THÈSE

en vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Discipline : INFORMATIQUE

délivrée par

l'Université Toulouse III - Paul Sabatier

présentée par

AXEL REYMONET

soutenue le 23 septembre 2008

Modélisation de connaissances à partir de textes pour une Recherche d'Information Sémantique

---

JURY

Sylvie CALABRETTO	*MCF INSA (HdR), LIRIS*	(rapporteure)
Gilles KASSEL	*Pr. Université de Picardie, LaRIA*	(rapporteur)
Marie-Pierre GLEIZES	*Pr. Université de Toulouse III, IRIT*	(présidente du jury)
Philippe LAUBLET	*MCF Université Paris IV, LaLIC*	(examinateur)
Josiane MOTHE	*Pr. Université de Toulouse II, IRIT*	(examinatrice)
Pierre ZWEIGENBAUM	*DR CNRS, LIMSI*	(examinateur)
Nathalie AUSSENAC-GILLES	*CR CNRS (HdR), IRIT*	(directrice de thèse)
Jérôme THOMAS	*Ingénieur de recherche, ACTIA*	(co-encadrant)

École doctorale : Mathématiques, Informatique, Télécommunications de Toulouse

Laboratoire d'accueil : Institut de Recherche en Informatique de Toulouse

Équipe d'accueil : Ingénierie des Connaissances, de la Cognition et de la Coopération

Axel Reymonet

MODÉLISATION DE CONNAISSANCES À PARTIR DE TEXTES POUR UNE RECHERCHE D'INFORMATION SÉMANTIQUE

Directrice de thèse :
Nathalie Aussenac-Gilles, CR CNRS, HdR

Résumé

Avec l'avènement d'Internet et des réseaux d'entreprise, les documents numériques ont subi de profondes transformations, tant dans la diversification de leur support (texte, image, son, vidéo), que dans la forte augmentation de leur nombre accessible informatiquement. La Recherche d'Information (RI) a alors pris une importance capitale : l'utilisateur en quête de données répondant à ses besoins veut disposer de logiciels capables d'exploiter les contenus textuels et de trouver automatiquement tout document pertinent pour la requête.

Pour comparer selon leur sens requête et documents, la RI sémantique nécessite deux opérations préalables : l'obtention d'un modèle des connaissances manipulées et, grâce à lui, l'indexation sémantique des données textuelles. Dans ce mémoire, nous étudions les modèles de Ressources Termino-Ontologiques (RTO) adaptés à la RI et développons un formalisme qui, contrairement aux approches classiques, décrit explicitement la relation entre termes du lexique et concepts de l'ontologie, tout en respectant le standard OWL-DL.

Nous abordons ensuite la problématique de maintenance d'une RTO pour la RI : quand un domaine évolue dans le temps, sa RTO correspondante doit être modifiée en conséquence. L'originalité de notre approche réside dans la mise en parallèle entre maintenance de RTO et indexation sémantique : l'ontographe définit des règles évaluant automatiquement la correction de la RTO en fonction des résultats d'indexation attendus ; appliquées aux documents à indexer, ces règles aident à repérer ceux qui témoignent de la nécessité de maintenance. L'outil présente alors ces documents avec des conseils de modification.

Notre dernière contribution inclut notre formalisme de RTO et le cycle de maintenance au sein d'un processus global de RI sémantique. Nous nous intéressons notamment à la comparaison sémantique d'un document à une requête en langue naturelle. Nous proposons une mesure de similarité tenant compte de la proximité taxonomique de deux notions, ainsi que de la manière dont chacune est reliée sémantiquement à d'autres éléments.

La pertinence de nos contributions a été principalement mise à l'épreuve par la réalisation et l'utilisation d'un prototype d'outil pour la RI sémantique dans le cadre d'un partenariat avec Actia, une société spécialiste du diagnostic automobile.

Mots-clés

Ingénierie des Connaissances, ontologie, ressource termino-ontologique, formalisme de modèle de connaissances, maintenance termino-ontologique, Recherche d'Information Sémantique, indexation sémantique, similarité / appariement sémantique

Institut de Recherche en Informatique de Toulouse - UMR 5505
Université Paul Sabatier, 118 route de Narbonne, 31062 TOULOUSE cedex 9

Axel Reymonet

KNOWLEDGE ENGINEERING FROM TEXTS FOR SEMANTIC INFORMATION RETRIEVAL

Supervisor :
Nathalie Aussenac-Gilles, CR CNRS, HdR

Abstract

With the spreading of Internet and local networks, numerical documents have been undergoing deep mutations, mainly due to the diversification of supports (text, image, sound, video) and their high number accessible by computers. Information Retrieval (IR) has thus become crucial : any user of a search engine wants it to be able to process textual contents to find automatically all documents relevant for their query.

In order to compare a query with a document, semantic IR needs two prior operations to be carried out : obtaining a model for the handled knowledge and using it to index semantically the textual data. In this thesis, we study Ontological and Terminological Ressources (OTR) adapted for IR and we develop a formalism which, unlike classical approaches, explicitly describes the relationship between terms and concepts, while respecting OWL-DL standard.

Afterwards, we broach the topic of maintaining an OTR for IR : when a domain evolves in time, its corresponding OTR must be modified accordingly. The originality of our approach lies in the parallel computing of OTR maintenance and semantic indexing : the engineer can define rules which evaluate automatically the correctness of the OTR with respect to the expected indexing results ; applied to the documents to be indexed, these rules help to spot the ones which show the necessity of maintaining the OTR. The tool then displays these documents with evolution advice.

Our last contribution consists in integrating our OTR formalism and the maintenance cycle into a global semantic IR process. We especially focus on the semantic matching between a document and a keyword based query. We propose a semantic similarity measure which takes into account both the taxonomical proximity of two notions and the way each one is semantically connected to other entities.

The relevance of our contributions was mainly tested by the implementation and use of a prototype tool for semantic IR as part of a partnership with Actia, a company specialized in automotive diagnosis.

Keywords

Knowledge Engineering, ontology, ontological and terminological resource, Knowledge model formalism, ontological and terminological maintenance, Semantic Information Retrieval, semantic indexing, semantic similarity / matching

Institut de Recherche en Informatique de Toulouse - UMR 5505
Université Paul Sabatier, 118 route de Narbonne, 31062 TOULOUSE cedex 9

Je dédie cette thèse
à la mémoire de mes grands-parents,
Isidé et Italo.

Remerciements

Financée par une bourse CIFRE, cette thèse s'est déroulée dans le cadre du laboratoire commun Autodiag, issu d'une coopération entre les laboratoires LAAS et IRIT d'une part, et la société ACTIA d'autre part. A cet égard, je tiens à remercier les directions de l'IRIT et d'ACTIA de m'avoir alloué les moyens financiers et matériels nécessaires pour mener à bien mes travaux de recherche.

Par ailleurs, j'exprime toute ma gratitude à mes deux encadrants Nathalie Aussenac-Gilles et Jérôme Thomas. En tant que directrice de ma thèse, Nathalie Aussenac-Gilles a toujours cherché - et souvent réussi - à m'insuffler sa passion pour l'Ingénierie des Connaissances et la recherche en général. Elle m'a notamment appris à remettre en question mes préjugés scientifiques et à faire preuve de rigueur dans mes analyses et mes contributions. Elle a également su répondre présente aux moments où j'avais le plus besoin de son avis éclairé. En parallèle, Jérôme Thomas, mon encadrant industriel et industrieux, n'a jamais rechigné à m'accompagner quotidiennement dans mes réflexions, même dans leur état le plus préliminaire. Malgré les diverses sollicitations dont j'ai pu l'assaillir, il a fait montre d'une patience à toute épreuve envers moi. De plus, il a su me prouver que, même dans le domaine de la recherche appliquée, l'honnêteté intellectuelle n'est pas un vain mot. Pour tout cela et bien plus encore, je les remercie tous deux infiniment.

De plus, je voudrais exprimer toute ma reconnaissance à l'ensemble de mon jury de thèse : tout d'abord, un grand merci à Sylvie Calabretto et Gilles Kassel qui ont eu l'amabilité d'accepter d'évaluer mes travaux et dont les remarques m'ont permis de mieux appréhender certains aspects théoriques liés à mes contributions. J'attends avec plaisir la prochaine occasion de revoir Gilles Kassel (peut-être aux prochaines journées de l'IC ?), qui nous permettra de poursuivre nos discussions sur l'importance des principes théoriques pour la construction d'ontologie. Je ne peux que regretter de n'avoir eu le plaisir de rencontrer Sylvie Calabretto en personne, mais je pense que nos pérégrinations scientifiques respectives nous amèneront à faire plus ample connaissance dans un futur proche. Je remercie également Josiane Mothe pour m'avoir fait prendre conscience, par ses commentaires et ses questions, de la richesse du domaine de la RI, ainsi que Philippe Laublet, pour avoir suivi et participé avec tant d'intérêt à mes travaux de recherche. Je me réjouis fort que, dans le cadre du projet Dynamo, nous soyons amenés tous trois à collaborer activement sur un sujet aussi passionnant que la gestion de la dynamicité des ontologies en contexte de RI. Enfin, je salue Marie-Pierre Gleizes et Pierre Zweigenbaum pour avoir accepté aussi cordialement de participer à l'examen de mes travaux, et pour m'avoir donné, à travers la pertinence de leurs

questions, l'occasion d'approfondir certains aspects pratiques de ma thèse.

Je dédie ensuite un remerciement à l'ensemble des personnes avec qui j'ai été amené à travailler au cours de ces quatre années : je pense notamment aux membres de l'équipe IC3 de l'IRIT, avec parmi eux Nathalie Hernandez, Mouna Kamel, Bernard Rothenburger, Jean-Luc Soubie ... Je salue aussi l'ensemble du personnel académique et industriel que j'ai pu côtoyer et avec qui j'ai pu collaborer dans le cadre d'Autodiag : Louise Travé-Massuyes, Audine Subias, Michel Combacau pour le LAAS, Hervé Poulard, Olivier Duffaut, Vincent Pujol, Arnaud Benhamou, Jean-Claude Fonté et Christian Desmoulins pour ACTIA. Je souhaite en outre exprimer le plaisir que j'ai eu, en tant qu'utilisateur, à interagir avec Sylvie Szulman dans le cadre de l'application Terminae. Les réflexions liées à cet outil de construction de RTO se sont avérées capitales pour moi puisqu'elles ont constitué un point de départ à l'élaboration d'un méta-modèle de RTO en OWL. Je remercie également Philippe Muller qui, non content d'avoir été pour moi un excellent tuteur de DEA, a été le premier à m'introduire auprès de Nathalie Aussenac-Gilles. C'est donc en partie grâce à lui que j'ai pu effectuer les travaux présentés dans ce manuscrit.

Arrive à présent le moment d'exprimer toute ma gratitude à l'ensemble de mes amis qui ont su me soutenir (voire me supporter !) pendant quatre longues années : en premier lieu, Elise, sans nul doute mon amie la plus chère, qui a toujours su trouver les mots justes lors de mes moments de doute ; ensuite, Florian et Kevin, mes deux compères à l'humour monty-pythonesque, auprès desquels il fait bon rire ; merci aussi à Siegfried et Hervé, les deux autres membres de la "dream-team recherche" à Actia, avec qui j'ai pu m'échapper sous-marinement durant de mémorables week-ends de plongée ; je salue également Sophie, la meilleure toulousaine belge d'Avignon que je connaisse, et Nicolas, le néo-liverpuldien jamais en reste de discussions passionnées sur la science ou la politique ... J'oublie sans doute de nombreux amis, j'espère qu'ils n'en prendront pas la mouche, leur sympathie m'a permis de franchir des montagnes.

La dernière partie de ces remerciements s'avère la plus difficile à rédiger, tant il est délicat d'exprimer par de simples mots tout ce que ces personnes m'ont apporté. Ces personnes, ce sont d'abord mes parents et ma sœur. C'est essentiellement grâce à leur soutien constant et inconditionnel que j'ai pu venir à bout de cette entreprise ardue que constitue la thèse. Je leur serai éternellement reconnaissant pour tout ce qu'ils ont fait afin que j'en arrive à ce point. En retour, j'espère de tout cœur pouvoir faire en sorte qu'ils n'aient jamais à rougir de l'homme qu'à travers moi, ils ont contribué à bâtir. Enfin, un immense merci à celle qui partage ma vie, Aurore, elle qui a réussi l'exploit de supporter au quotidien mes sautes d'humeur et qui a toujours tendu une oreille attentive et compatissante à mes tracas de thésard.

Table des matières

Introduction

0.1 Contexte scientifique

Avec le développement du WorldWide Web, un fort besoin de formalisation des données s'est fait sentir à travers la notion de méta-données [Berners-Lee, 1999]. Ces méta-données ont été introduites dans le but de représenter formellement les informations contenues dans les documents circulant sur le Web et de les rendre ainsi interprétables par les machines. Celles-ci pourraient ainsi effectuer à un niveau sémantique et de façon automatique (ou semi-automatique) des tâches d'intégration, de génération et de recherche d'information. L'ensemble des technologies visant à remplir cet objectif est regroupé dans la mouvance du Web Sémantique. A terme, une telle initiative doit permettre de disposer d'un Web plus "intelligent", avec un accès aux données plus intuitif pour les utilisateurs [Reynaud *et al.*, 2002].

Actuellement, l'ontologie est une des formes les plus populaires pour représenter formellement des connaissances comme les méta-données, fondement du Web Sémantique. Selon la définition de [Gruber, 1993], une ontologie est une formalisation explicite d'une conceptualisation partagée. Pratiquement, elle correspond à un consensus explicite accepté par une communauté donnée sur un sous-ensemble d'objets du monde regroupés entre eux en concepts selon certaines propriétés communes. Les notions de consensus explicite et de partage sont essentielles "*pour permettre l'exploitation des ressources du Web par différentes applications ou agents logiciels*" [Reynaud *et al.*, 2002]. De plus, l'aspect formel des ontologies, à travers leur représentation via des langages comme RDF(S) ou OWL, permet d'assurer les capacités de raisonnement nécessaires aux outils du Web Sémantique pour assister efficacement les utilisateurs humains.

En parallèle, avec la démocratisation d'Internet et la mise en place de réseaux d'entreprise toujours plus vastes, l'accès par ordinateur à des sources de connaissances textuelles a explosé au cours de la dernière décennie. Ce phénomène s'est avéré un formidable avantage pour de nombreuses communautés dont le domaine d'activité exige le partage, la sauvegarde et le recours à des connaissances spécifiques. Du fait de la quantité sans cesse croissante de documents disponibles, il est bien vite devenu capital de disposer de méthodes efficaces pour accéder de façon ciblée à l'information souhaitée. La Recherche d'Information (RI) s'est ainsi rapidement développée afin de répondre à ce besoin. Par le biais de techniques d'analyse statistique et/ou syntaxique des mots contenus dans les documents, la RI a su prouver son efficacité et atteindre une certaine maturité. Toutefois, celle-ci com-

mence à laisser entrevoir certaines limites en termes de précision et de rappel [1] : tout d'abord, si l'utilisateur emploie des termes différents mais synonymes de ceux d'un document, il n'obtiendra pas ce document en résultat (rappel décru) ; ensuite, si un terme de la requête possède plusieurs sens disjoints, certains textes retournés en résultat risquent de ne pas faire référence au bon sens du terme (précision décrue). Pour pallier ces limites, les modèles d'appariement entre une requête de l'utilisateur et un document ont peu à peu gagné en complexité, du modèle booléen basique [van Rijsbergen, 1979] vers des modèles de type vectoriel [Salton *et al.*, 1975], booléen flou [Baranyi *et al.*, 1998] ou probabiliste [Hofmann, 1999]. Une rapide analyse de ces modèles permet d'isoler une tendance générale : de fait, ceux-ci cherchent de plus en plus à évaluer la proximité d'une requête et d'un document non plus uniquement sur la base du lexique partagé mais aussi sur leur sens. Lorsque les procédés correspondants se fondent sur l'interprétation explicite des requêtes / documents grâce à une ressource extérieure passerelle entre le lexique et les idées, on peut alors parler de Recherche d'Information Sémantique.

Dans les deux cas, on constate que les ontologies jouent un rôle capital puisqu'elles constituent le cadre nécessaire pour stocker et manipuler des données sémantiques reliées entre elles par des relations de différents types (hypéronymie, méronymie ...). La synthèse de [Gomez-Pérez *et al.*, 2004] classe les ontologies en plusieurs familles : les ontologies de haut niveau, les ontologies de domaine, les ontologies de tâche et les ontologies applicatives. Tandis que les premières tendent à être largement réutilisables, les dernières préfèrent privilégier l'utilisabilité [Bachimont, 2004]. Utilisées dans un contexte de Recherche d'Information, les ontologies applicatives servent d'articulation entre, d'une part, un niveau sémantique avec les concepts du domaine modélisé et, d'autre part, un niveau linguistique avec les termes du corpus se référant à ces concepts. Par conséquent, elles se doivent de contenir une forte composante lexicale [Cimiano, 2006]. Dans le cadre de nos recherches, nous nous sommes intéressés à la nature des liens existant entre termes et concepts et que nous proposons un méta-modèle de Ressource Termino-Ontologique (RTO) avec une représentation explicite de ces relations. L'originalité de notre méta-modèle réside dans son pouvoir d'expressivité, puisque grâce à lui et contrairement aux approches classiques, il est possible de modéliser de façon simple certains phénomènes linguistiques comme l'anaphore ou la polysémie. De plus, le méta-modèle a été créé sur la base du langage OWL-DL, ce qui permet de garantir la calculabilité d'une ontologie exprimée dans ce formalisme (pour peu qu'elle respecte OWL-DL elle-même).

Par définition, une ontologie applicative correspond à une vision conceptuelle d'une tâche et d'un domaine partagée par une communauté d'utilisateurs (humains ou logiciels). Au cours du temps, de nombreux facteurs peuvent inciter à la révision d'un tel artefact : évolution de la tâche et/ou du domaine modélisé(s), modification du contexte d'utilisation, élargissement de la communauté d'utilisateurs ... Dans le cadre d'une Recherche d'Information Sémantique, la maintenance de l'ontologie est un processus d'autant plus crucial que les concepts et les relations du modèle interviennent directement dans les annotations sémantiques des documents de la base de recherche : ces annotations, que nous définissons comme un ensemble d'instances de concept et/ou de triplets de la forme

[1] La précision évalue la proportion de documents pertinents parmi ceux retrouvés par le système, le rappel la proportion de documents retrouvés parmi ceux pertinents.

$instance_{sujet} - relation - instance_{objet}$, représentent le contenu sémantique de la requête (ou du document) et sont répertoriées dans un index à des fins de RI. Ainsi, la moindre modification de l'ontologie remet en question les précédents résultats de la phase d'indexation [Luong, 2007]. A l'inverse, il peut être intéressant d'exploiter l'écart entre les résultats attendus et ceux obtenus sur le corpus de documents après une phase d'indexation sémantique : ce décalage peut en effet constituer une aide pour faire évoluer la RTO en adéquation avec le corpus. Cette idée présuppose notamment que le corpus soit suffisamment homogène pour que l'on puisse dégager un certain nombre de caractéristiques communes qui serviront de base pour définir des critères de bonne adéquation entre la RTO et le corpus. Nous nous proposons dans ce mémoire de démontrer la faisabilité d'un tel principe et de l'intégrer dans un processus de RI sémantique via un cycle de construction / maintenance de RTO parallèle à une phase semi-automatique d'indexation sémantique. La mise en parallèle de ces deux opérations devrait minimiser leurs coûts puisqu'ils devraient se recouvrer.

Enfin, nous nous intéresserons aux différents moyens d'accéder à une information ciblée dans un corpus de documents à partir d'une ontologie. Classiquement, deux approches sont retenues : orientée vers les Interfaces Homme-Machine, la première s'affranchit du problème de la formulation d'une requête en fournissant à l'utilisateur des outils de parcours de l'ontologie ; la seconde approche est fondée sur l'appariement sémantique global entre une requête et un document, estimé par une mesure qui peut prendre en compte la similarité conceptuelle, mais aussi la manière dont sont reliés entre eux les produits de l'étape d'indexation. Nous nous intéressons dans nos recherches à la seconde approche, et plus particulièrement aux moyens de rendre compte dans une mesure de similarité sémantique de la structure globale en graphes des annotations sémantiques, avec pour noeuds les entités conceptuelles et pour arêtes orientées les relations sémantiques. Nous proposons notamment une mesure de similarité sémantique entre deux instances de concept fondée sur la distinction entre les relations qui possèdent une contrainte de cardinalité non nulle pour la paire d'instances comparée (dites relations nécessaires sur la paire) et les autres relations (dites facultatives sur la paire). Si elle présuppose une forme de requête spécifique, notre approche permet de rendre compte de façon intuitive de l'influence de la proximité des instances voisines sur celle des instances comparées.

0.2 Contexte industriel

L'ensemble des recherches conduites au cours de notre thèse a été réalisé dans le cadre du projet MODE (Multiple Objective Diagnosis Engine). Ce projet a été accompli par les membres du laboratoire Autodiag, réunissant, d'une part, acteurs publics avec les équipes DISCO (DIagnostic, Supervision et COnduite des systèmes) et N2IS (Nano Ingénierie et Intégration des Systèmes) du LAAS (Laboratoire d'Automatique et d'Architecture des Systèmes) et l'équipe IC3 (Ingénierie des Connaissances, de la Cognition et de la Coopération) de l'IRIT (Institut de Recherche en Informatique de Toulouse), et, d'autre part, des acteurs privés avec la section Recherche de la société ACTIA. Les différents sous-projets associés à MODE partagent un même objectif pratique : aider un garagiste dans la réalisation d'une tâche de diagnostic par son interaction avec un outil logiciel mettant à profit un maximum d'informations liées au véhicule. De façon à mieux situer notre contribution au sein d'Au-

todiag, nous présentons brièvement le contexte d'étude, à savoir le diagnostic automobile, ainsi que les enjeux du projet MODE.

Avec les évolutions technologiques des quinze dernières années, l'automobile est devenue à la fois plus sûre (e.g. ABS, ESP [2], airbag ...) et plus pratique (e.g. assistance au stationnement, allumage automatique des feux, climatisation régulée ...). Toutefois, pour atteindre un tel niveau de performances, il aura fallu utiliser des composants de plus en plus complexes : initialement constitué uniquement de composants électromécaniques (e.g. moteur, direction, phare ...), un véhicule a peu à peu intégré des systèmes électroniques sous la forme de calculateurs. Chacun de ces calculateurs est généralement dédié à la gestion d'une fonctionnalité donnée du véhicule à partir de l'intégration de données évolutives et issues de mesures par capteurs. La prise en compte par l'électronique embarquée de cas d'utilisation de plus en plus complexes a amené à se préoccuper du problème de partage des données : par exemple, l'information relative à la vitesse instantanée du véhicule est récupérée par un calculateur donné mais peut servir à d'autres, notamment pour déterminer la vitesse de balayage des essuie-glace. C'est ainsi qu'une structure informatique est venue se rajouter dans l'organisation globale d'un système automobile de façon à permettre l'échange de données entre calculateurs via un bus de données. Dimensions électromécanique, électronique et informatique sont aujourd'hui les trois composantes constitutives d'un système mécatronique : tandis que les composants électroniques et informatiques permettent de piloter ce système, les composants électromécaniques réalisent - en accord avec les ordres donnés par la partie commande - la fonction pour laquelle ils sont destinés. Pour donner un ordre d'idée de l'importance de l'évolution du domaine, une Peugeot 406 commercialisée en 1995 contenait 7 calculateurs électroniques non connectés ; une Mercedes Classe S W220 commercialisée en 2002 en contenait 80, répartis sur 6 réseaux différents mais reliés entre eux.

Suite à cette complexification du domaine automobile, deux phénomènes ont pu être constatés. En premier lieu, le coût moyen des parties électronique et informatique a pris des proportions très importantes : tandis qu'ils représentaient à peu près 26% du coût total en 2002, les composants de ces types comptent désormais pour 48% dans le prix total d'un véhicule. Le second effet touche à l'évolution de la nature des pannes rencontrées et au champ de compétence d'un garagiste : comme la plupart (environ 80%) des pannes rencontrées concernent désormais des problèmes électroniques ou électriques, il devient de plus en plus difficile de s'appuyer uniquement sur une expertise humaine, par nature limitée dans sa connaissance fonctionnelle des systèmes complexes mis en jeu. De plus, bon nombre de garagistes ne sont pour l'instant formés que pour des problèmes mécaniques. Par conséquent, il est nécessaire qu'une aide logicielle puisse être proposée au réparateur de façon à combler ses lacunes en termes de connaissances requises pour mener à bien une tâche de diagnostic sur un véhicule récent.

