques est que les fonctions discriminantes ne parviennent pas à maîtriser le pouvoir d'expression des mots significatifs : les taux de bruit (la prise en compte de

termes non-pertinents) et de silence (l'omission de termes qui auraient pu s'avérer pertinents) restent relativement importants selon la taille des corpus traités [El Guedj and Nugues, 1997]. De plus, ces approches ne sont pas déterministes et peuvent donc donner des résultats différents à partir d'un même jeu de données.

Méthode symbolique : l'analyse formelle de concepts

L'analyse formelle de concepts (AFC) définie par [Wille, 1982] est une méthode mathématique permettant de structurer hiérarchiquement des concepts composés d'un ensemble d'objets partageant les mêmes propriétés. Cette classification hiérarchique est appelée treillis de concepts et est à la base d'une famille de méthodes de classification conceptuelle introduite par Barbut et Monjardet [Barbut and Monjardet, 1970]. Cette approche a ensuite été popularisée par Wille qui a utilisé la notion de treillis de concepts comme base de l'analyse formelle de concepts [Wille, 1982]. L'AFC a initialement trouvé des applications en intelligence artificielle pour la représentation et l'acquisition de connaissances [Ganter et al., 1985, Wille, 1982]. Dans ce qui suit, nous présentons les notions fondamentales de l'AFC provenant de [Ganter et al., 1999] et nous l'illustrons à travers un simple exemple reprenant un ensemble d'animaux caractérisés par un ensemble de propriétés :

L'AFC prend comme point de départ un « contexte formel » qui définit une relation binaire reliant deux types d'éléments : les « objets » et les « propriétés ».

Définition 1 (Contexte formel) *Un contexte formel est un triplet* $\mathbb{K} = (G,M,I)$ *où* G *est un ensemble d'objets,* M *est un ensemble de propriétés et* I *une relation binaire entre* G *et* M *vérifiant :*

1. $I \subseteq G \times M$.
2. $(g,m) \in I$ *signifie que l'objet* g *possède la propriété* m.

	a_des_poils	a_des_plumes	a_des_dents	a_des_nageoires	est_ovipare
Antilope	X		X		
Sanglier	X		X		
Poulet		X			X
Poisson			X	X	X

FIGURE 1.1 – **Contexte formel.**

La figure 1.1 présente un tableau illustrant le contexte formel $\mathbb{K}$ = (G,M,I) où G est un ensemble constitué d'animaux et M est un ensemble de propriétés. La corrélation (Antilope, a_des_dents) signifie que l'objet « Antilope » possède la

propriété « a_des_dents » et de façon duale que la propriété « a_des_dents » est possédée par l'objet « Antilope ».

Définition 2 (Connexion de Galois) *Soit $\mathbb{K}$ = (G,M,I) un contexte formel. Pour tout $A \subseteq G$ et $B \subseteq M$, on définit :*

1. *A' := { m∈M | ∀ g∈A | gIm},*
2. *B' := { g∈G | ∀ m∈B | gIm},*

La figure 1.2 illustre l'utilisation de la connexion de Galois. L'opérateur de dérivation (.)' définit une connexion de Galois entre l'ensemble de G, noté 2^G et l'ensemble des parties de M, noté 2^M. La première dérivation A' associe l'ensemble des propriétés vérifiées par l'ensemble des objets de A. Par exemple, les objets appartenant à A={Antilope, Sanglier} vérifient toutes les propriétés de A'={a_des_poils, a_des_dents}. De manière duale, le deuxième ensemble B', associe l'ensemble de propriétés de B avec l'ensemble des objets qui possèdent au moins toutes les propriétés de B. Par exemple B={a_des_poils, a_des_dents} et B'={Antilope, Sanglier}.

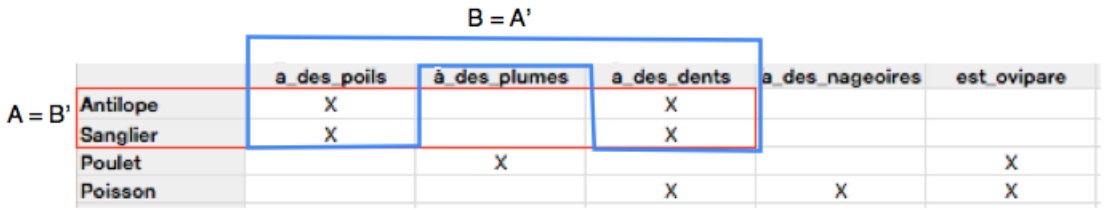


FIGURE 1.2 – **Connexion de galois.**

Les paires reliées par cette connexion de Galois forment ce qu'on appelle *les concepts formels* et sont définis comme suit.

Définition 3 (Concept formel) *Un concept formel d'un contexte $\mathbb{K}$ = (G,M,I) est une paire (A,B) avec : $A \subseteq G$ et $B \subseteq M$, A'=B et B'=A, où A' est l'ensemble de toutes les propriétés possédées par les objets A et de façon duale B' est l'ensemble de tous les objets possédant les propriétés de B. Les ensembles A et B sont aussi appelés respectivement* ***extension*** *et* ***intension*** *du concept formel C.*

A partir de la connexion de Galois illustrée par le figure 1.2, nous en déduisons un concept illustré par 1.3.

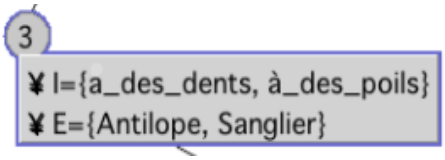


FIGURE 1.3 – **Concept formel.**

Définition 4 (Relation de subsomption) *Soit (A1,B1) et (A2,B2) deux concepts formels de B(G,M,I). (A1,B1)* $\sqsubseteq$ *(A2,B2)* $\Leftrightarrow$ *A1* $\subseteq$ *A2 (où de façon duale B2* $\subseteq$ *B1).*

La relation illustrée par la figure 1.4, s'appuie sur deux inclusions duales, entre ensembles d'objets et entre ensembles d'attributs. Ceci peut dès lors être interprété comme une relation de généralisation/spécialisation entre les concepts formels. Un concept est plus général qu'un autre concept s'il contient plus d'objets dans son extension. En contre partie, les attributs partagés par ces objets sont réduits. De façon duale, un concept est plus spécifique qu'un autre s'il contient moins d'objets dans son extension. Ces objets ont plus d'attributs en commun.

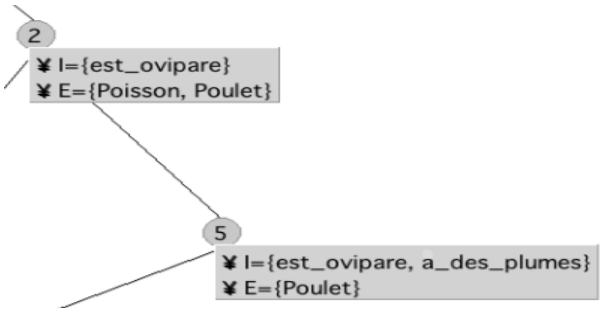


FIGURE 1.4 – **Relation de subsomption.**

Définition 5 (Treillis de concepts) *Soit (F,*$\sqsubseteq$*) le produit de deux treillis (*F_G*,*$\sqsubseteq$*) et (*F_M*,*$\sqsubseteq$*) appelés respectivement treillis des extensions et treillis des intentions. (F,*$\sqsubseteq$*) est le* ***treillis de concepts*** *associé à la relation binaire I sur G* $\times$ *M. L'ensemble de tous les concepts formels du contexte* $\mathbb{K}$ *= (G,M,I) muni de l'ordre partiel* $\sqsubseteq$ *est un treillis appelé* ***treillis de concepts*** *de* $\mathbb{K}$ *et noté* $\underline{\mathfrak{B}}$*(G,M,I).*

Ce treillis regroupe un ensemble de concepts formels composés chacun d'une paire (A,B) où A $\subseteq$ G et B $\subseteq$ M de façon à ce que A' = B et B' = A. Cette structure conceptuelle nous présente ces concepts comme étant des classes d'objets caractérisés par un ensemble de propriétés. De plus, l'ordre hiérarchique entre ceux-ci, permet de définir des liens de généralisation et de spécification

[Carpineto and Romano, 2000].

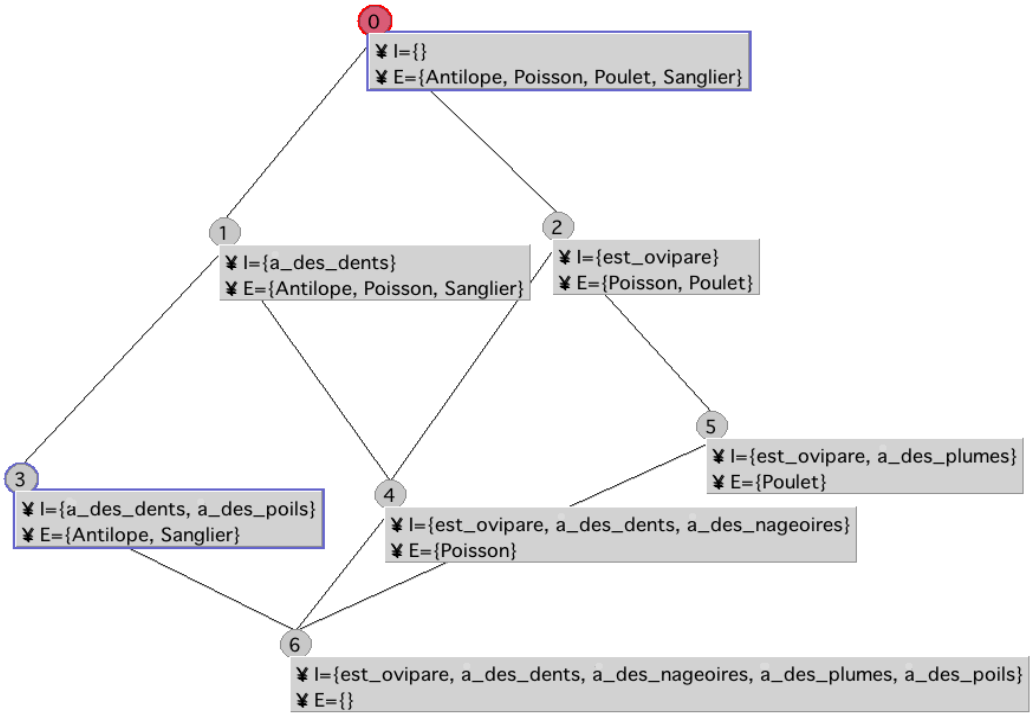


FIGURE 1.5 – **Treillis de concepts.**

Dans le domaine de la Recherche d'Informations (RI), l'AFC est largement utilisée. En effet, une analogie existe parmi les *objets X propriétés* et les *documents X termes*. Cette analogie a été explicitement mentionnée par [Godin et al., 1986] comme étant une application possible des treillis de concepts. Des collections de documents sont alors représentées sous la forme de contextes formels. Les objets du contexte sont des documents et les propriétés sont les termes d'indexation de ces documents. Chaque concept du treillis correspondant est vu comme un couple formé par une requête, dont les mots clés sont les termes contenus dans l'intension du concept. L'ensemble de documents pertinents pour cette requête sont les documents contenus dans l'extension du concept. A travers ce cas, la relation de subsomption entre concepts, nous permet le passage d'un concept (ou d'une requête) à un autre concept plus général ou plus spécifique. En effet, lorsque l'on parcourt le treillis de haut en bas, nous augmentons l'ensemble des propriétés ; ce qui correspond à une requête plus spécifique. De façon duale, lorsque nous parcourons le treillis de bas en haut, nous diminuons l'ensemble des propriétés et cela nous

donne une requête plus générale. Ainsi, les liens de spécialisation/généralisation entre les concepts permettent d'effectuer une recherche progressive dans le treillis. Cette idée provient de Carpineto [Carpineto and Romano, 1996] qui utilise la requête de l'utilisateur pour l'insérer dans le treillis de concepts déjà construit. Il ordonne ensuite les documents-réponses en considérant que plus les concepts sont proches de celui de la requête, plus les documents appartenant à leur extension sont pertinents. Pour illustrer ce type d'application, considérons le contexte formel défini ainsi que son treillis résultant par la figure 1.6. Ce contexte intègre différents moyens de transports abordés parmi un ensemble de revues/magazines. A partir de ce contexte, un exemple de requête d'utilisateur pourrait être donné par les termes :*motos, voiture, vélo*. Celle-ci signifierait que l'utilisateur souhaiterait obtenir la ou les revues contenant les termes spécifiés. Cette requête est alors définie en un concept formel : C_Q=($\{Query\}$, {Motos, Voiture, Vélo}) afin d'y être introduite dans le treillis. Pour cet exemple, l'insertion du concept C_Q dans le treillis entraîne l'ajout d'un nouveau concept propre à C_Q comme illustré par la figure 1.7. Néanmoins certaines requêtes ne nécessitent pas l'ajout de nouveaux concepts au treillis initial. En effet, lorsque les termes d'une requête sont identiques aux intentions d'un concept du treillis initial, l'insertion de la requête aura pour conséquences la modification de certains concepts du treillis. La figure 1.7 nous montre les modifications apportées au treillis ainsi que les concepts du treillis qui seront utilisés afin de fournir les résultats. Les entiers figurant à côté des concepts modifiés correspondent l'ordre des concepts considérés lors de la construction du résultat. Ainsi, le résultat retourné en réponse à la requête considérée est constitué des sources des ensembles R_0=$\{/\}$, R_1=$\{$Revue_1, Revue_2, Revue_4$\}$, R_2=$\{$Revue_1, Revue_2, Revue_4$\}$.

D'autres types de classifications hiérarchiques (arbres de décision) permettent de définir l'appartenance d'un objet à un concept à partir de ses propriétés mais ils n'ont pas d'héritage multiple et ils ne sont pas incrémentals. Selon les deux articles [Godin et al., 1995, Carpineto and Romano, 2000], la Recherche d'Informations à partir de treillis atteint des performances qui dépassent celle de la recherche booléenne classique. Une limite de cette approche est la complexité du treillis, à savoir son nombre de concepts, pour des contextes formels volumineux en termes d'objets et de propriétés. Cependant, cette limite maximale à travers des applications réelles n'est jamais atteinte [Carpineto and Romano, 1996] et certaines études [Stumme et al., 2002, Pernelle et al., 2002] ont défini différentes solutions visant à contrôler la taille des treillis correspondant à de grands contextes pour diminuer cette complexité.

Pour terminer, le logiciel **Galicia** [1][Valtchev et al., 2003] est un logiciel libre utilisé pour la construction de contextes formels ainsi que pour la génération et la visualisation de treillis de concepts. C'est également une plateforme qui contient une multitude d'algorithmes pour la construction de treillis.

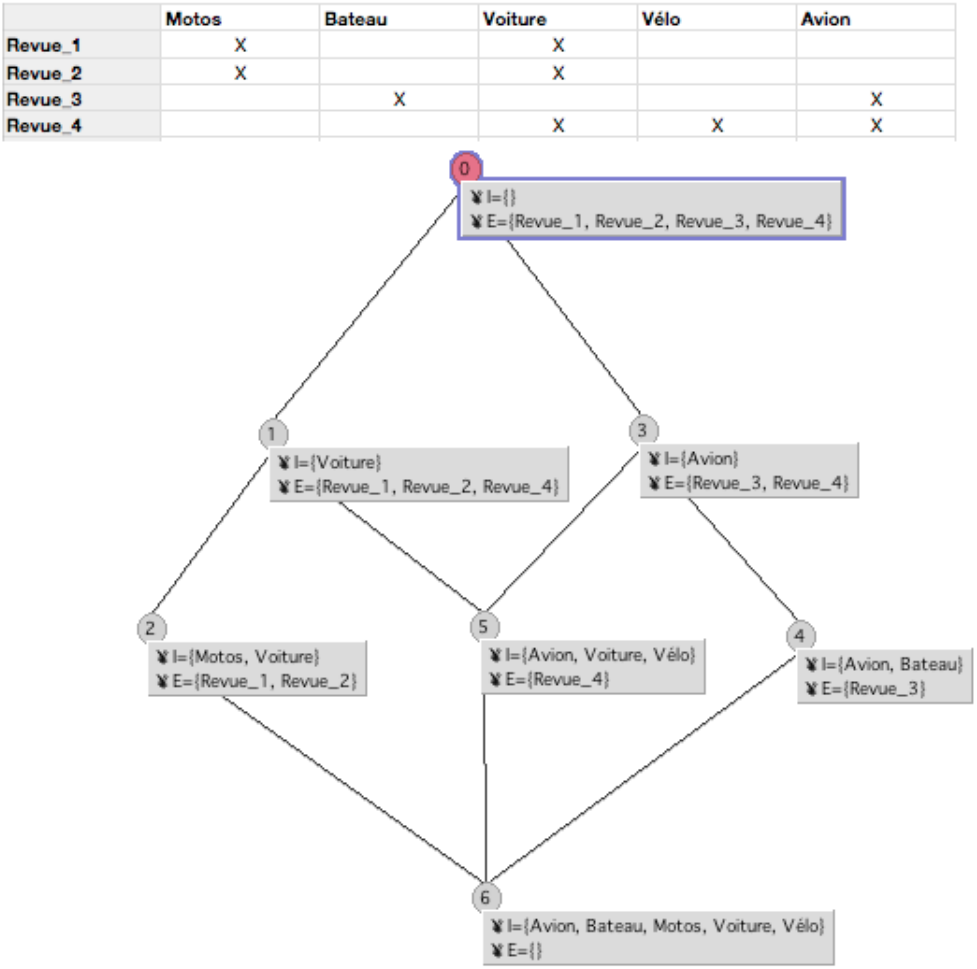


FIGURE 1.6 – **Contexte formel et treillis de concepts définis pour la recherche d'informations.**

1. Galicia : http ://www.iro.umontreal.ca/ galicia/publication.html. Date : 13/08/2010.

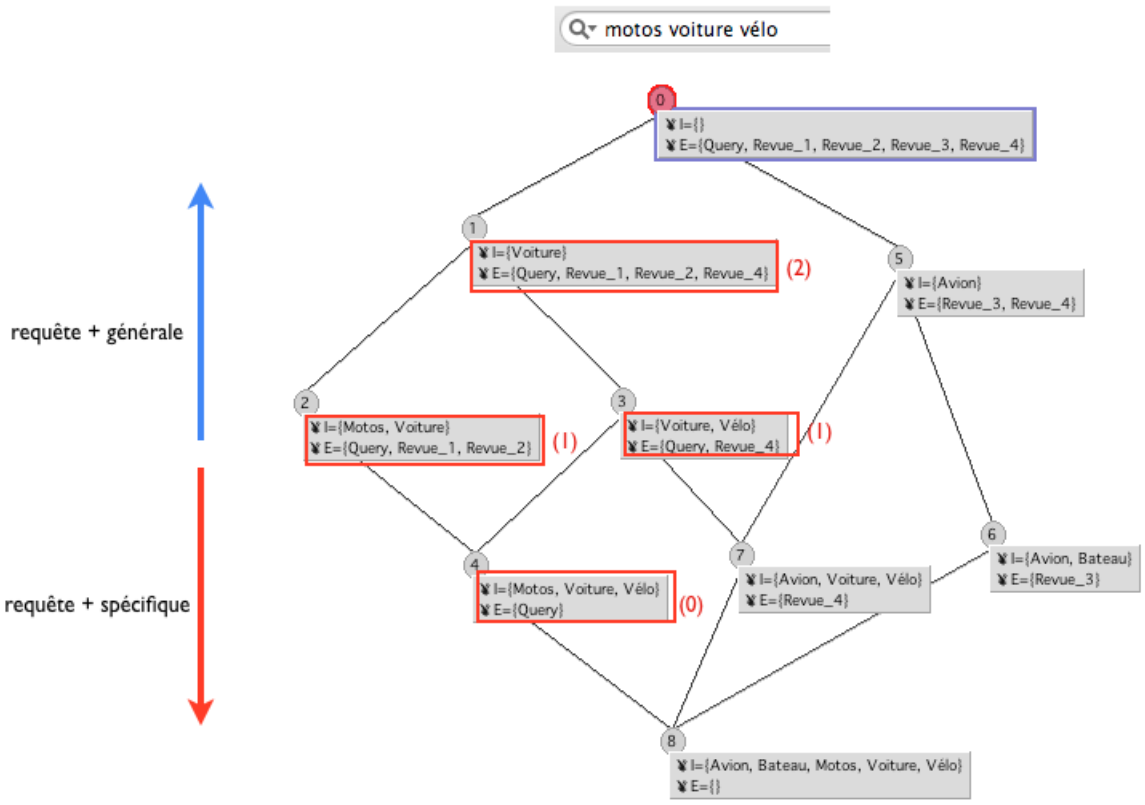


FIGURE 1.7 – **Treillis de concepts modifié par l'insertion de la requête.**

1.1.3 Les extensions de requête

Depuis ces dernières années, la quantité d'informations publiées sur le Web ne cesse d'augmenter et ce à une vitesse exponentielle. Néanmoins, nous ne pouvons pas avoir accès et utiliser cette information de manière efficace si celle-ci n'est pas correctement organisée et indexée. De nos jours, de nombreux moteurs de recherche ont été créés pour ce besoin. Malgré tout, la plupart de nos requêtes effectuées sur le Web nécessite une certaine expérience dans le choix des mots à formuler afin d'obtenir des résultats pertinents. De plus, certaines études effectuées à partir des « logs » issus des moteurs de recherche du Web, nous montrent que les requêtes formulées sont en moyenne inférieures à deux mots [Pfoser et al., 2000, Silverstein et al., 1998], d'où l'importance de les choisir correctement.

Néanmoins, il arrive souvent que certaines recherches d'informations n'aboutissent pas. Et ce, malgré une sémantique commune parmi les mots formulés par

l'utilisateur et ceux utilisés pour l'indexation des ressources. Ceci implique un grand écart entre les termes de la requête d'un utilisateur et ceux indexant un document. L'utilisateur peut donc parfois obtenir un grand nombre de documents parmi les résultats mais dont la majorité d'entre eux ne sont pas pertinents. Cette problématique fait l'objet de nombreux sujets d'étude dans la Recherche d'Informations (RI). Pour améliorer la pertinence des résultats retournés, il existe une méthode efficace utilisant l'extension de requête. Celle-ci offre la possibilité à l'utilisateur d'améliorer la pertinence des résultats retournés à travers l'ajout ou la suggestion de termes supplémentaires ayant une relation sémantique avec les termes initialement formulés et ceux contenus dans les documents concernés. De plus, les extensions de requête offrent l'avantage aux utilisateurs d'enrichir leurs propres connaissances à travers l'apprentissage de nouveaux mots sémantiquement similaires à ceux initialement formulés et dont ils n'avaient pas connaissance.

Bien que les extensions de requête visent à améliorer la recherche de documents, il leur arrive parfois de s'éloigner de la requête initiale en suggérant des termes ne correspondant pas aux attentes de l'utilisateur. Par exemple, lorsqu'un utilisateur effectue une extension sur le mot « jaguar », le moteur peut lui suggérer ou le remplacer par « félin » alors que l'utilisateur pensait plutôt à la célèbre marque automobile. En l'absence du jugement explicite de pertinence de l'utilisateur, cette stratégie pose essentiellement le problème de dérive du sujet de la requête. Ce phénomène particulier qui joue sur l'ambiguïté sémantique, se nomme le « *query drift* » auquel des solutions ont été apportées et sont expliquées par [Mitra et al., 1998, Buckley et al., 1995].

Il existe un grand nombre de manières d'utiliser les extensions de requête. Dans ce qui suit, nous présentons trois approches d'extensions de requête principalement utilisées en Recherche d'Informations.

Les différentes approches

1. L'approche statistique. La plupart des études menées pour ce type d'approche reposent sur les travaux de Karen Sparck Jones [Sparck Jones, 1971]. L'approche statistique consiste à classifier l'ensemble des ressources textuelles à partir des mots en commun qu'elles partagent. Chaque ressource est préalablement analysée pour en définir l'ensemble statistique des mots les plus fréquemment utilisés. L'ensemble des mots définis comme fréquents pour un certain nombre de documents forme un tout. A partir de ces clusters de mots, l'approche globale consiste à les utiliser pour étendre la requête initiale de l'utilisateur en une requête plus particulière. L'approche locale, se comporte comme l'approche globale mais s'ap-

plique à l'ensemble des documents possédant le meilleur rang (les plus pertinents) résultant de la requête initiale. La figure 1.8 montre une recherche effectuée sur le mot « jaguar » à partir du moteur de recherche Yahoo [2]. On constate que le moteur nous propose d'affiner nos résultats à travers diverses extensions se rapportant aussi bien à l'animal qu'à la marque automobile.

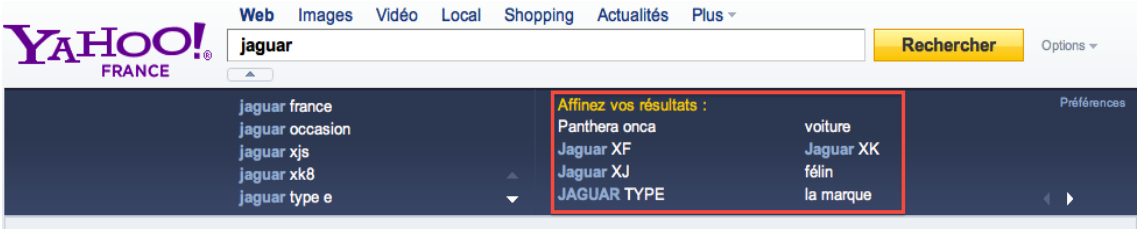


FIGURE 1.8 – **Extensions de requête à partir de statistiques.**

Cette approche offre de bons résultats en terme de robustesse [Xu and Croft, 1996] mais ne tient pas compte de l'aspect sémantique des termes qu'elle manipule et requiert un grand nombre de ressources.

2. L'approche définie à partir de logs. Dans cette approche, l'extension de requêtes se concrétise à partir de « logs » correspondant à un historique reprenant un ensemble de requêtes et de documents ayant été jugés pertinents et donc ayant été « cliqués » [Cui et al., 2002]. Grâce à des usages quotidiens, chaque moteur de recherche accumule une grande quantité de « logs » de requêtes. A partir de ceux-ci, il est possible d'extraire de nombreuses sessions de recherche. Une session de recherche est définie comme suit :

```
session: =  <query texte> [cliked document] *
```

Chaque session correspond à une requête formulée par l'utilisateur ainsi que les documents sur lesquels il a cliqué. L'idée centrale de cette approche est que si un ensemble de documents est souvent choisi pour les mêmes requêtes, alors ces documents sont liés aux termes de ces requêtes. Ainsi, des corrélations entre les termes de la requête probabiliste et les termes du document peuvent être établies sur la base des logs. Ensuite, ces corrélations probabilistes peuvent être utilisées pour sélectionner les termes d'expansion. Cette approche est une des approches utilisée par Google [3] comme nous montre la figure 1.9.

2. Yahoo : http ://fr.yahoo.com/. Date : 5/04/2010
3. Google : http ://www.google.be/. Date : 5/04/2010

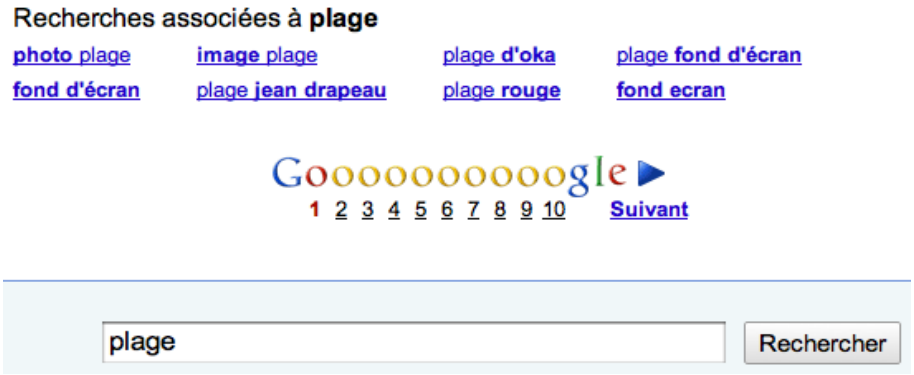


FIGURE 1.9 – **Extensions de requête à partir des logs.**

Une hypothèse importante de cette méthode est que les documents ayant été cliqués sont définis comme « pertinents » pour la requête. Bien que le fait d'avoir cliqué sur un document ne garantit pas un jugement pertinent explicite pour la Recherche d'Informations traditionnelle, le choix de l'utilisateur ne suggère qu'un certain degré de « pertinence » de ce document selon son besoin d'informations. Même si certains des clics sont erronés, nous pouvons convenir que la plupart des utilisateurs cliquent sur les documents qu'ils jugent pertinents. Par conséquent, les « logs » de requêtes peuvent être considérés comme une ressource très précieuse.

Cette approche permet d'adapter les requêtes facilement et de trouver les documents pertinents correspondant à une requête définie mais nécessite de disposer d'une quantité importante de logs.

3. L'approche linguistique. L'approche linguistique consiste à faire appel à des ressources linguistiques externes de type dictionnaire ou thesaurus afin de compléter les termes originaux de la requête en leur associant des termes sémantiquement proches. Le schéma traditionnel pour ce genre d'approche se subdivise en quatre étapes (figure 1.10) :

1. Retrouver les mots issus de la requête initiale au sein de la ressource linguistique.
2. A partir des mots correspondant syntaxiquement à ceux de la requête initiale, déterminer l'ensemble de mots sémantiquement proches suivant des relations définies par la ressource linguistique utilisée.
3. Filtrer l'ensemble des mots similaires sémantiquement qui ne font pas partie de l'index des ressources.

4. Suggérer à l'utilisateur les mots sémantiquement proches de ceux initialement formulés et se trouvant parmi l'index.

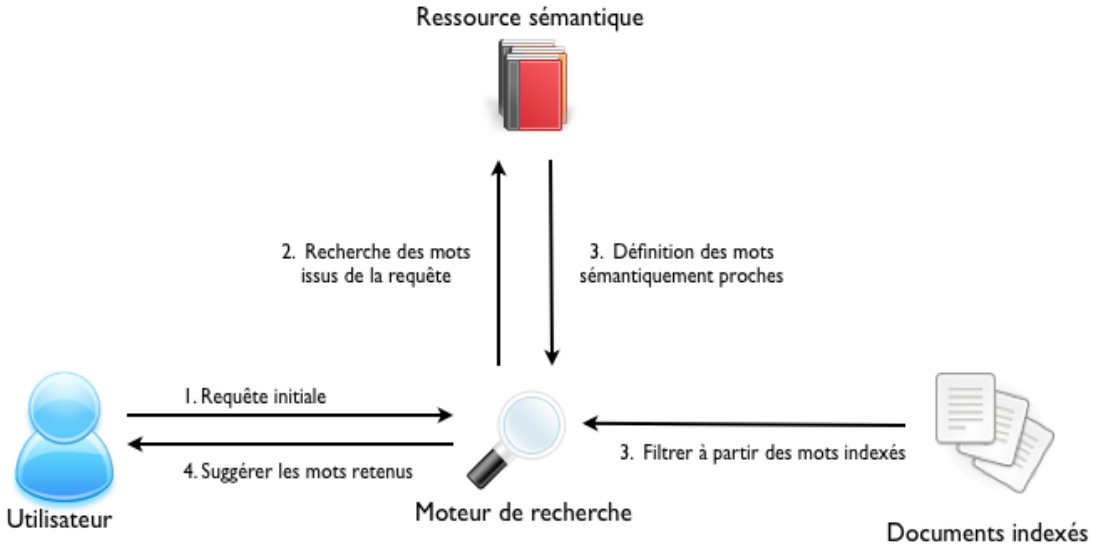


FIGURE 1.10 – **Extensions de requête à partir d'une ressource sémantique.**

De nombreux travaux en Recherche d'Informations (RI) utilisent cette approche. Une majorité d'entre eux étendent leur requête à partir de la ressource WordNet[Voorhees, 1994, Fellbaum et al., 1998]. WordNet [4] est une base de données lexicales qui a été développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. Cette ressource répertorie, classifie et met en relation, de diverses manières, le contenu sémantique et lexical de la langue anglaise. Ainsi avec cette ressource, il est possible d'enrichir la requête initiale à partir de relations lexicales contenues dans WordNet tels que ; la synonymie, l'antonymie, l'hyponymie, la métonymie ou encore la morphologie.

D'autres études telles que [Attar and Fraenkel, 1977, Qiu and Frei, 1993, Hersh et al., 2000] utilisent un thesaurus pour étendre leurs requêtes. Un thesaurus regroupe un ensemble terminologique de manière structurée et celui-ci est défini pour un domaine plus spécifique. D'une manière générale, la requête de l'utilisateur est étendue à partir des relations principales contenues par le thesaurus : hiérarchique, d'équivalence et d'association.

4. WordNet : http ://wordnet.princeton.edu/. Date : 5/04/2010

L'utilisation d'ontologies en Recherche d'Informations [Fu et al., 2005] permet la formulation de requêtes et l'indexation des contenus de documents à partir des mots-clés utilisés dans une ontologie. Ainsi, les termes de la requête sont associés aux concepts de l'ontologie afin de guider l'utilisateur dans le choix des termes qu'il souhaite étendre à travers de relations dites génériques (« instance_of », « part_of » et « is_a ») ou possédant une sémantique propre définie par l'ontologie concernée. Par exemple, certains travaux comme [Carpineto and Romano, 1996, Messai et al., 2005], utilisent la relation hiérarchique « is_a » afin d'affiner la requête de l'utilisateur par généralisation ou encore par spécialisation pour obtenir de meilleurs résultats dans la recherche de documents. De plus, les techniques utilisant les ontologies limitent le risque d'ambiguïté présent dans la requête car les termes de la requête sont associés aux concepts de l'ontologie réduisant ainsi le risque de bruit (le bruit faisant référence aux documents non pertinents).

1.2 La représentation de la connaissance

1.2.1 Introduction

D'une façon générale, l'utilisation de la connaissance en informatique vise à permettre aux machines de ne plus utiliser l'information de manière totalement aveugle mais, au contraire, de la comprendre et de pouvoir l'interpréter afin d'offrir une interaction et une coopération entre les utilisateurs et le système. Pour ce faire, il est nécessaire que le système puisse avoir non seulement accès aux termes manipulés par l'humain mais aussi d'en comprendre leurs sémantiques en vue d'une communication plus efficace.

Ainsi, l'ingénierie des connaissances (IC) est une discipline rassemblant un ensemble de méthodologies issues de l'intelligence artificielle et permettant d'acquérir, de modéliser, de stocker et de manipuler la connaissance au travers des mécanismes de raisonnements.

Représenter la connaissance a pour objectif de modéliser la connaissance d'un domaine en omettant certains détails non significatifs pour permettre une meilleure manipulation [Kayser, 1997]. Mais pour concrétiser cette représentation, il est nécessaire de partir de ressources renfermant cette connaissance et à partir desquelles cette connaissance est extraite.

Cependant, avant de parvenir à extraire et manipuler cette connaissance, il est impératif de comprendre comment l'informatique définit cette connaissance, d'où

celle-ci provient et comment elle est représentée. Pour commencer, nous établirons les différences sémantiques qui distinguent la connaissance d'une information et d'une donnée (section 1.2.2). Ensuite (section 1.2.3), nous parcourrons diverses ressources textuelles à partir desquelles la connaissance d'un domaine est représentée. Pour terminer, nous expliquerons les langages du Web sémantique ainsi que la logique de descriptions (LD) utilisées pour représenter, implémenter et interroger la connaissance d'un domaine (section 1.2.4).

1.2.2 Données - informations - connaissances

Il est important de bien distinguer les différences sémantiques se cachant derrière les termes : données, informations et connaissances selon leur utilisation dans le domaine informatique. De manière théorique Kayser [Kayser, 1997] définit ces trois termes tels que :

- les « données » sont le résultat d'observations ;
- les « informations » représentent le résultat d'une interprétation effectuée sur ces données ;
- les « connaissances » résultent de l'utilisation d'informations.

De manière beaucoup plus formelle [Schreiber et al., 1999] ;

- une « donnée » = signes + syntaxe ;
- une « information » = Données + une sémantique ;
- une « connaissance » = informations + capacité d'utiliser l'information.

Ainsi, on remarque qu'une donnée est une description élémentaire pouvant être représentée par un mot, une image, un symbole ou encore par un nombre. Cependant, une donnée à elle seule n'a aucun sens en tant que telle pour une machine ou pour un humain. Il est donc nécessaire de rattacher cette donnée à un contexte sémantique afin d'en tirer une information.

L'information contextualise une donnée en lui rajoutant une certaine sémantique, ce qui lui donne un sens. Ainsi, dire qu'un « chat est un félin » ou que « 20 est une température » ajoute une sémantique qui permet de comprendre et de lier la donnée avec le contexte dans laquelle elle s'inscrit et donc d'en tirer une information. L'utilisation d'un ensemble d'informations afin d'en déduire de nouvelles au moyen de raisonnements constitue la connaissance.

Le raisonnement appliqué sur des informations consiste à utiliser des mécanismes de déduction et d'induction. Ceux-ci sont largement implémentés et utilisés en intelligence artificielle (AI) afin d'en dégager des unités de connaissances souvent utilisées par les humains. Par exemple, si l'on sait que « un félin est un mammifère » et que l'on a l'information : « un chat est un félin », on peut dès lors déduire par inférence la connaissance qu'un chat est aussi un mammifère.

1.2.3 Ressources textuelles

Corpus de textes

Un corpus de textes constitue la ressource principale pour l'étude d'un domaine spécifique. Nous entendons par « corpus de textes », un ensemble de documents regroupés dans une optique précise. Sinclair [Sinclair, 1996] le définit comme étant « *une collection de morceaux de langage qui sont sélectionnés et organisés selon des critères linguistiques explicites pour servir d'échantillon du langage* ». Feldman [Feldman and Sanger, 2007] précise que « *la sélection d'un corpus peut être faite pour une application ou une tâche particulière* ». Dans ce travail, nous considérerons un corpus de textes comme étant un ensemble de documents traitant d'un même domaine spécifique afin d'en représenter au mieux la connaissance qu'ils possèdent.

La constitution d'un corpus de textes dépend de l'utilisation qui en sera faite. En ingénierie des connaissances (IC), les corpus de textes constituent l'une des ressources de données les plus importantes et les plus souvent utilisées pour l'extraction de connaissances dans le domaine de la linguistique et du traitement automatique des langues (TAL) afin d'assurer une compréhension des contenus des documents. Ainsi, si le but visé d'un tel corpus est d'en tirer les connaissances du domaine spécifique qu'il traite, alors deux solutions sont envisageables [Lame, 2002]. La première consiste à récolter l'ensemble des ressources textuelles à partir du Web au travers de requêtes posées à un moteur de recherche décrivant le domaine recherché. Cette méthode nécessite une attention et un contrôle particulier sur l'origine des documents récoltés. De plus, il arrive parfois, selon le domaine spécifique visé, de ne pas obtenir un nombre suffisant de documents faute de disponibilité publique ou encore du choix d'un domaine trop spécifique. La deuxième solution est de demander aux experts de définir eux-mêmes le corpus de textes traitant du domaine spécifique visé.

Thesaurus

Autrefois un thesaurus désignait, en grec ancien, « un trésor » représentant un dictionnaire de langue ancienne à but philologique ou archéologique mais au fil du temps, il finit par désigner un thesaurus documentaire. En informatique, les thesaurus sont toujours en quelque sorte un trésor visant à rassembler l'ensemble des informations disponibles et connues pour un domaine spécifique mais agencées de manières structurées. Ainsi, un thesaurus est un vocabulaire contrôlé qui regroupe un ensemble de concepts relatifs à un certain domaine. Il constitue un moyen de décrire ce domaine, d'en définir les concepts et de fixer la terminologie utilisée par un groupe de personnes. Les concepts représentés par des termes, peuvent être utilisés pour l'indexation de documents dans une banque de données bibliographiques ou dans un catalogue de centre de documentation à des fins de recherche documentaire. Dans ce cas, deux types de termes composent un thesaurus :

- les **descripteurs** sont utilisés pour indexer un document.

- les **non-descripteurs** ne peuvent être employés pour indexer un document et renvoient à un descripteur à utiliser. Ils correspondent à des synonymes, quasi-synonymes, abréviations ou variantes orthographiques du concept retenu comme descripteur. Les non-descripteurs font donc partie du thesaurus et peuvent être intégrés à un moteur de recherche afin de faciliter le repérage d'information.

Exemple :

Descripteur	**Non-Descripteur**
Abandon scolaire	Abandon d'études Abandon des études Abandon en cours d'études Décrochage Décrochage scolaire

Un concept, défini pour un thesaurus, peut être représenté en trois types de termes que contient un thesaurus :

- les termes **génériques** représentés généralement par le sigle **TG** et qui désignent les entités ou concepts principaux en référence aux autres termes et au domaine considéré.

– les termes **spécifiques** représentés généralement par le sigle **TS** et qui précisent et identifient les entités ou concepts particuliers à l'intérieur du champ sémantique d'un terme générique donné.

– les termes **équivalents** représentés généralement par le sigle **EP** sont une variante des termes spécifiques.

Les termes d'un thesaurus sont organisés hiérarchiquement. Cette hiérarchie permet de régler la précision de l'indexation ou de l'interrogation. L'indexation s'appuiera autant que possible sur l'identification des termes spécifiques (donc du niveau le plus bas possible), alors que la recherche selon les cas pourra faire appel aux termes génériques pour augmenter le nombre de réponses.

Les relations quant à elles se distinguent parmi plusieurs types :

– les relations *hiérarchiques*, base de la hiérarchie du thesaurus ;
  – **NT** définit un terme plus spécifique (*narrowed term*).
  – **BT** définit un terme plus général (*broader term*).

– les relations *d'équivalence*, base de l'univocité ;
  – **RT** définit un terme en relation (*related term*).

– les relations *spécifiques* ;
  – **UF** définit un terme utilisé (*used for*).
  – **U** définit un terme non utilisé (*unused*) et correspond à la relation inverse de UF.

Mais la plupart des thesaurus définissent également d'autres types de relations et de termes pouvant être ajoutés selon le domaine qui les concerne afin d'en enrichir et d'en améliorer leur usage. On peut notamment prévoir des équivalents linguistiques pour des thesaurus multilingues ainsi que des passerelles avec d'autres thesaurus du même domaine ou de domaines différents.

Cependant, cette structuration de l'information rend tout thesaurus peu formel et ne précise pas réellement et totalement les liens entre les termes de façon non ambigüe. Les relations entre termes restent vagues et ambigües. La relation « RT » est souvent difficile à exploiter, car elle connecte des termes en sous-entendant différents types de relations sémantiques. Par exemple, à travers une relation « RT », nous pouvons savoir que le terme « Main » est en relation avec le terme « Doigt », mais ne nous précise pas la sémantique de la relation. Il est souvent difficile de

déterminer les propriétés des relations « NT », « BT » qui peuvent regrouper les relations « est une instance de » ou « est une partie de ». Les thesaurus manquent également de consistance et peuvent contenir des informations contradictoires. Les thesaurus sont finalement difficilement exploitables dans des procédés automatiques à cause de ces différentes limites. Les ontologies, quant à elles, permettent de pallier ces manques.

Wordnet [5] est une base lexicale électronique développée depuis 1985 par l'université de Princeton par une équipe de linguistes. Ce thesaurus est une sorte de dictionnaire regroupant des termes synonymes en des ensembles *Synset*. Ceux-ci sont reliés au travers de différentes relations sémantiques et syntaxiques. Wordnet est défini comme une ressource linguistique et correspond à l'une des plus utilisées à ce jour.

Ontonet [6] est un système mis en place pour le développement d'ontologies lexicales à partir de Wordnet comprenant le niveau linguistique des termes ainsi que le niveau ontologique des concepts [REISCHER, 2007]. Ce système comprends deux composants majeures : le **Browser** offrant la possibilité de visualiser et de consulter la base lexicale de Wordnet et le **Weaver** permettant d'étendre ou de définir la construction d'un réseau syntaxique ou sémantique.

Proposée et maintenue par la NLM [7] (National Library of Medicine), Unified Medical Langage System [8] (**UMLS**) est la ressource terminologique la plus large actuellement disponible pour la médecine. Elle est le résultat de la fusion de plus d'une centaine de thesaurus, tels que MeSH [9] et SNOMED [10], décris en plusieurs langues, dont elle préserve les réseaux de relations entre termes. Ainsi, UMLS est qualifié comme méta-thesaurus et appelé aussi vocabulaires intégrés.

Ontologie

Il est difficile de nos jours d'attribuer une définition unique pouvant décrire avec exactitude ce qu'est une ontologie. Comme le décrit [Cimiano, 2006], le mot est employé dans des contextes différents et touche à de nombreux domaines tels que la philosophie, la linguistique ou encore l'intelligence artificielle.

5. WordNet : http ://wordnet.princeton.edu/. Date : 5/04/2010
6. Ontonet : http ://www.ontonet.de/. Date 16/07/2010.
7. NLM : http ://www.nlm.nih.gov/. Date :24/05/2010.
8. UMLS : http ://www.nlm.nih.gov/research/umls/. Date : 24/05/2010
9. MeSH : http ://www.ncbi.nlm.nih.gov/mesh. Date 24/05/2020.
10. SNOMED : http ://www.ihtsdo.org/snomed-ct/. Date : 24/05/2010.

En informatique, l'ontologie se définit comme « *une spécification explicite et formelle d'une conceptualisation partagée* » [Gruber et al., 1993] :

- l'expression « **spécification explicite et formelle** » signifie que le modèle en question doit être décrit de façon non ambigüe dans un langage formel pour pouvoir être manipulé par un logiciel aussi bien que par un humain.

- le terme « **conceptualisation** » correspond à un « modèle abstrait » d'une partie du monde réel sur lequel doit travailler le système considéré et qui se présente comme un ensemble de définitions de concepts muni de propriétés et de relations entre ces concepts.

- « **partagée** » signifie que les concepts définis le sont de façon consensuelle.

Autrement dit, une ontologie est un cadre formel pour la modèlisation des connaissances reposant sur des concepts définis par leurs attributs ainsi que par les relations avec les autres concepts qu'ils possèdent.

De manière formelle une ontologie est définie dans [Stumme and Maedche, 2001] comme une structure telle que :

$$\mathcal{O} := (C, \sqsubseteq_C, R, A, \tau) \tag{1.3}$$

- C est l'ensemble des concepts ;
- $\sqsubseteq_C$ est un ordre partiel sur C ;
- R ensemble des relations sur C x C ;
- τ ensemble des datatypes (entier, chaîne de caractères, etc...) ;
- A ensemble des propriétés sur C x τ ;

Un lexique pour un schéma d'ontologie $\mathcal{O}$:= (C, $\sqsubseteq_C$, R, A, τ) est une structure [Cimiano, 2006] :

$$Lex := (S_C, S_R, S_A, Ref_C, Ref_R, Ref_A) \tag{1.4}$$

où :
- les trois ensembles S_C, S_R, S_A sont appelés « instances » pour les concepts, les relations et les attributs respectivement,
- la relation $Ref_C \subseteq S_C$ x C est appelée référence lexicale pour les concepts,
- la relation $Ref_R \subseteq S_R$ x R est appelée référence lexicale pour les relations,
- la relation $Ref_A \subseteq S_A$ x A est appelée référence lexicale pour les attributs.

$$\forall s \in S_C, Ref_C(s) := \{c \in C | (s, c) \in Ref_C\}, \tag{1.5}$$

$$\forall c \in C, Ref_C^{-1}(s) := \{s \in S_C | (s, c) \in Ref_C\}. \tag{1.6}$$

Ref_R et Ref_A sont définies de la même façon que Ref_C.

Une ontologie formelle est ainsi constituée d'une collection de noms pour les types de concepts et de relations. La manipulation des concepts ou éléments de l'ontologie est guidée par l'ensemble des propriétés qui leur sont attachées et l'ensemble des relations qui définissent la structuration de l'ontologie. Cette structuration utilise essentiellement la relation de subsomption « is_a » définissant le lien de généralisation entre concepts et « choisie comme relation de structuration de l'arborescence ontologique » [Charlet et al., 2003]. D'autres relations permettent d'unir les concepts pour la construction d'une modèlisation conceptuelle plus complexe.

L'ontologie de la figure 1.11 nous présente une petite ontologie sur les animaux. Les concepts qu'elle contient sont représentés par des figures ovales tels que « Animal », « Mammifère », « Chien », « Plumes », etc. Les concepts les plus élémentaires dans un domaine devraient correspondre aux concepts racines des divers arbres taxonomiques. Néanmoins, dans le monde des ontologies, chaque individu est membre du concept « owl :Thing ». Ainsi, chaque concept défini par l'utilisateur est donc implicitement un sous-concept de « owl :Thing ». L'ensemble de ces concepts est structuré à travers la relation « is_a ». Les concepts « Pelage » et « Membre » sont liés respectivement au concept « Animal » au travers des relations « estRecouvertDe » et « seDéplaceAvec ». L'ontologie définit également l'attribut « enPossède » pour le concept « Membre », « estAgéDe » pour le concept « Animal » et représente les instances de concepts : « Nemo », « Brutus », « Willy », « Jack » et « Cocotte ».

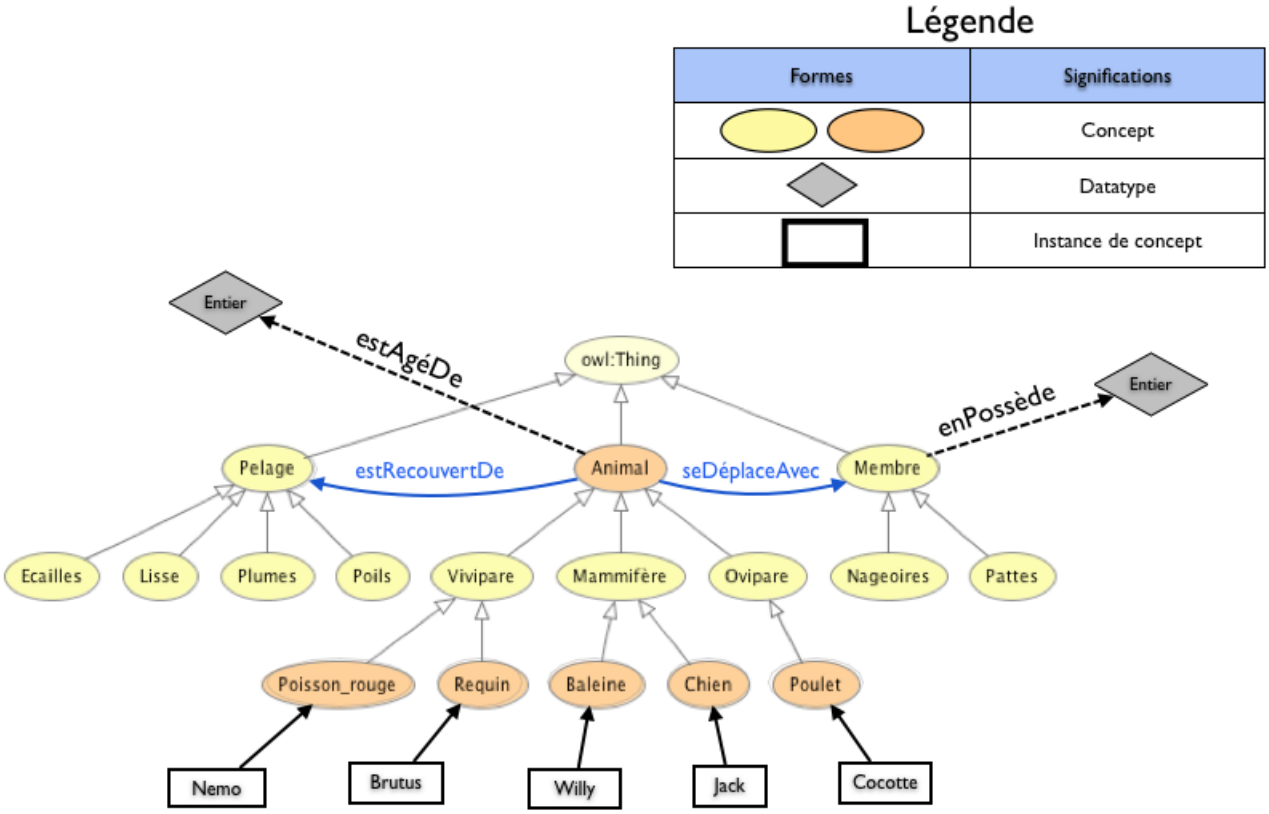


FIGURE 1.11 – **Exemple d'ontologie.**

1.2.4 Les langages du Web sémantique

L'apparition du Web apporta son lot de technologies et de langages de programmation pour la création et la publication de pages web (HTML, CSS, PHP, Javascript, etc...). Ces technologies permettent de représenter l'information au travers de pages web de façon totalement syntaxique. Ainsi, lorsque l'on crée un site internet, le webmaster doit se contenter d'agencer l'information qu'il souhaite publier à travers diverses balises, liens hypertextes ainsi que le code définissant l'affichage souhaité.

Dans l'un des articles fondateur du Web sémantique [Berners-Lee and Hendler, 2001], Hendler et Tim Berners-Lee le définissent à travers l'assertion suivante : « *The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation* ».

Leur vision exprime que le Web sémantique a pour but d'apporter du sens à l'information contenue dans les pages web. Cette contribution permettrait ainsi, que la sémantique contenue par des pages web puisse être interprétée et exploitée par les machines. L'objectif est donc de transformer les pages web afin que l'information qu'elles contiennent puisse être interprétée par nos machines et non plus uniquement lisible par des êtres humains. On peut dire que le potentiel du Web Sémantique aujourd'hui est comparable à celui du lien hypertexte aux débuts d'Internet.

Pour ce faire, le W3 Consortium [11] (W3C) a développé des standards permettant de rajouter du sens aux pages internet. Ainsi, la mise en oeuvre de cette vision se concrétisa au travers de l'apparition d'un ensemble de langages tels que RDF [12], RDFS [13] et OWL [14] reprenant une syntaxe XML et permettant de rendre utilisable les informations diffusées sur la toile par nos machines. Le gâteau sémantique (figure 1.12) nous montre l'ensemble des langages sémantiques mis en oeuvre par le W3C pour concrétiser la vision du Web sémantique.

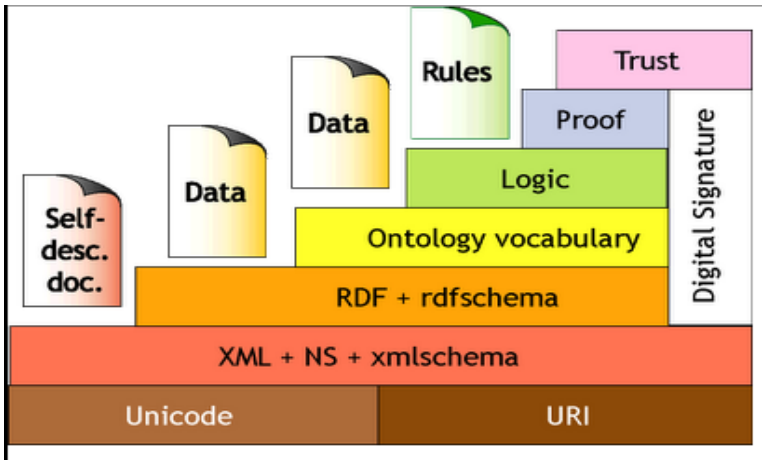


FIGURE 1.12 – **Le gâteau du web sémantique.**

La feuille de route du Web sémantique écrite en 1998 par Tim Berners-Lee [Berners-Lee, 1998] décrit un ensemble de spécifications en couches dont la représentation actuelle se trouve ci-dessus (figure 1.12). Dans la suite de cette partie, nous nous attarderons essentiellement sur les trois couches centrales à

11. W3C : http ://www.w3.org/. Date : 5/04/2010
12. http ://www.w3.org/RDF/. Date : 5/04/2010
13. http ://www.w3.org/TR/rdf-schema/. Date : 5/04/2010
14. http ://www.w3.org/TR/owl-features/. Date : 5/04/2010

savoir : RDF + rdfschema, Ontology vocabulary (owl) et Logic (logiques de descriptions).

Pour terminer, on peut considérer que le Web sémantique pourrait aboutir à un immense système réparti d'intelligence artificielle même si certaines personnes pensent que le Web sémantique, tel qu'on l'imagine, ne verra jamais le jour. En tout état de cause, les perspectives d'ouverture sont immenses.

Resource Description Framework

Resource Description Framework (RDF [15]) est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, afin de permettre le traitement automatique de celles-ci. Développé par le W3C, RDF est le langage de base du Web sémantique et se fonde sur une syntaxe RDF/XML.

En annotant des documents non structurés et en se servant d'interfaces pour des applications, RDF permet une certaine interopérabilité entre des applications échangeant de l'information non formalisée et non structurée sur le Web.

Un document RDF est structuré par un ensemble de triplets. Un triplet est une association : sujet, prédicat, objet où :

- le sujet représente la ressource à décrire ;
- le prédicat représente un type de propriété applicable à cette ressource ;
- l'objet représente une donnée ou une autre ressource : c'est la valeur de la propriété.

Un document RDF ainsi formé correspond à un multi-graphe orienté et étiqueté. Chaque triplet correspond alors à un arc orienté dont le label est le prédicat, le nœud source est le sujet et le nœud cible est l'objet. Par exemple, pour exprimer que « *le chien Jack est un animal recouvert de poils* »on aura ;

- Sujet : Jack
- Prédicat : estRecouvertDe
- Objet : Poils

15. http ://www.w3.org/RDF/. Date : 5/04/2010

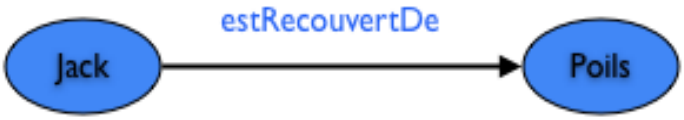


FIGURE 1.13 – **représentation du triplet sous forme de graphe.**

Appliqué dans le monde du web, le sujet correspond à une URI [16] (Uniform Ressource Identifier). Les URI sont une norme du W3C qui définit chaque ressource par une chaîne de caractère unique (comme la ressource rdf : « http ://www.w3schools.com/rdf »). Le prédicat est en quelque sorte une propriété possédée et appliquée à la ressource définie en tant que sujet et représentée au travers d'un nom tel que « estRecouvertDe ». L'objet d'une propriété est représentée par une valeur qui peut elle-même être ressource. Ainsi, la représentation du graphe (figure 1.13) en langage RDF nous donne :

```
<rdf:Description rdf:about= "http://www.exemple.com/Jack">
<estRecouvertDe> Poils </estRecouvertDe>
</rdf:Description>
```

Le langage RDF permet, à travers son formalisme, d'exprimer la logique du premier ordre avec des prédicats possédant une arrité de un et deux. Il permet également d'exprimer le quantificateur existentiel ainsi que la conjonction.