Le **projet MODE** s'inscrit dans un tel objectif d'aide au diagnostic automobile. Le principe développé consiste à faire collaborer un ensemble de stratégies différentes dans le but de couvrir un maximum de pannes connues. En tant qu'application industrielle potentielle, MODE essaie d'engendrer aussi peu de coûts supplémentaires que possibles pour l'obten-

[2] Ces acronymes correspondent respectivement à Antilock Braking System et Electronic Stability Program.

tion de données initiales. C'est pourquoi il se fonde uniquement sur des sources de connaissances déjà disponibles tout au long du cycle de vie d'un véhicule (cf fig. 1). Les différentes méthodes envisagées sont les suivantes :

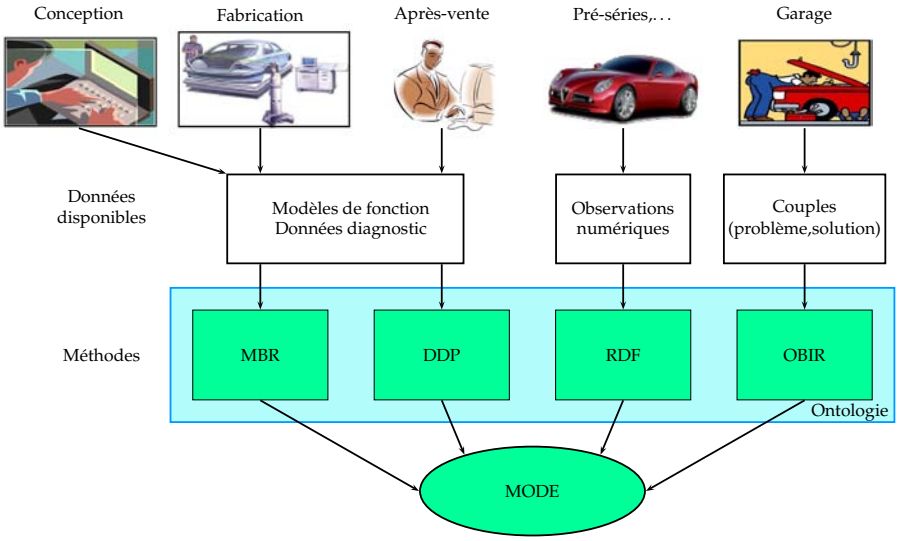

Figure 1 — Le projet MODE, ou l'exploitation de plusieurs sources de connaissances

- **le sous-projet OBIR** (Ontology Based Information Retrieval), qui nous concerne directement, cherche à fournir aux garagistes un moyen simple et intuitif (i.e. par une requête en langue naturelle) de vérifier dans une base de fiches d'incidents [3] si la panne traitée a déjà été précédemment résolue sur un modèle similaire de véhicule, auquel cas il est inutile de lancer d'autres méthodes plus complexes et le diagnostic courant aboutit immédiatement. Actuellement, la proportion de pannes récurrentes est relativement élevée. En effet, en raison de la forte augmentation du nombre de composants dans un véhicule et de la rapide évolution de son architecture générale, il est très délicat d'anticiper pendant la phase de conception des problèmes susceptibles d'apparaître suite à une utilisation normale du véhicule. Dans ce contexte, on voit combien il s'avère crucial de constituer une base de connaissances recensant les pannes rencontrées avec les méthodes de réparation associées et de disposer d'un moyen d'accès simple et efficace à ces données.
- le **sous-projet MBR** (Model Based Reasoning) a pour objectif d'isoler les composants remplaçables mis en cause dans une panne sur une fonction donnée. Pour cela, il utilise les modèles électriques et fonctionnels de bon fonctionnement de la fonction défaillante pour proposer au diagnostiqueur une séquence dynamique de tests optimisée en terme de coût de mise en place et de probabilité d'occurrence de faute.
- le **sous-projet RDF** (Reconnaissance De Formes) vise à améliorer les capacités d'autodiagnostic des calculateurs en tenant compte des corrélations potentielles existant entre différents paramètres (e.g. le régime du moteur et le débit d'air en admission). On distingue donc deux phases chronologiques distinctes : la phase d'apprentissage

[3] Cette base, qu'on appellera aussi base d'expérience, est composée de fichiers rédigés en langue naturelle et découpés en différents champs selon une syntaxe XML. Elle est décrite plus en détail en 5.1.1.1.

consiste à établir un ensemble de corrélations correctes à partir de tests de roulages en situation de fonctionnement nominal, tandis que la phase de détection se charge de détecter, lors d'une utilisation quotidienne, toute déviation des points de mesure. Cette méthode a pour avantage notable de ne requérir aucune connaissance sur le modèle de fonctionnement du système surveillé.

– le **sous-projet DDP** (Diagnostic Distribué Préventif) se focalise sur la détection de fautes intermittentes (et donc aléatoirement constatables) par un contrôle permanent des messages échangés par les calculateurs via le(s) réseau(x) informatique(s) de l'automobile. Cette approche permet notamment de repérer certaines pannes difficilement diagnosticables car non anticipées durant le processus de conception du modèle de véhicule.

L'emploi d'une ressource ontologique pour l'exploitation des fiches d'incidents est d'autant plus intéressante qu'elle pourrait servir ultérieurement de passerelle pour modéliser et partager dans un formalisme commun les informations détenues par chacun des sous-modules. Nous n'explorons toutefois pas cette possibilité dans notre mémoire car la problématique liée à la RI sémantique s'avère déjà très riche en soi et qu'une telle étude aurait nécessité de connaître précisément au préalable les types de connaissances manipulées par chaque sous-projet , ce qui n'était pas le cas (la plupart d'entre eux ont été menés en parallèle).

0.3 Organisation du mémoire

Le mémoire est organisé de la façon suivante : le **chapitre 1** s'intéresse à l'Ingénierie des Connaissances et plus particulièrement aux notions d'ontologie et de terminologie, aux relations qu'elles entretiennent, ainsi qu'à leurs principaux standards de représentation. Le **chapitre 2** définit la tâche de Recherche d'Information, la décrit dans son approche générale (sans recours aux ontologies), caractérise la Recherche d'Information Sémantique pour enfin comparer l'utilité et l'efficacité des deux approches.

Les deux chapitres suivants exposent nos contributions théoriques : le **chapitre 3** commence par situer nos travaux dans le cadre des ontologies de domaine, étudie la nature des paramètres influençant la phase de construction d'une ontologie et propose un méta-modèle de représentation en OWL-DL de RTO. Le **chapitre 4** s'attache à la conception d'une plate-forme de recherche sémantique permettant de résoudre le problème inhérent de maintenance d'une ontologie de domaine et d'effectuer un appariement sémantique entre une requête et un ensemble de documents.

Le manuscrit se termine par le (**chapitre 5**), dévolu à la description de nos apports au projet MODE dans le domaine du diagnostic automobile (construction d'une RTO à partir de compte-rendus de panne, prototype pour l'indexation de symptômes clients), à la mise en place et à l'application d'un protocole de validation de nos résultats.

Première partie

Etat de l'art

1 Entre langue et connaissance : les Ressources Termino-Ontologiques

1.1 Motivations

Dans le contexte de notre étude, nous nous sommes intéressé à l'Ingénierie des Connaissances (IC) pour la plus-value qu'elle peut apporter à un processus traditionnel[1] de Recherche d'Information (RI) sur un domaine particulier. Nous avons été guidé par le fait que la modélisation conjointe d'un niveau sémantique avec les notions du domaine en question d'une part, et d'un niveau lexical avec les termes caractéristiques de ce domaine d'autre part, peut apporter un gain simultané de précision et de rappel pour un système de RI [Meyer *et al.*, 1992]. En effet, un moteur de recherche utilisant ce paradigme en complément de méthodes plus classiques (comme la recherche par mots-clés) pourra faire le rapprochement entre des documents contenant des synonymes ou des termes de sens proches (accroissement potentiel du rappel). De même, il pourra éviter de mettre en relation deux textes contenant le même terme avec des sens différents (accroissement potentiel de la précision) [Vallet *et al.*, 2005]. Jusqu'à maintenant, les techniques de RI cherchant à interpréter les contenus textuels n'ont certes pas réussi à prouver leur supériorité sur des méthodes plus orientées vers l'exploitation directe de données statistiques. Toutefois, il se pourrait qu'elles les remplacent de façon avantageuse dans un contexte spécifique (celui notre cas d'étude), à savoir un **corpus de faible taille**, des **documents très courts** et un **domaine de connaissances limité** : sans un volume de données assez important, les traitements statistiques tendent à devenir peu efficaces, tandis que les coûts liés à l'interprétation sémantique des documents se font moins sentir. D'autres approches (ni statistiques, ni sémantiques) sont envisageables afin d'améliorer le processus de RI dans une situation de ce type. Néanmoins, nous avons délibérément fait le choix de nous intéresser uniquement aux techniques exploitant le sens des documents à travers une structure de données formalisant niveaux sémantique et lexical. C'est pourquoi nous faisons dans la section 1.2 un tour d'horizon de différents modèles conceptuels capables d'une telle expressivité, avec notamment les Ressources Termino-Ontologiques (RTO), également appelées ontologies lexicales.

[1] Par RI traditionnelle ou classique, nous entendons RI basée sur une indexation par mots simples issus des documents (voir la définition d'indexation en 2.1.3).

Dans un deuxième temps, notre parti-pris nous a amené à nous pencher sur le processus de construction de RTO. Si l'idée de se fonder sur un modèle conceptuel à composante lexicale paraît séduisante pour un moteur de recherche spécialisé, il est néanmoins indispensable de se soucier des coûts logistiques engendrés par l'étape préalable de construction de cette ressource. Pour la construction d'un modèle de connaissances, deux approches prévalent : une première fait l'hypothèse que de nombreuses ressources ontologiques sont disponibles sur Internet et elle cherche à sélectionner les plus appropriées afin de les intégrer dans le nouveau modèle [d'Aquin *et al.*, 2007, Uschold *et al.*, 1998]. La seconde approche considère les ressources textuelles d'un domaine comme autant d'indices dans la tâche de conception du modèle, elle cherche donc à les exploiter au mieux. On peut distinguer deux écoles au coeur de cette mouvance : la première souvent basée sur des techniques d'apprentissage [Maedche, 2002], met l'accent sur une chaîne de traitement aussi automatisée que possible, n'ayant recours à l'utilisateur qu'en phase finale de validation. A l'inverse, la deuxième école se base sur la coopération homme-machine et fait appel, outre à l'informatique, à des disciplines comme l'ergonomie ou la sociologie : elle estime que "*le système ne [doit] plus raisonner à la place des individus, mais doit donner à penser, s'intégrer dans les activités [de construction] en servant de médiateur*" [Aussenac-Gilles, 2005]. Nous faisons une description de ces différentes approches en 1.3. Dans cette même section, nous abordons une problématique connexe à la construction d'un modèle de connaissances, à savoir sa maintenance. En effet, toute application de RI utilisant une RTO s'appuie sur des indices textuels afin de vérifier la présence d'instances de certains concepts dans un texte. Il découle que la RTO associée doit pouvoir s'adapter aux évolutions des usages langagiers propres au domaine modélisé. De plus, de nombreux domaines technologiques voient l'apparition régulière de nouvelles notions, ce qui - sans maintenance de la RTO - entraînerait un décalage entre la réalité du domaine et sa modélisation.

Une dernière problématique relative aux RTO concerne leur format de représentation, dont le degré de formalisation dépend fortement de l'application visée. On peut envisager deux approches pour relier un fragment textuel au concept qu'il dénote : soit les éléments de lexique relatifs aux documents sont stockés directement dans le modèle ontologique, soit le système dispose d'outils de traitement du langage capables de retrouver ces éléments. Souhaitant soulager la chaîne de traitement correspondante et éviter d'utiliser des traitements spécifiques à une langue, nous avons délibérément opté pour la première approche. Nous abordons donc en 1.4 différents formalismes capables de représenter des connaissances sur plusieurs niveaux sémiotiques, du côté des terminologies (1.4.1) comme de celui des ontologies (1.4.2).

1.2 La Ressource Termino-Ontologique, carrefour pluridisciplinaire

1.2.1 Modèles conceptuels pour une Ingénierie des Connaissances

1.2.1.1 Evolution historique des applications à base cognitive

Avant d'aborder la notion de modèle conceptuel, commençons par donner une vue synthétique du domaine dans lequel nous nous situons. L'Ingénierie des Connaissances (IC) est une discipline connexe de l'Intelligence Artificielle apparue dans les années 70 à la suite de l'émergence d'applications à base de connaissances, dont le développement a soulevé des problèmes d'acquisition de connaissances.

Selon [Studer *et al.*, 1998], l'IC s'intéresse à des problématiques liées au processus de modélisation de connaissances, qu'ils définissent de la façon suivante :

> Nowadays there exists an overall consensus that the process of building a Knowledge-Based System (KBS) may be seen as a modeling activity. Building a KBS means building a computer model with the aim of realizing problem-solving capabilities comparable to a domain expert. It is not intended to create a cognitive adequate model, i.e. to simulate the cognitive processes of an expert in general, but to create a model which offers similar results in problem-solving for problems in the area of concern.

On constate que dans cette acception, l'IC s'intègre dans une démarche de construction d'un système de résolution de problème (RP) en tant que pivot entre les connaissances et leur formalisation. Toutefois, le domaine de l'IC a adopté plusieurs points de vue sur la tâche d'acquisition de connaissances au cours du temps : initialement centrée sur la modélisation de méthodes de RP, l'attention de l'IC s'est portée sur la modélisation des connaissances de domaine nécessaires, puis sur leur interaction avec l'utilisateur du système.

Historiquement, on constate que les préoccupations de l'IC ont évolué de pair avec les applications à base de connaissances [Aussenac-Gilles, 2005]. En effet, les premiers systèmes développés cherchaient à modéliser des connaissances de haut niveau appartenant à un expert de domaine, dans le but d'imiter de façon artificielle et automatique son raisonnement [Gaines et Shaw, 1980]. Les connaissances y étaient opérationnalisées sous forme de règles dans une base de connaissances. Par la suite, les applications ont pris de la distance avec les raisonnements suivis par les spécialistes : elles ont adopté des méthodes de résolution de problème originales et non plus nécessairement calquées sur celle de l'expert. L'objectif principal était alors de décrire de façon plus formelle des procédures de raisonnement potentiellement réutilisables dans d'autres contextes [Schreiber *et al.*, 1993]. Plus tard encore, les systèmes à base de connaissances ont évolué de façon à améliorer leur coopération avec les agents les utilisant [Hayes-Roth et Jacobstein, 1994]. Pour cela, ils se sont appliqués à prendre en compte les situations d'utilisation : profil de l'utilisateur, ergonomie des IHM (Interfaces Homme Machine), nature des données fournies/requises par d'autres modules ... Tout au long de ces évolutions, le domaine de l'IC s'est intéressé à la notion de modèle conceptuel [Hasling *et al.*, 1984, Vogel, 1988].

Un modèle conceptuel peut être défini comme un artefact formalisant par une abstraction normalisée un ensemble de connaissances, et ce de façon à être à la fois interprétable par un opérateur humain et utilisable dans un système formel [Linster, 1992]. Par la suite, nous nous intéressons à un type de modèle conceptuel particulier, les ontologies.

1.2.1.2 Les ontologies

Définitions L'ontologie est avant tout une notion datant de la philosophie antique grecque (avec Parménide et Aristote) qui a connu un net regain d'intérêt avec l'avènement de l'ère du numérique. A l'origine, elle désigne l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe. Elle est alors autant une discipline, un champ d'étude, que le support du résultat de cette étude. Toutefois, cette définition à vocation universelle ne correspond pas à l'acception actuelle d'ontologie en IC où il n'est plus fait référence qu'au support, à la représentation de notions et de leurs propriétés essentielles. Par un glissement de sens, l'ontologie est devenue un artefact concret représenté dans un système informatique. Gruber [Gruber, 1993] en donne une définition générale :
"*Ontologies are defined as a formal specification of a shared conceptualization.*". Concrètement, Gruber conçoit une ontologie comme un point de vue particulier et partagé par un ensemble de personnes (au moins les experts sollicités pendant la phase de construction et les utilisateurs de cet artefact) sur la/les parties du monde à modéliser ; de façon à ce qu'il soit uniformément interprétable, ce point de vue doit être formalisé dans un langage adéquat (i.e. répondant aux besoins de la communauté en termes d'expressivité / simplicité).

Un consensus relativement bien accepté en IC discerne dans une ontologie deux parties de types épistomologiques différents : d'une part la partie intensive (ou définitoire) qui définit entre autres les propriétés et relations communes aux différents ensembles d'objets modélisés, d'autre part la partie extensive (ou base de faits) qui décrit les objets mêmes. Selon ce paradigme, il est alors possible de voir la partie définitoire d'une ontologie comme un graphe simple orienté, puisqu'expliciter une vision d'une partie du monde consiste à :

- rassembler les objets à modéliser selon certaines caractéristiques pertinentes qu'ils possèdent en commun (leurs attributs), créant ainsi un ensemble de concepts [2], soit les sommets du graphe
- organiser ces concepts entre eux selon les relations qu'ils entretiennent ; celles-ci équivalent aux arcs orientés (dont l'origine est représentée par le domaine d'une relation et l'extrémité par le codomaine) dans le paradigme de graphe. On peut distinguer les relations taxonomiques (i.e. *est_une_sorte_de*) des relations transverses. La taxonomie joue un rôle particulièrement important : c'est le squelette de l'ontologie au cours de sa construction. Elle peut aussi s'appuyer sur la relation méronymique, selon le domaine à modéliser [Charlet, 2002].

A cette structure définissant le vocabulaire de base, vient s'ajouter un ensemble d'axiomes permettant d'énoncer les vérités du monde modélisé.

Au niveau de la base de faits, on trouvera la notion d'instance de concept qui correspond à une entité appartenant à la partie extensionnelle d'un concept. On notera que l'on peut

[2] Un concept peut être défini de façon extensionnelle (i.e. à partir d'un ensemble d'objets) et/ou de façon intensionnelle (i.e. avec un ensemble de conditions nécessaires et suffisantes).

associer plusieurs ensembles différents d'instances à une même ontologie selon les usages qu'on en fait : c'est pourquoi on a tendance à rassembler les instances en autant de bases de faits et à les séparer ainsi de la partie purement conceptuelle de l'ontologie.

Pour résumer, construire une ontologie revient à décider d'une interprétation pertinente à donner dans un contexte partagé aux objets du monde à modéliser, à envisager les liens existant entre les différentes interprétations et à figer ces choix dans le temps (jusqu'à la phase suivante de maintenance).

Plusieurs points de vue différents coexistent sur les ontologies et viennent compléter la définition générale donnée par Gruber :

- le courant logicien voit l'ontologie comme une théorie logique permettant de définir une vue partielle et orientée de notions du monde
  "*[An ontology is] a set of logical axioms designed to account for the intended meaning of a vocabulary*" [Guarino, 1998]
- le courant terminologique considère l'ontologie comme un moyen de rapprocher des termes selon leur sémantique et celle des relations qui les relient

- le courant pragmatique estime que l'ontologie est un simple artefact d'ingénierie capable de représenter de façon arbitraire sous une forme normalisée des ensembles d'objets différents selon des propriétés qu'ils partagent en commun et des relations mutuelles qu'ils entretiennent.
  "*Une ontologie est une spécification normalisée représentant les classes des objets reconnus comme existant dans un domaine*" [Charlet, 2002]

Typologie des ontologies Une ontologie est un artefact d'IC d'autant plus difficile à définir qu'il existe de grandes différences entre deux ontologies selon la nature des connaissances que chacune modélise et leurs niveaux de formalisation respectifs.

Une première catégorisation possible des ontologies peut se faire selon la richesse de la formalisation. En effet, une analyse préalable de l'objectif visé par l'application est nécessaire pour évaluer le compromis adapté à la situation entre le degré de formalisation souhaité pour l'ontologie et le temps estimé de modélisation. On distingue plusieurs niveaux de formalisation [Lassila et McGuinness, 2001][3], dans l'ordre croissant :

- les **taxinomies** correspondent à un arbre de concepts reliés entre eux par des relations *est_un*. Les concepts peuvent ou non être instanciés.
- les **"frames"** sont des ontologies qui s'inspirent du formalisme orienté objet avec les concepts représentés par des classes possédant des propriétés caractéristiques
- les ontologies capables d'exprimer les restrictions de valeur (e.g. la relation *manger* définie entre *Animal* et *Nourriture* est restreinte sur le domaine *Carnivore* au codomaine *Viande*)
- les ontologies capables d'exprimer des relations logiques autres (e.g. le concept *Carnivore* est équivalent au concept *Animal* avec une restriction de *manger* à *Viande*), ainsi que des axiomes (e.g. tout couple aux yeux bleus aura des enfants aux yeux bleus)

[3]Nous ne présentons qu'une partie des catégories décrites car nous estimons que les autres sont plus en rapport avec les modèles terminologiques.

Du point de vue des connaissances modélisées, on peut distinguer au moins 4 catégories ontologiques différentes [Gomez-Pérez *et al.*, 2004] :

- les **ontologies de haut niveau** contiennent des concepts de portée très générale et ont pour vocation de constituer le haut de la hiérarchie taxonomique de toute autre ontologie. Du fait de leur prétention à l'universalité et comme toutes les ontologies de ce type possèdent leurs propres critères de différentiation, le groupe de travail SUO (Standard Upper Ontology) de IEEE cherche à les unifier sous une seule méta-ontologie. On peut citer pour exemple les ontologies SUMO [Niles et Pease, 2001] ou DOLCE [Gangemi *et al.*, 2002].
- les **ontologies de domaine** modélisent des connaissances (concepts, relations, activités, principes ...) spécifiques à un domaine. L'ontologie Eng-Math [Gruber et Olsen, 1994] est ainsi une ontologie de l'ingénierie mathématique.
- les **ontologies de tâche** décrivent les notions nécessaires dans la réalisation d'une activité particulière (diagnostic, vente, publication ...). On peut notamment citer l'ontologie fonctionnelle pour la conception industrielle de [Kitamura et Mizoguchi, 2004].
- les **ontologies d'application** définissent toutes les notions nécessaires pour la réalisation d'une application spécifique.

Nous n'adhérons que partiellement à un tel découpage. En effet, nous contestons le présupposé lié à la catégorie des ontologies de domaine selon lequel il serait possible de construire un modèle du domaine sans prendre en compte l'application à laquelle est destinée l'ontologie. Nous aborderons plus en détail cette discussion en 3.1. Pour les ontologies de tâche, précisons qu'on peut distinguer une sous-catégorie à vocation générique (i.e. destinée à fournir un cadre pour décrire des tâches abstraites et/ou indépendantes du domaine ainsi que leurs méthodes de mise en oeuvre) d'une sous-catégorie d'ontologies de tâche liées à des domaines particuliers. Dans le cas d'ontologies d'application, on notera que certaines d'entre elles exigent de représenter des connaissances d'un domaine et/ou d'une tâche. On voit que la catégorisation proposée n'est pas une partition du monde ontologique au sens strict puisque le recouvrement entre certains ensembles n'est pas vide. En fait, on pourrait voir chacune de ces catégories comme une orientation possible vers laquelle peut tendre un modèle conceptuel.

Une autre remarque intéressante sur l'ensemble de ces catégories concerne leurs degrés d'utilisabilité et de réutilisabilité [Klinker *et al.*, 1991] : une ontologie de haut niveau requiert un certain nombre d'adaptations afin de pouvoir modéliser correctement un domaine (ou une tâche) particulier ; par contre, elle sera plus facilement réutilisable dans un domaine (ou une tâche) différent. A l'inverse, une ontologie d'application sera directement utilisable mais difficilement réutilisable dans d'autres contextes. Pour la suite, nous fixons notre contexte d'étude au cas particulier des ontologies d'application car dans le cadre du diagnostic automobile, nous jugeons plus pertinent de privilégier l'utilisabilité de la ressource à sa réutilisabilité.

Les rôles Un système à base de connaissance (SBC) a généralement pour objectif d'aider un utilisateur à réaliser une tâche dans un certain domaine d'application. La notion de rôle a été introduite en IC dans le but de distinguer dans un SBC le modèle du domaine du modèle de contrôle [Marcus et McDermott, 1989]. Le modèle de domaine contient les connaissances

spécifiques au domaine d'étude alors que le modèle de contrôle décompose la tâche principale en sous-tâches (modèle de tâche) et décrit comment réaliser celles-ci à travers les méthodes de résolution de problème (MRP) qui exploitent des connaissances indépendantes du domaine modélisé. Séparer ces deux types de connaissances a pour avantage de favoriser la réutilisabilité des MRP sur différents domaines. Dans ce contexte, on peut définir un rôle comme la fonction générique que joue un élément du domaine au cours du processus de raisonnement [Schreiber et de Hoog, 1999].

Comme le souligne [Reynaud *et al.*, 1997], la formalisation de la notion de rôle a progressivement gagné en richesse et en précision : initialement représenté dans le modèle conceptuel de façon implicite [Reynaud et Tort, 1997] ou comme une simple étiquette [Linster, 1992], le rôle est modélisé par la suite avec des critères syntaxiques et sémantiques dans des approches telles que [Roux et Laublet, 1995] ou [Aussenac-Gilles et Matta, 1994]. Si un consensus semblait exister sur la nécessité d'un degré de formalisation suffisant pour les rôles, il en était tout autrement sur la manière de les représenter. Ainsi, deux points de vue principaux s'affrontaient : [Guarino, 1997] souhaitait donner aux rôles un statut identique à celui des concepts [4], ramenant ainsi la construction d'un modèle conceptuel à la modélisation d'une ontologie de domaine et d'une ontologie de méthode. [van Heijst *et al.*, 1997] s'opposaient à cette vision car ils jugeaient qu'un rôle ne possède pas le même type épistémologique qu'un concept : ils reprochaient notamment à Guarino de représenter certains concepts (rigides) de domaine comme des spécialisations de rôles (par nature non rigides [5]) (e.g. le rôle `Symptôme` pris comme hypéronyme de `Fièvre`). [Kassel, 1999] réconciliait les deux points de vue en proposant de modéliser les rôles de MRP par une ontologie de rôle et de relier rôles et concepts du domaine par une relation *has-for-role* non taxonomique. Pour cela, il démontre la nécessité d'utiliser un langage fortement intensionnel (i.e. capable de représenter précisément les propriétés propres à un concept et non à ses instances). Une telle position permet entre autres de modéliser un rôle joué par une instance de concept (e.g. le métier de `Jean Dupont` est `Garagiste`) mais aussi un rôle joué par un concept (e.g. le concept de `Panne` peut jouer le rôle d'`Hypothèse` dans une tâche de diagnostic). On notera que cette proposition reste d'actualité puisque les travaux de [Bruaux, 2007] ont permis d'intégrer celle-ci au sein d'une méthode de construction d'ontologie d'application.

Récemment, certains travaux se sont intéressés à la nature contextuelle d'un rôle : par exemple, dans un contexte scolaire en présentiel, le rôle d'enseignant sera joué par un professeur ; dans un contexte d'étude à domicile, un logiciel pourra remplacer le professeur dans sa tâche. [Sunagawa *et al.*, 2005] décrit un environnement de construction d'ontologie qui permet de formaliser de tels contextes. Les auteurs y définissent le rôle par son (ses) contexte(s) son détenteur et la nature de celui-ci : dans l'exemple précédent, le rôle d'enseignant en école est assuré par un professeur qui s'avère être humain. L'article explique ensuite de façon précise comment organiser la hiérarchie des rôles en se basant sur les relations taxonomiques existant entre les détenteurs des rôles et les contextes. Cette approche a été reprise et adaptée pour le formalisme OWL dans [Kozaki *et al.*, 2007].

[4] Cette approche a par ailleurs donné lieu à la création de DOLCE [Gangemi *et al.*, 2002]

[5] Voir la définition de rigidité en 1.3.1

1.2.2 Modèles de RTO

1.2.2.1 Motivations en contexte de RI

Le rôle des ontologies dans un cadre de RI se situe à la charnière entre les connaissances et leur expression dans la langue : elles doivent à la fois être un support à la formulation de requêtes et une ressource pour définir des méta-données, pour annoter ou indexer des documents. Il est alors utile de se donner les moyens d'associer les concepts à leur expression linguistique pour en retrouver trace dans la langue. Plusieurs travaux comme [Skuce, 1993, Cimiano, 2006, Aussenac-Gilles *et al.*, 2006] mentionnent d'ailleurs la nécessité d'associer un lexique indépendant à une ontologie, de manière à étiqueter concepts et relations, le tout formant une ontologie à composante lexicale, soit pour définir les concepts (construction d'ontologie) soit pour caractériser les textes (annotation sémantique).

L'utilisation d'un lexique revient à inventorier les termes désignant concepts et relations : on parle de *composante lexicale d'une ontologie* [Maedche, 2002]. L'ontologie fige alors non seulement les concepts d'un domaine et leur définition, mais aussi les termes associés. Cette vision rejoint la tradition terminologique des années 40, qui posait une vue unificatrice sur le monde de la connaissance, découpé en domaines stables. Chaque domaine fixe un réseau de concepts et de termes qui en sont les représentations linguistiques.

Au sens classique, une terminologie liste les termes d'un domaine, et pour chacun d'eux, propose une fiche qui en décrit les usages, la (ou les) signification(s), ainsi que les relations entretenues avec des termes proches. Traditionnellement, on distingue trois types de ressources terminologiques au degré de formalisation croissant :

- un **vocabulaire contrôlé** est une simple liste de termes du domaine,
- un **glossaire** est une liste de définitions en langue naturelle (et donc non interprétables de façon logicielle) des principaux termes,
- un **thesaurus** est une représentation sémantiquement plus riche qui permet de mettre en relation deux termes. Typiquement, les liens permettent d'exprimer la synonymie ou la spécialisation [6]. Les termes ne sont ordonnés selon aucune hiérarchie explicite.

On notera que *stricto sensu*, aucune de ces ressources ne satisfait complètement la définition d'une terminologie. En effet, aucune n'associe à la fois une définition et des exemples d'usage à chacun des termes modélisés.

Avec la mise sur support informatique et la diversification des usages des terminologies dans les années 90, les terminologues se sont interrogés sur les notions de termes et de concepts, et sur leur articulation.

1.2.2.2 Des termes et des concepts

Dans le cadre des interactions constatées entre la terminologie et l'IC, il apparaît nécessaire de se pencher plus en détail sur la distinction faite entre terme et concept au sein de chaque approche. Au cours de cette réflexion, on s'appuiera sur la définition de concept telle qu'elle a été donnée en 1.2.1.2.

[6] Ce type de relation est différent de la subsomption et beaucoup moins strict : la spécialisation peut regrouper de façon indifférenciée la méronymie, l'hypéronymie ou l'instanciation, selon le thésaurus analysé.

La vision terminologique En premier lieu, il semble essentiel de résoudre l'apparente opposition sémantique entre les notions de syntagme et de terme. Un syntagme peut être défini comme un groupe de mots porteur d'une unité catégorielle (e.g. syntagme nominal, verbal ...) et fonctionnelle (e.g. sujet, complément d'objet ...) pour une grammaire donnée. Un terme correspond pour sa part à un élément du vocabulaire spécifique au domaine étudié. Ces deux définitions ont l'avantage de ne présumer en rien de la nature épistémologique du lien existant entre les deux notions. Sur ce point, deux courants s'affrontent en terminologie.

La première mouvance, historiquement la plus ancienne, a été initiée par Wüster dans les années 40. Elle défend l'idée selon laquelle le monde de la connaissance est découpé en domaines stables, dont chacun est équivalent à un réseau fixe de concepts, les termes étant les simples représentants linguistiques de ces concepts [Wüster, 1976]. Dans cette optique normative, le concept préexiste donc aux termes qui, sans lui, ne seraient que de simples syntagmes. Les concepts d'un domaine sont ainsi organisés selon des relations non linguistiques. De plus, cette vue traditionnelle fait l'hypothèse de la monosémie des termes pour un domaine de spécialité. Dans ce cadre, le but de la terminologie revient alors à rassembler les termes correspondant à un concept défini au préalable.

Avec le temps, les évolutions des pratiques terminologiques ont amené un certain nombre de critiques quant à la position "wüstérienne". De fait, la multiplication des types de ressources terminologiques (notamment avec l'apparition de la notion de Web Sémantique) et l'essor de la polysémie dû à l'entremêlement des domaines de spécialité mettent à mal les principes d'unicité et de fixité du réseau conceptuel [Condamines, 2003]. En réaction à ce phénomène, [Slodzian, 2000] propose alors de se fonder sur une étude des usages en corpus pour déterminer quels syntagmes appartiennent au vocabulaire de domaine (normaison), et décider ensuite d'une interprétation pour chacun des termes (normalisation). Par l'étude du fonctionnement réel des unités en discours selon une approche textuelle, l'auteur définit le "candidat-terme", artefact hybride qui n'obtiendra le statut définitif de terme qu'à l'expresse condition d'être accepté par un expert du domaine. Une telle conception de la terminologie avance que les connaissances pertinentes pour un domaine se trouvent dans les textes rédigés par une communauté scientifique ou technique, autant ou plus que chez l'auteur du terme ou dans les définitions qu'il a produites [Rastier, 1995]. L'analyse distributionnelle de [Harris, 1968] (qui permet de comparer les contextes d'apparition des termes d'un corpus par des techniques statistiques) reste emblématique d'une telle position.

La vision de l'IC Par définition, l'IC se préoccupe en priorité de la manipulation de représentations conceptuelles. Dans ce contexte, il est rarement nécessaire de définir ce qu'est un terme. Toutefois, dans le cadre du Web Sémantique (WS), cette notion joue un rôle capital. En effet, une grande partie des applications destinées au WS utilisent les ontologies comme un moyen de rendre compte de connaissances présentes dans des textes. Il devient alors essentiel qu'une ontologie dispose d'une composante lexicale qui lui permette de relier niveau textuel et niveau sémantique [Maedche, 2002].

Contrairement à la terminologie, l'IC ne s'intéresse qu'indirectement à la notion de terme, à travers ses propriétés les plus utiles pour son repérage dans un texte : sa langue, son label, ses variantes. Il en résulte que la plupart des formalismes ontolo-

giques prévoient une représentation minimale des termes (voir 3.2). Par ailleurs, la représentation du terme est souvent incluse dans celle du concept qu'il désigne (e.g. Terminae [Szulman et Biébow, 2004]), ce qui oriente fortement le modèle vers une hypothèse de monosémie.

On retrouve en IC une opposition de points de vue semblable à celle constatée en terminologie à travers les divergences existant entre ontologies formelles et ontologies d'application. [Aussenac-Gilles et Sörgel, 2005] font un rapprochement entre la démarche terminologique prônée par Wüster et les ontologies de haut niveau : les deux approches conçoivent les concepts comme non préalablement ancrés dans une réalité linguistique. A l'inverse, l'article considère qu'une partie seulement[7] des concepts et des relations d'une ontologie de tâche et/ou de domaine émerge de l'interprétation des termes d'un corpus approprié.

Dans ce contexte, nous avons choisi d'orienter nos recherches vers la représentation explicite de termes dans les ontologies d'application de façon à proposer un formalisme qui permettrait de manipuler aussi facilement les termes que les concepts et d'associer à ces termes suffisamment d'informations.

1.2.2.3 Un modèle terminologique à vocation ontologique : les BCT

Les Bases de Connaissances Terminologiques (BCT) ont été produites d'une réflexion sur les rapports entretenus entre un terme et un concept [Meyer *et al.*, 1992]. Elles constituent un enrichissement significatif car elles comportent une trace des informations conceptuelles relevées par le terminologue en identifiant les termes. Leur modèle original différencie un niveau linguistique d'un niveau conceptuel : on accède par les termes du domaine à une modélisation conceptuelle qui donne sens à ces termes [Aussenac-Gilles et Condamines, 2001]. Leur structure est proche de celle d'une ontologie [Szulman *et al.*, 2002] mais leur contenu n'a pas d'ambition ontologique, ce réseau ne prétend ni normaliser ni standardiser ni fixer définitivement les définitions des concepts du domaine concerné. Au contraire, un tel modèle rend compte des concepts tels qu'ils se dégagent de l'étude de la langue, en restituant une sémantique qui reste au plus près de l'usage linguistique.

Pratique terminologique, mise au point de BCT et modélisation d'ontologie en lien avec la langue remettent en question certaines hypothèses structuralistes des années 40 comme la non ambiguïté des termes dans chaque domaine, ou leur stabilité d'usage et de sens. Elles soulignent aussi que pour le repérage de concepts dans la langue, la diversité des expressions en langue des concepts : un concept peut se traduire par une paraphrase, être suggéré par des concepts proches, par les relations qu'il entretient avec d'autres concepts, etc. Dans ce cadre, ce n'est pas systématiquement un terme qu'il faudrait associer au concept, mais un ensemble d'outils comme des patrons d'extraction, des contextes grammaticaux ou sémantiques. La notion de BCT permet d'envisager cette évolution.

[7] Les auteurs insistent en effet sur la nécessité d'une intervention humaine pour valider les connaissances extraites et apporter des connaissances expertes au cours de la construction de l'ontologie.

1.3 Construction et Maintenance de RTO

Dans cette section, nous évoquons plusieurs méthodes issues de différents courants de pensée et destinées à la construction et/ou la maintenance d'une ressource ontologique (avec ou sans composante terminologique associée). Bien que pouvant être utilisées séparément, ces méthodes gagnent à être intégrées dans un processus commun car il est possible de tirer parti des spécificités de chacune d'elles : en 1.3.1, nous décrivons des principes génériques permettant d'élaborer une ontologie bien formée ; nous abordons ensuite la problématique de réutilisation de ressources existantes (1.3.2) pour nous intéresser enfin à différentes approches basées sur l'utilisation de textes du domaine pour la modélisation (1.3.3).

1.3.1 Principes théoriques de construction

Avec des techniques de plus en plus tournées vers l'automatisation de la tâche de construction d'ontologie, les ressources obtenues pêchent souvent par un manque de justification théorique des choix de modélisation. En réaction à cette tendance, plusieurs recherches ont pour but de mettre en pratique des "méta-critères" théoriques auparavant utilisés dans le cadre d'approches manuelles. Ces critères poussent le concepteur à expliciter les décisions qui jalonnent la tâche de construction du modèle.

Dans [Guarino et Welty, 2004], les auteurs reprennent plusieurs définitions à vocation philosophique de méta-propriétés qui permettent de révéler certaines conséquences logiques des choix de modélisation faits par le système et/ou l'ingénieur de la connaissance. L'ensemble de ces choix est réuni sous l'appellation d'*engagement ontologique*. Ces critères font partie intégrante de la méthodologie OntoClean, qui suppose que toute construction ontologique repose sur une réutilisation de l'ontologie de haut niveau Dolce [Gangemi *et al.*, 2002]. On peut citer notamment les critères suivants :

- la **rigidité**, pour une entité donnée, évalue si, par essence, chacune de ses instances le reste en toute situation (entité rigide, eg la notion d'*Humain*), si aucune ne le reste (entité anti-rigide, eg la notion d'*Etudiant* puisque tout étudiant peut perdre ce statut dans le temps) ou si on peut distinguer deux groupes d'instances se comportant de façon opposée (entité semi-rigide, eg la notion d'*Objet Rouge* puisqu'un classeur rouge sera toujours rouge alors qu'une pomme rouge peut être verte lorsqu'elle n'est pas mûre).
- posséder une **condition d'identité** garantit à une entité de pouvoir distinguer chacune de ses instances des autres. Par exemple, une *Voiture* possède deux conditions d'identité : son numéro de plaque minéralogique et son code VIN[8]. A l'inverse, un *Rétroviseur* ne possède aucune condition d'identité puisqu'il est impossible de distinguer deux rétroviseurs pour le même modèle de véhicule.
- l'existence d'un **critère d'unité** commun pour un concept assure que chacune de ses instances est soit un tout indivisible, soit décomposable selon le critère en un nombre fini de sous-parties indivisibles. Par exemple, le concept de *Moteur* est unitaire (le cri-

[8]Les codes VIN (Vehicule Identification Number) sont des codes alphanumériques uniques qui sont données à tous les véhicules automobiles.

tère commun est d'ordre fonctionnel : tout moteur est décomposable en pièces participant à l'objectif final de fournir un couple). Le concept de *Concessionnaire* n'est pas unitaire (parmi ses instances, on peut distinguer celles se rapportant à une personne physique et celles se rapportant à une personne morale ; il y a donc deux types différents de critères d'unicité). Enfin, le concept de *Carburant* est anti-unitaire.

[Guarino et Welty, 2004] utilisent ces notions dans la méthode OntoClean afin de formuler des contraintes de validité d'une ontologie : ainsi, tout concept anti-rigide ne peut subsumer un concept rigide, et tout concept subsumé par un père possédant des conditions d'identité et/ou d'unité doit hériter de celles-ci. Ces contraintes sont notamment réutilisées dans la méthode de spécification OntoSpec, destinée à apporter une aide à la construction d'ontologies semi-informelles [Kassel, 2005, Bruaux, 2007].

En plus de la notion d'engagement ontologique (définie ci-dessus) à laquelle il souscrit, [Bachimont, 2000] aborde un problème amont dans le processus de conception d'ontologies : avant même de restreindre l'extension d'un concept à un ensemble d'objets de l'univers d'interprétation, il est nécessaire de construire les primitives du domaine sur lesquelles la modélisation va s'appuyer. Cette phase préalable est fondée sur l'expression linguistique des connaissances du domaine : le processus de normalisation sémantique consiste à contraindre l'interprétation des libellés de concepts par leur analyse en contexte (ie en fonction des voisins dans l'arbre taxinomique). Au cours de la structuration des concepts, on applique alors des principes différentiels explicitant en fonction de ses voisins les identités et différences qui définissent chaque noeud. Ainsi, Bachimont définit l'engagement sémantique comme "*ensemble des prescriptions interprétatives qu'il faut respecter pour que le libellé fonctionne comme une primitive*". On retiendra principalement le principe de communauté avec le père (tout concept partage au moins un trait commun avec son père), le principe de différence avec le père (tout concept diffère de son père par au moins un trait) et le principe de différence avec les frères (tout concept diffère de ses frères par au moins un trait).

1.3.2 Construction par intégration de ressources existantes

Au même titre que la fusion, l'intégration d'ontologies s'appuie sur la réutilisation de ressources ontologiques existantes dans le but de construire de nouvelles ontologies. Elle s'avère toutefois une problématique moins abordée que la première. Formellement, on peut la définir comme le processus de construction d'une ontologie portant sur un sujet en se fondant sur une ou plusieurs ontologies portant sur des sujets différents (mais éventuellement reliés entre eux). L'intégration d'ontologies est bâtie sur l'intuition que réutiliser des ressources disponibles apporte un gain de temps pour la construction de nouvelles ressources et une meilleure interopérabilité entre les ressources.

[Pinto et Martins, 2001] décrivent une méthodologie permettant de mener à bien ce processus considéré comme exécutable en parallèle de la construction. L'article distingue plusieurs phases :

1. **Analyse des besoins** avec identification des sous-ontologies, des engagements ontologiques à respecter et des concepts essentiels du futur modèle,
2. **Sélection des ontologies intéressantes** selon des critères comme le domaine modélisé, le formalisme adopté, leurs engagements ontologiques ou leur disponibilité,

3. **Tri parmi les ontologies sélectionnées** selon les priorités de modélisation (langage de représentation, mécanismes de raisonnements possibles, maintenance du modèle, adéquation de la terminologie, compatibilité des modèles)

4. **Opérations d'intégration** sur les éléments de la/des ontologie(s) restante(s) (réutilisation directe, modification, spécialisation ou généralisation)

5. **Evaluation de l'ontologie produite**

Les étapes les plus difficiles à mettre en oeuvre sont celles de tri et d'intégration. Plusieurs approches ont été envisagées : [Pan *et al.*, 2006] introduit le paradigme des espaces ontologiques et utilise un algorithme de tableau distribué dans l'objectif de permettre l'import de certaines parties d'ontologie dans l'ontologie en cours d'élaboration. [Fernandez *et al.*, 2006] décrit le système CORE dans lequel des mesures de similarité à base lexicale (par un calcul de la distance de Levenshtein[9] entre deux mots associés à des concepts à comparer) ou taxinomique permettent d'évaluer la proximité d'une ontologie avec le modèle à construire. De même, la démarche adoptée par Terminae dans [Després et Szulman, 2008] utilise une technique de comparaison de chaînes de caractères pour rapprocher un concept de l'ontologie en construction d'un concept d'une ontologie générique existante ; dans leur calcul, les auteurs prennent en compte la proximité lexicale du premier à chacun des termes employés dans la description en langue naturelle du second.

Quelle que soit l'approche choisie, les auteurs affirment unanimement que l'intégration d'ontologie est un processus complexe dont il convient d'estimer prudemment le coût avant de choisir de l'utiliser. Par exemple, [Bontas *et al.*, 2005] rapporte l'expérience d'un projet pour lequel l'intégration d'ontologies existantes a engendré une perte de temps par rapport à un processus de construction partant d'un modèle vierge.

Même si nous sommes persuadé du bien-fondé de vouloir réutiliser des ressources existantes pour construire une ontologie, l'absence de ressource adéquate pour notre domaine d'application ainsi que la volonté de privilégier le critère d'utilisabilité en construisant une ressource adaptée aux besoins applicatifs (voir 1.2.1.2) nous ont poussé à nous orienter vers d'autres approches dans le cadre de nos recherches.

1.3.3 Construction à partir de textes

Nous classerons les méthodes de construction de RTO selon le critère du degré d'intervention de l'utilisateur : tandis que la première approche préfère automatiser le processus autant que faire se peut et considère l'ingénieur de la connaissance comme un opérateur de validation, la seconde privilégie l'étude des interactions possibles entre le système et l'utilisateur qui peut influencer à sa guise l'orientation du modèle en cours de construction. Notre tour d'horizon des travaux en relation avec cette problématique n'a pas pour prétention d'être exhaustif. Nous conseillons le lecteur en quête de plus de détails de se référer à [Cimiano, 2006] et à [Aussenac-Gilles, 2005].

[9] Si A, B sont deux mots, la distance de Levenshtein d est le nombre minimal de remplacements, ajouts et suppressions de lettres pour passer du mot A au mot B.

1.3.3.1 Approche automatique

Les motivations principales de l'Ontology Learning sont à chercher dans l'avènement du WS. En effet, le WS nécessite de disposer de nombreuses ontologies afin de représenter de façon consensuelle les méta-données de pages Web traitant de sujets très différents. Dans cette optique, l'Ontology Learning cherche à optimiser les coûts en temps et en ressources des tâches critiques de construction et de maintenance d'ontologie. Le but recherché est d'automatiser autant que possible ces processus par le biais de techniques issues de l'apprentissage et de l'exploitation de textes en tant que sources de connaissance.

On peut considérer la succession des différentes phases du processus d'Ontology Learning comme un cycle de bootstrap. Les étapes sont au nombre de 4, aucune n'étant obligatoire [Maedche, 2002] :

- **import et réutilisation de ressources existances** (voir 1.3.2)
- **extraction** d'ontologie qui permet de construire une première ébauche d'ontologie
- **réduction** (ontology pruning) pour supprimer des concepts inutiles dans l'ontologie
- **raffinage** pour obtenir des concepts et des relations spécifiques

Comme nous nous sommes détourné de l'idée de réutiliser des ontologies, nous n'aborderons pas la problématique d'*ontology pruning* dont l'objectif principal consiste à réduire une ressource trop générale à sa partie utile pour la modélisation. Nous allons donc nous focaliser sur les étapes d'extraction et de raffinage d'ontologie.

Pour une meilleure compréhension, nous séparons les techniques d'apprentissage utilisées pour l'extraction d'ontologie en 3 groupes selon la nature des objets proposés à l'utilisateur en sortie : concepts, relations taxonomiques, relations transverses.

Extraction de concepts La plupart des techniques destinées à dégager un ensemble de concepts sont fondées sur le repérage en corpus des termes prégnants du domaine et leur rapprochement sémantique. La première phase fait appel à des outils destinés à l'extraction de termes (e.g. Acabit ou Nomino). Parmi eux, certains logiciels comme Syntex [Bourigault *et al.*, 2005] ou Yatea [Aubin et Hamon, 2006] supposent une analyse syntaxique préalable du corpus afin de regrouper les mots en syntagmes. A l'inverse, d'autres extracteurs comme ANA [Enguehard et Pantera, 1994] préfèrent s'affranchir des contraintes linguistiques (eg disposer d'une grammaire de la langue) en utilisant des techniques d'apprentissage sur des listes de termes préalables.

Une fois les termes repérés, des mesures fréquentielles peuvent ensuite être appliquées pour faire émerger les plus importants. Une première mesure consiste à compter le nombre d'occurrences en corpus de chaque candidat-terme (fréquence). Or il se trouve que certains termes peu fréquents peuvent aussi avoir un intérêt dans la définition de concepts. Parmi les méthodes issues de la Recherche d'Information, la mesure du TF-IDF (Term Frequency-Inverse Document Frequency) est fondée sur l'intuition que les termes les plus intéressants sont ceux apparaissant souvent mais dans un nombre restreint de documents du corpus [Salton et Buckley, 1988]. Formellement, on l'exprime de la façon suivante :

$$TF\_IDF(t) = cf(t) * log\frac{|D|}{df(t)}$$

avec $cf(t)$ la fréquence en corpus du terme t et $df(t)$ le nombre de documents dans lesquels apparaît le terme. Pour évaluer l'importance d'un terme dans un domaine particulier, les travaux de [Velardi *et al.*, 2001] ou de [Drouin, 2003] proposent de comparer sa fréquence en corpus avec sa fréquence dans des corpus généralistes [10] ; ils montrent qu'une méthodologie reposant sur ce principe permet de mieux recenser les termes peu fréquents mais intéressants. En termes d'évaluation, il est impossible de juger une approche plus efficace que les autres, tant les résultats peuvent différer selon le corpus d'étude et/ou l'objectif applicatif de la ressource ontologique en cours de construction.

Extraction de relations taxonomiques L'objectif de cette étape consiste à ordonner autant que possible les concepts découverts sous forme d'un arbre taxinomique qui forme le squelette de l'ontologie à modéliser. Là encore, on peut distinguer deux types d'approches. L'approche statistique est fondée sur les valeurs de co-occurrence des termes pris deux à deux, c'est-à-dire l'apparition simultanée de deux termes dans une même fenêtre de texte. L'intuition suivie est que deux concepts fortement co-occurrents ont une probabilité plus grande d'être proches dans la taxonomie. On peut citer comme exemples d'application l'introduction de la matrice de co-occurrence entre tous les termes extraits avec [Maedche, 2002], la mise au point de règles fondées sur la co-occurrence des termes pour un système multi-agent [Ottens, 2007] ou l'emploi de probabilités conditionnelles d'occurrence avec [Sanderson et Croft, 1999].

L'approche linguistique consiste à utiliser des patrons lexico-syntaxiques d'extraction. L'idée, introduite par [Hearst, 1992], est de retrouver des relations de subsomption entre concepts en projetant sur le corpus sous forme d'expressions régulières les constructions syntaxiques attendues autour de certains termes représentatifs de l'hyperonymie : par exemple, la détection du patron "SN_1 ou tout autre SN_2" dans la phrase "*On pourra s'y rendre indifféremment en voiture ou tout autre moyen de transport*" permet de déduire que le concept `Automobile`, dénoté par le terme "voiture", est une forme de moyen de transport. Différents outils ont exploité avec succès cette idée, comme Prométhée [Morin, 1999] ou Caméléon [Séguéla, 2001].

Extraction de relations non taxonomiques Du fait de sa nature, ce processus peut être traité par les mêmes techniques permettant l'extraction de relations taxonomiques. Toutefois, subsiste un problème : il faut également pouvoir déterminer le type de chaque relation trouvée entre deux candidats termes. L'attitude adoptée par la plupart des applications face à ce problème délicat (et souvent résolu avec difficulté par ce genre d'approches) consiste à utiliser la co-occurrence pour rechercher un terme qui se retrouverait régulièrement aux alentours du couple de termes co-occurrents. Des hypothèses sur la nature syntaxique du terme désignant la relation peuvent également améliorer la convergence vers un terme particulier [Kavalec *et al.*, 2004].

[10] Cette démarche est aussi suivie par [Volz *et al.*, 2003] dans le but contraire, à savoir écarter tout concept dont les manifestations linguistiques en corpus n'auraient pas une fréquence relative significative.

1.3.3.2 Approche interactive

Même si les techniques utilisées sont sensiblement les mêmes, plusieurs travaux comme [Aussenac-Gilles *et al.*, 2008] préfèrent se différencier de l'approche précédente. En effet, les outils employés, essentiellement issus du domaine du Traitement Automatique de la Langue Naturelle (TALN), traitent de façon variée (statistique, syntaxique...) des données de type linguistique dans le but de suggérer des rapprochements à vocation sémantique. Toutefois, les partisans d'une approche interactive estiment que le passage au niveau sémantique n'est possible que par l'interprétation des résultats par un (ou plusieurs) opérateur(s) humain(s). Si l'approche automatique permet d'envisager la construction rapide d'ontologies relativement vastes à peu de frais, les structures sémantiques obtenues restent relativement superficielles (notamment au niveau du nombre de relations transverses découvertes). A l'inverse, l'approche interactive travaille en collaboration avec l'utilisateur afin d'obtenir des ontologies de taille moins importante mais bien plus riches par leur sémantique. On voit donc que selon la taille du domaine et le niveau de précision des détails à modéliser, chacune des deux approches sera plus ou moins recommandée.