Resource Description Framework Schema

Le langage RDFS (RDF-Schema) est une extension du langage RDF. Plus expressif, il ajoute la possibilité de définir des concepts, de prendre en compte les hiérarchies d'héritage de concepts et d'attributs ainsi que la définition du domaine et du co-domaine d'une relation. Nous reprenons les avantages de RDFS présentés dans [Horrocks et al., 2003] à travers l'ontologie illustrée en figure 1.11 :

- déclarer des concepts comme : « **Animal** », « **Pelage** », « **Chien** », etc... ;
- déclarer que « **Chien** » est un sous-concept de « **Mammifère** » ;
- déclarer que « **Jack** » est une instance de « **Chien** » ;

16. http ://fr.wikipedia.org/wiki/Uniform_Resource_Identifier. Date : 05/04/2010

– déclarer que « **estRecouvertDe** » est une relation entre « **Animal** » (domaine) et « **Pelage** » (co-domaine) ;
– déclarer que « **enPossède** » ; est un attribut de « **Membre** » ;
– déclarer que « **Jack** » est une instance de « **Chien** » et dont l'attribut « **estAgéDe** » possède comme valeur « **8** » ;

L'exemple décrit par la figure 1.14, nous présente en langage RDFS les concepts « **Animal** », « **Mammifère** » et « **Chien** » avec les propriétés « **estRecouvertDe** », « **seDéplaceAvec** » et l'instance « **Jack** » :

```
<rdfs:Class rdf:about="http://www.exemple.com/Animaux#Mammifère">
        <rdfs:subClassOf rdf:about="http://www.exemple.com/Animaux#Animal"/>
</rdfs:Class>

<rdfs:Class rdf:about="http://www.exemple.com/Animaux#Chien">

        <rdf:Property rdf:about= "http://www.exemple.com/Animaux#estRecouvertDe">
                <rdfs:Domain rdf:Resource= "http://www.exemple.com/Animaux#Poils"/>
        </rdf:Property>

        <rdf:Property rdf:about= "http://www.exemple.com/Animaux#seDéplaceAvec">
                <rdfs:Domain rdf:Resource= "http://www.exemple.com/Animaux#Pattes"/>
        </rdf:Property>

        <rdfs:subClassOf rdf:about="http://www.exemple.com/Animaux#Mammifère"/>
</rdfs:Class>

<Chien rdf:ID= "http://www.exemple.com/Animaux#Jack"/>
```

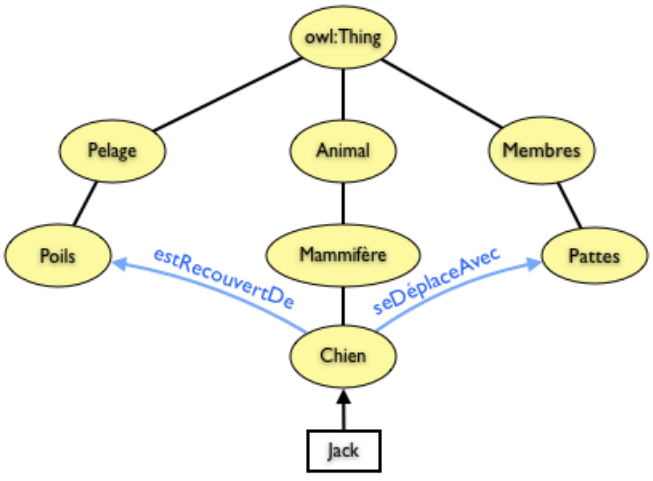


FIGURE 1.14 – **Description du concept « Chien » dans le langage RDFS.**

Cependant, malgré son expressivité plus importante que RDF, RDFS ne permet pas de savoir si un objet peut être à la fois **Mammifère** et **Ovipare**, ce qui serait quelque peu problématique...

Les logiques de descriptions

Les logiques de descriptions (LD) aussi appelées logiques descriptives [Baader et al., 2003] sont une famille de formalisme de représentation de connaissances qui sont utilisées pour modéliser la connaissance terminologique d'un domaine d'application de manière formelle et structurée. Le nom de logique de descriptions se rapporte, d'une part à la description de concepts utilisés pour décrire un domaine et d'autre part à la sémantique fondée sur la logique qui peut être donnée par une transcription en logique des prédicats du premier ordre. La logique de descriptions a été développée comme une extension des frames et des réseaux sémantiques qui ne possédaient pas de sémantique formelle fondée sur la logique. Les entités principales de ce formalisme sont les concepts, les instances (individus), et les rôles/attributs (ou propriétés). Un concept est représenté par un ensemble d'instances partageant les mêmes attributs/propriétés. Les trois idées suivantes ont largement façonné le développement des descriptions logiques :

- les éléments fondamentaux : les concepts atomiques, les rôles atomiques et les individus ;
- la puissance expressive du langage est limitée à travers l'utilisation d'un ensemble relativement restreint de constructeurs adéquats pour construire des définitions complexes de concepts à partir de concepts atomiques ;
- la possibilité de déduire de nouvelles connaissances à partir de concepts et des individus. Particulièrement, à partir de la relation de subsomption entre concepts et la relation d'instance entre concepts et individus.

La modélisation des connaissances d'un domaine avec les LD se réalise en deux niveaux. Le premier est le niveau terminologique « Terminological Box » ou TBox qui décrit les connaissances générales d'un domaine. Le second niveau est le niveau factuel « Assertional Box » ou ABox qui représente une configuration précise. Une TBox comprend la définition des concepts et des rôles, alors qu'une ABox décrit les individus en les nommant et en spécifiant en terme de concepts et de rôles, des assertions qui portent sur ces individus nommés. Plusieurs ABox peuvent être associées à une même TBox ; chacune représente une configuration constituée d'individus et utilise les concepts et rôles de la TBox pour l'exprimer. La TBox et la ABox constituent à elles deux une base de connaissances (BC). La sémantique associée à une BC en LD s'exprime généralement sous forme inspirée de la théorie

des modèles. A travers ces notions, les descriptions de concepts considérées dans ce mémoire se composent selon les règles de syntaxe décrites par la table 1.1.

A, B →	$\top$	(concept universel ou top)
	$\bot$	(concept vide ou bottom)
	P	(nom de concept)
	$\neg$P	(négation de concept)
	A $\sqcap$ B	(conjonction de concepts)
	A $\sqcup$ B	(disjonction de concepts)
	$\forall$r.C	(restriction universelle)
	$\exists$r.C	(restriction existentielle)

TABLE 1.1 – **Syntaxe des descriptions de concepts.**

La majorité des règles définies se rapporte à la théorie des ensembles, néanmoins nous détaillons de manière plus précise la restriction universelle et existentielle :

- La restriction existentielle exprime qu'un individu x appartient à $\exists$r.C si et seulement si x est relié par le rôle r à un individu y appartenant à C.
- La restriction universelle exprime qu'un individu x appartient à $\forall$r.C si et seulement si pour chaque y, si x et y sont reliés par la relation R, alors y est un C si et seulement si x n'est relié, par R, à rien d'autre qu'aux C.

Pour une meilleur compréhension, la figure 1.15 nous montre la distinction qu'il existe entre les deux restrictions.

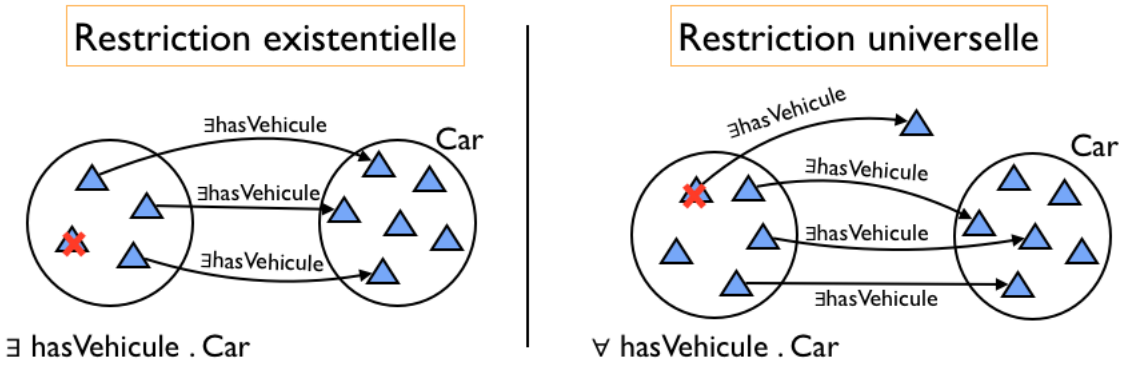


FIGURE 1.15 – **Schéma d'une restriction existentielle et universelle.**

Pour illustrer ces principes nous reprenons une partie de l'ontologie présentée en figure 1.11 que nous représentons en logique de descriptions (table 1.2).

TBox	**ABox**
Animal, Pelage $\sqsubseteq \top$	Poisson_Rouge(Bubulle)
Mammifère, Vivipare $\sqsubseteq$ Animal	Chien(Jack)
Chien $\sqsubseteq$ Mammifère	
Poisson $\sqsubseteq$ Vivipare	
Poils, Ecailles $\sqsubseteq$ Pelage	
Chien $\equiv$ Animal $\sqcap \forall$ estRecouvertDe.Poils	
Poisson_Rouge $\equiv$ Animal $\sqcap \forall$ estRecouvertDe.Ecailles	
estRecouvertDe $\sqsubseteq \top_R$	

TABLE 1.2 – **Exemple d'une base de connaissances composée d'une TBox et d'une ABox.**

Les entités atomiques

Les éléments de la TBox sont constitués des concepts atomiques et des rôles atomiques. Les noms commençant par une majuscule désignent les concepts (Animal, Mammifère, etc.), les autres désignent les rôles (estRecouvertDe).

Les concepts et rôles atomiques prédéfinis

Les LD prédéfinissent au minimum quatre concepts atomiques :
- le concept $\top$ et le rôle $\top_R$ correspondent tous les deux respectivement au concept et au rôle (propriété) universel les plus généraux.
- le concept $\bot$ ainsi que le rôle $\bot_R$ correspondent, quant à eux, au concept et au rôle les plus spécifiques.

Les entités composées

Les concepts et rôles atomiques peuvent être combinés au moyen de constructeurs pour former respectivement des concepts et des rôles composés. Les concepts « *Chien* » et « *Poisson_ Rouge* » sont des concepts « définis » car leurs conditions sont nécessaires et suffisantes. Les différentes LD se distinguent par les constructeurs qu'elles proposent. Plus les LD sont expressives, plus les chances sont grandes que les problèmes d'inférence soient non décidables ou de complexité très élevée. Par contre, les LD trop peu expressives démontrent une inaptitude à représenter des domaines complexes.

Ontology Web Language

Ontology Web Language (OWL) est un langage qui a été conçu afin d'être utilisé par les applications cherchant à traiter le contenu de l'information et non plus uniquement à présenter l'information. OWL facilite l'interprétation du contenu Web pour la machine par rapport aux langages XML,RDF et RDFS en fournissant un vocabulaire supplémentaire ainsi qu'une sémantique formelle. Cependant, OWL n'a pas été défini à partir de zéro et il a été conçu comme une extension du langage RDFS.

Comme pour le langage RDF et RDFS, [Horrocks et al., 2003] nous présentons les avantages d'OWL par rapport au langage RDFS :

- déclarer que les concepts « **Mammifère** », « **Ovipare** » et « **Vivipare** » sont disjoints ;
- déclarer que les instances « **Jack** » et « **Cocotte** » sont distinctes ;
- déclarer que « **estRecouvertDe** » est la relation inverse de « **recouvre** » ;
- déclarer que le concept « **Chien** » doit posséder au moins 1 valeur de la relation « **estRecouvertDe** » et de co-domaine « **Pelage** » ;
- déclarer que l'attribut « **estAgéDe** » est un attribut fonctionnel ce qui exprime que chaque individu possède une valeur de l'attribut « **estAgéDe** » ;

OWL est subdivisé en trois sous-langages définis en fonction de leur niveau d'expressivité. Cette subdivision résulte du fait que OWL ne pouvait être défini comme un seul langage destiné aux ontologies en raison de l'importance du nombre des exigences que le langage suscitait.

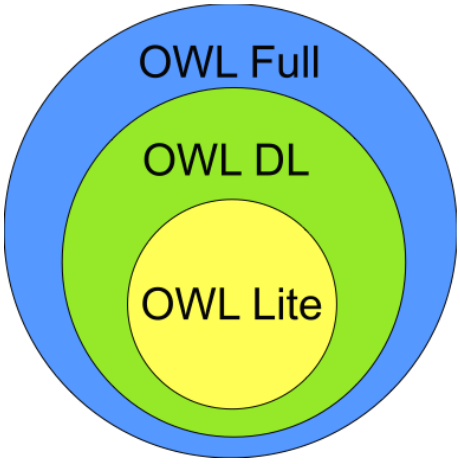


FIGURE 1.16 – **Le langage Owl.**

1. **Owl-Lite.** OWL-lite est le sous-langage le plus simple expressivement des trois. Il intègre la hiérarchie simple de classe ainsi que de simples contraintes possédant une limite de cardinalité de 0 ou 1. Par exemple, nous pourrions exprimer qu'un homme possède 0 ou 1 voiture mais nous ne pourrions pas exprimer qu'un homme puisse avoir plus d'une voiture en sa possession car sa cardinalité serait supérieur à un.

2. **Owl-DL.** Plus expressif que Owl-Lite, il est le plus souvent utilisé. Il restreint les constructions RDF possibles afin de rendre le langage décidable mais inversement lui fait perdre sa compatibilité avec un document RDF correct. De plus, il s'appuie sur les logiques de descriptions (« DL ») qui permettent à des raisonneurs tel que « Pellet [17] », d'utiliser des règles d'inférences sur les ontologies. Contrairement à Owl-Lite, la cardinalités n'est plus limité à 0 ou 1 et il permet d'exprimer le complément et l'union de classes.

3. **Owl-Full.** Celui-ci est le plus expressif de tous et contient toutes les primitives de RDF. Il est utilisé lorsque l'expressivité nécessaire est très élevée et pouvant être plus importante que la prise de décision. Owl-Full est le langage le plus général, n'imposant aucune restriction sur le vocabulaire (par exemple un même terme peut à la fois désigner une classe et une instance), ce qui le rend complètement compatible à RDF mais implique que Owl-Full soit indécidable.

Différences entre Owl-DL et Owl-Full. Leurs différences résident dans les restrictions sur l'utilisation de certaines des caractéristiques et sur l'utilisation des fonctionnalités RDF. Owl-Full permet de mélanger librement du Owl avec du RDF-Schema. Ce dernier n'applique pas une stricte séparation des classes, des propriétés, des instances et des attributs. A l'inverse, Owl-DL définit des contraintes sur le mélange avec RDF et exige que les classes, les propriétés et les attributs soient disjoints. Ainsi, une classe peut être traitée simultanément comme une collection d'individus et comme un individu si l'on utilise le lange Owl-Full mais ne serait pas accepté par le langage Owl-DL. De plus Owl-Full, permet un raisonnement moins prévisible que Owl-DL du fait qu'aucune implémentation complète du langage n'existe ce qui implique une moindre facilité d'exploitation. Pour plus d'informations sur OWL , se reporter au site du W3C [18].

Afin d'implémenter les ontologies dans le langage OWL, l'utilisation d'un éditeur d'ontologies est nécessaire. Le plus célèbre et le plus populaire reste l'éditeur

17. Pellet : http ://clarkparsia.com/pellet. Date 06/04/2010
18. http ://www.w3.org/TR/owl-features/. Date : 5/05/2010

Protégé [19]. Plateforme open-source, **Protégé** fournit une suite d'outils définie pour la construction et la gestion des ontologies. Il est également une librairie Java qui peut être étendue pour créer de véritables applications à bases de connaissances en utilisant un moteur d'inférence pour raisonner et déduire de nouveaux faits par application de règles d'inférence aux instances de l'ontologie et à l'ontologie elle même.

SPARQL

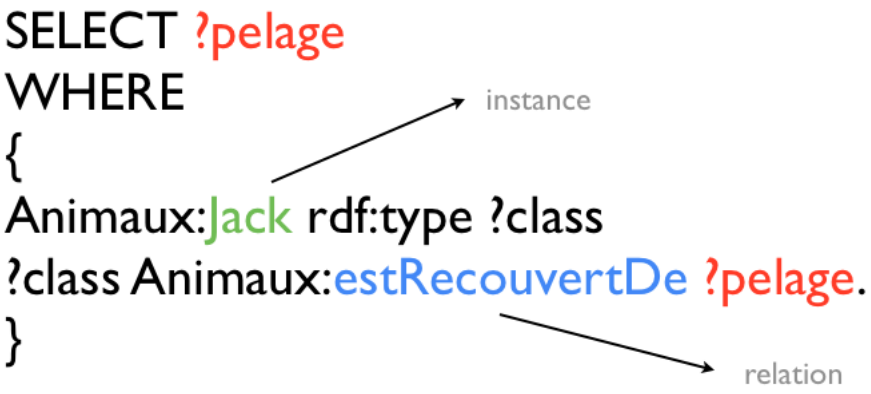
Le langage RDF ainsi que ses extensions (RDFS, OWL) sont pour la plupart utilisés pour représenter, entre autres, des informations personnelles, des réseaux sociaux, ainsi que pour fournir un moyen d'intégration par rapport aux sources d'informations disparates existantes.

Cette spécification définit la syntaxe et la sémantique du langage de requête SPARQL [20].

SPARQL est un langage d'interrogation et un protocole d'accès aux données pour le Web sémantique, devenu le 15 Janvier 2008, une recommandation W3C. Celui-ci définit la syntaxe et la sémantique nécessaires à l'expression de requêtes sur une base de données de type RDF, RDFS mais aussi OWL ainsi que la forme possible des résultats. De plus, la syntaxe SPARQL s'inspire fortement de celle de SQL pour la construction de ces requêtes tels que : « SELECT » et « WHERE ».

RDF est construit sur base d'un triplet composé d'un sujet, d'un prédicat et d'un objet. De la même façon, SPARQL construit ses requêtes à partir de ce même modèle. Ainsi, si nous souhaitons connaitre le type de pelage pour l'instance « **Jack** » à partir de la relation « **estRecouvertDe** », nous définissons pour ce faire la requête SPARQL suivante pour notre ontologie : « Animaux.owl »

19. Protégé : http ://protege.stanford.edu/. Date : 13/08/2010.
20. SPARQL : http ://www.w3.org/TR/rdf-sparql-query/. Date : 24/05/2010



La requête nous renverra la valeur : « Poils ».

1.3 La conception d'ontologies

1.3.1 Introduction

L'utilisation d'ontologies en informatique vise à intégrer une couche de connaissances aux systèmes informatiques afin de permettre des traitements élaborés de l'information qu'ils manipulent. Mais la conception d'une ontologie n'est pas une chose facile et nécessite une identification claire et précise des objectifs et des buts attendus par celle-ci avant même d'entamer sa conception. C'est d'ailleurs ce que la méthodologie proposée par Uschold et King [Uschold et al., 1998] définit à travers un ensemble de principes méthodologiques de construction d'ontologies. Ceux-ci visent à identifier clairement les motivations et les buts de l'ontologie à construire ainsi que le pourquoi de celle-ci avant même sa conception.

Après avoir répondu à ces questions, il est nécessaire de choisir une méthodologie de construction d'ontologies. Il existe une dizaine de méthodologies de construction d'ontologies dans la littérature mais la plupart de celles-ci peuvent se regrouper en deux catégories. La première vise à construire une ontologie de façon manuelle à travers l'intervention d'experts issus du domaine et sans l'utilisation de ressources textuelles spécifiques. Cependant, ce procédé de génération est très coûteux en temps et pose surtout des problèmes de maintenance et de mise à jour [Ding and Foo, 2002]. La deuxième approche, plus répandue, consiste à utiliser des ressources linguistiques du domaine telles que des corpus de textes, thesaurus ou d'autres ontologies de domaine. L'ontologie de domaine est ainsi construite à partir de connaissances textuelles et où la méthodologie utilisée se voit contrôlée par un

expert du domaine lors de chaque étape du processus de construction.

Le domaine de la Recherche d'Informations (RI) manipule essentiellement des documents textuels et des thesaurus ; notre étude se porte donc sur la conception d'ontologies à partir de ces deux types de ressources. Dans un premier temps, nous concentrons donc cet état de l'art sur la méthodologie permettant de partir des données contenues dans des ressources textuelles pour en produire de la connaissance. Ensuite, nous nous attarderons sur certaines méthodologies établies pour la construction d'ontologies à partir de corpus de textes ainsi qu'à partir de thesaurus.

1.3.2 Des données à la connaissance

Les ontologies formalisent et représentent la connaissance d'un domaine, mais avant d'en arriver là, il est nécessaire d'extraire cette connaissance à partir des ressources textuelles qui la contiennent. Ainsi, le processus permettant d'extraire des unités de connaissances à partir de données textuelles est appelé processus d'« Extraction de Connaissance à partir de Données » (ECD)[Fayyad et al., 1996]. Ce processus est constitué de plusieurs étapes qui couvrent la préparation des données, l'application de méthodes de fouille (data mining) et enfin la validation et l'évaluation des résultats. L'automatisation de l'ensemble de ces étapes se révèle une tâche difficile car l'expertise requise dans le domaine est trop importante. Par conséquent, le besoin d'interactivité est nécessaire.

C'est pourquoi ce processus est itératif et interactif et consiste à extraire des unités de connaissances à partir de données brutes issues des ressources du domaine. Ce processus se décompose en trois étapes (figure 1.17 [Piatetsky-Shapiro and Frawley, 1991, Napoli, 2005]) :

1. le prétraitement des données, qui consiste à définir l'ensemble des ressources du domaine à partir desquelles les unités de connaissances seront extraites.
2. la méthode de fouille symbolique ou numérique qui, appliquée sur les ressources, permet d'en extraire des patrons syntaxiques intéressants.
3. L'interprétation et l'évaluation des patrons par les experts du domaine afin d'en déduire des unités de connaissances.

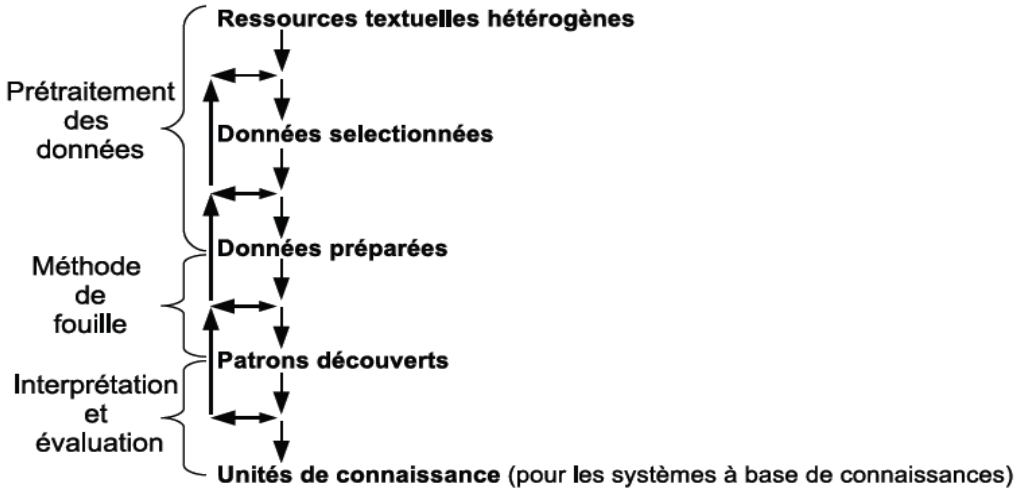


FIGURE 1.17 – **Processus d'extraction de connaissances à partir de données.**

Chaque étape du processus doit être contrôlée et validée par un expert ou par un analyste du domaine afin d'obtenir les connaissances répondant à l'objectif défini. Cet analyste interprète, analyse et sélectionne les unités de connaissances pour construire le modèle qu'il considérera comme modèle de connaissances du domaine (ontologie).

1.3.3 Construction d'ontologie à partir de ressources textuelles

Lorsque les connaissances à représenter sont issues de documents, l'ingénierie des connaissances (IC) s'appuie sur des méthodologies développées dans le domaine de la linguistique et du traitement automatique des langues afin d'assurer une compréhension efficace du contenu des documents considérés. L'extraction de connaissances à partir de textes (ECT) se fonde sur le processus d'extraction de connaissances à partir de données [Fayyad et al., 1996], tout en y apportant un certain nombre de particularités [Toussaint, 2004]. Une définition de ce processus est proposée par [Toussaint, 2004] :

Définition 6 (L'extraction de connaissances à partir de textes)
L'extraction de connaissances à partir de textes est un processus non trivial qui construit un modèle de connaissances valide, nouveau, potentiellement utile et au final compréhensible à partir de textes bruts.

L'objectif de ce processus vise à extraire des éléments à partir d'une collection de textes afin qu'ils puissent être interprétés comme des éléments de connaissances par un expert du domaine [Feldman and Sanger, 2007]. L'ensemble de ces éléments de connaissances nous permet de construire une ontologie du domaine. La figure 1.18 illustre ce processus. Celui-ci est, tout comme le processus d'ECD, itératif et incrémental et se décompose au travers de six étapes [Fayyad et al., 1996] :

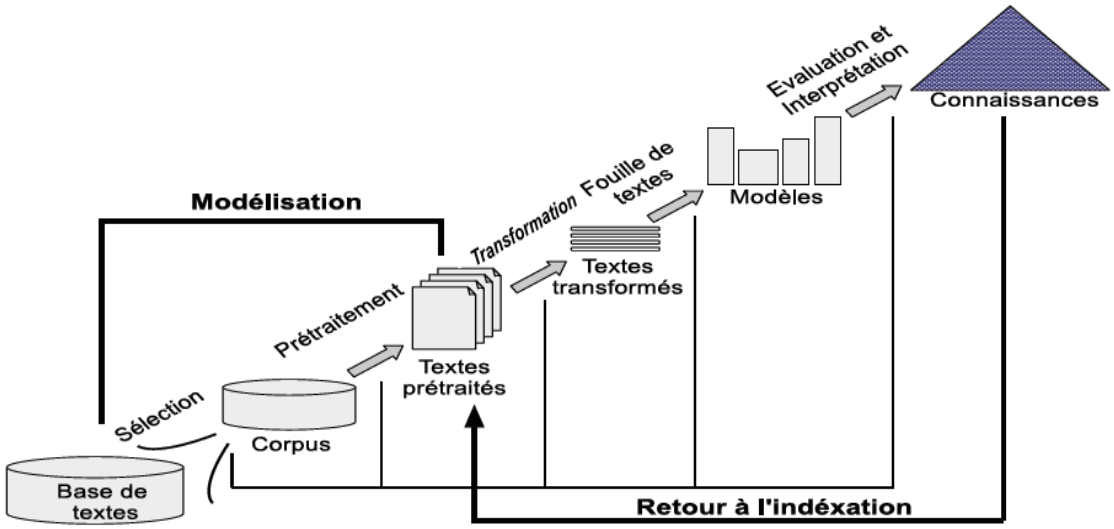


FIGURE 1.18 – **Processus d'extraction de connaissances à partir de textes (ECT).**

1. **La sélection**

   Cette étape consiste à récolter l'ensemble des textes du domaine à partir desquels la connaissance devant être formalisée y est contenue. Un corpus de textes (voir section 1.2.2) est alors constitué traitant du même domaine spécifique et à partir duquel les connaissances y seront extraites.

2. **Le prétraitement**

   Le prétraitement consiste à nettoyer les documents en supprimant certaines parties, styles et formulations de phrases insignifiantes et ne contenant aucunes données intéressantes pour le domaine. Par exemple les phrases comme « Cette partie reprend les principes fondamentaux utilisés dans ce mémoire

et se subdivise en trois sections » ou encore « Ce chapitre conclura ce mémoire et énoncera les différentes perspectives d'extension de ce travail » n'apportent aucune information supplémentaire sur le domaine et peuvent donc être supprimée. Un autre prétraitement consiste à combler certains manques d'informations pouvant survenir entre ce que l'expert pense et ce qu'il écrit. Par exemple, lorsqu'un expert écrit que « les jaguars vont très vite », il est nécessaire de reformuler la phrase afin de préciser le véritable sens du mot « jaguar » et ainsi de combler la perte d'information.

3. **La transformation**

Cette étape consiste à rechercher et à transformer, certaines structures de données contenues par les documents afin de mieux les adapter aux outils de fouille de textes choisis. Ces transformations peuvent être de différents types telles que définies explicitement par [Han et al., 2001] :

- Le « lissage » revient à réduire le bruit pouvant provenir des données contenues par les documents à travers diverses techniques tels que : le clustering, la regression (technique utilisée pour ajuster une équation à un ensemble de données) et le binning (technique réduisant les erreurs mineures provenant d'observations).

- L'« agrégation » vise à appliquer des opérations d'agrégation ou de synthèse sur des données d'un document. Par exemple, les données d'un document représentant les ventes journalières d'un commerce peuvent être agrégées et calculées de façon mensuelle ou annuelle.

- La « généralisation » des données s'appuye sur une hiérarchie de concepts afin de remplacer certaines données faisant références à un concept par leur parents et ainsi restreindre le nombre de données contenues dans un document. Par exemple, des noms de « rues » peuvent être généralisés à un concept de plus haut niveau tels que la « ville » ou le « pays ». De façon similaire, certaines valeurs numériques comme l'âge peuvent être également liées à des concepts de plus haut niveau comme « jeune », « âge-moyen » et « senior ».

- La « normalisation » qui consiste à ajuster des données d'un document de manière à entrer dans un intervalle spécifié.

4. **La fouille de textes**

   La fouille de textes a pour but d'acquérir les connaissances détenues par les textes en récupérant les régularités (ou irrégularités) de l'ensemble des données préparées. Pour effectuer cette tâche, il existe de nombreuses méthodes de fouilles mais le choix de celles-ci doit être établi selon l'objectif visé par l'expert. Certains objectifs de fouilles ont été relevés dans [Han et al., 2001] :

   – la recherche d'associations entre des attributs qui prennent des valeurs particulières de façon concomitante ;
   – la classification et la prédiction s'appuyant sur la définition d'un modèle à partir d'un jeu de données d'apprentissage ;
   – la construction de clusters qui regroupent les données selon des mesures de similarité ;
   – la détection de cas extrêmes révélant une forme d'irrégularité (pertinent dans le domaine médical) ;

   Les termes candidats pour représenter les concepts d'une ontologie peuvent être extraits selon deux approches : numérique ou symbolique (voir section 1.1.2). L'approche symbolique analyse le rôle grammatical des mots dans ces textes alors que l'approche statistique repose sur la fréquence d'apparition des mots dans les textes.