Par conséquent, cette mouvance préfère mettre l'accent sur l'adéquation entre l'utilisateur et le système et sur la complémentarité de différentes méthodes plutôt que sur leur automatisation. Dans ce but, les connaissances à modéliser seront puisées dans un corpus de textes caractéristiques du domaine ainsi que dans celles d'un (ou plusieurs) expert(s) du(des) domaine(s) [Aussenac-Gilles, 2005]. Contrairement à l'approche automatique pour laquelle l'expert tient un rôle de simple validateur en fin de traitements, l'approche interactive intègre sa présence dès le début du processus de construction ontologique. On peut alors considérer que ce processus se décompose en deux grandes phases successives :

- le **recueil d'indices** est principalement opéré par les méthodes de TALN appliquées au corpus et l'ingénieur de la connaissance y joue un rôle "mineur" (celui de choisir comment constituer le corpus)
- la **synthèse manuelle** des résultats accorde à l'ingénieur une place prépondérante puisqu'il est chargé d'intégrer les résultats de la phase précédente en accord avec les connaissances d'un (ou plusieurs) expert(s)

Naturellement, ces deux phases peuvent s'enchaîner de façon cyclique et il n'est pas nécessaire que tous les résultats des outils de TALN soient directement exploités par l'ingénieur de la connaissance. Celui-ci peut, selon ses besoins, se focaliser sur un sous-ensemble des résultats, commencer la modélisation d'une partie de l'ontologie, revenir aux données d'entrée pour vérifier la pertinence de la modélisation, et exploiter les autres résultats au cours d'une itération ultérieure du cycle.

Dans ce paradigme, on soulignera l'importance capitale que revêt l'articulation entre le niveau lexical et le niveau sémantique. En effet, dans le cas d'une approche supervisée, les choix du modélisateur sur la façon de rendre compte du sens contextuel d'un terme dans l'univers conceptuel influencent fortement la structure de l'ontologie résultante. Pour une démarche entièrement automatisée, ce sont uniquement des seuils sur des critères numériques qui évaluent l'intérêt d'un élément linguistique en vue de la création potentielle d'un (plusieurs) concept(s) dénoté(s) par cet élément.

1.3.4 Maintenance

1.3.4.1 Motivations

Dans un contexte d'utilisation d'ontologie d'un domaine technique, il est nécessaire d'envisager les méthodes existantes de mise à jour de la structure termino-ontologique. En effet, les concepts représentés ainsi que les termes qui les désignent dans les textes sont d'autant plus susceptibles d'évoluer au cours du temps que la technicité du domaine est grande et les besoins applicatifs changeants. De plus, toute modification du modèle doit nécessairement s'accompagner d'une réflexion préalable sur les conséquences d'un tel changement sur les résultats antérieurs de l'application. Comme nous souhaitons contribuer à résoudre ces problèmes, nous exposons ci-après quelques travaux antérieurs en relation.

1.3.4.2 Techniques de maintenance

La plupart des approches se préoccupent du cas d'utilisation particulier correspondant à l'ajout de nouvelles entités (termes et/ou concepts). Ainsi, on retrouvera dans cette catégorie des propositions comme celles de [Maedche, 2002], fondées sur des techniques combinées de traitement automatique du langage et d'apprentissage comparables à celles décrites en 1.3.3.1 : la situation de maintenance revient à les utiliser pour compléter des connaissances préalables. L'approche de [Faatz et Steinmetz, 2002] utilise la lexicalisation d'une relation entre deux concepts pour détecter l'apparition de nouveaux labels à attribuer à l'un des deux concepts en question. Elle a pour inconvénient principal de supposer l'existence et la fixité des patrons d'extraction associés aux relations. La méthode proposée par [Alfonseca et Manandhar, 2002] cherche à enrichir un réseau sémantique par l'analyse de pages Web relatives aux termes désignant ses concepts. Pour cela, les auteurs constituent des corpus à partir des réponses d'un moteur de recherche à des requêtes formées à partir des termes caractéristiques des synsets de chaque entrée WordNet [11]. Pour chaque ensemble de termes synonymes, ils conduisent ensuite une analyse statistique sur le corpus correspondant, ce qui permet d'isoler de nouveaux termes absents du réseau sémantique initial. Grâce au calcul de la matrice de co-occurrence entre nouveaux et anciens termes, ils proposent un éventuel redécoupage de la hiérarchie taxinomique incluant les éventuels nouveaux "concepts".

Une démarche possédant a priori une plus grande souplesse consiste à se fonder sur des systèmes multi-agents (SMA). Suivant cette approche, [Ottens, 2007] démontre que grâce à de tels systèmes, on peut implémenter -pour des coûts du même ordre de grandeur - des algorithmes similaires à ceux d'une approche classique de construction d'ontologie à partir de textes. A l'instar des démarches par apprentissage, les SMA ont l'avantage d'être dynamiques et de s'adapter aux contraintes extérieures et de proposer une chaîne de traitement capable de procéder de façon indifférenciée à la construction ou à la maintenance d'une structure ontologique. [Ottens, 2007] donne le rôle d'agent aux candidats-termes découverts dans le corpus d'étude et les dote de règles leur permettant d'interagir et de s'organiser en

[11] WordNet est une base de données lexicale développée par l'université de Princeton et disponible sur `http://wordnet.princeton.edu/`

taxonomie. L'ingénieur de la connaissance a une importance capitale dans la méthode : il peut à tout moment suspendre le cours des opérations pour imposer une modification sur la structure en formation. Aussi prometteuse qu'elle soit, cette approche possède encore quelques limites : la problématique de modification des relations transverses n'a pas encore été abordée et l'exécution de l'outil sur une ontologie existante engendre pour l'instant un nombre trop important de modifications, à tel point qu'il devient difficile de le qualifier d'outil de maintenance.

1.3.4.3 Causes et conséquences d'un processus de maintenance

Dans un contexte de RI sémantique, nous souhaitons souligner les influences réciproques exercée entre une évolution ontologique et sur le devenir des annotations conceptuelles reposant sur la même ontologie [12]. En effet, si l'implication semble évidente dans un sens (e.g. une opération de maintenance consistant en l'ajout de nouveaux concepts amènera souvent à une annotation plus riche et/ou plus précise), sa réciproque s'avère également vraie : il est possible d'évaluer l'adéquation d'une ontologie à un corpus (et donc le degré d'urgence à la réviser) par la nature et la qualité des annotations sémantiques obtenues dans le temps [Hernandez, 2005]. L'article de [Maynard *et al.*, 2007] se place dans ce paradigme et souligne la nécessité de disposer de méthodes automatiques visant simultanément à gérer la phase de réannotation consécutive à une évolution ontologique et à mesurer la nécessité de maintenance en fonction de l'évolution d'annotations du domaine modélisé. Pour éviter de réannoter inutilement une grande partie du corpus, les auteurs proposent d'utiliser un ensemble de règles génériques (manuelles ou automatiques) permettant de reclasser - si possible - les instances concernées sous les concepts adéquats. L'approche retenue pour la phase inverse repose sur l'utilisation de "folksonomies" (i.e. un ensemble évolutif d'annotations collaboratives librement créées par les rédacteurs et/ou lecteurs d'un texte sur le Web) de domaine : les auteurs commencent par partitionner l'ensemble des tags d'un domaine selon leurs degrés mutuels de co-occurrence et à les aligner aux concepts de l'ontologie susceptible d'être maintenue [13] ; ils comparent ensuite la répartition obtenue à la précédente, sachant qu'une trop grande différence entre les deux témoignera de la nécessité d'envisager une phase de maintenance de l'ontologie. Même si nous partageons l'idée selon laquelle les résultats d'un processus d'annotation sémantique permettent de juger des besoins d'une ontologie à être maintenue, nos recherches manipulent des ontologies de nature fort différente. En effet, les folksonomies constituent par nature une vision ad hoc d'un domaine : cet artefact n'est issu ni d'un consensus, ni d'un besoin applicatif unique et précis. Il nous paraît donc préjudiciable de vouloir en faire l'alignement avec une ontologie telle que nous la concevons, i.e. nécessairement orientée selon un point de vue et un besoin précis (cf 3.1).

De façon plus générale, une problématique qui nous semble actuellement sous-estimée dans la littérature consiste en la gestion/prédiction des conséquences d'une évolution ontologique. En effet, la modification d'une partie d'ontologie peut entraîner des incon-

[12] Même si cette notion sera plus précisément caractérisée par la suite en 2.2.1, on peut sommairement définir une annotation sémantique comme un processus (ou son résultat) consistant à signaler dans un texte la présence d'instances de concepts de l'ontologie. En tant que résultat, une annotation sémantique est donc assimilée ici à une instance de concept.

[13] Pour plus de détails sur cette procédure, se reporter à [Specia et Motta, 2007]

sistances difficilement détectables dans une autre région. Pour résoudre ce problème, [Stojanovic *et al.*, 2002] propose une approche fondée sur la notion de stratégie d'évolution. Après avoir fait la liste exhaustive des changements pouvant entraîner des inconsistances, les auteurs envisagent systématiquement toutes les stratégies possibles pour sauvegarder la consistance de l'ontologie. Par exemple, dans le cas d'un concept *C* supprimé, ses anciens sous-concepts peuvent être reconnectés au père direct de *C*, à la racine de la taxonomie ou ils peuvent même être supprimés. L'ensemble des choix faits par l'utilisateur pour résoudre les inconsistances engendrées par tous les changements problématiques constitue une stratégie d'évolution. L'idée consiste alors à demander à l'utilisateur la stratégie globale qu'il souhaite adopter de façon à ce que le système sache comment réagir par la suite face à une modification génératrice d'inconsistances dans l'ontologie. Une telle approche a pour avantage d'améliorer l'ergonomie des interactions entre l'opérateur humain et le système puisque l'utilisateur n'aura plus à se soucier d'une éventuelle inconsistance dans l'ontologie, cette tâche étant gérée par le système de façon totalement transparente. De même, un outil suivant cette démarche pourra facilement simuler les conséquences d'une modification de l'ontologie pour l'utilisateur, qui pourra alors décider de garder ou de rejeter la modification. On voit combien cette démarche complète celle prônée par [Aussenac-Gilles et Condamines, 2004] (et que nous défendons) quant au besoin de traçabilité des critères ontologiques et de l'utilisation linguistique des concepts en contexte. Par la suite, nous chercherons à prendre en compte de façon précise les répercussions que les évolutions de la RTO peuvent avoir sur les résultats d'un processus d'indexation sémantique (décrit en 2.2.1).

1.4 Les standards de représentation

1.4.1 Standards terminologiques

1.4.1.1 TMF

La norme ISO 16642 définit l'environnement TMF (Terminological Markup Framework), fondé sur les formats MARTIF et Geneter. TMF permet de décrire tous les éléments d'une terminologie avec un langage formel [Romary, 2001]. Celui-ci est constitué d'un méta-modèle et d'un ensemble de contraintes sur les catégories de données utilisées pour représenter les propriétés de chaque terme. Le respect de ce format a l'avantage de garantir la compatibilité mutuelle de deux TML (Terminological Markup Language) de syntaxe différente. Le méta-modèle de TMF représente la structure sous-jacente d'une terminologie sur plusieurs niveaux :

- les informations sémantiques (le concept)
- les réalisations linguistiques (les langues dans lesquelles est exprimé le concept)
- les informations lexicales (les termes associés au concept dans une certaine langue)

Pour décrire un terme, TMF recommande le recours aux catégories de données définies par la norme ISO 12620. Parmi les différentes sortes d'information, on peut trouver le type du terme, les informations grammaticales (catégorie syntaxique, genre, nombre ...), les usages, la formation (provenance, étymologie), la prononciation ou la morphologie.

Pendant la phase de conception de la terminologie, les objectifs applicatifs influencent directement le processus de sélection des propriétés de terme utiles. Il faut néanmoins prendre en compte l'équilibre souhaité entre le niveau d'expressivité de la terminologie et la complexité des traitements ultérieurs [14]. Dans le cadre de notre étude, nous choisissons de nous restreindre à la partie lexicale et textuelle d'un terme. Nous représenterons donc principalement le terme à travers ses usages (textes dans lesquels il apparaît et position exacte de ses occurrences).

Il est bon de noter ici que même dans le cas d'un domaine monosémique [15], la position d'un terme dans un texte est forcément reliée directement à sa représentation et non au concept qu'il désigne. Dans un contexte d'indexation sémantique, on pourrait penser que seule la localisation du concept nous importe, et pas celle des termes associés. Or nous considérons que l'ontologie est une représentation qui évolue avec le temps : l'apparition de nouveaux textes à indexer peut entraîner l'ajout ou la modification de concepts, ce qui aura pour conséquence probable la réorganisation des relations entre termes et concepts. Dans le domaine du diagnostic automobile, on peut par exemple envisager le cas de figure suivant : dans le corpus étudié, la fréquence d'occurrence élevée du terme "allumage du voyant" amène l'utilisateur à définir un concept qui sera dénoté par ce terme. Par la suite, de nouveaux textes ajoutés au corpus de départ peuvent suggérer l'existence de deux concepts plus spécifiques, à savoir l'allumage constant et le clignotement de voyant. Dans ce cas, il sera utile de revoir dans quel contexte le terme "allumage du voyant" est employé, ce qui pourra entraîner un changement de dénotation. On comprend ainsi la nécessité de pouvoir visualiser le contexte d'utilisation d'un terme (et non celui de tous les termes associés au même concept).

1.4.1.2 SKOS

Simple Knowledge Organisation System (SKOS [16]) dérive de RDF Schema (voir en 1.4.2.2) et a été conçu par le World Wide Consortium [17] dans le but de représenter et de partager de façon plus simple qu'avec un langage ontologique des vocabulaires contrôlés comme les classifications, les glossaires ou les thesauri. SKOS n'est pas encore une recommandation officielle mais le groupe de travail W3C pour le déploiement du Web Sémantique doit lui accorder ce statut courant 2008.

Un thésaurus SKOS est un ensemble de triplets RDFS fondés autour de la notion de `skos:Concept`. On peut définir un tel objet à l'aide d'une `skos:definition` rédigée en langue naturelle, on peut lui attacher des libellés préférés (`skos:prefLabel`) ainsi que d'autres libellés alternatifs (`skos:altLabel`). Ces labels sont les représentants concrets des termes et correspondent à de simples chaînes de caractères. Les instances de la classe `Concept` représentent les concepts du thésaurus. On peut les associer à l'aide de plusieurs propriétés comme `skos:broader`, `skos:narrower`, `skos:related`... Les deux

[14] Généralement, plus l'information stockée sera riche, moins les traitements devront être lourds.

[15] Comprendre "dont aucun terme ne peut être interprété de façon ambiguë".

[16] `http://www.w3.org/2004/02/skos/`

[17] W3C, consortium à l'origine de plusieurs recommendations de langages orientés Web à valeur de standards industriels. Voir `http://www.w3.org/`

premières relations organisent les concepts en une hiérarchie sans propriété d'héritage, chaque concept pouvant posséder plusieurs pères via la relation `skos:broader`. De plus, un thésaurus SKOS peut être utilisé pour une annotation sémantique de document grâce à la relation `skos:subject` allant du document vers un concept.

Plusieurs recherches sont en cours avec pour objectif de changer l'orientation de certains thésauri des termes vers les concepts en utilisant le formalisme de SKOS. Un tel phénomène marque un clivage avec la position défendue par l'influent standard ISO-2788 publié en 1986. De même, plusieurs standards récents comme ANSI/NISO Z39-19 admettent que les termes peuvent être considérés comme des libellés lexicaux représentant des concepts, mais leurs formats continuent à garder les termes comme objets de référence [NIS, 2003].

1.4.2 Standards ontologiques

Dans le cadre du Web Sémantique, il est capital d'avoir un formalisme commun pour la représentation d'ontologies, afin de permettre une meilleure interopérabilité dans le partage, la modification et l'intégration de telles structures. A cet égard, RDFS et OWL sont considérés comme les langages les plus adéquats car ils sont issus de recommandations du W3C et bénéficient d'une expressivité adaptée aux besoins de chacun. Tous deux s'appuient sur le langage de balisage XML, élément fondamental du Web Sémantique.

1.4.2.1 Les cartes topiques

Le formalisme des Topic Maps [18] a été créé il y a une dizaine d'années afin d'aider de façon logicielle à l'indexation documentaire. Adoptées comme norme ISO en 2000, les cartes topiques ont été dotées d'une syntaxe XML (à savoir XTM) puis associées à un langage de représentation de requêtes (TMQL). Elles mettent en avant plusieurs artefacts :

- Le **topic** constitue la réification d'un sujet quelconque dont la définition précise est volontairement occultée. Il s'ensuit qu'un topic peut renvoyer indifféremment à une classe ou à un objet du domaine modélisé. Tout topic peut être instance d'un (ou plusieurs) autre(s) topic(s).
- Tout topic possède trois caractéristiques : un ensemble de **noms** indépendant du topic, des **occurrences** qui correspondent aux identifiants vers les ressources (voire aux ressources elles-mêmes) en relation avec le topic, et un ensemble de liens N-aires d'**association** avec d'autres topics.
- La **portée** peut être vue comme le contexte dans lequel il faut interpréter les caractéristiques d'un topic. On peut citer par exemple la langue pour les noms, le niveau de confidentialité pour les occurrences ou la situation d'interprétation pour les associations.

On constate qu'outre la non-différentiation entre classe et objet, ce formalisme ne donne pas de sémantique explicite à chaque topic. C'est à l'interpréteur (humain ou logiciel) de la déduire à partir du contexte dans lequel il se trouve et des caractéristiques valables dans ce contexte. Cette souplesse de représentation est souhaitable et bienvenue dans certains cas, notamment lorsqu'il s'agit de faire cohabiter dans un modèle des points de vue diffé-

[18] `http://www.topicmaps.org/`

rents sur un domaine [Caussanel *et al.*, 2002]. Toutefois, une telle caractéristique nous paraît peu souhaitable pour nos recherches qui supposent un domaine consensuel dans lequel des expertises sont en mesure de faire autorité.

1.4.2.2 RDF et RDFS

RDF (Resource Description Framework) est un langage créé en 1999 et destiné à l'annotation de ressources sur le Web. Un document RDF est un ensemble de triplets de la forme <sujet,prédicat,objet>, chaque élément étant une URI, un littéral ou une variable. RDF possède plusieurs syntaxes (RDF/XML ou N3) et une sémantique formelle (exprimable en théorie des ensembles et en théorie des modèles) comparable à celle des graphes conceptuels simples [Sowa, 2000] : une information sera contenue dans un document RDF si et seulement si la formule logique qui lui est associable est conséquence de celle attachée au document RDF [Baget *et al.*, 2004]. En soi, RDF n'est pas un langage d'ontologie car il ne permet pas le typage des ressources annotées.

RDF Schema est une extension descriptive du vocabulaire de RDF qui permet la spécification de la classe dont une ressource est instance. La sémantique de ce langage (avec notamment les notions de classes et propriétés) est à rapprocher de celle de la programmation objet. Toutefois la modélisation est différente car à l'inverse d'un langage de programmation comme Java, RDFS met l'accent sur la définition de propriétés à partir de classes (domaine et codomaine). La richesse de RDFS est relativement limitée : on ne peut y exprimer la notion d'axiome, point de départ de tout système logique. Le parti-pris de ce langage n'est pas de se suffire à lui-même dans des tâches complexes mais de constituer une base de départ solide pour la définition de nouveaux formalismes [W3C, 2004a].

1.4.2.3 OWL

OWL est une évolution du langage Web DAML+OIL qui s'appuie sur RDFS. Il a été conçu "pour représenter explicitement la signification des termes des vocabulaires [au sens de la logique des prédicats] et les relations entre ces termes" [W3C, 2004b]. OWL dépasse RDFS par ses capacités à représenter une ontologie de façon interprétable par une machine. En effet, OWL introduit la possibilité pour une machine de raisonner sur la base de connaissances, ce qui lui permet d'inférer des connaissances implicites et détecter d'éventuelles incohérences. De plus, le vocabulaire d'OWL s'avère plus riche que celui de RDFS car il rajoute des relations entre classes, des propriétés de cardinalité, d'égalité, la définition de classe par énumération. OWL permet de gérer des niveaux de complexité différents à travers trois sous-langages à l'expressivité croissante :

- OWL Light, sous-ensemble minimal destiné à la construction de taxinomies,
- OWL DL (Description Logics), à la fois beaucoup plus expressif qu'OWL Light et garant de la complétude et de la décidabilité des calculs,
- OWL Full avec la liberté syntaxique de RDFS mais sans la complétude des calculs.

Du fait de son degré d'expressivité modulable et de sa grande popularité en tant que standard privilégié pour les ontologies du Web Sémantique, nous avons choisi de fonder notre approche sur ce langage, nous y reviendrons plus en détail par la suite.

1.5 Bilan

A travers ce premier chapitre, nous avons mis en place un cadre d'étude formel pour la modélisation de connaissances d'un domaine. Tout d'abord, nous avons choisi de nous intéresser à un artefact capable de représenter à la fois les connaissances et le lexique associé : la RTO. Pour la construction de cette ressource, nous retiendrons une approche à partir de textes, d'autant plus pertinente dans le contexte d'une RI sémantique que l'ingénieur de la connaissance dispose dès le départ d'un corpus qu'il souhaite pouvoir indexer par des concepts de la RTO. Nous insistons sur le rôle capital que nous donnerons au modélisateur au cours de la construction et la maintenance de la ressource : plutôt que l'envisager comme un simple validateur en fin de cycle, nous essaierons de l'insérer au coeur même des processus et ainsi améliorer par une meilleure interaction avec le système l'adéquation de la RTO résultante avec les besoins applicatifs. Le cadre de RI sémantique dans lequel nous nous plaçons nous pousse également à nous intéresser à la problématique de gestion de l'évolution de la RTO (avec notamment ses conséquences sur les index conceptuels existants). Enfin, de façon à disposer d'un formalisme riche et modulaire et assurer une interopérabilité souhaitable dans le contexte du Web Sémantique, nous avons choisi le standard OWL comme langage de représentation pour la suite.

2 Recherche d'information et ontologies

Dans ce chapitre, nous exposons les principales techniques employées dans le domaine de la Recherche d'Information (RI) textuelle et plus particulièrement celles liées à l'utilisation d'ontologies (ce que nous appellerons par la suite RI sémantique). Pour justifier notre démarche, nous définissons en 2.1 le processus de RI d'un point de vue informatique : Comment peut-on décomposer cette notion ? Quelles sont les techniques mises en œuvre ? Nous nous penchons ensuite plus en détail sur le sous-domaine de la RI sémantique et sur deux de ses principaux processus, à savoir l'indexation sémantique et la comparaison sémantique.

2.1 Recherche d'information : éléments de base

Avec les premiers travaux des années 1960 comme [Cleverdon, 1962] ou [Salton et Lesk, 1965], la RI appartenait principalement au domaine de l'ingénierie documentaire et s'intéressait à l'indexation de textes d'une collection donnée dans un but ultérieur de recherche dans cette collection. L'avènement d'Internet au début des années 90 a provoqué un bouleversement dans la pratique de la RI. En effet, la création de ce réseau a permis la mise à disposition de tous d'un moyen de partage d'informations simple et sans précédent. La quantité et la diversité des informations disponibles, ainsi que l'absence d'un modèle sous-jacent bien défini pour la structuration du Web, ont rapidement amené au problème de l'accès à l'information [Baeza-Yates et Ribeiro-Neto, 1999]. Dans ce contexte, la RI a immédiatement gagné en popularité. Désormais, elle a considérablement approfondi ses domaines d'intérêt, notamment sur des problématiques liées à la classification et catégorisation de documents, la visualisation de données, les interfaces homme-machine ... En parallèle, la notion de document s'est progressivement enrichie et peut désormais désigner des artefacts aussi divers que des textes, des images, des sons, des vidéos. Nous expliquons dans cette section en quoi consiste une tâche de RI textuelle dans son acception la plus classique, pour faire ensuite un inventaire des principales techniques impliquées à chaque étape du processus.

2.1.1 Objectifs et description de la tâche

Pour décomposer un processus de RI dans un ensemble de documents textuels (indifféremment une base documentaire ou le Web), envisageons un cas de recherche ad hoc. [Belew, 2000] conçoit la tâche étudiée comme un dialogue entre l'utilisateur et le système de RI (SRI), dont l'objet pour l'usager est l'acquisition de connaissance : l'utilisateur exprime ses besoins en information, le moteur de recherche lui fournit des éléments susceptibles de l'aider, l'utilisateur analyse alors les résultats et peut reformuler/préciser son besoin si nécessaire. Par la suite, nous préférons ne pas aborder l'étape d'analyse par l'utilisateur de l'aide fournie par le SRI car nous la jugeons trop éloignée de notre cadre d'étude. Nous nous concentrerons donc en priorité sur les deux premières étapes.

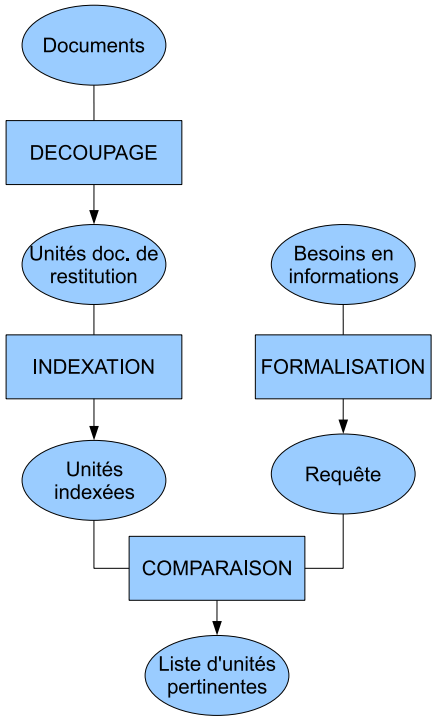

Figure 2.1 — Système de Recherche d'Information classique

2.1.2 Interaction entre humain et logiciel

Avant d'aborder les étapes d'indexation et de comparaison inhérentes à tout SRI, intéressons-nous aux interactions entre un SRI et son utilisateur à travers les problématiques d'expression des requêtes et de granularité des réponses.