5. **L'interprétation et l'évaluation**

   Au terme de la méthode de fouille, l'ensemble des unités extraites sont analysées et interprétées par un expert du domaine. Lorsqu'une unité est validée par l'expert en tant que connaissances du domaine, celle-ci est stockée et représentée dans la base de connaissances (ontologie du domaine).

1.3.4 Construction d'ontologie à partir d'un thesaurus

Une des difficultés majeures dans la migration des thesaurus vers des ontologies est de capturer la sémantique implicitement présente dans ces ressources utilisées habituellement par des personnes.

Une méthodologie proposée dans [Van Assem et al., 2004] permet d'assurer la migration d'un thesaurus en RDF/OWL. Cette méthodologie à la particularité de pouvoir s'adapter à n'importe quelle spécification de thesaurus. Les thesaurus comportant d'autres relations que les relations les plus utilisées (voir section 1.2.3)

sont aussi considérés. Cette méthode se veut donc générale et propose de capturer l'information implicitement présentée par le thesaurus. Elle repose sur quatre étapes principales que nous reprenons de [Van Assem et al., 2004] :

1. La préparation : cette étape consiste à analyser le thesaurus concerné à travers sa documentation, mais aussi à partir de son modèle conceptuel et des relations qu'il possède.

2. La transformation syntaxique : celle-ci se décompose en deux étapes. La première consiste à conserver la structure initiale du thesaurus en la transformant à partir d'éléments simples de RDFS(définition des classes, définition des noms des classes) et des types de données supportés par XML tels que les entiers, les dates, etc.). Cette étape doit se faire en préservant le sens des informations dans le thesaurus, en évitant la redondance d'informations et l'interprétation sur les données du thesaurus. La deuxième étape vise à expliciter la structure syntaxique de la future ontologie en ajoutant de nouveaux éléments qui étaient sous-entendus dans la représentation originale et qui ont besoin d'être formalisés dans la représentation conceptuelle. Par exemple, la mise en évidence du terme principal pour représenter un ensemble de synonymes peut se faire en mettant en gras le terme dans la représentation texte ; dans l'ontologie, cette distinction devra être intégrée en créant par exemple une nouvelle classe de termes.

3. La transformation sémantique se décompose, elle aussi, en deux étapes. La première est l'explicitation de la sémantique contenue dans la version initiale comme, par exemple, l'ajout de caractéristiques aux propriétés (transitivité : owl :TransitiveProperty, symétrie : Owl :SymmetricProperty). La deuxième est l'introduction d'interprétation comme, par exemple, l'ajout d'une nouvelle classe permettant de regrouper plusieurs classes comme étant ses filles. La dernière étape de standardisation consiste à lier le schéma de modélisation de l'ontologie proposée à un schéma standard visant à représenter un méta-modèle de thesaurus. Cette étape reste délicate, car aucun modèle ne présente jusqu'ici un consensus [Van Assem et al., 2004].

Cette méthodologie permet la séparation des différentes phases menant à la transformation du thesaurus. Ceci présente l'intérêt de limiter l'interprétation des éléments représentés dans le thesaurus et mène à la construction d'un nouveau thesaurus respectant la connaissance initialement représentée. Cependant, aucun outil n'est développé pour aider à sa mise en oeuvre.

Une autre méthodologie définie par [Chrisment et al., 2008] transforme un thesaurus pré-existant en une ontologie de domaine qui sera utilisée pour indexer sémantiquement une collection de documents. Cette méthode nécessite cinq étapes :

1. Spécifier les besoins auxquels doit répondre l'ontologie ;
2. Définir le choix du corpus de référence à partir duquel l'ontologie sera construite de façon automatisée ;
3. Etude des ressources : thesaurus et corpus ;
   - Extraction des termes et des relations du domaine à partir des ressources textuelles concernées,
   - Structuration des termes présents dans le thesaurus à partir des relations de celui-ci,
4. Normalisation des résultats obtenus : définition des concepts et des relations sémantiques à partir des termes et des relations lexicales obtenues. Au niveau de cette étape, le thesaurus peut être utilisé pour aider à la spécification des concepts ;
5. Formalisation : le réseau sémantique défini à l'étape précédente est traduit dans un langage formel tel que OWL ;

Cette méthodologie offre l'avantage de moins solliciter les interventions d'experts durant l'élaboration de l'ontologie. De plus, les processus spécifiés pour son élaboration sont simples à mettre en oeuvre et peuvent se faire de façon automatique.

[Hahn and Schulz, 2004] définissent une construction d'ontologies à partir du thesaurus UMLS. La particularité de leur approche consiste à utiliser les logiques de descriptions afin de pouvoir y effectuer du raisonnement. [Han and Choi, 2008] proposent une méthode permettant la conception d'un thesaurus condensé à partir de celui d'UMLS. Celui-ci offre l'avantage de ne contenir que les termes les plus utilisés pour un domaine donné et ainsi d'améliorer d'une part l'efficacité de Recherche d'Informations et d'autre part de réduire l'effort et le coût de la construction d'ontologies de domaine.

L'article [Schlangen et al., 2004] propose un système basé sur le Web sémantique utilisé dans le domaine pathologique, permettant la recherche à la fois de textes et d'images numériques pour le diagnostic, le diagnostic différentiel et les tâches d'enseignement. Ce système, mis en place, utilise de nombreuses ontologies afin d'annoter et d'enrichir les documents fournis en entrée et définit son domaine de connaissance à partir du thesaurus d'UMLS. L'utilisation du thesaurus comme source principale de connaissance du système, a du être personnalisé en raison de la complexité et de la grande quantité de connaissances qu'il représente. Cette

personnalisation permit de déterminer les bibliothèques pertinentes pour le domaine pathologique ainsi que le vocabulaire nécessaire. De plus, l'article fournit une méthodologie de traduction des concepts, représentant le domaine, issus du thesaurus en OWL. Celle-ci débute par la modélisation de chaque concepts en classe OWL avec une prise en compte de leur définition propre ainsi que de leurs synonymes issus d'UMLS. Par la suite, certaines relations propres aux concepts sont utilisées et représentée en OWL, ainsi que le type sémantique auquel chaque concepts appartient. L'ontologie générée à terme nécessitera le contrôle et la correction d'erreurs d'inconsistances provoquées par certaines relations entres concepts non-pertinentes. A travers cette méthodologie, l'article nous montre bien qu'UMLS peut nous fournir un thesaurus pour un domaine spécifique et que celui-ci doit être définis à partir d'un ensemble pertinents de termes du domaine visés et contenus dans UMLS. De plus, les relations sémantiques et hiérarchiques sont générées à partir des concepts d'UMLS, ce qui implique qu'il est important de bien délimiter les sources nécessaire et pertinentes du thesaurus pour le domaine de connaissance visé.

Chapitre 2

Méthodologie

2.1 Introduction

Ce travail s'inscrit dans le cadre du projet « e-Health » visant à promouvoir et à soutenir l'échange électronique et sécurisé de données entre tous les acteurs des soins de santé (médecins, hôpitaux, pharmaciens, patients, etc.) tout en respectant la protection de la vie privée et le secret médical.

Dans ce contexte, la diversité des acteurs de la santé conduit à une hétérogénéité des formats de données, des protocoles et des solutions logicielles. Le suivi et l'étude de la santé des populations impliquent la constitution de grandes quantités d'informations disséminées sur le territoire. Ainsi, cette variété de données rend la Recherche d'Informations complexe et peu efficace. De plus, la plupart des systèmes de recherche d'informations s'appuient sur l'extraction de termes dans les documents, termes qui servent de base pour l'accès à ces documents. Néanmoins, ces techniques ont l'inconvénient de reposer sur des termes qui peuvent être ambigus et de ne pas prendre en compte les liens sémantiques qui existent entre les termes et par conséquent, augmenter le risque de silence parmi les résultats.

Pour remédier à ce problème, nous présentons une approche permettant la conception d'un moteur sémantique fondé exclusivement sur l'utilisation de deux ontologies de domaines. La première représente la connaissance d'un domaine spécifique et a pour objectif d'étendre la requête de l'utilisateur selon trois types d'enrichissements. La deuxième ontologie vise à classifier l'ensemble des ressources du domaine concerné en fonction des termes qu'elles partagent. Pour cette dernière, nous utilisons une méthode de classification symbolique que nous représentons sous une ontologie afin de pouvoir y effectuer du raisonnement.

2.1.1 Présentation de l'approche

Notre méthodologie pour la conception d'un moteur de recherche sémantique se subdivise en quatre étapes (figure 2.1) :

1. **Choix du corpus de textes.**
   Cette partie énonce les différents choix de corpus de textes qui pourraient s'avérer être les plus pertinents pour cette méthodologie et qui devront représenter au mieux la connaissance du domaine choisi.

2. **Acquisition des connaissances du domaine.**
   Pour modéliser la connaissance de notre domaine, nous utilisons un mécanisme d'extraction de connaissances à partir de ressources textuelles (ECT). Ce processus de construction vise dans un premier temps à extraire l'ensemble des termes importants du domaine à partir des ressources textuelles retenues. Ensuite, cet ensemble terminologique est utilisé conjointement avec un thesaurus traitant du même domaine dans le but de structurer les termes extraits en une ontologie à partir des concepts, des relations entre concepts et de la structure hiérarchique du thesaurus utilisé.

3. **Indexation des documents.**
   Les documents ayant servi à la construction de notre ontologie structurant leurs termes constituent les ressources mises à disposition par notre moteur de recherche. Ces ressources doivent donc être indexées à partir des mêmes termes utilisés lors de la création de la base de connaissances et selon une technique d'indexation utilisant l'Analyse Formelle de Concepts. A partir de cette méthode de classification symbolique, nous obtenons un treillis de concepts dans lequel chaque concept correspondra à une classe de documents partageant un même ensemble de termes. Nous représentons par la suite ce treillis sous une ontologie destinée à la recherche de documents.

4. **Recherche de documents.**
   A travers l'utilisation conjointe des deux ontologies définies précédemment, nous effectuons une recherche documentaire proposant trois types d'extensions de requête. Ces extensions visent à enrichir la requête de l'utilisateur de nouveaux termes afin d'améliorer la pertinence des documents recherchés ainsi que d'apporter de nouvelles connaissances lors du processus de recherche.

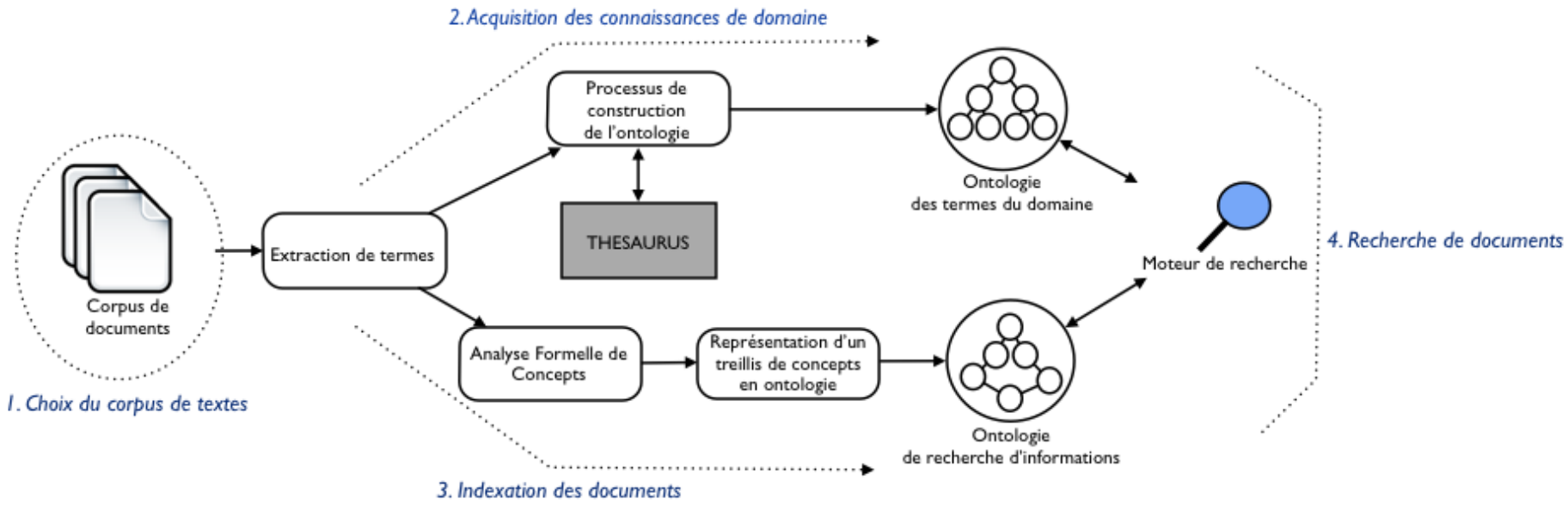


FIGURE 2.1 – **Vue globale de la méthodologie présentée.**

Dans la suite de ce chapitre nous expliquerons ce en quoi consiste chacune de ces étapes et nous les illustrerons également au travers d'un contexte défini dans le cadre d'un stage en entreprise. Le domaine étudié durant ce stage fut « la maladie de Parkinson ».

2.2 Choix du corpus de textes

La constitution d'un corpus de textes dépend fortement de l'objectif visé (voir section 1.2.3). Pour cette étape, le corpus de textes doit représenter au mieux la connaissance du domaine concerné. C'est pourquoi le choix des textes qui constituent le corpus est important car ils déterminent l'ensemble terminologiques du domaine étudié. Dans le contexte d'une Extraction de Connaissances à partir de Textes (ECT) (voir section 1.3.3), Toussaint [Toussaint, 2004] privilégie le choix de textes du type « résumé ». Les « résumés » sont des textes courts, contenant l'essentiel de l'information et possédant une densité terminologique élevée. Mais, le choix d'autres types de documents, autres que des résumés, peuvent être tout aussi pertinents et efficaces selon le domaine étudié et selon l'objectif du corpus visé. De plus, lors de la recherche de textes, il est bon de favoriser des textes provenant d'un maximum de sources différentes afin d'assurer une bonne couverture et une forte richesse terminologique pour représenter au mieux la connaissance du domaine.

Dans le cadre du stage, la définition du corpus fut composée essentiellement de « guidelines » traitant de la maladie de Parkinson. Le choix de ce type de ressource est motivé par leur structure clairement définie ainsi que pour leur vocabulaire relativement bien délimité et contrôlé par des experts du domaine. De plus, celles-ci sont pour la plupart concises et contiennent les informations essentielles sur la pathologie qu'elles traitent.

2.3 Acquisition des connaissances du domaine

Dans cette partie, nous développons le processus visant à représenter la connaissance du domaine à partir des textes constituant le corpus. Pour ce faire, nous procédons dans un premier temps par une méthode de fouille visant à extraire un ensemble terminologique représentant au mieux le domaine étudié. Ensuite, nous structurons cet ensemble terminologique à partir d'un thesaurus en une ontologie.

2.3.1 Méthode de fouille

Une fois le corpus constitué, nous procédons à une méthode de fouille afin d'en extraire les termes candidats qui détermineront les concepts de notre ontologie reflétant la connaissance du domaine. Dans le cas où la taille du corpus est importante, il convient d'utiliser un outil de traitement automatique de la langue (TAL) afin d'extraire l'ensemble terminologique représentant le domaine traité et ensuite de le valider par un expert du domaine. Lorsque le corpus n'est pas trop volumineux, l'extraction des termes peut se faire de manière manuelle à travers l'analyse et la lecture des documents appartenant au corpus. Une fois cet ensemble de termes extraits, chacun de ceux-ci est ensuite normalisé afin de ne retenir que le sens premier de chaque terme et d'en faciliter la recherche au sein du thesaurus lors du processus de construction de l'ontologie.

La figure 2.2 illustre un échantillon de 12 termes ayant été extraits et normalisés à partir des ressources textuelles rassemblées pour la maladie de Parkinson.

```
1  parkinson
2  tremor
3  levodopa
4  pain
5  falls
6  stress
7  diet therapy
8  dementia
9  dopamine
10 depression
11 basal ganglia
12 hypothalamus
```

FIGURE 2.2 – **Echantillon de 12 termes extraits des « guidelines » de la maladie de Parkinson.**

2.3.2 Processus de construction de l'ontologie

Pour construire la base de connaissances (ontologie) représentant le domaine étudié, nous reprennons la méthodologie de Chrisment [Chrisment et al., 2008] présentée en section 1.3.4. Néanmoins, nous utilisons celle-ci uniquement afin de structurer les termes extraits du domaine à partir des concepts et des relations définies au sein du thesaurus.

Les thesaurus ont l'avantage de reposer sur un ensemble de termes qui ont été identifiés et contrôlés par des experts comme étant représentatifs du domaine. Cependant, il ne faut pas perdre de vue les différences fondamentales qui distinguent les thesaurus des ontologies dont la principale étant qu'ils ne sont pas formels. Pour la conception de l'ontologie représentant le domaine, il va de soi d'utiliser un thesaurus traitant du même domaine. Pour la maladie de Parkinson, nous avons choisi le grand thesaurus biomédical, Unified Medical Langage System (UMLS).

Le thesaurus UMLS se compose d'un ensemble de vocabulaire et de concepts contrôlé par des experts du domaine. D'après son dernier bulletin annuel datant de l'année 2009, UMLS regroupe un ensemble de 2,1 millions de concepts médicaux définis à partir de plus de 140 sources terminologiques et de sous-thesaurus qui contiennent plus de 8 millions de termes issus de 17 langues. Ce thesaurus se compose de 3 parties :

- « **Metathesaurus.** » Le « *Metathesaurus* » représente la base du vocabulaire de UMLS contenant l'ensemble de tous les concepts biomédicaux ainsi que de nombreuses informations sur ceux-ci. La construction de celui-ci a été faite à partir de nombreuses versions électroniques ainsi que d'autres thesaurus ce qui explique son appellation « *Metathesaurus* » ; mais également provenant de diverses classifications ainsi que de listes de termes utilisés et contrôlés dans les soins publics, les facturations des services de santé, les statistiques de santé publique, etc. Tous les concepts repris dans le « *Metathesaurus* » sont assignés à au moins un type sémantique provenant du « Semantic Network ».

- « **Semantic Network.** » Le réseau sémantique est un graphe composé de noeuds représentant des concepts ainsi que des arcs traduisant les relations sémantiques pouvant exister entre ces concepts. Les concepts du réseau, appelés « types sémantiques », sont définis de façon unique et visent à représenter une catégorisation consistante de l'ensemble des concepts appartenant au « *Metathesaurus* ». Ainsi, le réseau sémantique correspond à une « metahiérarchie » à laquelle tous les concepts du « *Metathesaurus* » sont assignés. Ce réseau compte plus de 135 types sémantiques reliés entre eux par plus de 54 types de relations différentes. La figure 2.4 nous montre une partie du réseau sémantique de UMLS.

- « **SPECIALIST Lexicon.** » Le « SPECIALIST Lexicon » est un lexique général anglais incluant de nombreux termes biomédicaux. Ce Lexique, destiné à des spécialistes a été développé pour fournir des informations lexicales nécessaires à des processus de traitement automatique de la langue (TAL).

Ainsi, il contient une entrée lexicale pour chaque mot ou expression possédant une variation pouvant être syntaxique, morphologique ou orthographique.

Dans le cadre du processus de construction nous n'avons utilisé que le « Metathesaurus » ainsi que le « Semantic Network ». Pour chaque terme extrait des documents, nous récupérons dans un premier temps le concept du « Metathesaurus » auquel il correspond. A partir de ce dernier, nous récupérons le type sémantique auquel il appartient ainsi que l'ensemble des ancêtres dont il hérite jusqu'à la racine du « Semantic Network ». La figure 2.3 illustre ce processus pour le terme « parkinson ». De manière incrémentale, nous construisons l'ontologie par l'ajout successif de sous-arbres de telle façon que, lorsque deux sous-arbres partagent un même ancêtre, ceux-ci sont fusionnés pour ne faire qu'un. Pour finir, nous enrichissons l'ontologie résultante avec les relations existantes parmi l'ensemble des types sémantiques issus de UMLS.

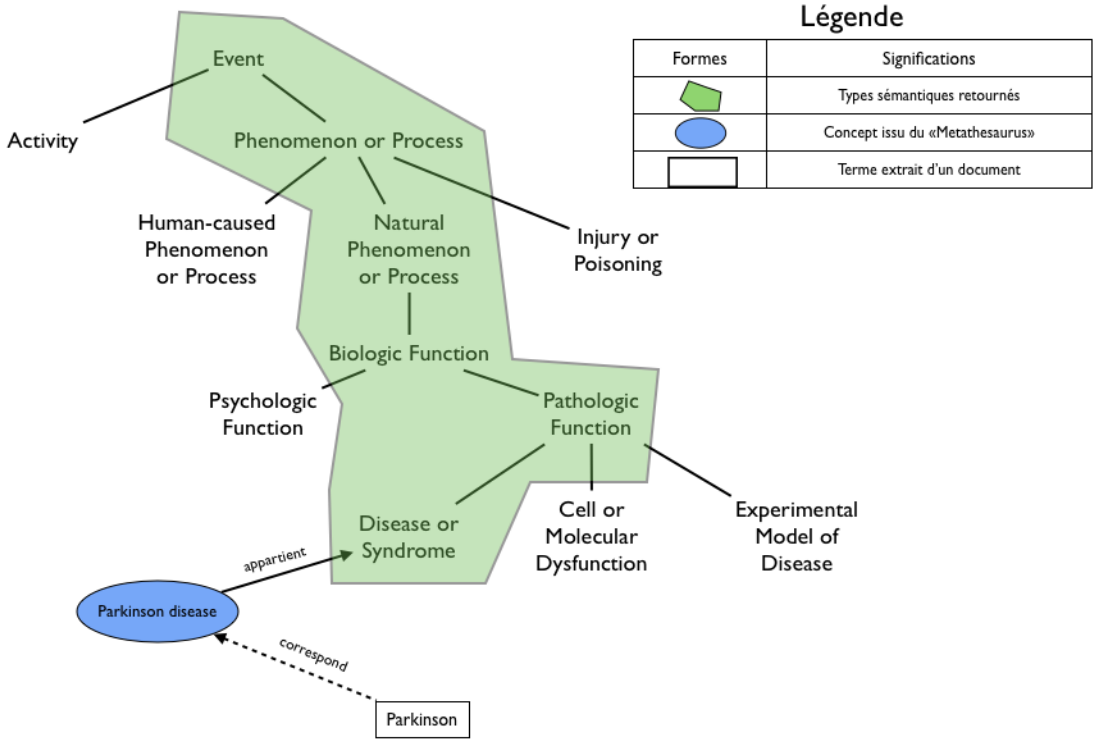


FIGURE 2.3 – **Exemple de récupération des ancêtres hiérarchiques à partir du terme « parkinson » et de son concept « *Parkinson disease* ».**

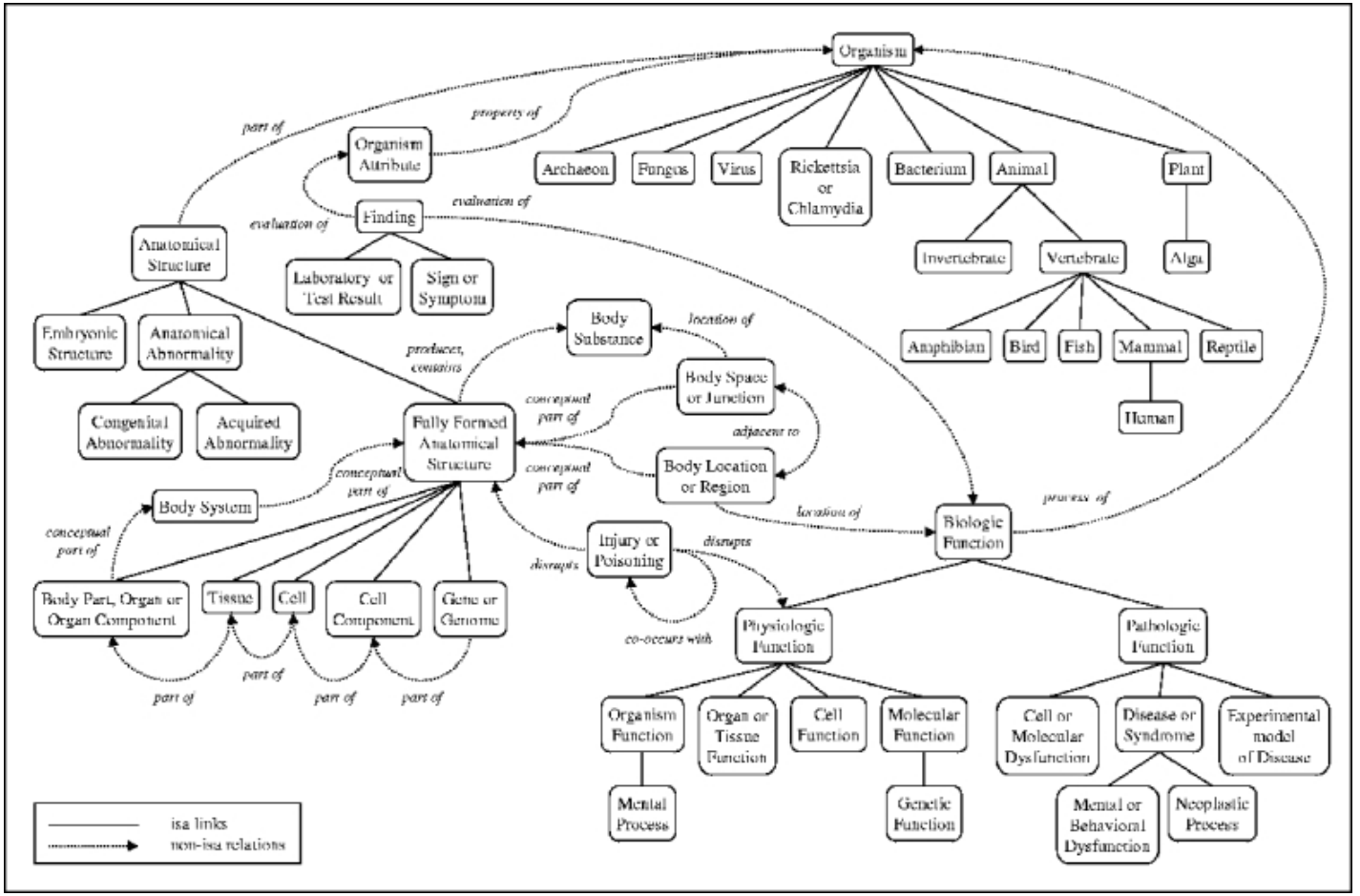


FIGURE 2.4 – **Partie du réseau sémantique de UMLS.**

A partir des 12 termes extraits des « guidelines », nous construisons l'ontologie avec le thesaurus UMLS. La figure 2.5 nous présente l'ensemble des concepts de l'ontologie obtenues à l'aide de l'outil « Protégé » [1] » (logiciel défini pour la 'édition et la gestion d'ontologies).

En observant l'ontologie résultante (figure 2.5), on peut constater que les concepts qui représentent l'ensemble des terminaisons de l'ontologie correspondent tous aux concepts extraits du « Metathesaurus » de UMLS et dont les instances sont les termes qui ont été extraits des « guidelines ». Les autres concepts correspondent quant à eux aux types sémantiques extraits de UMLS et sont reliées entre eux à travers un ensemble de propriétés correspondant aux relations sémantiques extraites et contenues dans le thesaurus UMLS (voir figure 2.7).

1. Protégé : http ://protege.stanford.edu/. Date 4/05/2010

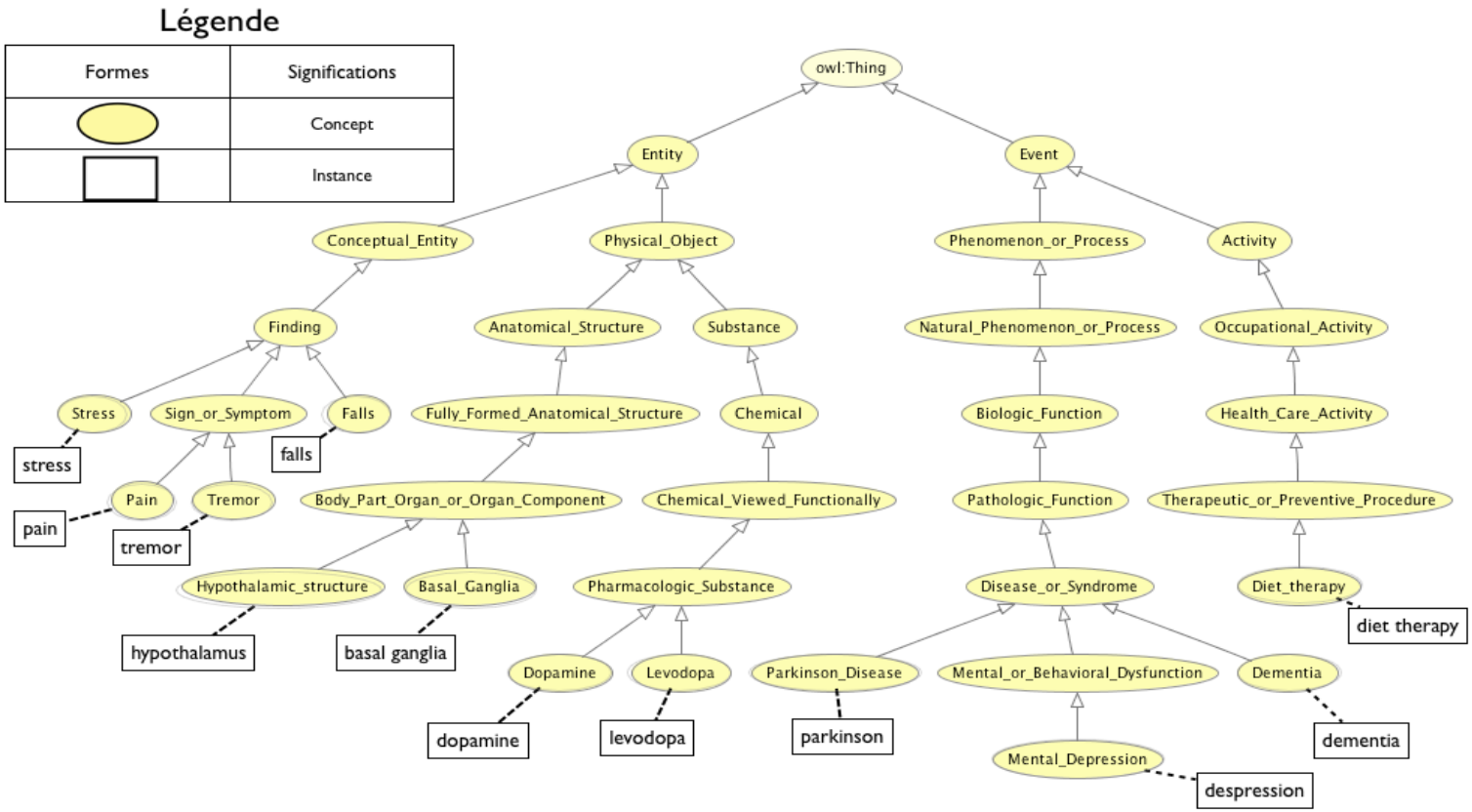


FIGURE 2.5 – **Ontologie structurant l'échantillon les termes de la maladie de Parkinson.**

Pour cette ontologie structurant les 12 termes, nous avons récupéré 19 propriétés, autrement dit, 19 sortes de relations sémantiques distinctes venant de UMLS (figure 2.6). C'est au moyen de celles-ci que nous pourrons, par la suite, proposer un enrichissement de requête par relation lors de la recherche de documents (voir section 2.5).

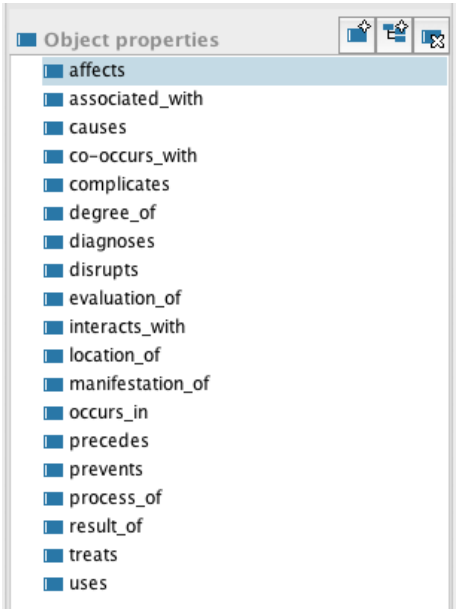


FIGURE 2.6 – **Propriétés récupérées de UMLS.**

Lorsqu'un concept, issue de l'ontologie, possède une relation sémantique avec un autre, celle-ci est définie au travers d'une description logique. Ainsi, toujours à travers l'utilisation de l'outil « Protégé [2] », la figure 2.7 montre que le concept « *Substance* » hérite du concept « *Physical_Object* » mais aussi que celui-ci est en relation au travers de la propriété « *causes* » avec les concepts « *Mental_or_Behavioral_Dysfunction* », « *Disease_or_Syndrome* » et « *Pathologic_Function* ».

2. Protégé : http ://protege.stanford.edu/. Date 4/05/2010

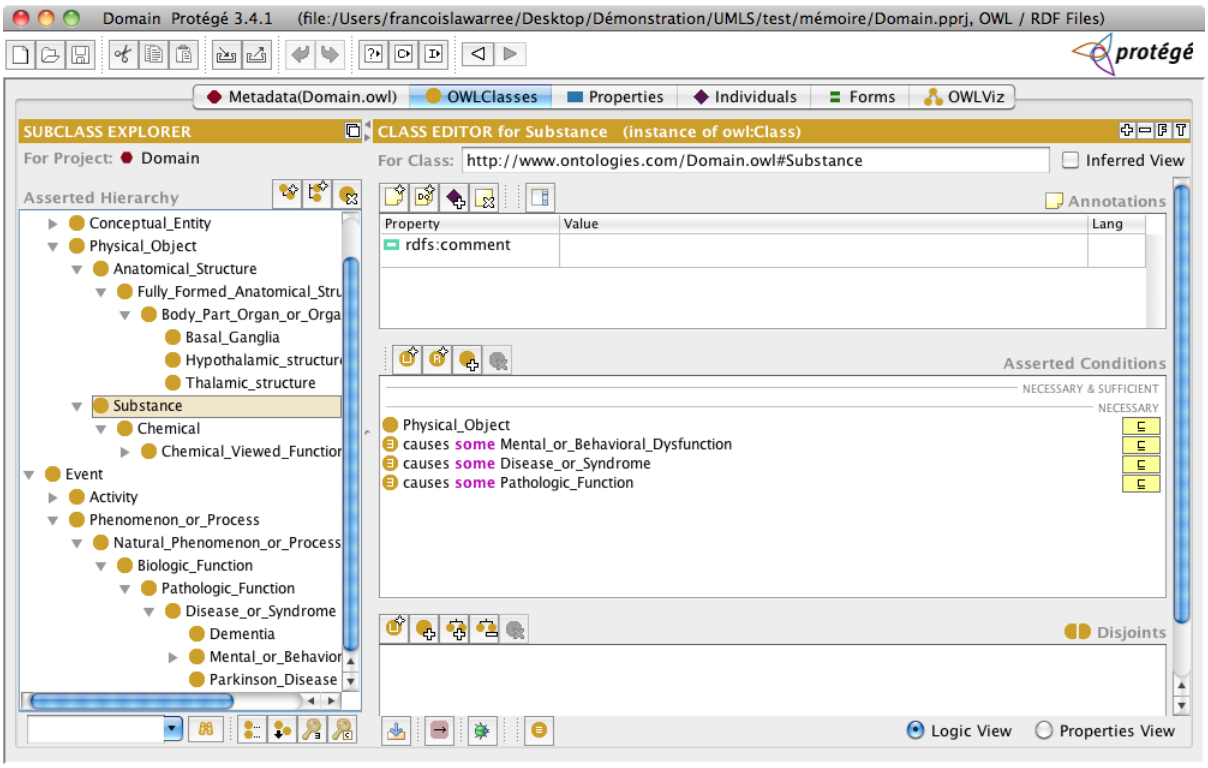


FIGURE 2.7 – **Représentation des propriétés d'un concept.**

Il est important de noter que lorsqu'un concept possède une propriété qui le relie à un autre, il en est de même pour l'ensemble de ses sous-concepts par héritage. Ainsi les sous-concepts « *Chemical* » et « *Chemical_ Viewed_ Functionally* » héritent, eux aussi, des propriétés possédées par le concept « *Substance* ».

Au terme de ce processus, nous obtenons une ontologie de domaine comprenant un ensemble de concepts structurées et liées entre eux à partir des concepts et de relations issus du thesaurus. Les termes du domaine sont définis en tant qu'instances des concepts auxquels ils correspondent.

2.4 Indexation des documents

Afin de permettre la recherche documentaire sur le corpus de textes, il nous est nécessaire d'indexer les documents utilisés avec les termes ayant été extraits pour représenter la connaissance de notre domaine. A cet effet, nous utilisons une approche symbolique appelée Analyse Formelle de Concepts (AFC) (voir section 1.1.2) afin de classifier les documents du corpus. Par la suite, nous représentons le

treillis de concepts résultant de l'AFC en une ontologie afin de pouvoir raisonner sur celle-ci et permettre d'y effectuer une recherche de documents.

2.4.1 Méthode de fouille

Dans cette partie, l'objectif est de regrouper l'ensemble des documents du corpus en fonction des termes communs qu'ils partagent. Pour effectuer cette tâche, nous utilisons comme méthode de fouille l'Analyse Formelle de Concepts. Pour utiliser cette technique mathématique, il est nécessaire de tenir compte de la relation binaire pouvant exister entre les textes utilisés et les termes qui en ont été extraits.

Ainsi, chacun des textes que comprend le corpus, est mis en relation avec chacun des termes que celui-ci contient sous la forme d'une paire « (texte, terme) ». L'ensemble des paires constituées sont ensuite utilisées pour en définir le contexte formel $\mathbb{K}$=(G,M,I). Le contexte formel illustré par la figure 2.8 est celui définit pour les 12 termes précédemment extraits des « guidelines » de notre corpus.

	parkinson	tremor	levodopa	pain	falls	stress	diet therapy	dementia	dopamine	depression	basal ganglia	hypothalamus
Guideline_1	x	x	x			x	x	x	x	x		
Guideline_2	x	x	x	x	x			x	x	x	x	
Guideline_3	x	x	x	x	x	x	x	x	x	x		x
Guideline_4	x	x	x			x		x	x	x		
Guideline_5	x	x	x	x	x	x		x	x	x	x	x

FIGURE 2.8 – **Contexte formel pour la maladie de Parkinson.**

La relation (Guidelines_3, tremor) issue du contexte signifie que la ressource « *Guideline_3* » contient le terme « *tremor* » et de façon duale que le terme « *tremor* » est contenu par la ressource « *Guideline_3* ».

A partir de ce contexte formel, le treillis de concepts présenté par la figure 2.9 est généré. Les détails de certains concepts issus de la figure 2.9 n'ont pas été affichés afin de ne pas surcharger la lisibilité du treillis illustré. Le graphe résultant (treillis de concepts) organise, classe et représente de manière équivalente, les données du contexte formel. Il regroupe les textes partageant un même ensemble de termes en un concept formellement défini. De plus la représentation graphique du treillis facilite la compréhension et l'interprétation des relations entre concepts.

Cette représentation à aussi pour avantage de permettre de retrouver le contexte formel initial à partir d'un treillis de concepts et inversement.

L'étape suivante consiste à représenter le treillis de concepts par une ontologie. Ce choix se justifie d'une part afin d'assurer l'utilisation exclusive d'ontologies par le moteur de recherche et d'autre part afin de raisonner sur l'ontologie obtenue tout en la préservant d'insertions si l'on envisageait de se contenter du treillis de concepts. En effet, à partir d'un treillis de concepts, il nous est déjà possible de fournir aux utilisateurs les ressources pouvant répondre le plus précisément à leurs requêtes en reprennant l'idée de Carpineto [Carpineto and Romano, 1996] (présentée en section 1.1.2). Celle-ci consiste a considérer chaque concept du treillis comme une requête formée par la conjonction des termes d'index du concept (les éléments de son intension). Cependant, cette façon de procéder suppose que la requête de l'utilisateur existe déjà dans le treillis et implique l'utilisation d'algorithmes de construction incrémentaux de treillis de concepts afin de permettre l'insertion des requêtes dans un treillis déjà construit. En représentant le treillis de concepts par une ontologie, nous évitons ainsi toutes modifications/insertions de celui-ci lors de la Recherche d'Informations, préférant l'utilisation du langage d'interrogation SPARQL sur l'ontologie.

2.4.2 Représentation du treillis de concepts en une ontologie

Cette étape vise à représenter le treillis résultant de l'étape précédente par une ontologie. Néanmoins, il est important de préciser que l'ontologie qui sera créée n'est pas faite dans le but de représenter la connaissance du domaine comme l'ontologie construite précédemment mais bien comme moyen afin de pouvoir y effectuer du raisonnement. Pour ce faire nous reprenons l'idée de Bendaoud [Bendaoud et al., 2008] en formalisant le passage du treillis de concepts en une ontologie à partir de la table 2.1.

Treillis de concepts	**Ontologie**
Propriété m ∈ M	Concept défini α(m)≡ ∃m.⊤ dans la TBox
Objet g ∈ G	Instance α(g) dans la ABox
Concept c = (X,Y) ∈ C	Concept défini α(g) dans la TBox, s.t., α(g) ≡$\sqcap_{m \in Y} \alpha$(m)
Relation de subsomption $C_1 \sqsubseteq C_2$	Inclusion générale de concepts $\alpha(C_1) \sqsubseteq \alpha(C_2)$

TABLE 2.1 – **Formalisme du passage du treillis à l'ontologie.**

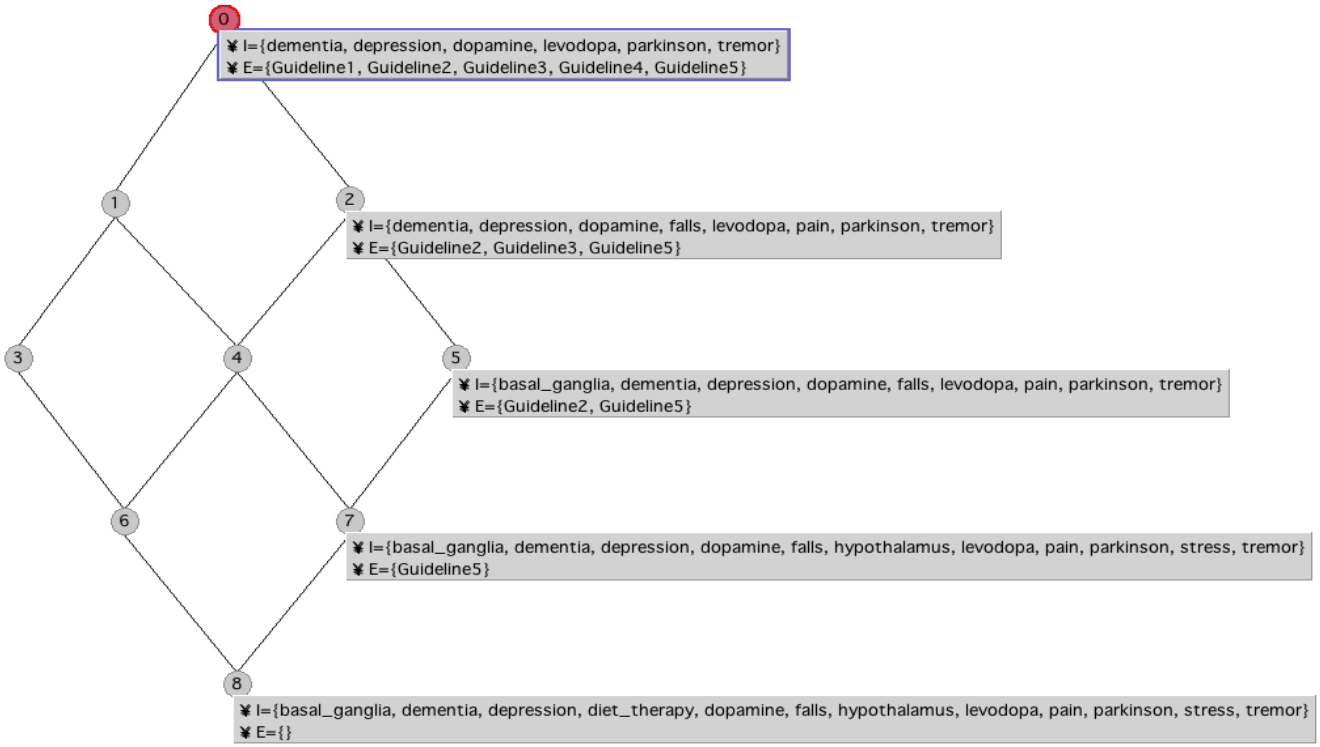


FIGURE 2.9 – **Treillis de concepts résultant pour la maladie de Parkinson.**

A partir de cette formalisation, l'ensemble des concepts formels issus du treillis sont représentés en concepts définis en logique de description (voir section 1.2.4). Pour illustrer cette représentation nous l'appliquons sur les concepts *C2* et *C5* du treillis représenté par la figure 2.9.

TBox	**ABox**
$\alpha(C_2) \equiv \exists$dementia.$\top \sqcap \exists$depression.$\top \sqcap \exists$dopamine.$\top \sqcap$ $\exists$falls.$\top \sqcap \exists$pain.$\top \sqcap \exists$levodopa.$\top \sqcap$ $\exists$parkinson.$\top \sqcap \exists$tremor.$\top$	C2(Guideline2) C2(Guideline3) C2(Guideline5)
	C5(Guideline2)
$\alpha(C_5) \equiv \exists$basal_ganglia.$\top \sqcap \exists$dementia.$\top \sqcap \exists$depression.$\top \sqcap$ $\exists$dopamine.$\top \sqcap \exists$falls.$\top \sqcap \exists$pain.$\top \sqcap$ $\exists$levodopa.$\top \sqcap \exists$parkinson.$\top \sqcap \exists$tremor.$\top$	C5(Guideline5)
$C_5 \sqsubseteq C_2$	

TABLE 2.2 – **Représentation des concepts formels « C2 » et « C5 » en logique de description.**

Une fois l'ensemble des concepts du treillis représentés en logique de description, nous procédons à l'implémentation de ceux-ci en langage OWL. La figure 2.10, nous présente l'implémentation du concept C5. La première partie de la figure correspond à la définition propre du concept (owl :Class) défini (owl :equivalentClass) par un ensemble de restrictions (owl :Restriction) existentielles (owl :someValuesFrom) de propriétés (owl :onProperty) tels que *dementia, depression, dopamine.* La deuxième partie définit les objets *Guideline2, Guideline5* en tant qu'instances du concept C5.

```
<owl:Class rdf:ID="C5">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty rdf:resource="#basal_ganglia"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#dementia"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#depression"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#dopamine"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#falls"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#levodopa"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#pain"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#parkinson"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#tremor"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