2.1.2.1 Formulation des besoins

Cette étape est relativement délicate car l'utilisateur doit être capable de formuler dans un langage donné les lacunes cognitives à combler. Ces lacunes peuvent être partielles (e.g. l'utilisateur connaît vaguement le domaine de la finance mais souhaiterait en savoir plus sur les placements possibles pour son épargne) ou totales, auquel cas il devient difficile

d'exprimer clairement un besoin informationnel[1]. Il est important de faire la distinction entre la nature des besoins formulés par l'utilisateur et la requête, traduction de ces besoins dans le langage manipulé par le SRI [Mothe, 2000]. Différentes approches existent : alors que certaines exigent que l'utilisateur formule ses besoins directement dans une syntaxe particulière liée au langage de requête, d'autres préfèrent guider l'utilisateur par une interface adaptée ou en le contraignant à employer des termes d'un vocabulaire contrôlé ; au détriment de devoir rajouter une phase de post-traitement de la requête, une dernière catégorie permet à l'utilisateur de formuler ses besoins en langage libre (i.e. suite de mots ne respectant pas nécessairement la structure syntaxique d'une phrase en langue naturelle).

2.1.2.2 Documents et unités documentaires

Si l'on revient dans les années 1960 aux origines de la RI avec l'ingénierie documentaire, on constate que la tâche principale d'un SRI consistait à apparier une requête avec une ou plusieurs descriptions d'ouvrages (appelées documents secondaires) afin de permettre à un usager d'accéder à ceux-ci. Par la suite, l'avènement d'Internet a amené la RI à s'intéresser au contenu même d'un document textuel (i.e. au document primaire) dans le but d'en connaître les différents thèmes de façon plus précise. Du fait de la grande taille de certains documents disponibles sur le Web, il est devenu urgent de se préoccuper de la nature des résultats à présenter à un utilisateur d'un SRI. De fait, il semble peu pertinent de retourner l'intégralité d'un document lorsque l'information recherchée n'y tient qu'en quelques lignes. Pour remédier à ce problème, il est nécessaire de distinguer un document de ses différentes unités de restitution [Mothe, 2000]. De même, on peut également définir une unité documentaire d'indexation, correspondant aux portions de document utilisées par le SRI au cours de la comparaison avec la requête. Pour plus de détails sur la problématique des unités de restitution en RI, on pourra consulter plusieurs études menées autour de ce thème comme [Callan, 1994], [Hernandez, 2005] ou [Fourel *et al.*, 1998]. Dans nos travaux, du fait de la structuration formelle et de la brièveté des documents auxquels nous nous sommes intéressé, nous avons choisi comme unité d'indexation d'un document un ensemble (éventuellement la totalité) de champs contenus dans le document et comme unité de restitution le document lui-même.

2.1.3 Indexation des documents

2.1.3.1 Origine et principe

Le processus d'indexation en RI dérive de la notion d'index en ingénierie documentaire, dont le but est de fournir une vue synthétique des thèmes d'un document [Baeza-Yates et Ribeiro-Neto, 1999]. Dans son acception documentaire, un *index* se définit comme l'*association entre une nomenclature* (i.e. une liste de descripteurs dont la structure peut se rapprocher de celle d'un thésaurus) *et un ensemble de renvois* (références à une ou plusieurs pages, voire à d'autres descripteurs) [Mekki et Nazarenko, 2001]. Alors que certains

[1]Dans ce cas de figure, il sera plus simple pour l'utilisateur de naviguer dans des documents relatifs au(x) domaine(s) approprié(s) pour arriver à son objectif sans avoir à l'expliciter [Hernandez, 2005]. C'est la problématique connexe de l'exploration d'informations, que nous n'abordons pas dans ce mémoire.

considèrent comme nécessaire la normalisation des connaissances manipulées dans un index [Gros et Assadi, 1997], d'autres jugent plus pertinent de ne pas contraindre celles-ci afin de garantir la diversité des points de vue [Bourigault et Charlet, 1999].

Avec l'apparition d'Internet se fait sentir le besoin de pouvoir retrouver rapidement des documents traitant d'un sujet donné. Du fait de la croissance exponentielle du nombre de documents disponibles sur le Web, une approche manuelle du problème telle qu'employée par l'ingénierie documentaire n'est pas viable. La RI adapte donc la notion d'index pour en faire le socle de base d'une approche automatisée. Dans ce contexte, le processus d'indexation consiste à extraire d'un document ou d'une requête une représentation caractéristique de sa sémantique [Baziz, 2005]. Le résultat correspond à un ensemble de termes appelé langage d'indexation. Ce langage peut s'avérer être un lexique figé (langage contrôlé) ou non (langage libre), auquel cas l'index se compose des termes principaux du document, bien souvent extraits de façon automatique. Dans un contexte de RI, on peut alors attribuer à la notion d'index une définition plus opérationnelle : il s'agit d'une *structure recensant de façon univoque* (e.g. via les URI) *et pour chaque terme du langage d'indexation les ressources textuelles mentionnant ce terme ainsi qu'éventuellement, la (les) position(s) relative(s) de son (ses) occurrence(s)*. A ce niveau, nous préférons avertir le lecteur d'une ambiguïté lexicale : l'appellation de "terme" pour un descripteur d'index dans un cadre de RI ne correspond pas à la notion homonyme abordée tout au long du précédent chapitre. Par "*terme*", on comprendra ici "*groupe de mots possédant une unité sémantique*[2]". Le terme tel qu'on l'entend ici peut donc se réduire à sa portion congrue, à savoir une simple chaîne de caractères.

On peut définir l'indexation automatique d'un document selon deux étapes : pendant une première phase, le SRI sélectionne les termes caractéristiques au contenu textuel ; il peut ensuite juger du poids relatif de chacun d'eux de façon à affiner sa perception de la sémantique du document. Même si nous avons déjà mentionné certaines techniques liées en 1.3.3.1, nous détaillons ci-après les principes de repérage et de pondération de termes avec un langage d'indexation libre. En 2.1.5, nous abordons les principales limites d'une telle approche, ainsi que les avantages et inconvénients à recourir à un processus d'indexation par un langage contrôlé.

2.1.3.2 Repérage des termes

Si l'on se base sur la figure 2.2, on constate que le repérage de termes est le résultat de l'enchaînement de plusieurs modules plus ou moins facultatifs :

- l'**analyse syntaxique** consiste en l'application d'une grammaire sur le contenu textuel pour en dégager les syntagmes nominaux, verbaux ...
- l'**utilisation d'un anti-dictionnaire** permet d'éliminer les "mots vides" de l'index (e.g. "*le | la | les*", "*et | ou*", "*être | avoir*" ...) et de réduire ainsi sa taille. Les mots vides correspondent à des termes pouvant apparaître dans n'importe quel texte et par conséquent non discriminants pour la tâche de RI.
- la **lemmatisation** consiste à associer un mot à sa forme canonique, à savoir l'infinitif pour un verbe et le masculin et/ou singulier pour le reste (e.g. "*mangeront*" donne "*manger*", "*délicates*" donne "*délicat*").

[2] On ne présume pas quant à la nature implicite ou explicite de cette sémantique.

– La **radicalisation**, opération plus extrême que la lemmatisation, représente toute variante d'un terme par une forme unique. Pour remplir cet objectif, plusieurs stratégies sont possibles, comme l'utilisation de tables de correspondance, la suppression de préfixes et suffixes, la troncature ... Là encore, au détriment des temps de traitement, de tels processus permettent de réduire la taille finale de l'index et de rassembler de facto certaines expressions lexicalement proches. Par exemple, selon un algorithme adapté de [Porter, 1980] pour la langue française, le mot "*itération*" est transformé pour la suite des traitements en la chaîne de caractères "*iter*".
– l'**analyse statistique** sert, entre autres, à regrouper en termes des mots dont la sémantique émerge de l'interprétation de leur juxtaposition. Elle a usuellement recours à la notion de co-occurrence de mots (i.e. leur apparition simultanée) pour essayer de dégager de nouveaux termes. Une forte co-occurrence entre deux mots peut s'expliquer de différentes manières : soit ces mots ne sont interprétables qu'en étant combinés (e.g. "*arbre à cames*"), soit leurs sens en contexte sont liés par une relation sémantique (e.g. les mots "*chat*" et *"souris*" car généralement, les chats chassent les souris), soit ils désignent des notions de sens proche (e.g. "*train*" et "*auto*" dans un contexte du type "*Train et auto sont les moyens de transport les moins onéreux.*")

Parmi l'ensemble des processus possibles, seule l'analyse statistique est réellement indépendante de la langue dans laquelle est rédigé le document à indexer.

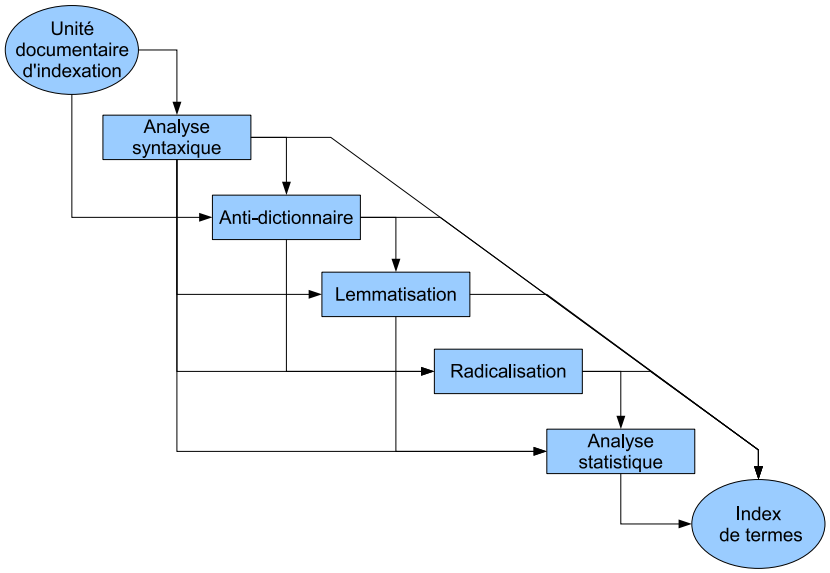

Figure 2.2 — Processus générique de reconnaissance de termes adapté de [Baeza-Yates et Ribeiro-Neto, 1999]

2.1.3.3 Détermination de l'importance d'un terme

Pour aller plus loin dans la caractérisation d'un document à partir des mots qu'il contient, certains SRI choisissent d'estimer numériquement leur importance relative. Généralement, le calcul du poids d'un terme relativement à un document et un ensemble de textes se fonde à la fois sur son importance au sein du document (pondération locale) et sur son potentiel à différencier le document en question des autres (pondération globale). En

effet, un terme fortement discriminant et représentatif du document sera d'autant plus susceptible de permettre de retrouver un document pertinent (lorsque la requête le contient).

Pour mesurer l'importance locale d'un terme, la plupart des approches utilise sa fréquence d'occurrence dans le texte. De fait, la conjecture de Luhn [Luhn, 1958] détecte une corrélation entre ces deux notions. Elle se fonde pour cela sur la loi de Zipf[3] [Zipf, 1949] et sur certaines observations : les mots très fréquents dans un document ont tendance à se retrouver aussi nombreux dans l'ensemble des textes à indexer (et sont donc peu discriminants) alors que les mots très rares ne permettent généralement pas d'observer de régularités d'usage. Après avoir défini un intervalle de rangs appropriés, on peut alors calculer la pondération locale d'un terme à partir de sa fréquence. Certaines formules conçoivent le poids du terme comme directement proportionnel à sa fréquence dans le document (*term frequency*), certaines établissent un rapport logarithmique, d'autres encore normalisent le poids en fonction de la fréquence maximale constatée dans l'ensemble des documents de la collection. On notera tout de même que la loi de Zipf est une approximation qui ne semble refléter la réalité que dans le cas de termes composés d'un seul mot. Des recherches comme celles de [Egghe, 1999] ont été menées afin d'affiner cette loi et de permettre des calculs d'importance locale de termes multi-mots plus appropriés.

La pondération globale mesure directement le potentiel discriminant d'un terme en considérant qu'il sera d'autant plus élevé que le nombre de documents dans lesquels le terme apparaît sera faible. La mesure la plus commune, appelée *Inverse of Document Frequency*, se définit ainsi pour le terme t [Spärck-Jones, 1972] :

$$IDF(t) = -log\frac{df(t)}{|D|}$$

avec $df(t)$ le nombre de documents dans lesquels apparaît t et $|D|$ le nombre total de documents. Certains travaux comme [Singhal *et al.*, 1995] ou [Robertson et Walker, 1997] proposent des mesures proches d'IDF mais plus adaptées à des collections de documents à taille variable car elles prennent en compte la taille relative d'un document par rapport à la moyenne et évitent ainsi de favoriser les textes longs.

2.1.4 Calcul de similarité requête/document

Un SRI peut se baser sur plusieurs grands types de modèles afin de mesurer l'adéquation entre une requête et un ensemble de mots-clés issus de l'indexation d'un document. Nous reprenons ici en partie la description de ces modèles réalisée dans [Baziz, 2005], auquel le lecteur est conseillé de se référer pour une énumération plus détaillée.

2.1.4.1 Modèles booléens

Le **modèle booléen simple** est historiquement un des plus anciens, mais aussi un des plus restrictifs. Il permet d'exprimer une requête sous la forme de conjonctions, de disjonctions et/ou de négations de termes [van Rijsbergen, 1979]. Un document sera retrouvé

[3]Cette loi empirique affirme que fréquence et rang d'un mot (i.e. sa position dans la liste des mots classés par fréquence décroissante) sont inversement proportionnels.

par un SRI booléen à partir d'une requête conjonction (respectivement disjonction) de deux termes si et seulement si il contient les deux termes (resp. un des deux). Une requête constituée par une négation de terme sera satisfaite par tout document ne contenant pas ce terme. Dans ce modèle, l'évaluation de la pertinence est binaire : un document est pertinent ou non. Par définition, un tel modèle permet de garantir des résultats qui répondent exactement à la requête. Toutefois, lorsque l'utilisateur n'a qu'une vague idée de son besoin en informations, il est souhaitable qu'il puisse avoir accès à des documents ne répondant que partiellement à sa requête. Le **modèle booléen étendu** a été proposé dans [Salton *et al.*, 1982] afin de répondre à ce besoin et pour ce faire, il introduit le poids des termes de la requête dans le calcul de similarité. De façon similaire, le **modèle booléen flou** permet de représenter une pertinence partielle dans l'appariement requête-document [Baranyi *et al.*, 1998] : la similarité entre une conjonction (resp. disjonction) de deux termes et un document sera égale à la similarité minimale (resp. maximale) entre chacun des termes et le document.

2.1.4.2 Modèles vectoriels

Proposé par [Salton *et al.*, 1975], le **modèle vectoriel** définit un document/une requête dans l'espace constitué par l'ensemble des termes qui l'indexent. Chacune des coordonnées correspond au poids du terme associé, selon une des formules présentées en 2.1.3.3. Par extension, les documents sur lesquels le SRI opère sont représentés par une matrice numérique de taille $N \times M$ avec M le nombre de documents et N le nombre total de termes dans tous les documents. Pour calculer la similarité entre une requête et un document, on retrouve alors les mesures traditionnelles de proximité en calcul vectoriel comme le produit scalaire, le cosinus, la mesure de Jaccard ... Un des principaux inconvénients de ce modèle réside dans l'obligation pour un texte de contenir au moins un des mots de la requête pour pouvoir être retrouvé (car les dimensions de l'espace vectoriel étant orthogonales, les termes sont considérés comme indépendants).

Penchons nous maintenant sur un modèle vectoriel particulier, à savoir l'approche par **indexation sémantique latente** ou **LSI**. La LSI part du constat selon lequel la matrice représentant les documents en fonction des termes est une matrice creuse. En effet, parmi l'ensemble des termes employés dans les documents, seule une petite partie est utilisée dans chaque texte. La LSI utilise alors la technique de décomposition en valeurs singulières pour réduire l'espace de représentation des documents [Furnas *et al.*, 1988]. Ce faisant, elle regroupe les mots en plusieurs classes, faisant ainsi apparaître un type d'entités opérant un lien entre les documents et les mots. Ces entités ont pour caractéristique de regrouper les mots utilisés dans des contextes proches. L'appariement entre requête et document s'appuie alors sur ces entités. Ce modèle, tout en se basant sur des critères purement statistiques, peut être considérée comme une façon implicite de donner une sémantique aux documents, puisqu'avec elle, une requête et un document ne partageant pourtant aucun mot en commun, pourraient être jugés proches. Toutefois, du fait qu'elle exploite les contextes d'utilisation des termes, la LSI reste limitée à des collections de documents relativement homogènes (en taille) et de taille suffisante (sans cela, le vocabulaire employé n'est pas assez diversifié pour garantir une approximation correcte). De plus, l'approche par LSI se heurte à des problèmes de passage à l'échelle, puisque la décomposition par valeurs singulières est d'autant plus

coûteuse que les dimensions de la matrice termes-documents sont élevées (cas typique d'un grand nombre de documents, comme celui mobilisé au cours d'une RI sur Internet).

2.1.4.3 Modèles probabilistes

L'approche classique d'un modèle probabiliste consiste à évaluer par une probabilité le degré de pertinence que peut avoir un document par rapport à une requête [Robertson, 1977]. Le calcul d'une telle probabilité, que l'on peut noter $P(Pert|D_i)$, utilise les principes fondamentaux de la théorie des probabilités :

$$P(Pert|D_i) = \frac{P(Pert)}{P(D_i)} * P(D_i|Pert) \tag{2.1}$$

où $P(Pert)$ désigne la probabilité qu'un document quelconque soit pertinent pour la requête, $P(D_i)$ la probabilité d'obtenir le document D_i. Cette probabilité peut se décomposer comme suit :

$$P(D_i) = P(D_i|Pert) * P(Pert) + P(D_i|\overline{Pert}) * P(\overline{Pert}) \tag{2.2}$$

Pour calculer $P(D_i|Pert)$ (ou $P(D_i|\overline{Pert})$), on considère que le document D_i peut être représenté par les m termes t_j qui l'indexent (à la manière des approches vectorielles) et on suppose que ces termes sont mutuellement indépendants. On a alors

$$P(D_i|Pert) = \prod_{j=1}^{j=m} P(t_j|Pert) \tag{2.3}$$

avec $P(t_j|Pert) = \frac{r_j}{R}$ (et $P(t_j|\overline{Pert}) = \frac{n_j - r_j}{N-R}$) où R est le nombre de documents pertinents pour la requête, r_j le nombre de documents pertinents contenant le terme t_j, N le nombre total de documents et n_j le nombre de documents contenant t_j.

En étudiant un échantillon représentatif de documents, il est alors possible d'estimer par l'emploi d'une loi de distribution (e.g. la loi de Poisson) les probabilités $P(t_j|Pert)$ et d'en déduire $P(Pert|D_i)$ pour chaque document à indexer. Les documents sont alors classés par ordre décroissant selon ces valeurs et retournés à l'utilisateur. Par la suite, malgré le succès effectif de cette approche, plusieurs travaux comme [Robertson *et al.*, 1981] se sont appliqués à relaxer l'hypothèse d'indépendance des termes. En effet, comme [Lewis, 1998] le souligne, l'approche probabiliste classique ne prend en compte ni la fréquence d'occurrence d'un terme dans un document ni la longueur de ce document, ce qui devrait - a priori - amener des imprécisions dans le modèle.

Le modèle "Document Network" La démarche présentée dans [Turtle et Croft, 1991] s'inspire des réseaux bayésiens et du modèle de Dempster-Shafer. Elle propose de représenter le processus de RI par la fusion de deux graphes acycliques orientés dont les origines correspondent aux documents de la base de recherche, la feuille représente la requête de l'utilisateur et les arcs reliant les noeuds symbolisent les probabilités conditionnelles d'observer un événement en fonction d'un autre. Comme on peut le voir dans la figure 2.3, les auteurs distinguent deux sous-réseaux à l'intérieur d'un tel graphe :

– le **réseau de documents** est un artefact construit a priori par une phase d'indexation, la probabilité d'observer un terme donné étant conditionnée par le poids (cf 2.1.3.3) de celui-ci dans chacun des documents.
– le **réseau de la requête** est construit dynamiquement. Pour cela, la requête de l'utilisateur (initialement sous la forme de texte libre ou de type booléen) est décomposée en plusieurs termes primitifs dont la présence dans un document sera considérée comme une preuve que la requête de l'utilisateur aura été satisfaite. Selon la nature des connecteurs logiques utilisés dans la requête, sa probabilité d'être satisfaite en fonction des termes primitifs rencontrés sera calculée de façon différente.

La liaison entre les deux réseaux se fait par la mise en correspondance des termes primitifs avec les termes indexants. Cette connexion permet de représenter les cas d'identité (i.e. un terme trouvé dans un sous-ensemble de documents correspond à un élément de la requête), mais aussi des proximités d'ordre plus "sémantique". Le processus dans sa globalité consiste alors à calculer systématiquement la probabilité de satisfaction de la requête en fonction de chaque document.

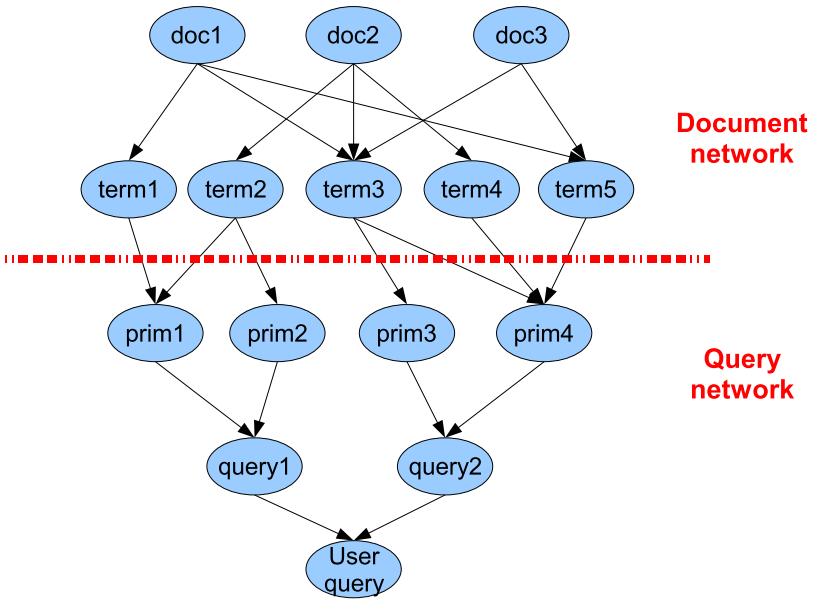

Figure 2.3 — Représentation simplifiée d'un réseau d'inférence

Le système InQuery décrit dans [Broglio *et al.*, 1993] implémente un tel modèle. Il utilise des techniques classiques pour l'indexation des termes des documents (analyse syntaxique, anti-dictionnaire, radicalisation) mais essaie aussi de rapprocher certains termes désignant des objets de même nature (villes, pays, compagnies, personnes) par l'emploi de règles heuristiques. De même, pour l'étape de mise en correspondance du réseau de requête et du réseau de documents, InQuery pondère le lien existant entre un terme indexant et un terme primitif selon la proximité de leurs contextes d'usage : de cette façon, la présence d'un terme dans un texte pourra permettre à ce document de satisfaire une requête contenant un terme différent mais au contexte d'usage proche. On constate donc qu'ici encore, la volonté est de parvenir à une interprétation sémantique partielle de façon à améliorer les résultats par rapport à un SRI classique.

2.1.5 Limites de l'approche par indexation libre

Parmi les approches de RI utilisant un langage d'indexation libre, nous pouvons citer - entre autres - deux approches légèrement différentes, à savoir les méthodes fondées sur l'occurrence de mots-clés dans les requêtes et les textes d'une part, et les méthodes exploitant en sus la co-occurrence de ces mêmes mots-clés d'autre part. Nous examinons ci-après les limites de ces deux types d'approches par rapport à des techniques d'indexation par un langage contrôlé.

Le principal reproche que l'on peut faire à la première approche de RI par mots-clés réside dans son postulat de base : la comparaison entre une requête et un texte se fait sur leur contenu lexical et non sémantique. S'ensuivent certaines conséquences malheureuses, dont la dépendance de ces techniques au lexique employé tant dans la requête que dans les documents : deux requêtes similaires mais employant des synonymes n'auront pas forcément la même liste de documents susceptibles d'être pertinents (rappel [4] décru) ; de même, les termes d'une requête ne seront pas désambiguïsés par ce genre de système, amenant certains textes à être retournés à l'utilisateur de façon erronée (précision [4] décrue).

Pour remédier à ces problèmes, des techniques faisant appel à la notion de co-occurrence sont apparues : par une étude statistique du contexte d'utilisation des termes, elles permettent de comparer ceux-ci selon l'hypothèse que deux termes partageant un même contexte lexical auront des sens proches. Ce principe permet à un SRI de s'affranchir en partie de l'influence du lexique sur les résultats de recherche. L'efficacité des premières approches fondées sur la co-occurrence restait à confirmer [Chen *et al.*, 1998] : alors que certaines expériences rapportaient de bons résultats, d'autres concluaient sur le peu d'utilité d'exploiter la co-occurrence. On peut tenter d'expliquer le manque de cohérence dans ces résultats par la nature variable des requêtes (e.g. des requêtes courtes peuvent empêcher un emploi profitable des techniques de co-occurrence), par l'adéquation variable du vocabulaire de l'utilisateur avec celui des documents, et par la nécessité d'envisager une phase de désambiguïsation de la requête. Plusieurs travaux récents comme [Baziz, 2005] montrent des résultats plus positifs dans des recherches liées à la question.