```
<C5 rdf:ID="Guideline2"/>
<C5 rdf:ID="Guideline5"/>
```

FIGURE 2.10 – **Représentation du concept « C5 » en Owl.**

L'utilisation d'un moteur d'inférence appliqué sur l'ontologie résultante nous permet de restituer visuellement, à l'aide de l'outil Protégé [3], les relations de subsomption représentées par le treillis. Ainsi, comme nous le montre la figure 2.11,

3. Protégé : http ://protege.stanford.edu/. Date 4/05/2010

le concept « C5 » inclut le concept « C2 » à partir de l'ensemble des propriétés de ce dernier représentant un sous-ensemble des propriétés de « C5 ».

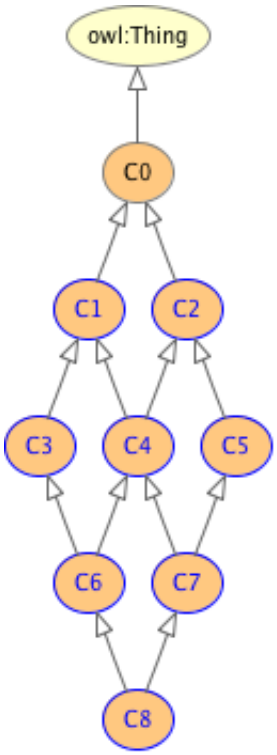


FIGURE 2.11 – **Ontologie de recherche d'informations pour la maladie de Parkinson.**

A partir de cette ontologie de recherche d'informations, il nous est désormais possible d'acquérir les textes pouvant répondre à une conjonction de termes émanant d'une requête d'utilisateur. Pour ce faire, nous utilisons le langage d'interrogation sémantique SPARQL (voir section 1.2.4) afin de questionner l'ontologie. Dès lors, lorsqu'une requête est formulée par un utilisateur, nous la traduisons en SPARQL et interrogeons l'ontologie de recherche d'informations pour fournir l'ensemble des instances (textes) de concepts vérifiant l'ensemble des propriétés (termes de la requête) demandées. Dans l'exemple présenté ci-dessous (figure 2.12), nous présentons une requête SPARQL générée pour la recherche de documents possédant les termes « basal ganglia » et « parkinson ». Celle-ci déduira les concepts possédant les propriétés correspondant aux termes et renverra les instances des concepts retenus. Pour cet exemple, les instances « Guideline2 » et « Guideline5 » seront retournées.

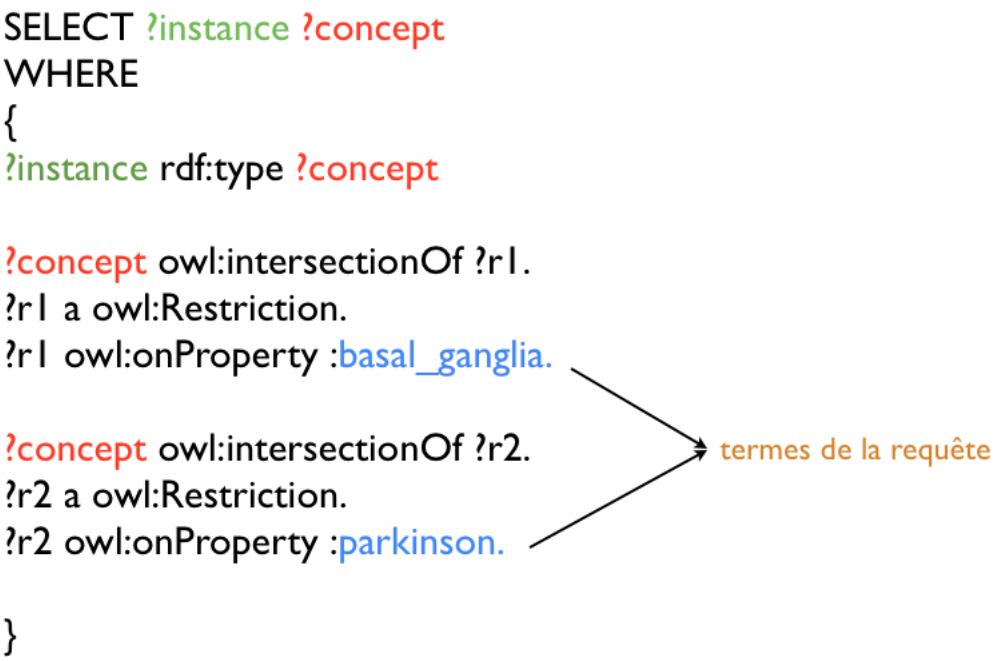


FIGURE 2.12 – **Exemple d'une requête SPARQL pour une recherche sur « basal_ganglia » et « parkinson ».**

2.5 Recherche de documents

Après avoir construit nos deux ontologies, il nous est désormais possible de commencer la recherche documentaires à partir des propriétés souhaitées et exprimées par la requête de l'utilisateur. Avant d'entamer les explications sur la façon dont nous allons procéder, nous rappelons dans un premier temps les différents objectifs des deux ontologies ainsi que leur définition respective afin de correctement les distinguer.

L'ontologie structurant les termes du domaine est utilisée afin d'enrichir la requête de l'utilisateur par l'utilisation d'extensions de requête (voir section 1.1.3). Les extensions de requête consistent à reformuler une requête en y ajoutant des termes à partir d'une ou de plusieurs ressources sémantiques (ontologies de domaine, thesaurus, taxonomies, etc.). Les termes ajoutés doivent être en relation sémantique avec ceux issus de la requête initiale. Le raffinement de requête présenté dans ce travail, offre la possibilité à l'utilisateur d'enrichir sa requête au travers de trois types d'enrichissements différents :

– **Enrichissement par spécialisation** enrichit la requête de l'utilisateur de

termes plus spécifiques. Ceux-ci sont obtenus à partir des descendants du concept correspondant au terme de la requête dans l'ontologie.

- **Enrichissement par généralisation** enrichit la requête de l'utilisateur de termes plus généraux. Ceux-ci sont obtenus à partir des ancêtres du concept correspondant au terme de la requête dans l'ontologie.

- **Enrichissement par relation** offre à l'utilisateur la possibilité d'enrichir la requête de termes partageant une relation sémantique, définie par l'ontologie, avec un terme issu de sa requête initiale.

Dans ce mémoire, nous nous inspirons de l'approche de Carpineto [Carpineto and Romano, 1996](voir section 1.1.3), mais nous y apportons en plus des enrichissements par généralisation et par spécialisation une enrichissement de requête utilisant les relations sémantiques propres de l'ontologie. Celles-ci offrent à l'utilisateur la possibilité d'acquérir la connaissance formalisée par l'ontologie et d'en bénéficier lors de sa recherche.

L'ontologie de recherche d'informations permet d'obtenir l'ensemble des documents vérifiant la ou les propriétés de la requête initiale de l'utilisateur ou à partir d'une requête ayant été préalablement enrichie.

En utilisant conjointement ces deux ontologies, nous effectuons la recherche de documents à travers quatre étapes illustrées par la figure 2.13 :

1. Cette première étape permet à l'utilisateur de choisir le type d'enrichissement qu'il souhaite appliquer aux termes issus de sa requête initiale. Le choix de l'un des trois enrichissements va impliquer l'utilisation de l'ontologie structurant les termes du domaine afin de rechercher le terme issu de la requête et d'y appliquer l'enrichissement choisi.
2. Le résultat de la première étape est ensuite retourné à l'utilisateur afin d'enrichir sa requête d'un ou plusieurs termes correspondant tous au domaine et appartenant à l'une ou l'autre ressource du corpus indexé.
3. Une fois l'enrichissement effectué, nous interrogeons par une requête SPARQL, l'ontologie de recherche d'informations à partir de l'ensemble des termes de la requête.
4. Cette dernière étape nous retourne l'ensemble des documents vérifiant les termes contenus par la requête.

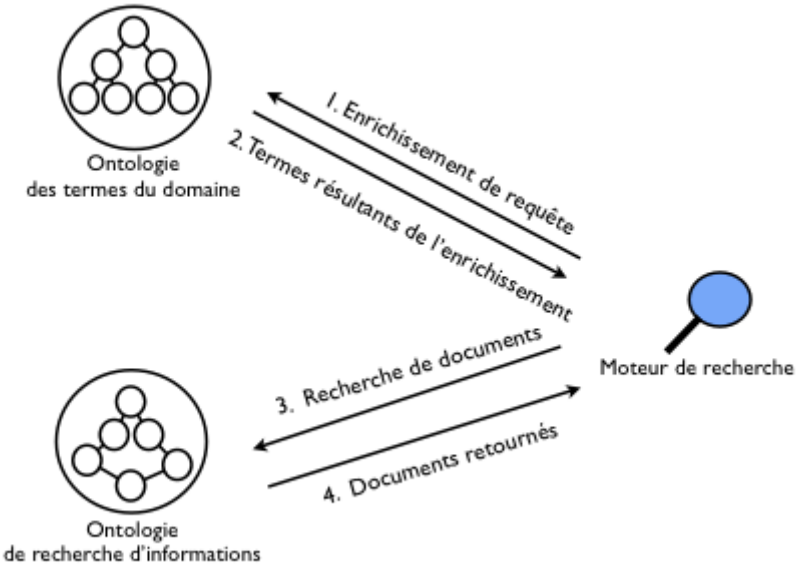


FIGURE 2.13 – **Approche générale d'une recherche documentaire à partir d'ontologies de domaines.**

Pour expliquer l'idée des enrichissements proposés, nous exprimons les deux ontologies créées de manière formelle. Ensuite, nous expliquons le fonctionnement de la recherche documentaire pour chacun des enrichissements de requête présentés et nous les illustrons à partir d'exemples définis pour le domaine de la maladie de Parkinson.

De manière formelle, nous considérons :

– **l'ontologie structurant les termes du domaine** comme une structure telle que :

$$\mathcal{O}_1 := (C_1, \sqsubseteq_C, R_1, A_1) \tag{2.1}$$

où C_1 est l'ensemble des concepts, $\sqsubseteq_C$ est un ordre partiel sur C_1, R_1 est l'ensemble des relations définies sur C_1 x C_1 et A_1 l'ensemble des propriétés.

Nous en définissons également son lexique tel que :

$$Lex(\mathcal{O}_1) := (S_{1C}, S_{1R}, S_{1A}, Ref_{1C}, Ref_{1R}, Ref_{1A}) \tag{2.2}$$

où :

- les trois ensembles S_{1C}, S_{1R}, S_{1A} sont appelés « instances » pour les concepts, les relations et les attributs respectivement,
- la relation $Ref_{1C} \subseteq S_{1C}$ x C_1 est appelée référence lexicale pour les concepts,
- la relation $Ref_{1R} \subseteq S_{1R}$ x R_1 est appelée référence lexicale pour les relations,
- la relation $Ref_{1A} \subseteq S_{1A}$ x A_1 est appelée référence lexicale pour les attributs.

Dans notre contexte, les concepts appartenant à l'ensemble C_1 correspondent tous à ceux extraits du thesaurus utilisés. Les instances faisant partie de S_{1C} sont les termes ayant été extraits des documents.

- **l'ontologie de recherche d'informations** comme une structure telle que :

$$\mathcal{O}_2 := (C_2, \sqsubseteq_C, R_2, A_2) \tag{2.3}$$

où C_2 est l'ensemble des concepts, $\sqsubseteq_C$ est un ordre partiel sur C_2, R_2 est l'ensemble des relations définies sur C_2 x C_2 et A_2 l'ensemble des propriétés.

Son lexique est une structure tel que :

$$Lex(\mathcal{O}_2) := (S_{2C}, S_{2R}, S_{2A}, Ref_{2C}, Ref_{2R}, Ref_{2A}) \tag{2.4}$$

où :

- les trois ensembles S_2C, S_R, S_2A sont appelés « instances » pour les concepts, les relations et les attributs respectivement,
- la relation $Ref_{2C} \subseteq S_{1C}$ x C_2 est appelée référence lexicale pour les concepts,
- la relation $Ref_{2R} \subseteq S_{1R}$ x R_2 est appelée référence lexicale pour les relations,
- la relation $Ref_{2A} \subseteq S_{1A}$ x A_2 est appelée référence lexicale pour les attributs.

Pour cette ontologie, la définition de chaque concept $c_i \in C_2$ est composé d'une paire telle que $c_i :=(o_i, p_i,)$ avec :

- $o_i \subseteq S_{2C}$ et où S_{2C} correspond à l'ensemble des instances de l'ontologie. Ces instances représentent l'ensemble des documents du domaine concerné.

– $p_i \subseteq A_2$ et qui représentent l'ensemble des propriétés de l'ontologie. Ces propriétés correspondent, quant à elles, à l'ensemble des termes du domaine ayant été extraits des documents.

A travers cette formalisation, nous montrons que les termes utilisés pour représenter la connaissance du domaine ne sont pas considérés de manière univoque selon l'ontologie concernée. La figure 2.14 nous montre cette distinction et nous exprime que l'ontologie structurant les termes représente ceux-ci en tant que instances de concepts, tandis que l'ontologie de recherche d'informations, les représente comme des propriétés de concepts. Ceci implique que l'ensemble regroupant les instances défini pour l'ontologie structurant les termes soit identique à celui représentant les propriétés de l'ontologie de recherche d'informations : $S_{1C} \equiv A_2$. Cette distinction résulte du choix d'avoir utilisé les mêmes principes que ceux définis pour les treillis de concepts par les travaux de Carpineto[Carpineto and Romano, 1996]. Néanmoins, il aurait été tout a fait valable de considérer les termes de l'ontologie de recherche d'informations en tant qu'instances de concepts et les documents comme propriétés de concepts.

Dès lors, pour permettre un enrichissement de requête pertinent, nous étendons la requête initiale de l'utilisateur à partir des instances de l'ontologie structurant les termes du domaine.

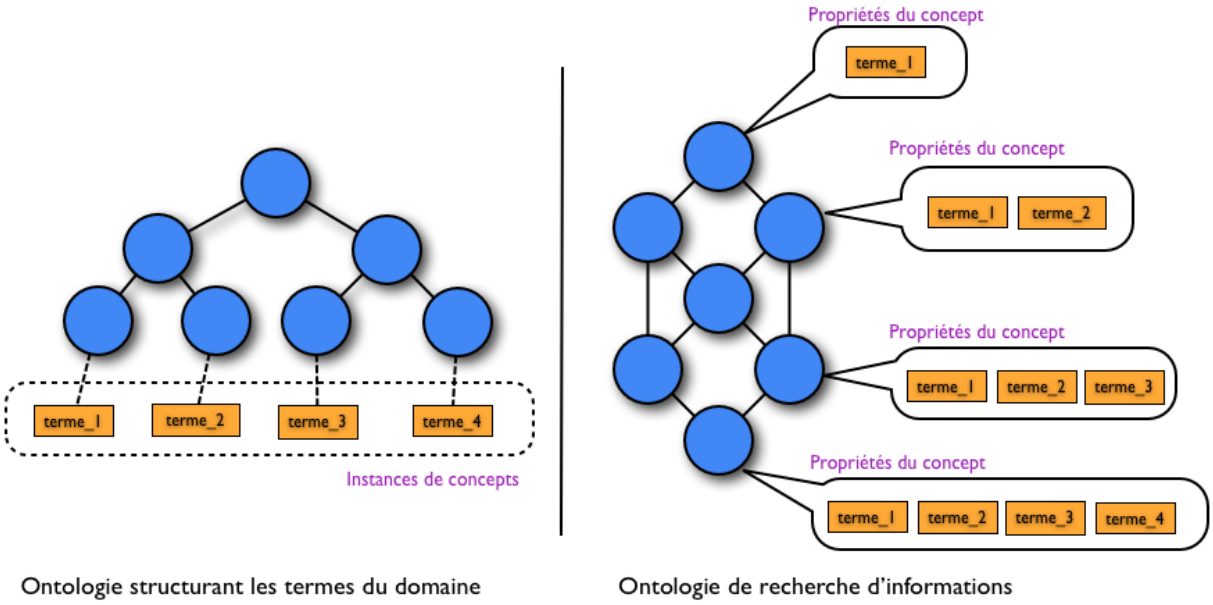


FIGURE 2.14 – **Définition d'un terme selon l'ontologie concernée.**

2.5.1 Enrichissement par spécialisation

L'enrichissement de requête par spécialisation consiste à enrichir la requête de l'utilisateur d'un ensemble de termes plus spécifiques. Ce mode d'enrichissement s'avère particulièrement utile lorsque l'utilisateur souhaite connaître un ensemble terminologique découlant d'un même terme et reflétant une catégorie. De plus, ce type d'enrichissement peut amener de nouvelles connaissances à travers la découverte de termes pas encore connus par l'utilisateur. Pour ce faire, nous localisons dans un premier temps le concept de l'ontologie correspondant pour chacun des termes figurant dans la requête initiale : $c_{Req} \in C_1$. Grâce à la relation hiérarchique qui structure les concepts de l'ontologie, l'ensemble des instances des concepts qui subsument le concept c_{Req} sont aussi instances de c_{Req} par héritage. Par conséquent, à partir du concept c_{Req}, nous récupérons l'ensemble de ces instances et celles des concepts qu'il subsume.

De manière formelle, la spécialisation dans l'ontologie structurant les termes du domaine se définit comme :

$$specialisation(c_{Req}) := instances(c_{Req}) \tag{2.5}$$

Exemple Ce type d'enrichissement trouve facilement son utilité dans la plupart des domaines offrant non seulement la possibilité à l'utilisateur d'enrichir sa requête, mais aussi de lui permettre d'acquérir de nouvelles connaissances à travers la découverte de nouveaux termes. Dans le cadre de la maladie de Parkinson, nous illustrons celle-ci en montrant comment obtenir l'ensemble des symptômes pouvant être diagnostiqués pour cette maladie (voir figure 2.15). Ainsi, pour répondre à cette question, nous effectuons une spécialisation à partir du terme « symptom ». Le résultat de cet enrichissement nous donnera l'ensemble des symptômes appartenant au domaine et possédés parmi les documents indexés. La recherche de ce terme dans notre ontologie nous retourne le concept c_{Req}={Sign_or_Symptom}. A partir de ce concept, nous en déduisons l'ensemble des instances de tous les concepts qu'il subsume soit « Pain » et « Tremor ». Ces instances seront par la suite ajoutées à la requête initiale.

- Requête initiale : {symptom}
- Requête enrichie : {symptom, pain, tremor}

2.5.2 Enrichissement par généralisation

Comme déjà précisé préalablement, l'enrichissement par généralisation vise à enrichir la requête de l'utilisateur d'un ensemble de termes plus généraux. Tout comme pour l'enrichissement précédent, il est nécessaire d'identifier le concept de l'ontologie pour chacun des termes figurant dans la requête initiale : $c_{Req} \in C_1$. A partir de ce concept, nous récupérons l'ancêtre le subsumant de deux niveaux supérieurs : $c_{Anc} \in C_1$ et où $c_{Req} \sqsubseteq c_{Anc}$. Le choix de remonter de deux niveaux s'explique d'une part afin de ne pas se contenter des instances des concepts « frères » du concept c_{Req} et d'autre part afin d'éviter de remonter trop haut dans l'ontologie et ainsi de rester proche du sens du concept correspondant à la requête. Ainsi, a partir des concepts c_{Req} et c_{Anc}, nous récupèrons l'ensemble des instances qu'ils possèdent pour ensuite ne retourner que celles du concept c_{Anc} n'étant pas possédées par le concept c_{Req}.

De manière formelle, la généralisation dans l'ontologie structurant les termes du domaine se définit comme :

$$generalisation(c_{Req}) := instances(c_{Anc}) \backslash instances(c_{Req}) \; sachant \; que \; c_{Req} \sqsubseteq_2 c_{Anc} \tag{2.6}$$

Exemple Pour le domaine de la maladie de Parkinson, nous détaillons ce type d'extension en partant d'une requête contenant le terme *{depression}* et illustré en figure 2.16. Pour commencer, nous récupérons le concept c_{Req}={Mental_Depression} correspondant au terme de la requête ainsi que son ancêtre le subsumant de deux niveaux soit c_{Anc}={Disease_or_Syndrome}. A partir de ces deux concepts, nous récuperons les instances de chacun d'entre eux. L'ensemble complémentaire des instances du concept *{Mental_Depression}* parmi l'ensemble des instances du concept *{Disease_or_Syndrome}* nous retournera les instances *{parkinson, dementia}* qui enrichiront par la suite la requête initiale de l'utilisateur.

- Requête initiale : {depression}
- Requête enrichie : {depression, parkinson, dementia}

2.5.3 Enrichissement par relation

Cette dernière extension diffère des deux précédentes, car celle-ci nécessite d'une part un terme défini par l'utilisateur et d'autre part une relation choisie parmi l'ensemble de celles possédées par l'ontologie structurant les termes du domaine et que nous représenterons par ; $r_{req} \in R_1$. Tout comme pour les deux précédents types d'enrichissement, nous commençons par rechercher le concept c_{req} correspondant au terme fourni par l'utilisateur. Ensuite, nous vérifions si le concept c_{req} possède la propriété r_{req} demandée par l'utilisateur. Notez qu'il n'est pas nécessaire de parcourir l'ensemble des concepts ancêtres subsumant le concept c_{req} car l'ensemble de leurs propriétés sont héritées. Dès lors, chacun des concepts c_i vérifiant la relation $r_{req}(c_{req},c_i)$ est ensuite spécialisé. Les instances appartenant aux concepts résultant des spécialisations respectives sont ensuite ajoutées à la requête initiale.

De manière formelle, l'extension par relation dans l'ontologie structurant les termes du domaine se définit comme :

$$relation(c_{Req}, r_{Req}) := instances(spec(c_1, c_2..., c_n)) \; tel \; que :$$

$$\forall \; i \in n, \; r_{Req}(c_{Anc}, c_i) \; et \; c_{Req} \sqsubseteq c_{Anc} \tag{2.7}$$

Exemple Pour ce cas de figure, nous considérons le souhait de savoir quels seraient les termes vérifiant la relation « **causes** » avec le terme « dopamine » qui se trouve être l'une des substances pharmacologiques prescrites pour la pathologie de Parkinson. La construction d'une telle requête nécessite pour l'utilisateur de préciser le terme « dopamine » ainsi que de choisir la relation « **causes** » faisant partie des 19 relations sémantiques représentées dans l'ontologie structurant les termes (voir figure 2.6). Nous recherchons le terme dans notre ontologie et nous obtenons le concept c_{Req}={Mental_Depression}. Ce concept hérite de l'ensemble des propriétés des concepts qui le subsument dont le concept {Substance}. Ainsi, comme nous le montrait la figure 2.7, le concept {Substance} vérifie la relation « **causes** » avec les concepts « *Mental_or_Behavioral_Dysfunction* », « *Disease_or_Syndrome* » et « *Pathologic_Function* ». Par conséquent, le concept c_{Req} les vérifie aussi. Une spécialisation est appliquée sur les concepts vérifiant la relation et enrichira par la suite la requête initiale à partir des instances appartenant aux concepts résultants.

- Requête initiale : {dopamine} avec la relation {causes}
- Requête enrichie : {dopamine, dementia, depression, parkinson}

2.5.4 Conclusion

Dans ce chapitre, nous avons présenté une méthodologie permettant la conception d'un moteur de recherche sémantique défini pour un domaine. Nous avons détaillé les grandes étapes menant à l'élaboration de ce moteur à travers la conception de deux ontologies en prenant pour exemple le domaine de la maladie de Parkinson. Comme point de départ, nous avons extrait un ensemble terminologique représentant notre domaine à partir d'un corpus de textes. Ensuite, nous avons entamé la conception de la première ontologie visant à structurer les termes du domaine à travers l'utilisation d'un thesaurus. Pour la deuxième ontologie, nous avons utilisé la méthode de fouille : Analyse Formelle de Concepts, regroupant nos documents en fonction des termes qu'ils partagent en un treillis de concepts. Par la suite, nous avons représenté le treillis de concepts résultant en une ontologie dans le but de pouvoir y faire du raisonnement. Pour terminer, en utilisant conjointement les deux ontologies, nous avons montré la processus de recherche proposant à l'utilisateur trois types d'enrichissements de requête.

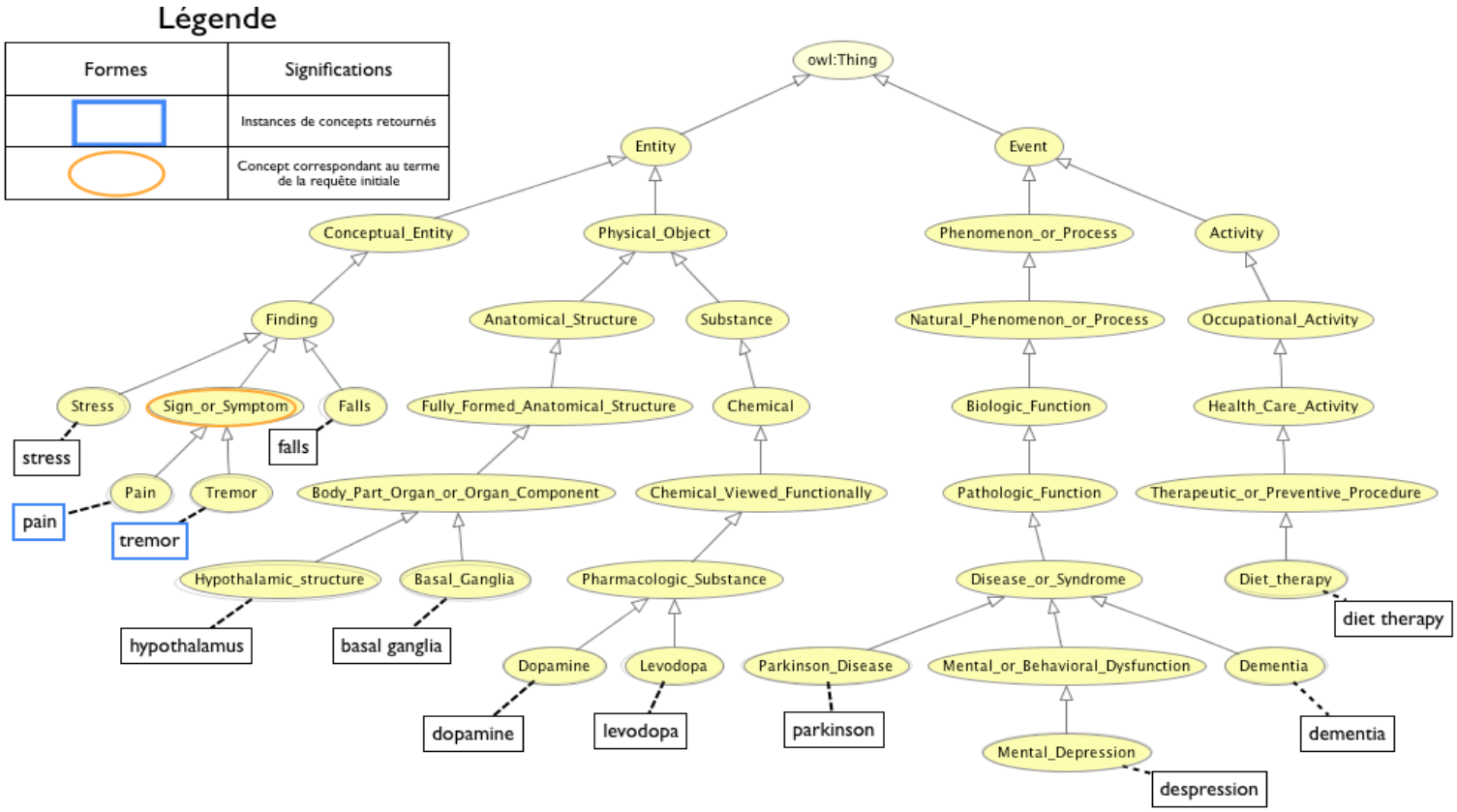


FIGURE 2.15 – **Enrichissement de termes par spécialisation à partir de l'ontologie structurant les termes de la maladie de Parkinson.**

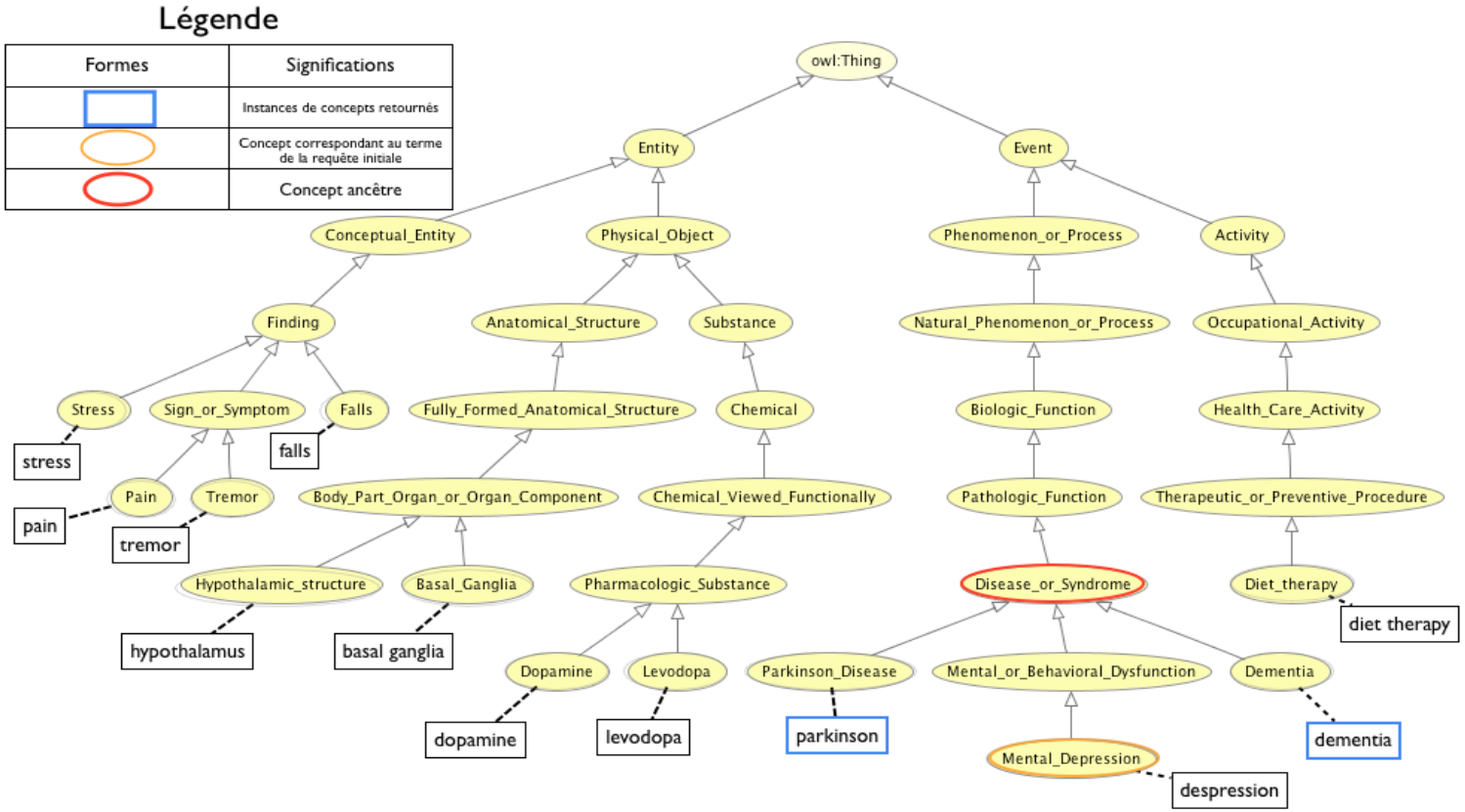


FIGURE 2.16 – **Enrichissement de termes par généralisation à partir de l'ontologie structurant les termes de la maladie de Parkinson.**

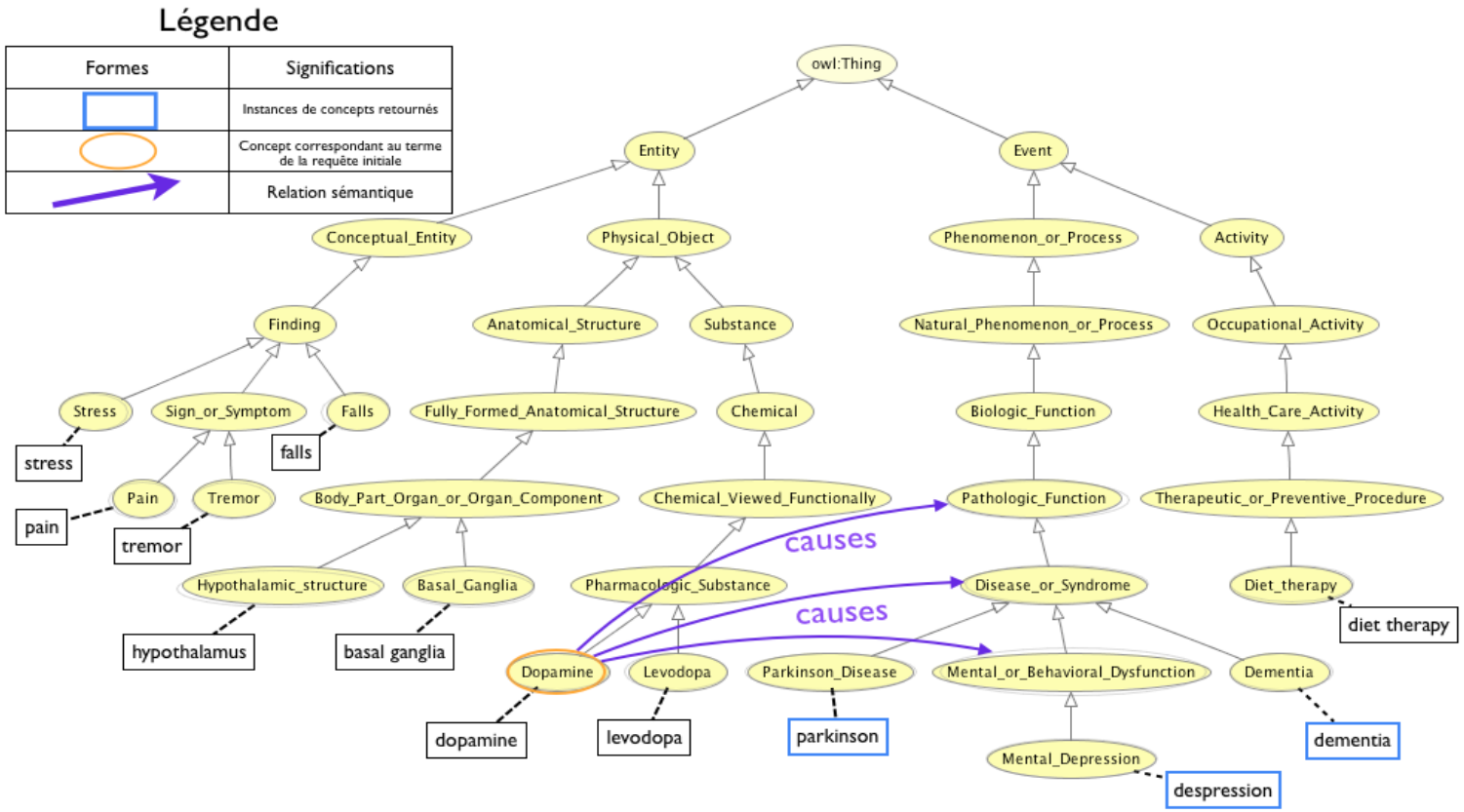


FIGURE 2.17 – **Enrichissement de termes par relation à partir de l'ontologie structurant les termes de la maladie de Parkinson.**

Chapitre 3

Développement de prototypes

3.1 Introduction

Pour tester la validité de notre méthodologie, nous avons implémenté un ensemble de trois prototypes se rapportant chacun à une étape spécifique de notre approche. Développé dans le cadre d'un stage en entreprise effectué au CETIC (Centre d'Excellence en Technologies de l'Information et de la Communication), ces prototypes ont tous été implémentés à l'aide du langage Java afin d'être facilement portable sur différentes plateformes et compatible avec les différentes librairies et langages utilisés. De plus, ceux-ci ont été testés sur un jeu d'essai comprenant 9 ressources textuelles traitant toutes du domaine de la maladie de Parkinson.

Les trois prototypes développés sont les suivants :

– « **UMLS2OWL** » : il vise à construire une ontologie en structurant un ensemble de termes à partir du thesaurus UMLS. Ce prototype se rapporte au processus de construction de l'ontologie défini lors de l'acquisition des connaissances du domaine (voir section 2.3.2).

– « **TREILLIS2OWL** » : ce prototype a pour but de représenter un treillis de concepts en une ontologie et de rendre ainsi automatique le processus de notre approche définie en section 2.4.2.

– « **SearchEngine** » : ce dernier prototype représente le moteur de recherche sémantique utilisant conjointement les deux ontologies résultant des deux prototypes précédents et permettant ainsi d'effectuer une recherche documentaire (section 2.5) à partir de plusieurs extensions de requête.

Avant de poursuivre, il est bon de noter que les prototypes « TREILLIS2OWL » et « SearchEngine » ne dépendent pas d'un quelconque domaine spécifique. Contrairement au prototype « UMLS2OWL » qui lui a été défini pour le thesaurus UMLS et donc pour la conception d'ontologies représentant des connaissances se rapportant à un domaine biomédical.

Dans ce chapitre, nous présentons pour commencer le cadre technologique à partir duquel les prototypes ont été construits. Ensuite, nous détaillons les différents prototypes à travers leur architecture, leur conception ainsi qu'à travers leur utilisation.

3.2 Cadre technologique utilisé

Dans cette section, nous présentons l'ensemble des technologies et langages utilisés pour la conception des prototypes développés.

- Java JDK 1.6. Le langage de programmation Java a été choisi pour le développement des trois prototypes en raison de sa portabilité ainsi que pour permettre l'utilisation du Framework Jena et du langage JSP décrit ci-dessous.

- SAOP [1] (Simple Object Access Protocol). C'est un protocole permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications. Ce protocole est utilisé dans le modèle client-serveur et permet de gérer les différents messages entre ces entités. Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP mais peut également se faire par un autre protocole comme SMTP. Ce protocole a dû être utilisé pour communiquer avec le service web proposé par UMLS.

- Jena Framework [2]. C'est un framework java utilisé pour la construction d'applications Web sémantique. Il fournit un environnement de programmation défini pour les langages du Web sémantique tels que RDF, RDFS, OWL et SPARQL mais aussi un ensemble de règles basées sur un moteur d'inférence.

- JSP [3] (Java Server Page). C'est une technique basée sur Java qui permet

1. SAOP : http ://www.w3.org/TR/soap/. Date 21/04/2010
2. Jena Framework : http ://jena.sourceforge.net/. Date 21/04/2010
3. JSP : http ://java.sun.com/products/jsp/. Date 21/04/2010

aux développeurs de générer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique. Les pages Web crées en JSP permettent ainsi d'appeler les différentes méthodes et classes définies pour chacune des ontologies utilisées par le moteur de recherche.

3.3 Le prototype « UMLS2OWL »

3.3.1 Architecture

Ce premier prototype « UMLS2OWL » vise à construire une ontologie structurant un ensemble terminologique fourni en entrée à partir des concepts du « *Metathesaurus* » et des types sémantiques d'UMLS. La figure 3.1 nous présente l'architecture de ce prototype nécessitant un fichier en entrée comprenant l'ensemble des termes reflétant au mieux le domaine visé et fournissant en sortie l'ontologie structurant ces mêmes termes à partir du thesaurus UMLS.

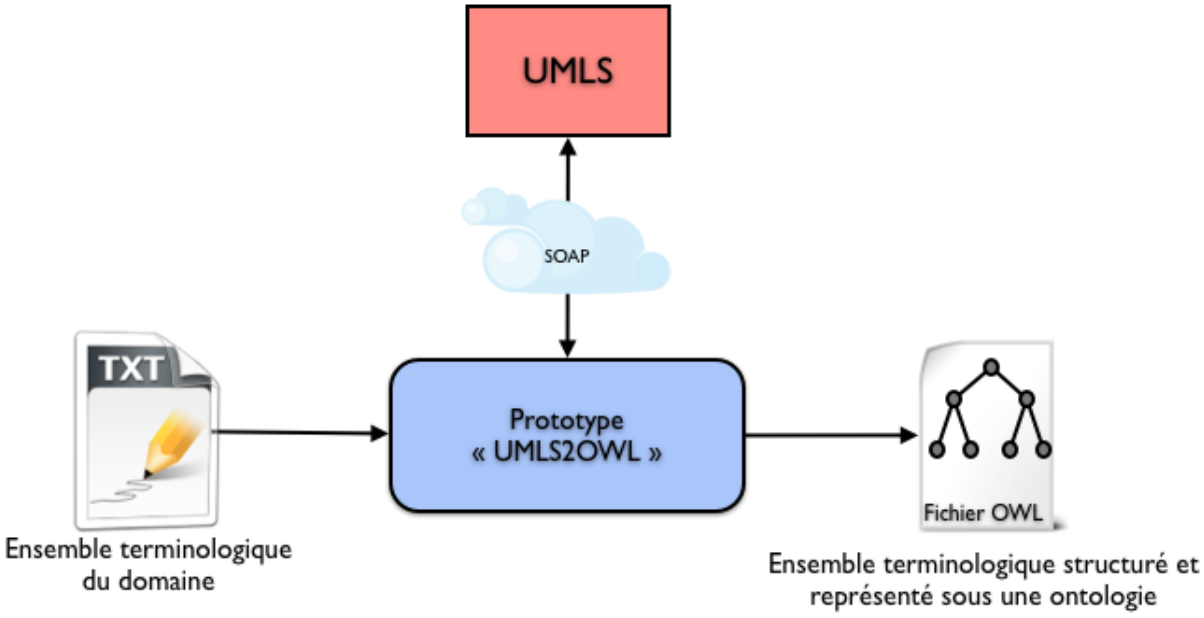


FIGURE 3.1 – **Data flow diagram du prototype « UMLS2OWL »**

3.3.2 Conception

L'ensemble du vocabulaire UMLS constitue une ressource riche et idéale pour la conception d'ontologies liées à un domaine biomédical, mais sa complexité et sa taille accentuent sa difficulté de compréhension et d'utilisation. C'est pourquoi le prototype proposé a été developpé afin de faciliter la manipulation du thesaurus UMLS et aussi de permettre à tout utilisateur de pouvoir construire une ontologie de domaine facilement et rapidement à partir d'un ensemble de termes choisis.

Afin que notre prototype puisse intéragir avec les données du thesaurus, nous avons installé une API [4] (Application Programming Interface) mise à disposition pour les développeurs et permettant l'utilisation du service web d'UMLS.

Celle-ci s'effectue en trois étapes et est détaillée dans le « Developer's Guide » [5] d'UMLS :

1. Obtention du protocole SOAP2 1.2 (Simple Object Access Protocol) afin de permettre l'échange de messages avec le service Web d'UMLS.
2. Téléchargement de l'authentification « *WSDL* » [6] (Web Services Description Language) à partir du site d'UMLS.
3. Téléchargement du « *UMLS WSDL* » également à partir du site UMLS.

Une fois ces étapes effectuées, il est alors possible d'obtenir les « stubs clients » (classes permettant de communiquer avec les services Web d'UMLS) au moyen de deux lignes de commande précisées par la figure 3.2 ainsi que sur le site « Developer's Guide » [7] :

4. API : http ://fr.wikipedia.org/wiki/Interface_de_programmation. Date : 24/07/2010.
5. Developer's Guide UMLS : http ://umlsks.nlm.nih.gov/DocPortlet/html/dGuide/appls.html Date : 24/07/2010.
6. WSDL : http ://www.w3.org/TR/wsdl. Date : 24/07/2010.
7. Developer's Guide UMLS : http ://umlsks.nlm.nih.gov/DocPortlet/html/dGuide/appls.html Date : 24/07/2010.

```
prompt> java -classpath <CLASSPATH> \
                  org.apache.axis.wsdl.WSDL2Java \
                  -o . \
                  -d Session \
                  -s -S true \
                  -Nurn:authorization.umlsks.nlm.nih.gov
gov.nih.nlm.umlsks.authorization \
                  <CAS_WSDL_LOCATION>

prompt> java -classpath <CLASSPATH> \
                  org.apache.axis.wsdl.WSDL2Java \
                  -o . \
                  -d Session \
                  -s -S true \
                  -Nurn:umlsks.nlm.nih.gov gov.nih.nlm.umlsks \
                  <UMLSKS_WSDL_LOCATION>
```

FIGURE 3.2 – **Lignes de code nécessaires pour l'obtention des classes « stub client » afin d'intéragir avec le service web d'UMLS.**

Ces classes sont par la suite utilisées et intégrées dans l'environnement de développement du prototype. Ainsi, il nous est possible d'interagir avec le thesaurus UMLS nous permettant de rechercher, d'accéder et d'extraire les concepts du thesaurus. Cependant, pour que notre prototype puisse structurer les concepts extraits d'UMLS en ontologie, il nous est nécessaire de pouvoir créer et générer du code OWL. Pour effectuer cette tâche, nous avons utilisé le Framework Jena permettant de concevoir et de manipuler des ontologies et d'en générer leur code OWL. L'utilisation de Jena se fait par un simple téléchargement des librairies à partir du site [8] et l'intégration de celles-ci à l'environnement de développement.

La figure 3.3 présente le diagramme de classes du prototype reflétant ses classes principales :

- **Umls_Connection** : classe qui à pour but d'établir la connexion avec le service web d'UMLS. De plus, cette classe utilise le pattern « Singleton » afin qu'une seule et même instance soit utilisée par l'ensemble des autres classes.

- **Metathesaurus** : classe qui regroupe l'ensemble des requêtes utilisées pour récupérer les propriétés d'un concept issu du « *Metathesaurus* » d'UMLS.

- **SemanticNetwork** : classe qui regroupe l'ensemble des requêtes effectuées

8. Jena Framework : http ://jena.sourceforge.net/. Date 21/04/2010

sur le réseau sémantique d'UMLS.

– **OntologyManagement** : classe qui va s'occuper de constituer l'ontologie de termes à partir des concepts extraits et de leurs types sémantiques afin de générer le code OWL.

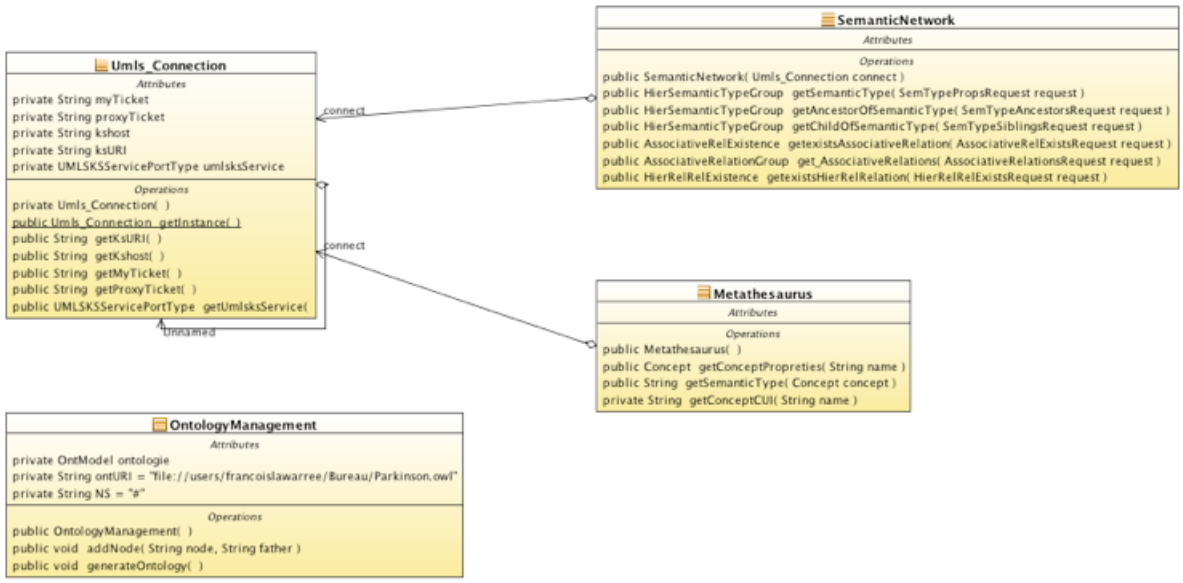


FIGURE 3.3 – **Diagramme de classes du prototype « UMLS2OWL ».**

3.3.3 Utilisation

Nous détaillons l'utilisation de ce prototype au travers de son interface (figure 3.4) et en présentant les différentes étapes nécessaires pour la bonne utilisation de celui-ci :

1. La première étape nécessite pour l'utilisateur de spécifier le chemin d'accès du fichier contenant l'ensemble terminologique du domaine médical concerné (« Input file »). Une fois le fichier choisi, l'interface définit par défaut le chemin du fichier résultat (« Output folder ») à partir du répertoire d'où provient le fichier de départ. Par la suite, il est possible de lancer la création de l'ontologie par l'exécution du bouton « *Create New Ontology* ». Cette action créera l'ontologie d'une manière logique à partir de l 'API Jena sans

pour autant en avoir générer le fichier OWL. Au terme de cette première étape, un fichier contenant les termes n'ayant pas été trouvés dans UMLS est créé.

2. La deuxième étape permet à l'utilisateur d'enrichir l'ontologie, créée logiquement, de nouveaux termes en plus que ceux définis dans le fichier input. Cette étape s'avère également utile lorsque certains termes définis dans le fichier de départ n'ont pu être trouvés par UMLS et laisse donc libre choix à l'utilisateur d'en ajouter d'autres équivalents sémantiquement.

3. La troisième étape est une « check box » offrant la possibilité à l'utilisateur de tenir compte des relations sémantiques pouvant exister parmi les concepts extraits de UMLS et de les ajouter à l'ontologie contenue en mémoire.

4. Cette dernière étape générera le fichier résultat du prototype qui contiendra le code OWL de l'ontologie créée.

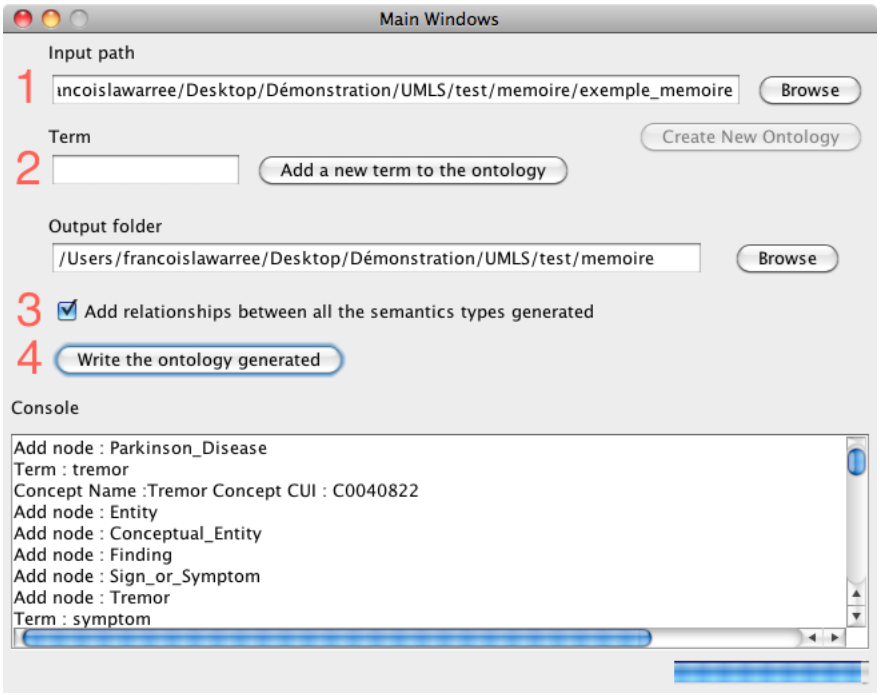


FIGURE 3.4 – **Interface du prototype « UMLS2OWL ».**

3.4 Le prototype « TREILLIS2OWL »

3.4.1 Architecture

Le prototype « TREILLIS2OWL » a pour objectif de représenter un treillis de concepts, généré à partir du logiciel Galicia [9] et contenu dans un fichier de type « texte », en une ontologie.

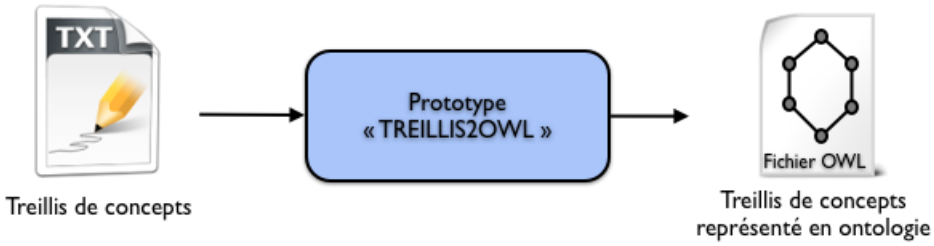


FIGURE 3.5 – **Data flow diagram du prototype « TREILLIS2OWL ».**

3.4.2 Conception

Pour la conception de ce prototype, nous avons utilisé le framework Jena afin de construire, de manière logique, l'ontologie représentant le treillis donné en entrée et afin d'en générer le code OWL.

Le développement du prototype « TREILLIS2OWL » a été fait en fonction de la structure du fichier qu'il prend en entrée. En effet, celui-ci contient la représentation textuelle d'un treillis de concepts dont les données sont agencées et encodées de manière précise. La figure 3.6 nous illustre le contenu du fichier se rapportant au treillis de concepts de la figure 2.9. Chaque ligne du fichier correspond à la description d'un concept formel et se compose de trois parties :

- le nom du concept ;
- l'ensemble des objects appartenant au concept ;
- et l'ensemble des propriétés vérifiées par le concept ;

9. Galicia : http ://www.iro.umontreal.ca/ galicia/. Date : 28/07/2010

```
C0 : {Texte_1, Texte_2, Texte_3, Texte_4} x {}
C1 : {Texte_1, Texte_2, Texte_4} x {terme_3}
C2 : {Texte_3, Texte_4} x {terme_5}
C3 : {Texte_1, Texte_2} x {terme_1, terme_3}
C4 : {Texte_4} x {terme_3, terme_4, terme_5}
C5 : {Texte_3} x {terme_2, terme_5}
C6 : {} x {terme_1, terme_2, terme_3, terme_4, terme_5}
```

FIGURE 3.6 – **Structure d'un fichier en entrée pour le prototype « TREILLIS2OWL ».**

Chacune de ces trois parties sont traitées et traduites par le prototype en langage OWL. Ainsi, le nom d'un concept sera représenté par une classe OWL (owl :Class), l'ensemble des objets d'un concept deviendront des instances de classe et l'ensemble des propriétés seront considérées en tant que propriétés (owl :onProperty) de classes. A titre d'exemple, la figure 3.7 nous montre de façon simplifiée le traitement effectué par le prototype pour le concept formel « C3 ».

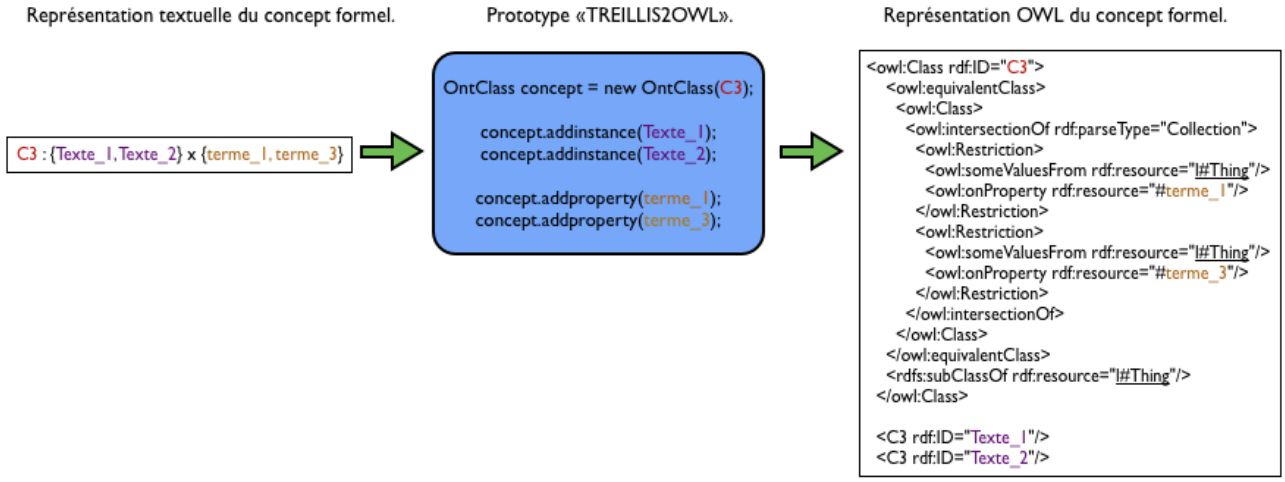


FIGURE 3.7 – **Représentation du concept « C3 » en langage OWL à partir du prototype « TREILLIS2OWL ».**

3.4.3 Utilisation

Ce prototype ne possède pas d'interface. L'utilisation de celui-ci consiste simplement à spécifier le chemin d'accès contenant le treillis de concept décrit dans un fichier texte et généré à l'aide de l'outil Galicia [10]. L'exécution retournera l'on-

10. Galicia : http ://www.iro.umontreal.ca/ galicia/. Date : 28/07/2010

tologie résultante dans le même répertoire que le fichier donné en entrée.

3.5 Le prototype « SearchEngine »

3.5.1 Architecture

Ce prototype représente le moteur de recherche qui utilise conjointement le résultat des deux prototypes précédents. Ainsi, pour que notre programme « SearchEngine » puisse fonctionner, celui-ci nécessite l'utilisation de deux ontologies. L'une sera utilisée afin d'effectuer la recherche d'informations proprement dite, l'autre permettra d'améliorer la pertinence des résultats selon trois types d'enrichissements de requête choisis en fonction des besoins de l'utilisateur.

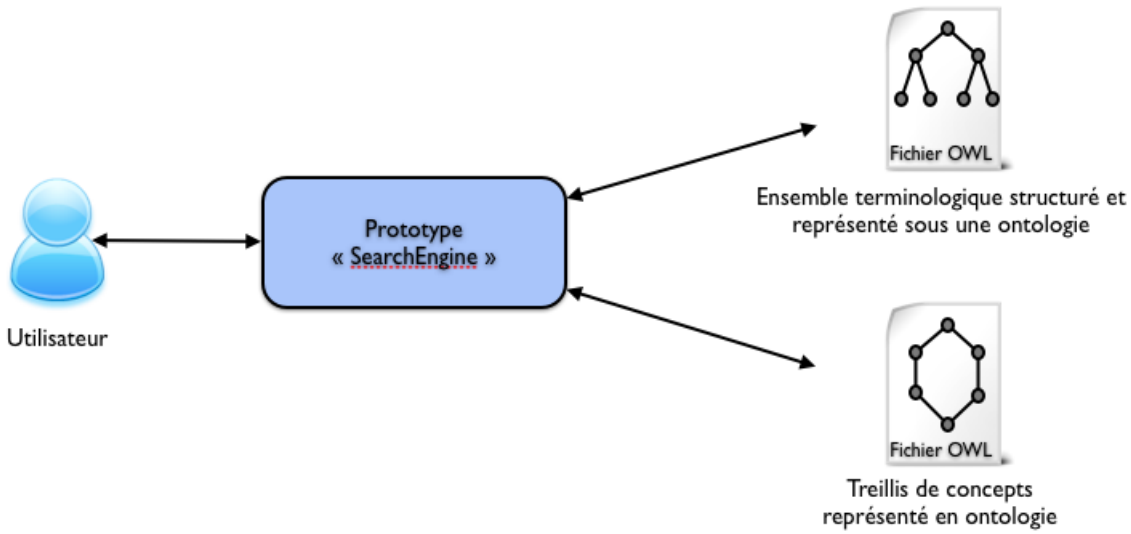


FIGURE 3.8 – **Data flow diagram du prototype « SearchEngine »**

3.5.2 Conception

Pour l'environnement de développement de cet outil nous avons utilisé une fois encore le framework Jena pour permettre au prototype de raisonner et d'interagir avec les deux ontologies. Nous voulions également que celui-ci puisse être accessible à travers le web, c'est pourquoi nous avons opté pour l'utilisation du langage

JSP [11] offrant la possibilité à des pages HTML [12] de générer leur contenu dynamiquement par l'appel des classes Java.

Pour visualiser la conception de notre moteur de recherche, la figure 3.8 nous illustre son diagramme de classes. Les classes du package « Ontologies » correspondent aux deux classes qui raisonneront sur les deux ontologies utilisées. L'autre package nommé « JSP Pages » contient l'ensemble des pages web qui communiqueront avec les classes du package « Ontologies » à partir de la technologie JSP. L'illustration des pages web du moteur de recherche est présentée dans la section 3.5.3.

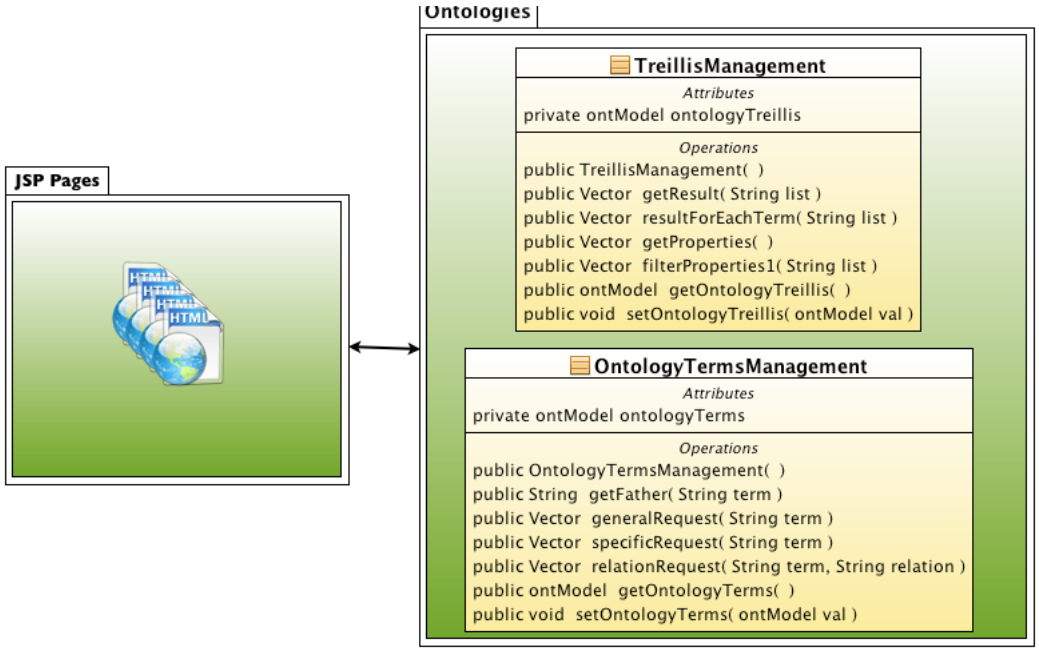


FIGURE 3.9 – **Diagramme de classes du prototype « SearchEngine ».**

3.5.3 Utilisation

Dans cette partie nous illustrons différentes captures d'écran représentant les pages web de notre moteur de recherche sémantique et dévoilant les étapes devant être effectuées pour une recherche d'informations.

La recherche d'informations s'effectue au travers de quatre pages web :

11. JSP : http ://java.sun.com/products/jsp/. Date 21/04/2010
12. HTML : http ://www.w3.org/MarkUp/. Date : 28.07/2010

1. La figure 3.10 illustre la première page du moteur et invite l'utilisateur à y introduire le ou les termes qu'il souhaite rechercher.

2. A partir de l'ensemble des termes entrés par l'utilisateur, la deuxième page génère dynamiquement un tableau laissant libre choix à l'utilisateur du type d'enrichissement de requête qu'il souhaite appliquer pour chacun des termes introduits ; par « defaut », par « généralisation », par « spécialisation », par « relation » (figure 3.11).

3. La troisième page présente les termes obtenus pour l'ensemble des extensions de requêtes choisies à l'étape précédante. L'utilisateur peut ainsi, choisir le ou les termes à partir desquels il souhaite enrichir sa requête initiale (figure 3.12).

4. La dernière page illustre les documents retournés et vérifiant un ou plusieurs termes de la requête enrichie de l'utilisateur (figure 3.13).

FIGURE 3.10 – **Première page du prototype « SearchEngine » : introduction des termes initiaux.**

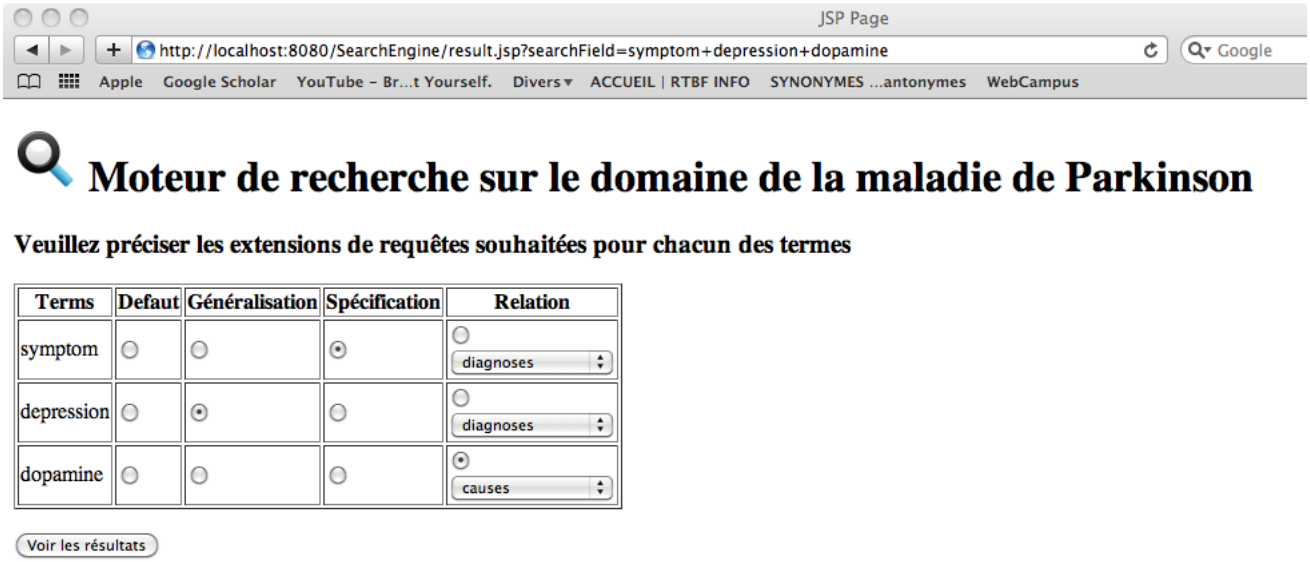


FIGURE 3.11 – **Deuxième page du prototype « SearchEngine » : choix des extensions de requête.**

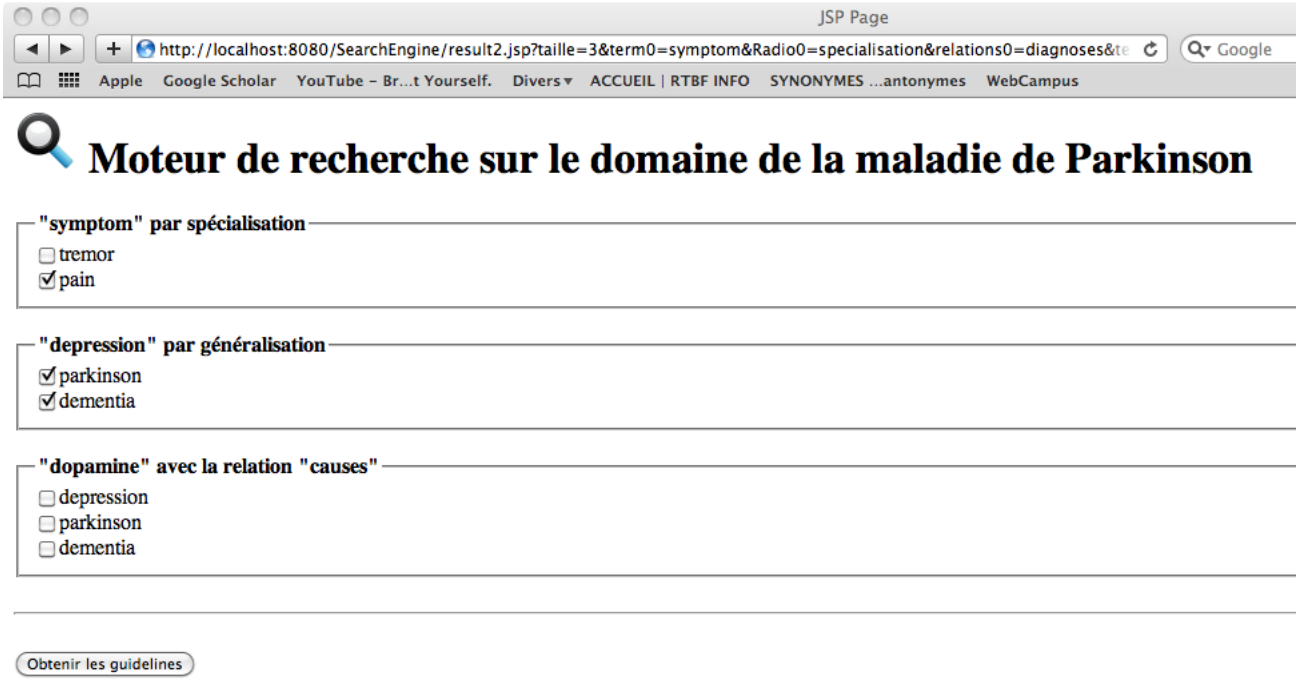


FIGURE 3.12 – **Troisième page du prototype « SearchEngine » : résultats des extensions de requête et choix des termes à enrichir.**

FIGURE 3.13 – **Quatrième page du prototype « SearchEngine » : résultats de la requête enrichie.**

3.5.4 Conclusion

Dans ce chapitre, nous avons présenté un ensemble de trois prototypes ayant été conçus dans le cadre d'un stage effectué au CETIC. Chacun des prototypes présenté se rapporte un l'une des étapes de notre méthodologie détaillée dans le chapitre deux. Pour chacun d'entre eux, nous avons présenté leur architecture générale, les étapes menant à leur conception et leur utilisation à travers chaque interface.

Chapitre 4

Expérimentation

Dans ce chapitre, nous présentons une expérimentation de la méthodologie développée dans le cadre d'un stage au CETIC. L'expérimentation est définie pour le domaine de la maladie de Parkinson et se rapporte au contexte du projet « e-Health » de la région Wallonne.

Dans le cadre de ce stage, la définition de notre corpus de départ fut composée essentiellement de « guidelines » traitant de la maladie de Parkinson. Le choix de ce type de ressources est motivé par leur structure clairement définie ainsi que pour leur vocabulaire relativement bien délimité et contrôlé par des experts du domaine. De plus, ces ressources sont pour la plupart concises et contiennent les informations essentielles sur la pathologie qu'elles traitent. L'ensemble des « guidelines » composant le corpus ont toutes été récoltées à partir de différents sites médicaux trouvés sur internet. Nous avons privilégié les « guidelines » rédigées en anglais afin de faciliter la correspondance terminologique de celles-ci avec le vocabulaire provenant du thesaurus médical utilisé et décrit majoritairement en anglais.

A partir de ces ressources textuelles, nous en avons extrait un ensemble terminologique de domaine de manière manuelle à travers la lecture et l'analyse de l'ensemble du corpus. A cet effet, 250 termes se rapportant à la maladie de Parkinson ont été extraits.

Par la suite, nous avons entamé la construction de l'ontologie structurant les termes à l'aide du prototype « UMLS2OWL » implémenté pour ce processus et spécifique au thesaurus UMLS. Lors de la construction de cette ontologie, nous avons constaté qu'un grand nombre de termes extraits pouvaient se rapporter à un ou plusieurs concepts du thesaurus. Par exemple, lorsque l'on effectue une recherche dans UMLS pour le terme « depression », on constate que celui-ci intervient dans plusieurs concepts comme « Actual depression », « Depressed - symptom »,

« Depressed mood », « Depression motion », « Depressive disorder », « Depressive episode unspecified » ou encore « Mental Depression ». De plus, un concept extrait de UMLS peut lui-même être rattaché à plusieurs types sémantiques selon le contexte dans lequel il s'inscrit. Pour résoudre ce problème, nous laissons à l'utilisateur la possibilité d'effectuer un choix lorsqu'une telle situation survient. Ainsi, nous lui proposons de déterminer le concept ou le type sémantique le plus approprié en fonction du contexte dans lequel le terme extrait est utilisé. Lors de la construction de notre ontologie partant de 250 termes, nous avons relevé en moyenne une soixantaine d'interactions ce qui allonge fortement le temps employé pour construire notre ontologie (5 heures). De plus, le choix d'utiliser le service web d' UMLS au lieu d'une installation sur un support numérique ralentit également le temps consacré à la construction de l'ontologie. Néanmoins, ces choix s'avèrent indispensables afin de représenter au mieux la connaissance du domaine tout en disposant de la dernière version du thesaurus UMLS.

Au terme de la construction, l'ontologie structurant les termes de notre domaine comprenait 244 classes dont 178 correspondaient aux termes extraits et 66 aux types sémantiques extraits de UMLS (voir figure 6.4 en annexe). Cependant, certains des termes extraits ne possédaient pas de correspondance avec les concepts d'UMLS (60 termes parmi les 250). Ceux-ci sont alors définis en tant qu'instances du concept le plus général (« Top ») de l'ontologie afin de préserver le même ensemble de termes considérés par les deux ontologies (voir section 2.5).

La construction de l'ontologie définie pour la recherche d'informations fut, quant à elle, beaucoup plus rapide que pour l'ontologie structurant les termes. Pour ce faire, nous avons utilisé le logiciel Galicia nous permettant de construire le contexte formel défini pour nos « guidelines » et les 250 termes extraits. Toujours à partir de l'outil Galicia, nous générons ensuite le treillis de concepts correspondant de manière textuelle et représenté dans un fichier texte. Celui-ci est ensuite utilisé par notre prototype « TREILLIS2OWL » qui le traduit automatiquement en une ontologie.

L'étape finale de notre méthodologie consiste à mettre en œuvre la recherche de documents au travers d'un moteur sémantique utilisant conjointement nos deux ontologies. A cet effet, nous avons développé le prototype « SearchEngine » défini pour le domaine de la maladie de Parkinson. La configuration de celui-ci nécessite simplement de préciser les deux chemins d'accès conformes aux ontologies précédemment définies. Lorsque nous avons testé celui-ci, nous avons constaté que les termes définis initialement par notre requête se devaient d'être syntaxiquement identiques à ceux contenus par nos deux ontologies afin que des résultats puissent

être retournés. Dès lors, nous avons préféré améliorer la correspondance en vérifiant si la chaîne de caractères d'un terme de la requête est incluse parmi les noms définis dans les deux ontologies. Par exemple, le terme « Organ » n'aurait pu être trouvé au sein de l'ontologie structurant les termes en figure 2.5 car le concept s'y rapportant était nommé par « Body_Part_Organ_or_Organ_Component ». Cependant, cette amélioration peut conduire à un nombre trop important de résultats si le terme initial de la requête est trop court.

Au terme de l'élaboration de notre moteur de recherche, nous avons testé celui-ci au travers de quelques requêtes faisant intervenir les différents types d'enrichissements. Les premiers résultats récoltés améliorent de façon significative la pertinence et la précision des documents tout en offrant à l'utilisateur la possibilité de découvrir de nouveaux termes et donc de nouvelles connaissances lors du processus de recherche. Néanmoins, il nous est difficile de comparer notre moteur sémantique défini pour la maladie de Parkinson avec un autre système de recherche. Ceci est dû :

- à leur indexation syntaxique ;
- au fait qu'ils ne sont pas limités à un ensemble de documents ;
- qu'ils ne sont pas limités à un domaine précis ;
- à la particularité du domaine de la maladie de Parkinson.

Dans cette étude de cas, nous sommes partis d'un ensemble composé de six « guidelines » que nous avons lu et analysé afin d'en extraire les termes représentant notre domaine. Dans le cadre d'un passage à l'échelle prenant en compte une masse importante de ressources textuelles, il ne serait pas envisageable de procéder à une telle extraction manuelle. Néanmoins, celle-ci devrait être privilégiée dans un premier temps afin de définir une base de connaissance suffisante pour le domaine traité. Par la suite, l'utilisation d'outils d'extraction de termes automatiques devrait être envisagée. Dans ce cas, il sera nécessaire de mettre à jour les deux ontologies à partir des ressources et des termes ajoutés. Or, dans le cadre de nos outils développés, ceci impliquerait l'adaptation des prototypes « UMLS2OWL » et « TREILLIS2OWL ». Le prototype « UMLS2OWL » devrait pouvoir charger une ontologie de domaine et la compléter des nouveaux termes du domaine trouvés et contrôlés. Le prototype « TREILLIS2OWL » devrait également permettre de mettre à jour l'ontologie de recherche d'informations en prenant en compte les nouvelles ressources devant être indexées.

Chapitre 5

Conclusion et perspectives

Dans ce mémoire, nous avons proposé une méthodologie pour la conception d'un moteur de recherche sémantique permettant d'effectuer une recherche de documents fondée sur l'utilisation de deux ontologies d'un domaine. Pour ce faire, nous avons montré et expliqué les différentes étapes nécessaires pour mettre en œuvre sa conception et permettre de tenir compte de la sémantique contenue parmi les ressources textuelles du domaine indexées.

La première ontologie a pour objectif de représenter la base de connaissance du domaine. Sa conception repose sur un ensemble de termes extraits à partir de ressources textuelles du domaine et structurés à travers l'utilisation d'un thesaurus. Grâce à cette ontologie, nous augmentons la pertinence des documents retournés au moyen de trois types d'enrichissements de requête, dont l'un s'appuyant sur les relations sémantiques propres de l'ontologie.

La seconde ontologie indexe l'ensemble terminologique représentant le domaine avec l'ensemble des ressources textuelles utilisées. Nous avons représenté celles-ci à partir d'un treillis de concepts résultant d'une méthode de classification symbolique : AFC. Cette classification, une fois représentée en ontologie offre la possibilité d'effectuer du raisonnement sur celle-ci.

Au final, l'utilisation de ces deux ontologies nous permet, d'une part d'améliorer la pertinence des résultats grâce à trois types d'enrichissements et d'autre part, à retourner l'ensemble des documents vérifiant la requête à partir d'une représentation formelle et cohérente (AFC).

Les prototypes qui ont été développés pour ce travail supportent bien notre démarche. Néanmoins, l'utilisation du prototype « UMLS2OWL » défini pour structurer les termes du domaine en ontologie, n'en reste pas moins long et interactif

comme processus. Ceci en raison du choix d'avoir utilisé le service web d'UMLS afin de disposer de la version la plus à jour du thesaurus d'une part et d'autre part en raison des nombreuses interactions qu'il implique avec l'utilisateur afin de préciser le contexte sémantique de certains termes.

Dans ce travail il est important de préciser que la méthodologie présentée est valable pour un domaine spécifique et il serait présomptueux et pratiquement impossible à l'heure actuelle de généraliser celle-ci à tous domaines confondus. Dans ce cas, la masse de connaissances à traiter serait tellement gigantesque que la technologie d'aujourd'hui serait incapable de l'absorber et de la traiter dans un délai raisonnable. Néanmoins, une perspective intéressante serait d'appliquer notre méthodologie à un ensemble de sous-domaines spécifiques appartenant tous à un même domaine. Par exemple, dans le domaine médical, il serait envisageable de définir un moteur de recherche sémantique regroupant un ensemble d'ontologies définies pour des pathologies spécifiques et ainsi permettre d'effectuer des recherches de documents selon un axe préalablement choisi.

Lors du choix des ressources textuelles utilisées pour notre méthodologie, nous privilégeons dans un premier temps, l'utilisation de certains types de ressources (résumés, guidelines) afin de représenter le plus fidèlement possible la connaissance du domaine visé. Dans un deuxième temps, d'autres types de ressources plus variées peuvent être considérées afin d'enrichir de manière incrémentale les ontologies de domaine pour peu que la sémantique contenue par celles-ci soit en relation avec le domaine concerné. Ainsi, dans le contexte médical, notre moteur sémantique permettrait à terme de proposer une recherche d'informations pour divers types de documents : guidelines, rapports, dossier médicaux, radios, images, etc.

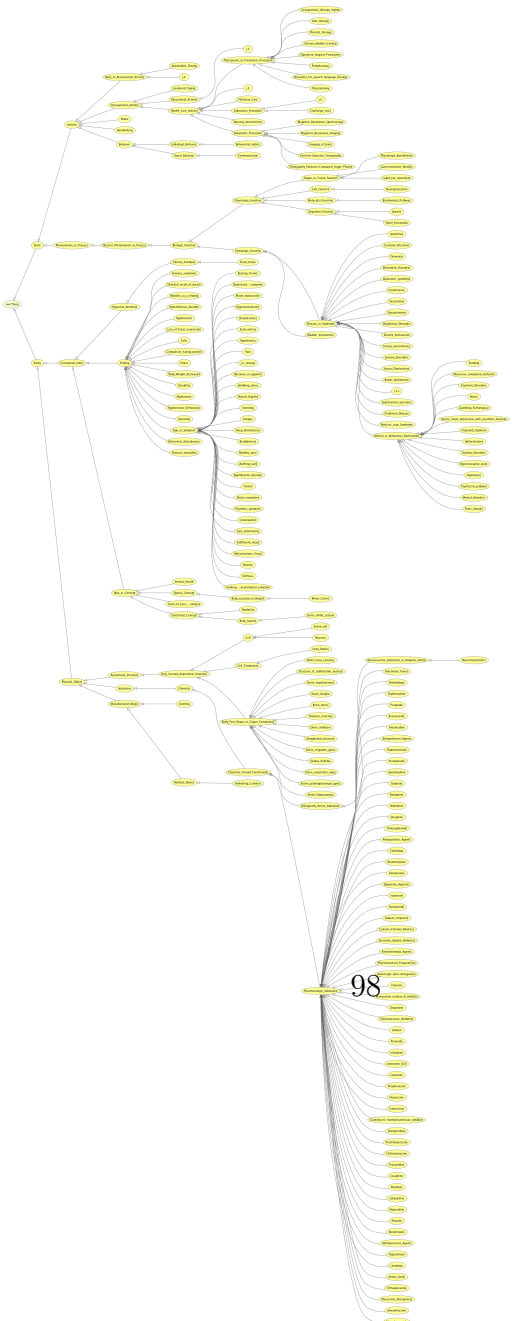
Dans notre approche, nous favorisons une extraction terminologique dite « manuelle » pour la création des ontologies du domaine. Par la suite, une autre perspective serait d'envisager une extraction terminologique dite « automatique » où les nouveaux termes seraient contrôlés et ajoutés aux ontologies de façon incrémentale.

Pour terminer, il serait également utile de réutiliser l'idée de Capineto [Carpineto and Romano, 2000] afin de classer les documents retournés selon leur degré de pertinence en fonction des termes contenus dans la requête.

Chapitre 6

Annexes

6.1 Ontologie structurant les termes pour le domaine de la maladie de Parkinson



6.2 Algorithme d'enrichissement de requête

6.2.1 Enrichissement par spécialisation