Indépendamment de ces considérations, l'exploitation de mesures statistiques pour identifier et comparer les termes les plus importants d'un corpus peut, dans certaines situations, se révéler inefficace : en effet, si les documents de la base de recherche sont trop concis ou en nombre trop restreint, il sera plus difficile (voire impossible) de se fonder sur des données numériques comme la fréquence d'apparition d'un groupe de mots (que ce soit dans un document ou sur l'ensemble du corpus). Nous situons justement nos travaux dans ce contexte particulier, puisque notre cas d'étude est constitué d'un ensemble très limité de documents comportant chacun peu de mots [5]. C'est pourquoi nous examinons par la suite les apports potentiels d'un processus d'indexation fondé sur des techniques non statistiques.

Le choix d'une phase d'indexation via un langage contrôlé (ou indexation sémantique) comporte, dans notre contexte d'étude, certains avantages. De fait, il permet de disposer d'une sémantique explicite dans laquelle représenter une requête ou un texte. Contraire-

[4] Pour une définition des notions de rappel et précision, voir en 5.2.3.1.

[5] Pour une description plus détaillée de la base de recherche, voir en 5.1.1.1.

ment à l'approche par co-occurrence de termes, l'indexation sémantique permet de contrôler précisément le processus de comparaison entre deux notions (cf 2.2.2). De plus, l'emploi d'une ressource sémantique formelle extérieure à la base de recherche permet d'apporter une valeur ajoutée en termes de raisonnement logique. En effet, la détection de certaines connaissances explicitement présentes dans un texte ou une requête peut amener le SRI à déduire des connaissances implicites que n'aurait pu inférer un SRI traditionnel [Maynard *et al.*, 2005].

Si la RI sémantique semble posséder plusieurs avantages sur une RI par mots-clés, elle demande également un investissement humain supplémentaire. En effet, elle nécessite qu'une RTO adaptée soit disponible au préalable afin d'établir un lien entre les niveaux lexical et sémantique. On peut distinguer deux cas de figure : soit les documents sur lesquels s'effectue la tâche de RI sont à vocation généraliste, soit ils se cantonnent à un domaine particulier. Dans le premier cas, du fait de sa vaste portée, la RTO existe habituellement déjà. La ressource la plus communément utilisée (e.g. [Richardson et Smeaton, 1995], [Mandala *et al.*, 1998] ou [Baziz *et al.*, 2005]) est WordNet[6], base de données lexicale qui fait correspondre à un terme l'ensemble de ses sens par des listes de synonymes (appelées synsets). Chaque sens de terme participe éventuellement à des relations sémantiques (hypéronymie, méronymie ...) avec d'autres synsets. La principale difficulté rencontrée par la RI sémantique sur une base de recherche généraliste réside dans la désambiguïsation des termes rencontrés [Baziz, 2005]. Dans le cas où la RI sémantique est conduite sur une base de recherche limitée à un domaine, l'étape la plus délicate est la construction même de la RTO. Celle-ci donne un point de vue spécifique sur le domaine et doit s'adapter à l'objectif applicatif. Elle peut être construite à partir de textes du domaine, par réutilisation ou par adaptation de ressources existantes (voir en 1.3). Dans la section suivante, nous décrivons en détail la tâche de RI sémantique dans le paradigme d'une base de recherche relative à un domaine.

2.2 La Recherche d'Information sémantique appliquée un domaine

Après avoir introduit les éléments de bases nécessaires au déroulement d'un processus de RI, nous nous focalisons désormais sur la RI sémantique appliquée à un domaine particulier. Dans un premier temps, nous explicitons les étapes d'indexation (2.2.1) et de comparaison sémantique (2.2.2). Nous envisageons par la suite les effets que peut avoir l'utilisation d'une ressource évolutive sur les artefacts résultant du processus d'indexation, ce qui nous amène à considérer une phase de gestion de ces conséquences (2.2.3). Nous examinerons alors les différents outils existants qui permettent de gérer au moins une des trois étapes précédemment évoquées (2.2.4).

[6] `http://wordnet.princeton.edu/`

2.2.1 Indexation sémantique

2.2.1.1 Définition

Avant de décrire l'ensemble des étapes qui constitue un processus d'indexation sémantique, nous commençons par nous interroger sur la terminologie à employer. En effet, de nombreux travaux comme [Amardeilh, 2007], [Luong, 2007] ou [Bechhofer *et al.*, 2002] se préoccupent d'annotation sémantique, notion qui semble correspondre dans les faits à notre vision de l'indexation sémantique.

Tout d'abord, définissons précisément le sens que nous donnons à cette notion : nous adhérons au point de vue de [Calabretto et Prié, 2004] qui considèrent l'indexation comme "[permettant d']associer à un document des descripteurs qui permettront de le retrouver". Par extension, une indexation sémantique peut alors être vue comme un processus d'indexation pour lequel les descripteurs d'une ressource textuelle correspondent à son contenu sémantique et sont exprimés dans un langage donné.

Dans ce paradigme, une annotation (en tant que résultat du processus homonyme) peut être définie de la façon suivante :

> "Une annotation est à la base une note critique ou explicative accompagnant un texte, et par extension, une quelconque marque de lecture portée sur un document, que celui-ci soit textuel ou image." [Prié et Garlatti, 2004]

Par essence, l'objet-annotation est donc une méta-donnée, i.e une donnée sur une autre donnée, à savoir le fragment textuel cible. La principale différence entre les activités d'annotation et d'indexation se situe en fait au niveau de leurs motivations respectives : alors que l'annotation ne présume en rien de l'objectif pour lequel on l'emploie, l'indexation poursuit un but précis, à savoir permettre de retrouver ultérieurement le texte [Prié, 2000]. On en déduit que toute indexation sémantique est de facto un processus d'annotation sémantique. Dans le contexte du Web Sémantique, on peut également citer la composition de documents comme un usage possible de l'annotation sémantique [Prié et Garlatti, 2004]. Par la suite, comme nos travaux se situent dans un contexte de RI, nous ferons systématiquement référence à la notion d'indexation sémantique (que par abus de langage, nous pourrons éventuellement désigner par le terme d'annotation sémantique). Comme nous utilisons les ontologies comme langage de représentation pour l'index, nous ne distinguerons pas comme le fait [Baziz, 2005] l'indexation sémantique (approche se fondant sur le sens des mots) de l'indexation conceptuelle (approche se fondant sur une structure conceptuelle). Nous emploierons donc indifféremment les deux termes.

En tant que processus d'annotation, l'indexation sémantique peut être analysée selon plusieurs dimensions [Marshall, 1998], avec entre autres :

- son **objectif** est de constituer un moyen d'accès simplifié au contenu du document analysé (et non de l'enrichir comme dans le cas de certaines annotations libres).
- la **portée** d'une annotation sémantique ne se limite généralement pas à un individu, elle est destinée à être partagée et utilisée par un ensemble d'agents (humains et/ou logiciels).
- pour les mêmes raisons, le **niveau d'explicitation** d'une annotation sémantique est relativement élevé. En effet, elle doit être intelligible par plusieurs agents, chacun ayant

un état mental potentiellement différent de celui du producteur de l'annotation.

- le **degré de formalisation** peut varier selon le langage d'annotation et le schéma de méta-données choisis. On peut en effet choisir de stocker les annotations hors de la ressource sémantique. On peut alors distinguer le langage permettant de représenter une annotation sémantique du langage permettant d'exprimer les contraintes sur une ou plusieurs annotations. Par exemple, les travaux de [Amardeilh, 2007] se fondent sur RDF pour formaliser les annotations tandis que XTM (XML Topic Maps) est utilisé pour le réseau sémantique. Comme l'indexation sémantique vise à représenter le contenu sémantique d'un texte de façon à ce que des agents logiciels puissent le manipuler librement, ce processus possède un degré de formalisation nécessairement plus poussé qu'une activité d'annotation libre. Pour une description des langages disponibles actuellement pour la représentation d'ontologies et/ou d'annotations sémantiques, le lecteur pourra se reporter en 1.4.
- la **durée de vie** d'une annotation sémantique dépend fortement de la stabilité du vocabulaire contrôlé et de la structure conceptuelle sur lesquels elle est fondée. Si ceux-ci sont modifiés en réaction à une évolution des usages linguistiques ou du domaine modélisé, il faudra ajouter, supprimer ou modifier certaines annotations de façon à garantir leur cohérence avec la structure conceptuelle.

Dans [Prié et Garlatti, 2004], en plus des dimensions intrinsèques à un processus d'indexation (comme le degré d'automatisation ou les granularités d'indexation et de restitution déjà abordées en 2.1.2.2) les auteurs caractérisent l'indexation sémantique par le critère supplémentaire du **stockage** des annotations. En effet, deux possibilités sont envisageables :

- stockage interne au texte annoté : cette approche présuppose un accès centralisé aux documents à indexer et un nombre restreint d'agents autorisés à modifier les annotations (ou tout du moins un mécanisme de gestion de configuration). Ses deux avantages résident dans l'absence de nécessité de mémoriser la position du fragment annoté (comme en général, l'annotation encapsule celui-ci, il suffit de savoir repérer cette dernière pour connaître la position du fragment) et dans le repérage immédiat des annotations incohérentes en cas d'évolution des textes indexés. Comme inconvénients majeurs, on peut citer principalement l'impossibilité d'annoter sous cette forme un document qui serait uniquement accessible en lecture (i.e. une grande partie des documents disponibles sur Internet) ainsi que la nécessité de parcourir l'intégralité d'un document pour en connaître ses annotations.
- stockage externe : à l'inverse, ce choix méthodologique impose pour toute annotation sémantique de gérer explicitement la localisation du document annoté (et l'éventuelle position relative du fragment indexé), ainsi que la phase de maintenance des annotations (suite à une évolution de l'ontologie ou des textes). En contre-partie, il devient possible d'indexer sémantiquement n'importe quel document (pour peu qu'il soit dans le domaine de représentation de l'ontologie) et de disposer de plusieurs points de vue différents sur le même texte.

En outre, nous souhaitons mentionner un paramètre d'importance peu abordé dans la littérature, à savoir la **nature possible de l'artefact ontologique désigné par un terme** des documents à indexer. Plusieurs objets ontologiques de nature différente peuvent être mis en jeu dans un processus de RI sémantique (tab. 2.1) : des concepts, des instances, des attributs

ou des relations. L'emploi de chacun de ces types est associé à un but précis pour l'indexation sémantique. Ainsi, le repérage de termes associés à des instances de concepts est utile pour la reconnaissance d'entités nommées (NER, pour "Named Entity Recognition"). Dans nos recherches, nous avons préféré ne pas nous focaliser sur ce genre d'indexation dont le but consiste en général à repérer de façon incrémentale des entités nommées à l'aide de patrons lexico-syntaxiques (en cela, le repérage des relations ontologiques correspondantes a une importance non négligeable dans ces approches). On notera que selon le contexte, certains termes dénotant un concept de l'ontologie peuvent se référer soit à ce concept soit à une de ses instances. Ce phénomène dépend en fait de la portée (universelle/spécifique) donnée au terme en contexte.

Connotation absolue d'un terme repéré	**Nature de l'artefact désigné**	**Intérêt du repérage du terme**	**Exemple**
concept	concept	découverte de connaissances générales	"Un thermocontact contrôle tout motoventilateur"
	instance	découverte de connaissances spécifiques	"Le motoventilateur du véhicule est trop bruyant"
instance	instance	reconnaissance d'entités nommées (NER)	"Actia est une société du domaine du diagnostic auto"
attribut	attribut	repérage de valeur d'attribut	"La température du moteur atteint **100°C**"
relation	relation	reconnaissance de concepts ou instances par patron lexico-syntaxique	`Panne` + affecte + `Système`

Tableau 2.1 — Nature des objets ontologiques dénotables par un terme donné

2.2.1.2 Description du processus

Le processus d'indexation sémantique se décompose en deux étapes : la première consiste à déterminer quels artefacts ontologiques sont mis en jeu dans les documents à indexer, tandis que la deuxième étape évalue l'importance relative de chacun des objets détectés dans un contenu textuel.

Pour l'identification des entités de l'ontologie mises en jeu dans chaque document, une approche manuelle semble peu adaptée, surtout dans le contexte du Web Sémantique : dans [Erdmann *et al.*, 2000], les auteurs jugent celle-ci trop coûteuse en temps et non exempte d'erreurs. De façon plus pragmatique, une majorité d'outils essaie donc d'automatiser ce processus autant que possible. Pour ce faire, deux grands types de démarches complémentaires émergent :

- **la projection de la RTO sur les documents** commence par s'appuyer sur des techniques syntaxiques et/ou statistiques d'extraction de termes similaires à celles mises en oeuvre pour une indexation classique (voir en 2.1.3.2). Une fois repérés les termes constitutifs de chaque document, une comparaison avec les termes liés aux entités on-

tologiques permet de déterminer quels artefacts ontologiques sont mis en oeuvre par document. En cas de conflit d'appariement (i.e. deux termes de la RTO sont potentiellement retrouvés à des positions se recouvrant dans le texte), la priorité est généralement donnée au terme le plus long car on considère qu'il correspond au concept plus spécifique [Baziz, 2005]. Lorsqu'un terme polysémique de la RTO est retrouvé dans un texte, il est nécessaire de le désambiguïser, c'est-à-dire de déterminer le sens utilisé dans le contexte local au document [Sanderson, 2000]. Plusieurs stratégies de désambiguïsation sont possibles [Hernandez, 2005] : soit le texte est indexé par tous les concepts que le terme peut dénoter, soit le concept le plus fréquent dans le texte est choisi, soit le concept est choisi en fonction de sa proximité sémantique (voir 2.2.2) avec les autres concepts reconnus dans son voisinage (pouvant varier d'une fenêtre d'une dizaine de mots autour du terme ambigu jusqu'à l'ensemble du texte).

– **l'utilisation d'un moteur d'extraction d'information** consiste à appliquer un ensemble de patrons d'extraction, sachant qu'un patron peut mêler des informations de type lexical, syntaxique et/ou sémantique. Leur projection sur une collection de documents permet de découvrir de nouvelles lexicalisations de concepts ou de nouvelles instances de concept. Ces patrons peuvent être soit construits manuellement avant leur application, soit proposés suite à une phase d'apprentissage supervisé [Amardeilh, 2007]. Ce genre d'approche est généralement semi-automatique et permet à l'utilisateur de valider les propositions du système avant qu'elles ne soient ajoutées à l'ensemble des annotations sémantiques d'une part, et à la base de faits associée (pour les nouvelles instances de concept) ou à la RTO (pour les nouvelles lexicalisations) d'autre part.

Une fois les concepts et/ou instances repérés, il est possible d'affiner le résultat du processus d'indexation sémantique en déterminant l'importance relative de chacun par le calcul d'un poids numérique. Là encore, plusieurs types de facteurs peuvent être utilisés séparément ou combinés pour obtenir cette valeur :

– les **paramètres d'ordre statistique** se fondent sur la fréquence d'occurrence d'un concept/instance et les notions de pondérations locale et globale déjà mentionnées en 2.1.3.3 pour l'indexation classique [Vallet *et al.*, 2005]. Avec le passage à un niveau sémantique, se pose un nouveau problème : pour évaluer correctement la fréquence d'occurrence d'un artefact ontologique, il est nécessaire de comptabiliser toute expression s'y référant de façon plus ou moins directe (phénomène de deixis comme l'anaphore grammaticale). De même, un terme (même s'il est systématiquement désambiguïsé), à travers ses diverses occurrences, peut faire référence à des instances différentes d'un même type.
– les **paramètres d'ordre structurel** prennent en compte la structure des documents indexés et attribuent une même importance à des entités sémantiques situées dans une même sous-structure. A l'inverse, une entité sémantique repérée dans un titre aura logiquement plus d'importance qu'une entité repérée dans une note de bas de page.
– les **paramètres d'ordre sémantique** cherchent à exploiter le contexte sémantique dans lequel est détecté un artefact ontologique. Pour cela, ils calculent la proximité sémantique de l'artefact avec les autres annotations présentes dans le texte. Plus l'artefact en sera sémantiquement proche, plus son poids augmentera.

2.2.2 Proximité sémantique entre une requête et un document

2.2.2.1 Formalisation sémantique des besoins de l'utilisateur

Avec la représentation des informations indexées sur un niveau sémantique, la RI gagne en ambition. En effet, un processus de RI sémantique peut repérer de façon beaucoup plus précise et moins ambiguë des informations présentes dans un texte. De là, selon la précision du besoin de l'utilisateur, on peut envisager que le SRI retourne non plus un ensemble de documents potentiellement intéressants, mais une réponse concrète à une requête suffisamment précise. Dans ce cadre, il est nécessaire que le formalisme de représentation de la requête soit assez riche pour formuler une question sur un élément représenté dans la base de faits (i.e. les instances de l'ontologie) ou dans l'ontologie.

Le SRI de type Question / Réponse Parmi les approches cherchant à fournir en sortie du SRI une réponse plutôt qu'un document, un premier groupe se fonde sur des questions posées intégralement en langue naturelle (LN). Ainsi, les travaux de [Cimiano *et al.*, 2007] développent une méthode générique de traduction de requête de la LN vers une logique de description (LD), facilement utilisable pour interroger (via SPARQL) une base de faits représentée en OWL. Les auteurs utilisent en fait les bases du lambda-calcul pour transformer la structure syntaxique d'une phrase de LN en un formalisme proche des LD. Au préalable, ils séparent le lexique en une partie indépendante du domaine d'application de l'ontologie (e.g. les déterminants, les pronoms interrogatifs, les prépositions spatio-temporelles ...) et une partie propre au domaine. Cette seconde partie doit être alignée aux entités ontologiques adéquates. Pour venir en aide à l'opérateur humain réalisant cette étape, les auteurs imaginent un processus cyclique d'alignement du lexique spécialisé avec l'ontologie de domaine : le modélisateur commence par associer le lexique qu'il juge nécessaire aux entités de l'ontologie ; lorsque le système est incapable de traduire en LD une question d'un utilisateur, le vocabulaire inconnu est stocké pour être présenté au modélisateur au cours de la phase de maintenance suivante. Le processus s'arrête lorsqu'une couverture suffisante des questions est atteinte.

Un deuxième type de recherches se préoccupe également d'obtenir pour un SRI des requêtes hautement formalisées, tout en utilisant en entrée un mode d'expression auquel les utilisateurs sont habitués, à savoir l'interrogation par mots-clés. En effet, les approches obligeant les utilisateurs à exprimer leur besoin dans un langage de requête particulier, ainsi que celles qui imposent la définition de la requête par une interface liée au contenu ontologique, semblent trop contraignantes pour un utilisateur moyen (et a fortiori pour un néophyte). Certaines études utilisent pour cela des ressources sémantiques génériques (comme [Royo *et al.*, 2005] avec WordNet). D'autres se focalisent sur des ontologies de domaine, beaucoup plus porteuses de sens. Ainsi, le principe développé par [Lei *et al.*, 2006] consiste à envisager l'occurrence de tout mot-clé de la requête comme une référence potentielle vers une instance, un concept ou une relation ; des règles heuristiques déterminent ensuite, selon le type des entités mises en jeu dans la requête, comment les combiner pour reconstruire le sens global. De leur côté, les travaux de [Tran *et al.*, 2007] préfèrent une approche orientée vers les graphes : une fois repérées les entités de l'ontologie mentionnées

par les mots-clés de la requête (via une phase de RI classique implémentée avec Lucene), leur SRI cherche à les connecter entre elles par un chemin dont la taille ne doit pas dépasser un certain seuil (paramétrable). Si l'opération de connexion réussit, le sous-graphe ainsi obtenu est traduit dans un formalisme de LD. La principale limite du prototype proposé n'est pas liée à l'étape de traduction de la requête mais au fait qu'il ne gère pas automatiquement (pour l'instant) la phase de désambiguïsation au cours de l'indexation sémantique.

Le SRI de type portail vers les documents Les approches précédemment abordées font l'hypothèse commune que la requête formulée par l'utilisateur est assez précise pour que le SRI puisse y répondre par des instances de la base d'annotations. Toutefois, ce cas de figure est assez éloigné et difficilement comparable à l'objectif rempli par une approche de RI traditionnelle, à savoir retourner à l'utilisateur une liste ordonnée de documents sémantiquement proches de sa requête.

La principale différence de ce type de SRI avec celui précedemment abordé réside dans la nécessité de représenter une requête avec un certain degré d'imprécision consécutif aux lacunes cognitives de l'utilisateur. A partir du moment où le besoin de l'utilisateur ne peut être clairement circonscrit, il apparaît logique que plusieurs documents puissent convenir à la requête de façon plus ou moins précise. Concernant les formats possibles pour la représentation de la requête, comme celle-ci s'avère par définition imprécise, il semble inutile de recourir à un texte en langue naturelle. Une approche possible consiste à laisser l'utilisateur exprimer ses besoins par l'emploi d'un langage libre (i.e. ne respectant pas nécessairement de règles syntaxiques). Cette solution a l'avantage d'être bien acceptée par la plupart des utilisateurs, ceux-ci étant habitués à exposer leurs besoins dans un format similaire à travers les SRI commerciaux classiques comme Google. A l'inverse, d'autres approches comme celle de [Guarino *et al.*, 1999] préfèrent miser sur une interface plus contrainte qui pousse l'utilisateur à exprimer explicitement les entités ontologiques qui l'intéressent.

On notera que l'indexation sémantique permet également d'améliorer les techniques d'expansion de requête. Celles-ci consistent usuellement à rajouter des mots-clés à une requête de façon à améliorer le rappel d'un système. En effet, la théorie sous-tend que l'ajout à la requête de termes choisis pour leur proximité pseudo-sémantique[7] avec un mot-clé initial permet de retrouver des textes contenant les termes proches. Toutefois, l'expansion de requête peut a priori exercer une influence négative sur la précision du SRI, notamment au cas où le terme est ambigu. Avec une RTO, le SRI dispose de connaissances sémantiques avérées potentiellement capables d'identifier le sens exact d'un mot-clé ambigu dans le contexte d'une requête. Dans [Vallet *et al.*, 2005], les auteurs considèrent que l'intérêt principal d'utiliser des RTO dans un processus de RI réside dans la manipulation de connaissances explicites via les règles d'inférence auxquelles elles sont soumises et leur organisation en hiérarchie de subsomption. Ils comparent alors une RI sémantique à un processus de RI avec un mécanisme d'expansion de requête solide (car moins générateur de bruit). Leur approche pour construire un SRI générique efficace consiste donc à croiser des techniques classiques de RI fondées sur un modèle vectoriel avec des techniques sémantiques de RI sur les domaines pour lesquels une RTO est disponible.

[7] Celle-ci est souvent calculée au moyen de techniques statistiques comme l'exploitation de co-occurrences déjà mentionnée en 2.1.3.2.

2.2.2.2 Similarité sémantique inter-conceptuelle

Pour estimer un degré d'adéquation entre requête et document, il faut disposer d'une mesure de proximité sémantique qui permet de classer les documents par ordre décroissant de pertinence. Dans le contexte d'une RI sémantique, ce genre de mesure s'appuie entre autres sur la notion de similarité sémantique qui quantifie le degré de ressemblance essentielle de deux concepts par une valeur numérique.

Dans cette partie, nous décrivons donc plusieurs mesures de similarité sémantique entre deux concepts au sein d'une même ontologie. Nous essaierons de présenter les approches selon leur complexité croissante afin de bien apprécier les enjeux associés. Comme le fait [Hernandez, 2005], nous distinguerons les mesures fondées sur le nombre de liens séparant deux concepts, celles évaluant leur quantité d'information partagée, et celles combinant les deux points de vue. Nous aborderons enfin le cas particulier des similarités différenciant dans leur calcul chacune des relations selon sa nature sémantique. Pour la plupart des mesures que nous citerons, nous utiliserons la taxonomie partielle représentée en fig 2.4.

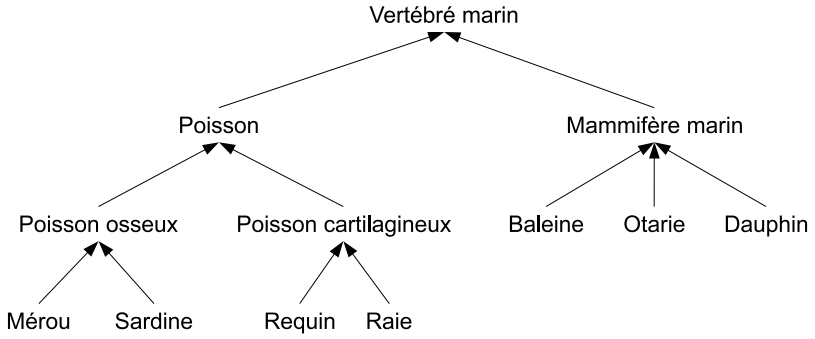

Figure 2.4 — Représentation partielle de la classification biologique des vertébrés marins

Nous ne devons toutefois pas perdre de vue que si la notion de similarité sémantique est utilisée au cours du calcul de proximité entre une requête et un document, elle n'est pas suffisante. En effet, la requête, tout comme le document comparé, est indexée par un ensemble d'entités ontologiques qui peuvent être en relation sémantique mutuelle. Outre un moyen de comparer ces entités deux à deux, nous devons donc posséder des méthodes permettant de faire l'alignement des graphes formés par les deux ensembles d'annotations et de combiner les différentes similarités sémantiques pour obtenir une évaluation numérique globale de la proximité entre requête et document. Nous aborderons cette discussion ultérieurement en 2.2.2.3.