```
public Vector specifRequest(String term){

    Vector <String> res = new Vector<String>();


    String queryString =
        "PREFIX   rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> \n"+
        "PREFIX   owl:<http://www.w3.org/2002/07/owl#> \n"+
        "PREFIX   rdfs:<http://www.w3.org/2000/01/rdf-schema#> \n"+
        "PREFIX   :<"+modelinf.getNsPrefixURI("")+"> \n"+
        "SELECT DISTINCT ?instance \n" +
        "WHERE { \n" +
        "    ?instance rdf:type ?sibling. \n"+
        "    ?sibling rdfs:subClassOf ?object \n" +
        "FILTER regex(STR(?object), \""+term+"\", \"i\")\n" +
        "      }\n";


    Query query = QueryFactory.create(queryString);
    // Execute the query and obtain results
    QueryExecution qe = QueryExecutionFactory.create(query, modelinf);
    ResultSet results = qe.execSelect();

   while(results.hasNext()){
        String tmp = results.next().getResource("instance").getLocalName();
        res.add(tmp);
   }

    // Important - free up resources used running the query
    qe.close();

    return res;

}
```

FIGURE 6.2 – **Code java utilisé en vue d'un enrichissement par spécialisation.**

6.2.2 Enrichissement par généralisation

```
public Vector generalRequest(String term){

    String father = this.getfather(term);
    String grandfather = this.getfather(father);

    Vector specifFather = this.specifRequest(father);
    Vector specifgrandFather = this.specifRequest(grandfather);

    for (int i=0; i<specifFather.size();i++){

            if (specifgrandFather.contains(specifFather.get(i)));
             {
                specifgrandFather.remove(specifFather.get(i));
             }
        }

    return specifgrandFather;
}
```

FIGURE 6.3 – **Code java utilisé en vue d'un enrichissement par généralisation.**

6.2.3 Enrichissement par relation

```
public Vector relationsRequest(String term, String relation){

     Vector <String> res = new Vector<String>();

     String queryString =
         "PREFIX   rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> \n"+
         "PREFIX   owl:<http://www.w3.org/2002/07/owl#> \n"+
         "PREFIX   rdfs:<http://www.w3.org/2000/01/rdf-schema#> \n"+
         "PREFIX   :<"+modelinf.getNsPrefixURI("")+"> \n"+
         "SELECT DISTINCT ?concept \n" +
         "WHERE { \n" +
         "    ?object rdfs:subClassOf ?x. \n" +
         "FILTER regex(STR(?object), \""+term+"\", \"i\")\n" +
         "    ?x owl:onProperty :"+relation+". \n" +
         "    ?x owl:someValuesFrom ?concept. \n" +
         "      }\n";

         System.out.println(queryString);

     Query query = QueryFactory.create(queryString);

     // Execute the query and obtain results
     QueryExecution qe = QueryExecutionFactory.create(query, modelinf);
     ResultSet results = qe.execSelect();

    while(results.hasNext()){
         String tmp = results.next().getResource("concept").getLocalName();
         res.add(tmp);
    }
     // Important - free up resources used running the query
     qe.close();
     return res;


 }
```

FIGURE 6.4 – **Code java utilisé en vue d'un enrichissement par relation.**

Glossaire

AFC L'analyse formelle de concepts (AFC) est une méthode mathématique permettant de structurer hiérarchiquement des concepts composés d'un ensemble d'objets partageant les mêmes propriétés.

CETIC Centre d'Excellence en Technologies de l'Information et de la Communication.

Concept Dans ce mémoire nous distinguons l'utilisation du terme « concept » selon les deux définitions suivantes :

1. **Les concepts d'une ontologie** résultent d'une « conceptualisation » qui correspond à un « modèle abstrait » d'une partie du monde réel sur lequel doit travailler le système considéré et qui se présente comme un ensemble de définitions de concepts muni de propriétés et de relations entre ces concepts.
2. **Les concepts formels** sont des concepts présentés dans le contexte de l'AFC et sont définis à la section 1.1.2.

Corpus de textes Ensemble de documents regroupés dans une optique précise.

Extension de requête Enrichissement de requête qui consiste à rajouter un certain nombre de termes à la requête initiale afin de guider l'utilisateur dans sa recherche d'informations mais aussi d'améliorer la pertinence des documents retournés.

FUNDP Facultés Universitaires de Namur.

Jena Jena est un framework java utilisé pour la construction d'applications Web sémantique. Il fournit un environnement de programmation défini pour les

langages du Web sémantique tels que RDF, RDFS, OWL et SPARQL mais aussi un ensemble de règles basées sur un moteur d'inférence.

JSP Le Java Server Page est une technique basée sur Java qui offre aux développeurs la possibilité de générer dynamiquement du code HTML, XML ou tout autre type de page web.

Logiques de descriptions Les logiques de descriptions aussi appelées logiques descriptives sont une famille de formalisme de représentation de connaissances qui sont utilisées pour modéliser la connaissance terminologique d'un domaine d'application de manière formelle et structurée.

Ontologie En informatique, l'ontologie se définit comme « *une spécification explicite et formelle d'une conceptualisation partagée* ». Dans ce mémoire nous élaborons la construction de deux ontologies :

1. **Ontologie structurant les termes du domaine** : cette ontologie représente la connaissance d'un domaine et est définie à partir des termes du domaine structurés entres-eux.
2. **Ontologie de recherche d'informations** : cette ontologie n'est pas une vraie ontologie car elle n'est pas conçue pour représenter la connaissance d'un domaine mais bien afin d'indexer des documents et leurs termes.

OWL Ontology Web Language (OWL) est un langage qui a été conçu afin d'être utilisé par les applications cherchant à traiter le contenu de l'information et non plus uniquement à présenter l'information. OWL facilite l'interprétation du contenu Web pour la machine par rapport aux langages XML,RDF et RDFS en fournissant un vocabulaire supplémentaire ainsi qu'une sémantique formelle.

Protégé Plateforme open-source, Protégé est un logiciel qui fournit une suite d'outils définie pour la construction et la gestion des ontologies.

RDF Resource Description Framework est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs méta-données afin de permettre le traitement automatique de celles-ci.

RDFS Le langage RDFS (RDF-Schema) est une extension du langage RDF. Plus expressif, il ajoute la possibilité de définir des concepts, de prendre en compte les hiérarchies d'héritage de concepts et d'attributs ainsi que la définition du domaine et du co-domaine d'une relation.

Recherche d'informations La Recherche d'Informations (RI) est une science qui comprend un ensemble de méthodes, de procédures et de techniques permettant, en fonction de critères de recherche propres à l'usager, de sélectionner l'information dans un ou plusieurs fonds documentaires plus ou moins structurés.

Sémantique La sématique d'un mot (ou terme) exprime le sens ou la signification de celui-ci.

SPARQL SPARQL est un langage d'interrogation et un protocole d'accès aux données pour le Web sémantique. Celui-ci définit la syntaxe et la sémantique nécessaires à l'expression de requêtes sur une base de données de type RDF, RDFS mais aussi OWL ainsi que la forme possible des résultats.

Syntaxe La syntaxe d'un mot (ou terme) se rapporte au respect ou au non-respect grammatical de celui-ci dans un langage formel.

Thesaurus Dans le domaine informatique, le thesaurus est un vocabulaire contrôlé qui regroupe un ensemble de concepts relatifs à un certain domaine.

UMLS Constitue le plus grand thesaurus biomédical et se compose d'un ensemble de vocabulaire et de concepts contrôlé par des experts du domaine (Unified Medical Langage System).

Table des figures

Liste des tableaux

Bibliographie

[Attar and Fraenkel, 1977] Attar, R. and Fraenkel, A. (1977). Local feedback in full-text retrieval systems. *Journal of the ACM (JACM)*, 24(3) :397–417.

[Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., and Nardi, D. (2003). *The description logic handbook : theory, implementation, and applications*. Cambridge Univ Pr.

[Barbut and Monjardet, 1970] Barbut, M. and Monjardet, B. (1970). Ordre et classification, algebre et combinatoire. *Paris, Hachette*, 1 :176.

[Bendaoud et al., 2008] Bendaoud, R., Toussaint, Y., and Napoli, A. (2008). PACTOLE : A methodology and a system for semi-automatically enriching an ontology from a collection of texts. *Conceptual Structures : Knowledge Visualization and Reasoning*, pages 203–216.

[Berners-Lee, 1998] Berners-Lee, T. (1998). Semantic web road map.

[Berners-Lee and Hendler, 2001] Berners-Lee, T. and Hendler, J. (2001). Scientific publishing on the semantic web. *Nature*, 410 :1023–1024.

[Buckley et al., 1995] Buckley, C., Salton, G., Allan, J., and Singhal, A. (1995). Automatic query expansion using SMART : TREC 3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 69–80. Diane Pub Co.

[Carpineto and Romano, 1996] Carpineto, C. and Romano, G. (1996). A lattice conceptual clustering system and its application to browsing retrieval. *Machine learning*, 24(2) :95–122.

[Carpineto and Romano, 2000] Carpineto, C. and Romano, G. (2000). Order-theoretical ranking. *Journal of the American Society for Information Science*, 51(7) :587–601.

[Charlet et al., 2003] Charlet, J., Bachimont, B., and Troncy, R. (2003). Ontologies pour le Web sémantique. *Web sémantique : Action spécifique CNRS/STIC*.

[Chrisment et al., 2008] Chrisment, C., Hernandez, N., Mothe, J., and Genova, F. (2008). Enrichissement sémantique pour la recherche d'information : méthodologie de transformation d'un thesaurus en une ontologie de domaine.

[Cimiano, 2006] Cimiano, P. (2006). *Ontology learning and population from text : Algorithms, evaluation and applications*. Springer Verlag.

[Crouch, 1988] Crouch, C. (1988). A cluster-based approach to thesaurus construction. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 309–320. ACM.

[Cui et al., 2002] Cui, H., Wen, J., Nie, J., and Ma, W. (2002). Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web*, page 332. ACM.

[Curran and Moens, 2002] Curran, J. and Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition*, pages 59–67.

[Ding and Foo, 2002] Ding, Y. and Foo, S. (2002). Ontology research and development. Part 1-a review of ontology generation. *Journal of information science*, 28(2) :123.

[El Guedj and Nugues, 1997] El Guedj, P. and Nugues, P. (1997). Analyse syntaxique combinant deux formalismes au sein d'un chart hiérarchique. *Humankybernetik*.

[Fayyad et al., 1996] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). From data mining to knowledge discovery in databases. *Communications of the ACM*, 39(11) :24–26.

[Feigenbaum, 1961] Feigenbaum, E. (1961). The simulation of verbal learning behavior. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 121–132. ACM.

[Feldman and Sanger, 2007] Feldman, R. and Sanger, J. (2007). The Text Mining Handbook–Advanced Approaches in Analyzing Unstructured Data. *Computational Linguistics*, 34(1).

[Fellbaum et al., 1998] Fellbaum, C. et al. (1998). *WordNet : An electronic lexical database*. MIT press Cambridge, MA.

[Fisher, 1987] Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2) :139–172.

[Fu et al., 2005] Fu, G., Jones, C., and Abdelmoty, A. (2005). Ontology-based spatial query expansion in information retrieval. *On the Move to Meaningful Internet Systems 2005 : CoopIS, DOA, and ODBASE*, pages 1466–1482.

[Ganter et al., 1985] Ganter, B., Stahl, J., and Wille, R. (1985). Conceptual measurement and many-valued contexts.

[Ganter et al., 1999] Ganter, B., Wille, R., and Wille, R. (1999). *Formal concept analysis*. Springer Berlin.

[Gennari et al., 1989] Gennari, J., Langley, P., and Fisher, D. (1989). Models of incremental concept formation. *Artificial intelligence*, 40(1-3) :11–61.

[Godin et al., 1995] Godin, R., Mineau, G., Missaoui, R., and Mili, H. (1995). Méthodes de classification conceptuelle basées sur les treillis de galois et applications.

[Godin et al., 1986] Godin, R., Saunders, E., and Gecsei, J. (1986). Lattice model of browsable data spaces. *INFO. SCI.*, 40(2) :89–116.

[Gruber et al., 1993] Gruber, T. et al. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5 :199–199.

[Hahn and Schulz, 2004] Hahn, U. and Schulz, S. (2004). Building a very large ontology from medical thesauri. *Handbook on ontologies*, pages 133–150.

[Han et al., 2001] Han, J., Kamber, M., and Tung, A. (2001). Spatial clustering methods in data mining : A survey. *Geographic Data Mining and Knowledge Discovery. Taylor and Francis*, 21.

[Han and Choi, 2008] Han, S. and Choi, J. (2008). Construction of a Condensed Thesaurus for Building Radiology Ontology. *Relation*, 10(1.90) :7338.

[Hersh et al., 2000] Hersh, W., Price, S., and Donohoe, L. (2000). Assessing thesaurus-based query expansion using the UMLS Metathesaurus. In *Proceedings of the AMIA Symposium*, page 344. American Medical Informatics Association.

[Horrocks et al., 2003] Horrocks, I., Patel-Schneider, P., and Van Harmelen, F. (2003). From SHIQ and RDF to OWL : The making of a web ontology language. *Web semantics : science, services and agents on the World Wide Web*, 1(1) :7–26.

[Kayser, 1997] Kayser, D. (1997). La représentation des connaissances, hermes. *Collection Informatique*.

[Lame, 2002] Lame, G. (2002). *Construction d'ontologie apartir de textes. Une ontologie de droit dedieeala recherche d'information sur le Web*. PhD thesis, PhD dissertation, Ecole des mines de Paris, Paris France, December 2002 (http ://www. cri. ensmp. fr/).

[Messai et al., 2005] Messai, N., Devignes, M., Napoli, A., and Sma "ıl-Tabbone, M. (2005). Querying a bioinformatic data sources registry with concept lattices. *Lecture Notes in Computer Science*, 3596 :323.

[Mitra et al., 1998] Mitra, M., Singhal, A., and Buckley, C. (1998). Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–214. ACM.

[Napoli, 2005] Napoli, A. (2005). A smooth introduction to symbolic methods for knowledge discovery. *Handbook of Categorization in Cognitive Science*, pages 913–933.

[Pernelle et al., 2002] Pernelle, N., Rousset, M., Soldano, H., and Ventos, V. (2002). Zoom : a nested galois lattices-based system for conceptual clustering. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2) :157–187.

[Pfoser et al., 2000] Pfoser, D., Jensen, C., and Theodoridis, Y. (2000). Novel approaches in query processing for moving object trajectories. In *Proceedings of the 26th International Conference on Very Large Data Bases*, page 406. Morgan Kaufmann Publishers Inc.

[Piatetsky-Shapiro and Frawley, 1991] Piatetsky-Shapiro, G. and Frawley, W. (1991). *Knowledge discovery in databases*. Aaai Pr.

[Qiu and Frei, 1993] Qiu, Y. and Frei, H. (1993). Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169. ACM.

[REISCHER, 2007] REISCHER, J. (2007). OntoNet–a WordNet-based ontological-lexical development system. In *Proceedings of the GLDV-07 Workshop on Lexical-Semantic and Ontological Resources*, volume 13, page 2007.

[Salton, 1968] Salton, G. (1968). *Automatic information organization and retrieval.* McGraw Hill Text.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval* 1. *Information processing & management*, 24(5) :513–523.

[Schlangen et al., 2004] Schlangen, D., Stede, M., and Bontas, E. (2004). Feeding owl : Extracting and representing the content of pathology reports. In *RDF/RDFS and OWL in Language Technology : 4th ACL Workshop on NLP and XML.*

[Schreiber et al., 1999] Schreiber, G., Akkermans, H., and Anjewierden, A. (1999). *Knowledge engineering and management : the CommonKADS methodology.* the MIT Press.

[Silverstein et al., 1998] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. (1998). Analysis of a very large AltaVista query log. *Compaq Systems Research Center.*

[Sinclair, 1996] Sinclair, J. (1996). Preliminary recommendations on corpus typology. *EAGLES Document TCWG-CTYP/P (available from http ://www. ilc. pi. cnr. it/EAGLES/corpustyp/corpustyp. html).*

[Sparck Jones, 1971] Sparck Jones, K. (1971). *Automatic keyword classification for information retrieval*. London.

[Stumme and Maedche, 2001] Stumme, G. and Maedche, A. (2001). Fca-merge : Bottom-up merging of ontologies. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 17, pages 225–234. Citeseer.

[Stumme et al., 2002] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., and Lakhal, L. (2002). Computing iceberg concept lattices with TITANIC. *Data & Knowledge Engineering*, 42(2) :189–222.

[Toussaint, 2004] Toussaint, Y. (2004). Extraction de connaissances à partir de textes structurés. *Documents numérique*.

[Uschold et al., 1998] Uschold, M., King, M., Moralee, S., and Zorgios, Y. (1998). The enterprise ontology. *The knowledge engineering review*, 13(01) :31–89.

[Valtchev et al., 2003] Valtchev, P., Grosser, D., Roume, C., and Hacene, M. (2003). Galicia : an open platform for lattices. In *Using Conceptual Structures : Contributions to 11th Intl. Conference on Conceptual Structures (ICCS'03)*, pages 241–254. Citeseer.

[Van Assem et al., 2004] Van Assem, M., Menken, M., Schreiber, G., Wielemaker, J., and Wielinga, B. (2004). A method for converting thesauri to RDF/OWL. *The Semantic Web–ISWC*, pages 17–31.

[Voorhees, 1994] Voorhees, E. (1994). Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69. Springer-Verlag New York, Inc.

[Wille, 1982] Wille, R. (1982). Restructuring lattice theory : an approach based on hierarchies of concepts. *Formal Concept Analysis*, pages 314–339.

[Xu and Croft, 1996] Xu, J. and Croft, W. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, page 11. ACM.

[Yu and Salton, 1977] Yu, C. and Salton, G. (1977). Effective information retrieval using term accuracy. *Communications of the ACM*, 20(3) :135–142.