Les mesures par calcul de longueur de chemin L'ensemble des mesures de ce type ont pour spécificité commune de manipuler l'ontologie comme un graphe orienté dont les sommets représentent les concepts et les arêtes les relations sémantiques. Le principe consiste à considérer que deux concepts seront d'autant plus proches que le nombre d'arêtes les séparant sera faible. Ainsi, la mesure développée par [Rada *et al.*, 1989] applique ce principe pour les relations taxonomiques à travers la technique nommée "edge counting". En normalisant

celle-ci entre 0 et 1, on obtient la formule suivante :

$$sim_{edge\_counting}(C_1, C_2) = -log\frac{Dist_{edge\_counting}(C_1, C_2)}{2 * Depth_{max}}$$

avec $Depth_{max}$ correspondant à la profondeur taxonomique maximale de l'ontologie. En reprenant l'exemple des vertébrés marins (fig. 2.4), on a alors :

$$Depth_{max} = 4$$

$$sim_{edge\_counting}(Merou, Sardine) = -log\frac{2}{2*4} \approx 0.6$$

$$sim_{edge\_counting}(Merou, Requin) = -log\frac{4}{2*4} \approx 0.3$$

$$sim_{edge\_counting}(Merou, Baleine) = -log\frac{5}{2*4} \approx 0.2$$

$$sim_{edge\_counting}(Merou, Mammifere) = -log\frac{4}{2*4} \approx 0.3$$

$$sim_{edge\_counting}(Osseux, Cartilagineux) = -log\frac{2}{2*4} \approx 0.6$$

Se plaçant dans le même paradigme, les recherches de [Wu et Palmer, 1994] prennent en compte deux phénomènes supplémentaires : tout d'abord, il paraît contre-intuitif avec la technique d'edge counting d'obtenir une similarité entre poissons cartilagineux et osseux identique à celle entre un mérou et une sardine (qui sont tous deux des poissons osseux). De même, il semblerait logique qu'un mérou et un requin soit considérés comme plus semblables qu'un mérou et un mammifère car mérou et requin sont tous deux des poissons. Lors du calcul de similarité, Wu et Palmer proposent donc de prendre en compte à la fois la profondeur des concepts comparés et la proximité relative de leur père commun C_0 (i.e. le concept subsumant le plus spécifique commun aux deux concepts). Le calcul de similarité associé est le suivant :

$$sim_{Wu\_Palmer}(C_1, C_2) = \frac{2 * Depth(C_0)}{Depth(C_1) + Depth(C_2)}$$

De cette façon, à nombre égal de relations de subsomption séparant deux concepts, la similarité entre deux concepts A et B dont le père commun est plus spécifique que celui de C et D sera plus élevée. Sur l'exemple de la figure 2.4, on retrouve les comportements recherchés :

$$sim_{Wu\_Palmer}(Merou, Sardine) = \frac{2*3}{4+4} = 0.75$$

$$sim_{Wu\_Palmer}(Osseux, Cartilagineux) = \frac{2*2}{3+3} \approx 0.67$$

$$sim_{Wu\_Palmer}(Merou, Requin) = \frac{2*2}{4+4} = 0.5$$

$$sim_{Wu\_Palmer}(Merou, Mammifere) = \frac{2*1}{4+2} \approx 0.33$$

L'inconvénient principal de ces mesures de similarités réside dans l'hypothèse forte selon laquelle toutes les relations de subsomption de l'ontologie interviennent dans une même

proportion au cours du calcul de similarité. Toutefois, le niveau de granularité des relations de subsomption peut être variable au sein d'une même ontologie. Dans l'exemple, on voit que la distinction faite entre poisson cartilagineux et poisson osseux au niveau du poisson est plus fine que celle faite entre baleine, dauphin et phoque au niveau des mammifères marins : de fait, pour ce dernier cas, l'ontologie "omet" les niveaux de distinction intermédiaires de cétacé (pour la baleine et le dauphin) et de pinnipède (pour l'otarie). Certaines approches comme celle de [Richardson et Smeaton, 1995] ont cherché à contourner la difficulté en émettant l'hypothèse que les concepts d'une partie dense de l'ontologie seraient mutuellement plus proches que les autres. Toutefois, l'exemple de la figure 2.4 va justement à l'encontre de cette supposition. Nous considérons en effet que les ontologies de domaine et/ou de tâche sont orientées par un besoin applicatif précis [8], et qu'elles ne modélisent qu'une partie des connaissances du domaine et/ou de la tâche, à savoir celles utiles pour l'application. Dans l'exemple précédent, on peut supposer, selon l'application visée, qu'il n'était pas utile de représenter les concepts de cétacé et de pinnipède.

Les mesures par calcul d'information commune Supposant cette valeur quantifiable, plusieurs mesures essaient d'évaluer l'information partagée par deux concepts afin de mesurer leur similarité. D'un point de vue purement théorique, l'information propre à un concept se retrouve au niveau de ses attributs et des relations sémantiques qu'il entretient avec les autres concepts de l'ontologie. Toutefois, mesurer la similarité de deux concepts à travers ce type d'informations présuppose un degré de formalisation élevé de l'ontologie qui n'est pas nécessairement garanti en pratique [Rodriguez, 2000]. Plusieurs approches cherchent donc à donner une approximation de l'information partagée par deux concepts en utilisant des techniques moins contraignantes. Les premiers travaux dans ce sens, initiés par [Resnik, 1995], fondent leur estimation sur la notion de contenu en information (CI) d'un concept. Le CI correspond en fait à la probabilité moyenne de détecter un concept dans un document. Pour un concept donné, Resnik l'évalue par la proportion d'occurrences de termes désignant le concept ou un de ses sous-concepts. Formellement, on a :

$$CI(C) = -log\frac{\sum_{t \in\ word(C \cup subConcepts(C))}\ n_{occ}(t)}{\sum_{w \in T}\ n_{occ}(w)}$$

avec $t \in\ word(C)$ indiquant que t peut désigner C, $n_{occ}(t)$ le nombre total d'occurrences d'un terme t sur l'ensemble des documents et T le lexique associé à l'ontologie. On constate tout d'abord que le concept le plus général de l'ontologie aura ainsi un CI nul et qu'en parcourant dans la profondeur un "chemin" de subsomption, les concepts auront un CI croissant (le numérateur croît nécessairement quand la profondeur diminue et la fonction $-log(x)$ est décroissante). Ce comportement correspond bien à l'intuition selon laquelle un concept abstrait est moins informatif qu'un concept plus spécifique. Si l'on reprend l'exemple des vertébrés marins et que l'on considère que chacun des concepts intervient 10 fois dans les documents de la base de recherche, on obtient des valeurs de CI indiquées sur la fig 2.5.

On peut noter que l'approche de Resnik est sensible au degré de polysémie des termes du domaine : en effet, dans le calcul du CI d'un concept C donné, Resnik prend en compte toute occurrence de terme pouvant désigner C sans se soucier de savoir si dans le contexte,

[8] Nous avancerons de façon détaillée des arguments étayant cette thèse en 3.1.

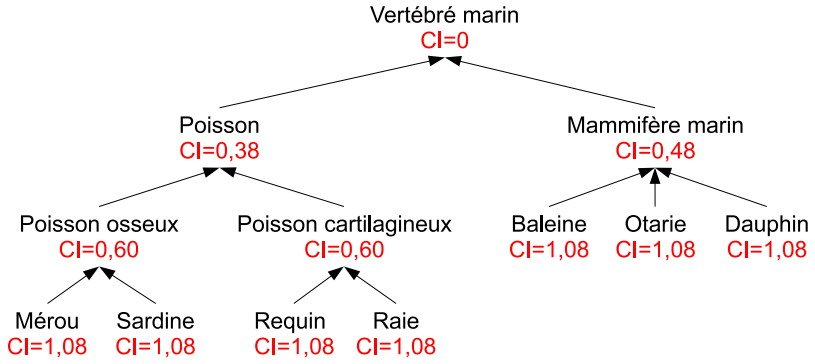

Figure 2.5 — Valeur du Contenu d'Information pour la taxonomie partielle des vertébrés marins

elle ne désigne pas un autre concept (au cas où le terme aurait plusieurs sens). A partir de là, deux approches sont possibles pour gérer cet effet : soit disposer d'un module de désambiguïsation appliqué à chaque occurrence de terme (fidèle à l'idée de départ mais difficile et coûteux en temps de traitement), soit adapter le calcul de CI en donnant moins d'importance aux termes polysémiques (moins fidèle mais plus simple).

La similarité entre deux concepts correspond, selon Resnik, au CI commun aux deux concepts, à savoir le subsumant commun dont le CI est le plus élevé :

$$sim_{Resnik}(C_1, C_2) = max_{\ C \in (Ancestor(C_1) \cap Ancestor(C_2))} CI(C)$$

avec $\forall i\ C_i \in Ancestor(C_i)$. D'après nos constatations précédentes, cette similarité correspond alors, dans le cas d'une hiérarchie sans héritage multiple, au CI du concept commun subsumeur le plus spécifique (déjà mentionné pour les mesures liées à la distance "graphique"). On remarque qu'en utilisant des informations liées aux occurrences des concepts dans les textes, l'approche de Resnik permet d'interpréter les différences de CI entre un concept et son père comme un poids associé au lien taxonomique correspondant : $w_{link(C_1,C_2)} = |CI(C_1) - CI(C_2)|$

On peut néanmoins voir deux limites à cette similarité. Tout d'abord, contrairement à la plupart des autres mesures, elle n'est pas normée : ses valeurs peuvent varier entre 0 et l'infini. Même si dans le cas d'une RI, la similarité ne représente qu'un outil permettant le classement relatif de documents en réponse à une requête, il est tout de même préférable qu'une mesure de similarité vérifie l'égalité

$$\forall (C_1, C_2)\ tq\ C_1 \neq C_2,\ sim_{theo}(C_1, C_1) = sim_{theo}(C_2, C_2)$$

Dans ce sens, on pourrait songer à n'utiliser que la probabilité d'occurrence d'un concept sans la moduler d'une fonction logarithmique, mais [Resnik, 1995] rapporte des résultats moins probants avec cette solution. Le second problème rencontré par la similarité de Resnik est lié au fait que sa valeur est complètement indépendante de la position relative des deux concepts en dehors de la position du subsumeur commun le plus spécifique (ainsi on a le résultat contre-intuitif : $sim_{Resnik}(Merou, Requin) = sim_{Resnik}(Poisson, Osseux)$).

Pour pallier ces manques, les travaux de [Lin, 1998] proposent une mesure dont la forme rappelle fortement celle de Wu et Palmer (au détail près que ceux-ci manipulent unique-

ment des données relatives à la profondeur d'un concept). Cette similarité prend la forme suivante :

$$sim_{Lin}(C_1, C_2) = \frac{2 * CI(C_0)}{CI(C_1) + CI(C_2)}$$

avec C_0 subsumeur commun à C_1 et C_2 le plus spécifique. On a alors bien :

$$\forall C \; sim_{Lin}(C, C) = 1$$

$$sim_{Lin}(Merou, Requin) = \frac{2 * CI(Poisson)}{CI(Merou) + CI(Requin)} \approx 0.35$$

$$sim_{Lin}(Poisson, Osseux) = \frac{2 * CI(Poisson)}{CI(Poisson) + CI(Osseux)} \approx 0.78$$

Les mesures mixtes Les mesures mixtes, comme leur nom l'indique, essaient de combiner les deux approches vues précédemment. Pour cela, leur principe consiste à calculer le chemin le plus court entre deux concepts en tenant compte du poids respectif de chaque lien traversé. Dans les recherches de [Jiang et Conrath, 1997], ce poids est calculé à partir de plusieurs facteurs :

- la **différence de contenu en information** entre le père et le fils[9] est révélatrice de son importance en termes de distance (plus le fils contient d'informations supplémentaires, plus il est éloigné de son père)
- la **profondeur du lien** renvoie à l'intuition selon laquelle les différences entre fils et père sont plus fondamentales sur les étages supérieurs du réseau formé par les liens de même nature. Par exemple, pour la relation hypéronymique, la différence entre un vertébré marin et un poisson est plus importante qu'entre un poisson osseux et une sardine.
- la **densité locale de ramification** au niveau du concept père permet d'estimer si la partie de l'ontologie concernée par la relation est relativement dense, auquel cas le poids du lien traversé est moindre car moins informatif (voir l'hypothèse mentionnée pour les mesures fondées sur le parcours de liens).
- la **nature sémantique du lien** permet d'affecter des poids différents à des relations selon l'utilité de leur type (hypéronymie, méronymie, autre ...) pour le rapprochement.

La formule globale proposée par Jiang et Conrath pour le calcul du poids d'un lien avec pour domaine C_p et codomaine C_c est la suivante :

$$w(C_c, C_p) = (\beta + (1 - \beta)\frac{E_{avg}}{E(C_p)})(\frac{depth(C_p) + 1}{depth(C_p)})^{\alpha} \; [IC(C_c) - IC(C_p)] * T(C_c, C_p)$$

avec $E(C)$ le nombre de relations partant de C, $T(C_i, C_j)$ le poids du lien reliant C_j à C_i, $\alpha \geq 0$ et $0 \leq \beta \leq 1$ deux paramètres variables. La distance du plus court chemin entre C_1 et C_2 ($SP(C_1, C_2)$) est alors

$$D_{Jiang\_Conrath}(C_1, C_2) = \sum_{C \in SP(C_1, C_2)} w(C, parent(C))$$

[9] Ici, nous ne cherchons en aucun cas à présumer de la nature du lien entre les deux concepts. Le terme "père" désigne le concept domaine de la relation tandis que le terme "fils" fait référence au codomaine de la relation.

A partir de cette formule, la similarité s'exprime simplement par

$$sim_{Jiang\_Conrath}(C_1, C_2) = 2 * D_{max} - D_{Jiang\_Conrath}(C_1, C_2)$$

La principale critique que nous pouvons faire de cette approche réside dans l'imprécision dont font preuve les auteurs quant à l'applicabilité de leur méthode pour des relations non taxonomiques. En effet, si leur mesure semble prévoir et permettre ce genre de calculs, Jiang et Conrath ne l'appliquent concrètement qu'au cas simplifié d'une taxonomie (avec $\forall(C_c, C_p)\ T(C_c, C_p) = 1$). De plus, comme le fait remarquer [Thieu *et al.*, 2004], le calcul à la manière de Resnik de la différence de contenu informationnel entre deux concepts n'est valable que s'il existe un rapport de subsomption entre eux : le CI d'un concept spécifique prend en compte à la fois la quantité d'informations héritées du concept plus général et la quantité d'informations propres au concept ; si les deux concepts ne partagent pas les mêmes traits taxonomiques, le calcul "à la Resnik" donnera des résultats incorrects.

La prise en compte de la nature sémantique d'une relation ontologique Après avoir décrit la tentative de [Jiang et Conrath, 1997] de prise en compte du type des relations sémantiques pour le calcul de similarité sémantique, nous jugeons pertinent de nous interroger sur l'intérêt des relations non taxonomiques pour déterminer la proximité en intension de deux concepts. En effet, dans l'absolu, il n'est pas évident qu'il existe un intérêt à rapprocher deux concepts relativement éloignés dans la hiérarchie taxonomique (en quelque sorte différents "par essence"). En étendant l'exemple des vertébrés marins à l'ensemble du monde aquatique, on peut imaginer une ontologie dans laquelle existent des relations de prédation entre animaux. Même si le calmar s'avère être une proie de choix pour le mérou, il peut sembler étrange à un utilisateur qui recherche des informations sur le mérou de voir le SRI lui retourner des documents traitant du calmar. On voit donc que la prise en compte d'une relation non taxonomique dans le calcul de similarité sémantique n'a de sens que si elle fait écho à un besoin de l'utilisateur. Ainsi, si celui-ci cherche à connaître la place du mérou dans la chaîne alimentaire, alors il sera logique que la relation de prédation soit utilisée au cours de l'appariement entre la requête et les documents.

D'un point de vue pratique, il est capital d'analyser de façon plus détaillée l'interaction entre relations taxonomiques et relations "transverses" au cours du calcul de similarité. Ainsi, les travaux de [Mazuel et Sabouret, 2007] soulignent le fait que la généralisation que les auteurs souhaitent apporter à la mesure de Jiang et Conrath pose le problème de l'unicité de chemin. En effet, le passage d'une structure arborescente (la taxonomie) à une structure de graphe entraîne la perte de cette propriété. Si plusieurs chemins existent, ils ne sont toutefois pas tous sémantiquement acceptables : par exemple, le poumon est une partie du dauphin, celui-ci mange des sardines, mais même si ce chemin semble être le plus court entre les concepts Poumon et Sardine, il ne correspond à aucune réalité et il serait aberrant de l'envisager. Mazuel et Sabouret proposent alors de considérer comme acceptable tout chemin composé en priorité d'un nombre indéfini de relations transverses d'un type commun et suivi d'un nombre indéfini de relations de subsomption. Ils définissent alors leur propre mesure de distance à partir de celle de Jiang et Conrath :

$$dist_{ont}(C_1, C_2) = min_{t \in \mathcal{C}, X \in \mathcal{R}}\{TC_X * (\frac{|sp_X(C_1, t)| - 1}{|sp_X(C_1, t)|}) + dist_{Jiang\_Conrath\_simple}(t, C_2)\}$$

avec TC_X étant le poids associé aux relations de type X, $sp_X(C_1, t)$ le plus court chemin entre $C1$ et t via des relations de type X. Dans [Mazuel et Sabouret, 2008], les auteurs mettent leur mesure à l'épreuve en utilisant WordNet et en comparant leurs résultats à une référence humaine. Ils rapportent des résultats encourageants sur un sous-ensemble de paires de mots à comparer mais mentionnent aussi l'aspect trop strict de leur critère de correction sémantique pour un chemin.

2.2.2.3 Appariement sémantique entre réseaux d'instances

Jusqu'ici, nous avons peu mis l'accent sur le fait que pour une RI sémantique, une requête, du fait qu'elle est interprétée, s'avère une source de données bien plus riche qu'une simple juxtaposition de mots-clés (cas d'un processus de RI classique). Suite à l'étape d'indexation sémantique, nous considérons en effet qu'un texte (ou, de façon symétrique, une requête) est annoté par des instances de concepts, mais aussi par des relations sémantiques entre ces instances. Outre les techniques de calcul de similarité conceptuelle, il est donc légitime de s'intéresser à des moyens d'évaluer la proximité sémantique d'une requête avec un document en fonction de la nature des concepts mis en jeu, mais aussi de leur organisation en réseau(x) par le jeu de relations sémantiques. Ci-après, nous présentons et analysons un ensemble d'approches qui tiennent compte de la présence de relations sémantiques pour le calcul de proximité dans le cadre d'un moteur de recherche sémantique.

Mécanisme d'activation propagée La méthode d'activation propagée est une technique relativement générique et peut s'appliquer à plusieurs domaines dans lesquels les informations (ainsi que leurs relations mutuelles) sont représentées sous la forme de réseaux (réseaux bayésiens, réseaux sémantiques ...). Le principe consiste à affecter des valeurs numériques d'activation (ou poids) à un ensemble de nœuds initiaux, puis à simuler leur propagation progressive dans l'ensemble du réseau grâce aux liens existant entre les nœuds. Un coefficient d'atténuation (pouvant dépendre de plusieurs facteurs comme le type de lien traversé, la distance minimale aux nœuds initiaux ...) permet d'affecter aux nœuds du réseau des valeurs d'activation inversement proportionnelles à la distance les séparant des nœuds initiaux, ce qui permet de garantir que la propagation se termine après un certain nombre d'itérations.

Les travaux présentés dans **[Rocha *et al.*, 2004]** utilisent l'activation propagée sur une ontologie de domaine dans le but d'enrichir automatiquement la requête d'un utilisateur à un moteur de recherche classique. De cette façon, les auteurs cherchent à améliorer le rappel, l'activation propagée permettant, dans ce cas, de retrouver tout document avec des références à des concepts proches de ceux retrouvés dans la requête (sans pour autant posséder nécessairement de référence directe à ceux-ci). Dans un premier temps, l'article présente plusieurs mesures pour déterminer l'importance relative d'une relation sémantique de type R entre deux instances de concept dans l'ontologie :

- la mesure par regroupement (*cluster*) évalue, pour une relation entre les instances de concept I_j et I_k, la proportion d'instances associées à I_j qui sont aussi en relation avec I_k dans le modèle ; cette mesure permet de rendre compte de l'intuition selon laquelle deux instances qui possèdent de nombreuses relations en commun sont d'autant plus

proches.
- la mesure de spécificité, inversement proportionnelle à la racine carrée du nombre de relations de type R pointant sur I_k, permet de modéliser le fait qu'une relation sémantique est d'autant plus importante qu'elle est peu entourée.
- la mesure hybride, qui combine les deux mesures précédentes en les multipliant, est annoncée par les auteurs comme étant a posteriori la plus efficace.

L'algorithme principal proposé commence par associer une valeur d'activation non nulle à toute instance retrouvée dans la requête (via une phase d'indexation classique). Cette valeur est renseignée en entrée du système en fonction du poids du nœud dans la requête, calculé pendant l'étape de RI (cf 2.1.3.3). L'algorithme est ensuite appliqué itérativement et dans l'ordre décroissant des valeurs d'activation des instances retrouvées :
- de façon à éviter toute propagation incontrôlée, le nœud traité doit éventuellement vérifier plusieurs contraintes (si elles ont été définies) : le type du nœud ne doit pas faire partie d'une liste d'exclusion, le nombre maximal de nœuds pouvant être traversés ne doit pas être dépassé, le nœud ne doit pas être relié à plus d'un certain nombre de nœuds.
- pour une relation donnée entre deux instances, la valeur transmise au voisin dépend du facteur d'atténuation, de l'importance absolue de la relation (fixée initialement par un ingénieur de la connaissance selon son type), de son importance relative (voir plus haut), et du facteur d'atténuation.
- le nœud courant est placé dans la liste des résultats, il ne sera plus parcouru mais sa valeur d'activation peut être modifiée après application de l'algorithme à d'autres nœuds.
- le nœud voisin activé, s'il reçoit une valeur non nulle, est ajouté à la liste des nœuds à parcourir.

L'algorithme s'arrête lorsqu'il n'existe plus aucun nœud à parcourir ou que la liste des résultats atteint une certaine taille paramétrable. Rocha et ses collègues considèrent la présentation à l'utilisateur des nœuds résultats comme une fin en soi, mais nous pouvons facilement imaginer que disposer d'une indexation sémantique de la requête plus riche permet à terme de retrouver plus de documents potentiellement intéressants.

Appariement de graphes conceptuels La théorie des graphes conceptuels (GC) est un formalisme général de représentation de connaissances développé par [Sowa, 1984] et fondé sur des considérations linguistiques, psychologiques et philosophiques. Même si les GC ne font pas partie des standards ontologiques actuels, il a été démontré dans plusieurs études comme [Corby *et al.*, 2000] ou [Dieng-Kuntz et Corby, 2005] qu'ils partagent de nombreuses caractéristiques avec les modèles RDF(S) et qu'une correspondance formelle peut être mise en place entre ces deux langages. De ce fait, plusieurs travaux appliquent les résultats existant sur les GC (avec notamment leurs mécanismes de projection) au cas de l'ontologie.

Les recherches présentées dans **[Zhong *et al.*, 2002]** s'intéressent ainsi aux GC pour mettre en place un processus de RI sémantique sur un domaine particulier. En guise de ressource sémantique, les auteurs utilisent le réseau WordNet associé à une ontologie du domaine. L'article décrit principalement une technique permettant de calculer la similarité globale entre un GC traduisant une requête de l'utilisateur et un GC représentant les annota-

tions sémantiques associées à un document textuel. Pour ce faire, il s'appuie sur une mesure de similarité entre concepts et une entre relations sémantiques. La mesure inter-conceptuelle présentée est comparable à la mesure de Wu et Palmer (présentée en 2.2.2.2) en ce que :

- deux concepts frères de profondeur N dans la taxonomie seront plus proches que deux frères de profondeur inférieure (i.e. plus généraux),
- deux concepts frères seront plus éloignés l'un de l'autre que chacun des deux de son père respectif (i.e. son hypéronyme direct).

Pour la similarité entre deux relations sémantiques, les auteurs en construisent une de type booléen :

- similarité maximale si la relation de la requête subsume la relation du document,
- nulle sinon.

Zhong et ses collègues combinent ces deux mesures dans une mesure asymétrique[10] de similarité entre deux GC symbolisant la requête et le document à comparer. Cette mesure est calculée de façon récursive à partir de la similarité conceptuelle entre chacun des deux nœuds initiaux. En effet, l'approche présuppose pour la requête (respectivement pour le document) l'existence d'un élément principal, renseigné au moment de la saisie de la requête (resp. découvert pendant une phase d'indexation sémantique adaptée). La formule de similarité est la suivante :

$$Sim_g(i^0_{req}, i^0_{doc}) = w(C^0_{req}) * sim_c(C^0_{req}, C^0_{doc}) + Max_{map_k}(\sum_j w(rel^j_{req}) * sim_r(rel^j_{req}, map_k(rel^j_{req})) * Sim_g(i^{rel^j_{req}}_{req}, i^{map_k(rel^j_{req})}_{doc}))$$

$$\text{avec } w(C^0_{req}) + \sum_j w(rel^j_{req}) = 1 \text{ (poids du nœud et des relations en partant)}$$

Les entités i^0_{req} et i^0_{doc} désignent les entrées initiales des deux GC, C^0_{req} et C^0_{doc} symbolisent leurs types respectifs, la fonction map_k apparie toute relation sémantique associée au nœud courant de la requête à une relation sémantique associée au nœud courant du document à comparer, et enfin $i^{rel^j_{req}}_{req}$ correspond au nœud associé par la relation rel^j_{req} au nœud courant de la requête.

Zhong et ses collègues se fondent sur la mesure récursive Sim_g pour l'algorithme de leur moteur de recherche sémantique, qui peut se résumer de la façon suivante :

1. Indexation sémantique de la requête
2. Transformation (grâce à un logiciel dédié) des annotations en un graphe conceptuel d'entrée C_0
3. Pour tout document ayant C_0 ou un hyponyme pour entrée de graphe, calcul de la similarité entre les deux graphes conceptuels avec Sim_g
4. Ordonnancement des documents par similarité de graphes décroissante et présentation à l'utilisateur

[10] Cette asymétrie provient de l'observation suivante : pour une requête mettant en jeu un concept C, il est logique de remonter systématiquement les documents indexés par un ou plusieurs hyponymes de C (de façon simplifiée, ce sont des cas particuliers de C) ; le contraire n'est pas systématique.

En examinant la structure récursive de la formule de similarité, on pourrait craindre une complexité algorithmique de l'ordre de celle du problème NP-complet d'appariement de sous-graphes maximaux. Toutefois, en considérant que les cycles dans une ontologie sont assez rares pour ne pas être pris en compte et en contraignant l'appariement sur une paire de nœuds initiale, l'article démontre que la complexité maximale de l'approche est en $O(n^4)$ avec n le nombre d'arcs intervenant dans la requête.

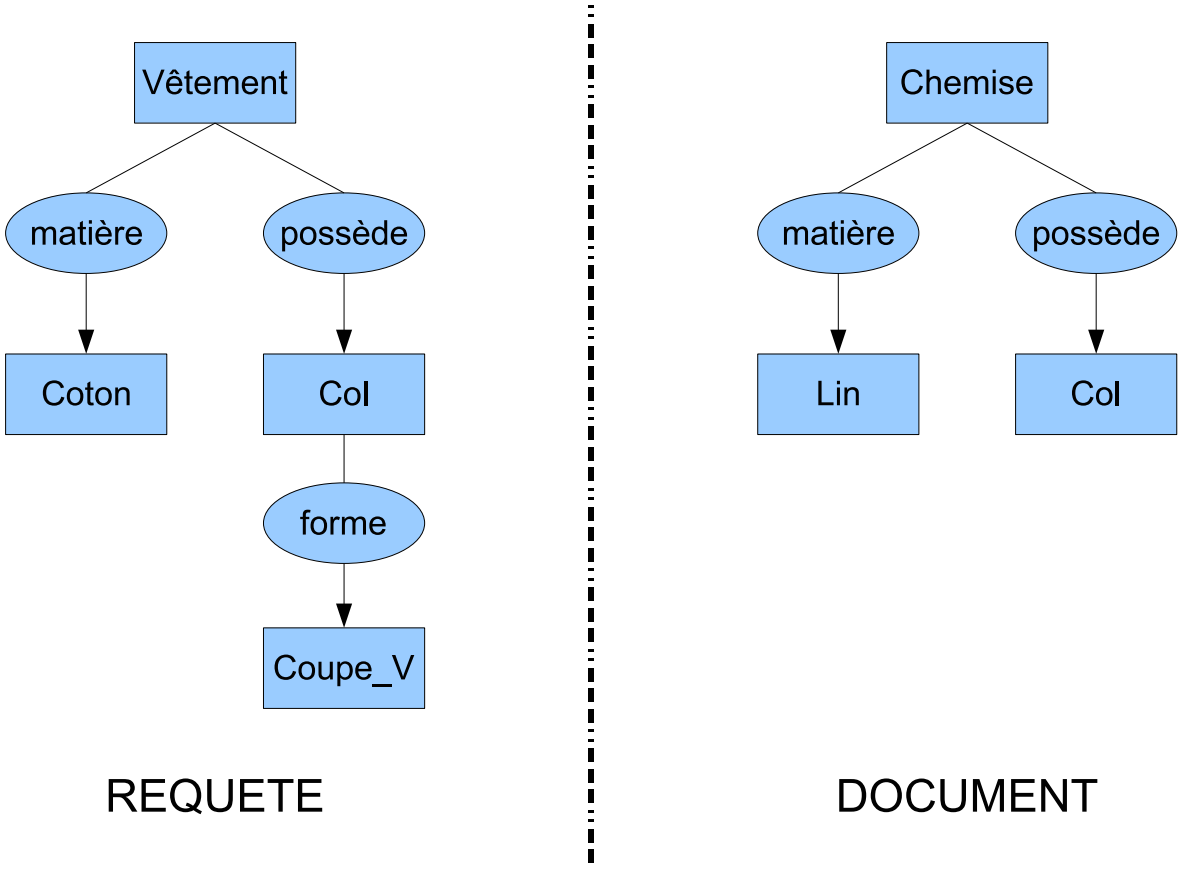

Figure 2.6 — Exemple de graphes à apparier

Dans leur algorithme d'appariement, les auteurs gèrent notamment l'absence de relation sémantique appariable dans un document par rapport à la requête en comparant le sous-graphe problématique de la requête à un sous-graphe par défaut construit artificiellement. Pour illustrer ce point, prenons l'exemple de l'appariement des deux graphes de la figure 2.6. Lorsque l'algorithme arrive au calcul de la similarité entre les deux sous-graphes de sommet initial `Col`, il ne trouve pas de sous-graphe associé au document pointant par une relation compatible avec `forme` vers un concept proche de `Coupe_V`. Pour remplacer l'information manquante [11], l'algorithme ajoute artificiellement une relation `forme` reliée au concept `Coupe_Col` hypéronyme de `Coupe_V`, avant de continuer le calcul en mesurant la similarité entre ces deux concepts. Cet exemple met à nouveau en avant l'asymétrie de la mesure de similarité développée puisque dans la situation contraire (i.e. seul le document

[11] Tout col de chemise a nécessairement une coupe spécifique.

possède une information sur la forme du col), ce sous-graphe serait ignoré pour le calcul.

Le rapport **[Corby *et al.*, 2005]** présente la fonctionnalité de recherche sémantique approchée du moteur Corese. Ce logiciel, qui possède un langage formel de requête spécifique [Corby *et al.*, 2004], accepte en entrée des requêtes sous la forme d'une conjonction de triplets de la forme `sujet-prédicat-objet`. Par exemple, la requête suivante cherche les noms de toutes les personnes ayant rédigé une thèse, ainsi que le titre de celle-ci :

```
?p rdf:type kmp:Person
?p kmp:name ?n
?p kmp:author ?doc
?doc rdf:type kmp:Thesis
?doc kmp:Title ?t
```

Dans son fonctionnement traditionnel, Corese utilise des opérations de projection liées au formalisme des GC pour retrouver des documents contenant les informations sémantiques cherchées dans la requête. Pour qu'une projection exacte de requête réussisse sur un document, il est nécessaire que les appariements des nœuds des deux graphes soient compatibles (i.e. qu'une instance d'un concept C de la requête soit appariée avec une instance de C ou d'un de ses hyponymes), ainsi que les relations existant entre les nœuds de chaque ensemble (i.e. s'il existe dans la requête une relation R entre les instances de concept i_1 et i_2, alors il doit exister une relation R ou une sous-propriété de R entre les instances du document appariées à i_1 et i_2).

Dans leurs travaux, Corby et ses collègues s'intéressent à la relaxation de ces contraintes de façon à pouvoir obtenir des résultats approchés à une requête suite à une recherche infructueuse. Ils définissent une mesure de distance sémantique fortement similaire à celle de [Zhong *et al.*, 2002], qui, pour deux concepts "frères", décroît avec leur profondeur dans la hiérarchie taxonomique. En outre, ils souhaitent prendre en compte de façon spécifique la relation `rdfs:seeAlso` qui peut relier deux concepts ou même deux relations sémantiques : comme sa présence atteste de l'existence d'un lien de parenté non explicité entre les deux entités reliées, les auteurs adaptent la définition de leur distance sémantique pour qu'elle soit plus faible entre deux concepts s'ils sont connectés par la relation `rdfs:seeAlso`.

L'article distingue deux types d'approximation : l'approximation ontologique et l'approximation structurelle. Pour la première, il définit un opérateur de projection approchée qui préserve l'adjacence et l'ordre sur les arêtes (i.e. deux relations appariées ont nécessairement le même nombre de variables et la i^{eme} variable de la première sera appariée à la i^{eme} variable de la seconde), qui autorise l'appariement de nœuds suffisamment proches selon la distance sémantique préalablement définie et qui permet l'appariement de relations connectées par la relation `rdfs:seeAlso`. L'approximation structurelle a été envisagée par Corby et ses collègues de façon à gommer les divergences de structure relationnelle globale entre les annotations sémantiques de la requête et celles du document. En effet, ils jugent que l'utilisateur n'a pas forcément connaissance des relations potentielles entre deux concepts et peut souhaiter visualiser tout document contenant des instances de ces concepts, quelle que soit la (les) relation(s) sémantique(s) entre elles. Par exemple, l'utilisateur peut souhaiter connaître le nom de chercheurs ayant des affinités avec un domaine scientifique particu-

lier ; les personnes retournées par le système peuvent avoir été choisies parce qu'elles travaillent sur le domaine voulu, ou qu'elles encadrent un doctorant sur un sujet d'un domaine connexe ... Pour permettre un tel comportement du moteur Corese, les auteurs autorisent, dans le cas où la projection d'une relation sémantique R de la requête sur un document ne correspond à aucune relation indexée dans ce texte, que R puisse être appariée à un sous-graphe du document dont les nœuds aux extrémités peuvent être appariés avec les nœuds associés à R dans la requête. Comme on peut le voir dans l'exemple de la figure 2.7, les deux documents sont retournés par Corese par rapport à la requête : le document 1 correspond à un appariement de graphe exact, alors que le graphe correspondant au document 2 est une projection structurellement approchée de la requête.

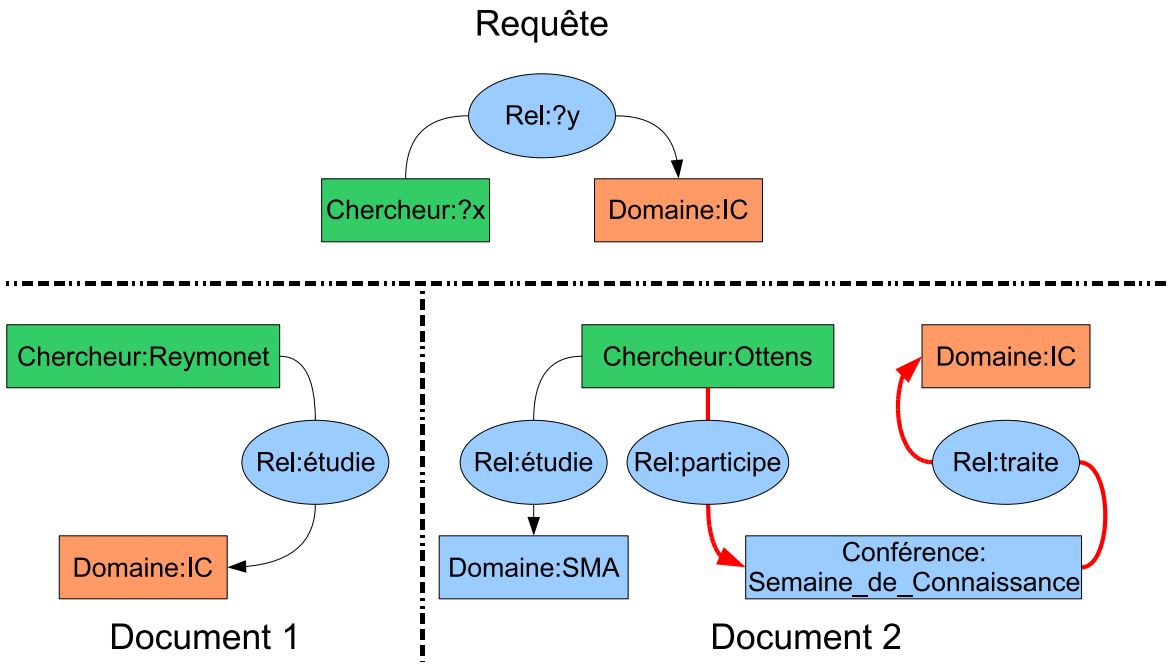

Figure 2.7 — Approximation structurelle dans Corese

Mesure de similarité pour une logique du premier ordre L'approche présentée dans **[Bisson, 1993]** et succinctement résumée dans **[Bisson, 2000]**, bien qu'elle concerne uniquement le calcul de similarité entre deux formules de la logique du premier ordre, s'avère intéressante pour nous : en effet, nous cherchons à évaluer la proximité dans une ontologie entre des conjonctions d'annotations sémantiques de la forme `sujet-prédicat-objet`, directement traduisibles en logique des prédicats. Pour mieux comprendre les mesures proposées par Bisson, reprenons un de ses exemples : soient les quatre prédicats `père`, `sexe`, `age` et `pays`, ainsi que les quatre constantes `paul`, `john`, `yves` et `ann`. Considérons les formules conjonctives suivantes :

$$F_1 = pere(paul, yves) \wedge sexe(yves, M) \wedge age(paul, 33) \wedge age(yves, 13) \wedge pays(paul, france)$$

$$F_2 = pere(john, ann) \wedge sexe(ann, F) \wedge age(john, 58) \wedge age(ann, 28) \wedge sexe(john, M)$$

Pour comparer Paul et John, une première méthode abordée par Bisson consiste à compter le nombre de prédicats dans lesquels les deux instances interviennent à la même position, rapporté au nombre maximal de prédicats faisant intervenir chaque instance. Ainsi, on obtient :

$$OSim(paul, john) = \frac{2}{3} \approx 0.67$$

(les 2 prédicats communs sont `père` et `age`, `paul` et `john` intervenant dans 3 prédicats chacun)
Comme chaque prédicat peut avoir une importance relative différente pour le calcul de similarité, Bisson propose de raffiner la mesure en la pondérant par les poids de chaque relation mise en jeu. Ainsi, si les poids des prédicats sont les suivants :

$$w(pere) = 5$$

$$w(sexe) = 3$$

$$w(age) = 2$$

$$w(pays) = 1$$

alors on obtient :

$$OSim'(paul, john) = \frac{5+2}{Max(5+2+1\ ;\ 5+2+3)} = 0.7$$

Bisson souhaite pousser plus loin la réflexion : si la prise en compte des prédicats communs à deux entités permet d'évaluer en partie leur proximité, l'auteur souligne l'intérêt intuitif de prendre en compte la similarité entre les autres paires d'entités intervenant dans la relation. Ainsi, `paul` et `john` seront d'autant plus similaires que leurs enfants se ressembleront. De même, deux personnes seront d'autant plus semblables que leurs âges seront proches. A partir de ces remarques, Bisson propose une mesure de similarité absolue entre deux entités x et y, calculée en fonction de la similarité des entités qui leur sont reliées par des prédicats. Si nous rapportons cette mesure au domaine des ontologies en OWL où les entités correspondent à des instances de concepts et les prédicats soit à des propriétés d'objet (ou relations entre concepts) soit à des propriétés de type de données (ou propriétés), nous obtenons la formule suivante :

$$\begin{aligned} ASim(x,y) = Mapp_k[& \sum_{rel_i}^{communes} w(rel_i) * (1 + RSim(x^k_{rel_i}, y^k_{rel_i}))] \\ & + \sum_{prop_j}^{communes} w(prop_j) * VSim(prop_j(x), prop_j(y)) \end{aligned}$$

avec $w(rel_i)$ et $w(prop_j)$ les poids associés à la relation i et à la propriété j, x_{rel_i} l'instance reliée à x via rel_i , $prop_j(x)$ la valeur de la propriété j pour l'entité x. Au cas où plusieurs appariements d'entités seraient possibles (e.g. on peut apparier la formule $rel_1(x,a)$ soit à $rel_1(y_1,b)$ soit à $rel_1(y_2,c)$), la fonction $Mapp_k$ permet de sélectionner l'appariement maximisant la similarité absolue finale. La mesure de similarité $VSim$ calcule une proximité relative entre deux valeurs, elle dépend donc du type de ces valeurs (chaîne de caractères, entier,

décimal, type énuméré ...). La mesure *RSim* entre deux instances est directement calculée à partir de la similarité absolue :

$$RSim(x,y) = \frac{ASim(x,y)}{Max(\sum_{rel_i}^{x} w(rel_i) + \sum_{prop_j}^{x} w(prop_j)\ ;\ \sum_{rel_i}^{y} w(rel_i) + \sum_{prop_j}^{y} w(prop_j))}$$

On voit donc que les mesures de similarité absolue et relative sont récursives croisées : pour connaître la similarité entre deux entités, il faut connaître celle entre chaque paire d'entités voisines (à travers le même prédicat) respectives, et vice versa. Pour résoudre ce système d'équations, Bisson propose d'employer la méthode itérative de Jacobi [Golub et van Loan, 1983].

Tandis que la similarité absolue permet de favoriser l'appariement des objets les plus importants, la similarité relative permet de favoriser l'appariement des objets les plus similaires. Comme les deux aspects sont importants pour l'objectif final (i.e. l'appariement de deux ensembles d'entités), Bisson définit une unique mesure de similarité *FSim*, produit des deux premières :

$$FSim(x,y) = ASim(x,y) * RSim(x,y)$$

Il en résulte la similarité globale *ESim* suivante :

$$ESim(E_1,E_2) = \frac{Max_k[\sum_{x^k,y^k}^{E_1,E_2} FSim(x^k,y^k)]}{Max[\sum_{x}^{E_1}(\sum_{rel_i}^{x} w(rel_i) + \sum_{prop_j}^{x} w(prop_j))\ ;\ \sum_{y}^{E_2}(\sum_{rel_i}^{y} w(rel_i) + \sum_{prop_j}^{y} w(prop_j))]}$$

Le numérateur de *ESim* est calculé en sommant la valeur *FSim* des couples (x,y) de $E_1 \times E_2$ appariés par ordre décroissant.

Forces et faiblesses des approches présentées Nous revenons ici sur les avantages et inconvénients de chacune des méthodes que nous venons d'aborder, dans le but ultérieur de nous inspirer de celles que nous aurons jugées intéressantes et de repérer les points à améliorer.

Les travaux de **[Rocha *et al.*, 2004]**, avec le mécanisme d'activation propagée, permettent la prise en compte de plusieurs instances de concepts simultanément dans une requête ou un document. En effet, il suffit pour cela d'initialiser les nœuds adéquats avant le début de la propagation. L'approche a également pour avantage d'aller dans le sens de l'intuition en favorisant les instances étant atteintes par plusieurs chemins de la requête. Enfin, même si les relations sémantiques potentielles entre les instances de la requête ne constituent pas des données d'entrée, elles sont prises en compte au cours de la circulation des flux d'activation. Toujours à ce propos, il semble capital que l'ontographe ait au préalable réglé le poids absolu de chaque relation, afin d'éviter que la propagation ne se fasse sans contrôle. Cependant, les contraintes supplémentaires qui peuvent être définies dans cet objectif semblent trop arbitraires (nombre maximal de nœuds parcourus, flot interrompu sur un nœud avec trop de relations) pour garantir un résultat final intuitif vis-à-vis de la requête utilisateur. Enfin, un défaut inhérent à la méthode de Rocha et de ses collègues réside dans l'impossibilité de sélectionner des documents contenant des instances de concepts qui sont pourtant taxonomiquement proches de ceux sollicités dans la requête : de fait, l'activation propagée

permet de prendre en compte les instances de relations transverses, mais sans considération des liens hypéronymiques implicites.

L'approche défendue par **[Zhong *et al.*, 2002]** possède un certain nombre de qualités : en plus de permettre la comparaison de réseaux d'instances de concepts, elle autorise l'approximation d'une instance par une instance d'un concept différent (contrairement à la méthode précédente). De plus, les auteurs démontrent une complexité de calcul très raisonnable, garantie par la nécessité de désigner l'instance de concept centrale à la requête et par l'élagage de branches inutiles dans l'algorithme itératif (au niveau des paires de relations sémantiques sans lien de parenté). L'inconvénient principal de la méthode correspond à une gestion insatisfaisante du calcul de similarité dans le cas où un sous-graphe de la requête n'a aucun équivalent au niveau du document comparé (cf fig. 2.6). En effet, le choix d'ajouter artificiellement au graphe du document la relation manquante vers une instance factice de l'hyperonyme direct du concept présent dans la requête est assez discutable : si l'on reprend l'exemple de la figure 2.6, un document identique mais mentionnant une forme de col ronde sera forcément moins similaire à la requête que le premier document car la similarité conceptuelle telle que les auteurs la calculent entre un concept et son hypéronyme est nécessairement plus grande qu'entre ce concept et un de ses frères. Selon le nombre de formes possibles pour un col et la similarité relative entre un col en V et un col rond, il peut paraître étonnant de préférer systématiquement un document ne mentionnant aucune forme de col plutôt qu'un document mentionnant une forme de col différente mais qui pourrait plus proche du col mentionné dans la requête que la plupart des autres cols.

Un des intérêts des travaux de **[Corby *et al.*, 2005]**, comparativement aux autres approches décrites ici, concerne la prise en compte spécifique de la relation `rdfs:seeAlso`. Elle permet de rapprocher deux entités sémantiques (concept ou relation), et ce quelle que soit leur position relative dans l'ontologie. Cette technique peut notamment s'avérer un moyen temporaire (avant maintenance [12]) et efficace de mémoriser dans l'ontologie une évolution dans la similarité entre deux entités. L'inconvénient majeur de la méthode présentée concerne l'approximation structurelle : les auteurs permettent à l'utilisateur de spécifier explicitement une requête approchée au niveau de sa structure, et ce n'est que dans cette situation qu'est autorisé l'appariement avec des documents dont la structure relationnelle des annotations sémantiques diffère. Il peut être pourtant intéressant de concevoir un mécanisme d'approximation structurelle valable quelles que soient les instances de relations sémantiques présentes dans la requête.

Enfin, l'approche développée par **[Bisson, 1993]** peut être vue comme le pendant de [Zhong *et al.*, 2002] : avec une idée de départ similaire, Bisson n'émet aucune contrainte sur la forme des graphes, et notamment aucune sur l'existence d'un nœud central autour duquel calculer la similarité sémantique globale. Toutefois, ceci se fait au détriment de la complexité des calculs : pour chaque comparaison, le système conçu par Bisson doit systématiquement résoudre un système d'équations non nécessairement linéaires. Néanmoins, le point qui nous semble le plus discutable est lié à l'ordre induit des résultats de similarité. En effet, les recherches de Bisson se fondent sur une hypothèse implicite qui n'est selon

[12] Nous considérons en effet qu'une ontologie utilisée pour une RI sémantique a été au préalable construite et/ou adaptée selon des critères de différenciation taxonomique directement liés à la tâche d'appariement sémantique.

nous pas toujours assurée : d'après Bisson, lorsque deux personnes sont comparées, le fait qu'elles soient toutes deux pères (ou mères) d'un enfant les rend plus proches mutuellement que chacune avec une tierce personne sans enfant. Si l'idée semble intuitive, nous avons la certitude qu'elle n'est vraie que dans certaines circonstances particulières et qu'elle ne saurait constituer le cas général : nous avons tendance à rapprocher une personne ayant des enfants à quelqu'un ayant aussi des enfants plutôt qu'à une personne sans enfant car nous jugeons implicitement qu'elles partagent une expérience commune. Cependant, ce jugement n'est valable que dans le cas où la similarité ontologique prend explicitement en compte ce point commun. Dans le cas contraire, il est nécessaire de calculer au préalable la similarité sémantique entre les enfants, ce qui peut amener, si tout oppose les deux enfants, à considérer que chacun des pères (ou mères) est plus proche d'une personne sans enfant que de l'autre père (ou mère).

2.2.3 Dynamisme des ontologies et gestion des conséquences

Même si nous avons déjà abordé en partie cette problématique en 1.3.4, nous souhaitons revenir plus en détail sur la conséquence de la nature évolutive des ontologies en termes de gestion des annotations sémantiques. Malgré l'importance de telles considérations dans le contexte du Web Sémantique et de l'utilisation croissante, la littérature correspondante est relativement peu abondante.

Si les premières études en rapport se fondaient sur un parallèle entre les ontologies et les bases de données (BD), les travaux de [Noy et Klein, 2004] démontrent que les deux types d'artefacts sont bien différents : contrairement aux BD, les ontologies incluent leur propre sémantique, elles sont construites et maintenues selon un processus fortement décentralisé. De plus, une ontologie se fonde sur un modèle beaucoup plus riche que celui d'une BD (avec notamment le mécanisme des méta-classes), et elle a plus souvent tendance à importer/être importée par d'autres ontologies, ce qui rend une opération d'évolution bien plus délicate (nécessité de propagation). Les instances de la base de faits ainsi que l'ensemble des artefacts de l'ontologie doivent éventuellement être modifiés de façon à éviter l'apparition d'inconsistances logiques. En outre, les auteurs soulignent la nécessité de distinguer deux types d'opérations d'évolution sur une ontologie :

- une opération **élémentaire** correspond à une action atomique non décomposable (e.g. modification d'un label, ajout/suppression d'un concept ou d'une relation ...)
- une opération **composite** est constituée d'une liste d'opération élémentaires (e.g. fusionner deux concepts, affecter le domaine d'une propriété à un père du concept domaine courant ...)

A priori, on pourrait être tenté de considérer la catégorie de changement composite comme un ajout de sucre syntaxique mais ce n'est pas le cas. En effet, modéliser une opération d'évolution composite comme une simple succession d'opérations élémentaires pourrait entraîner une perte inutile d'informations : par exemple, si, pour affecter le domaine d'une propriété d'attribut à un de ses pères taxonomiques, on supprimait le lien avec la propriété et on en créait un nouveau pour le père, on perdrait les valeurs d'attribut affectées aux instances de l'ancien domaine. L'article de [Noy et Klein, 2004] envisage alors systématiquement les conséquences en termes de préservation d'informations que peut avoir chaque opération

de changement sur l'ontologie. Même si l'analyse proposée s'avère pertinente, les auteurs n'envisagent pas d'autre stratégie d'évolution que celle centrée sur la préservation des données : par exemple, dans le cas d'ontologies de grande taille, il peut être plus intéressant de choisir une stratégie fondée sur la rapidité des traitements associés, auquel cas, après une modification de l'ontologie, il sera peut-être plus efficace de supprimer certaines instances plutôt que d'envisager pour elles une phase de redistribution.

Un autre point intéressant pour la gestion des conséquences d'évolution concerne son degré de synchronisme avec chaque changement. En effet, on peut envisager soit une phase systématique d'adaptation de l'ontologie et des systèmes utilisateurs après chaque modification opérée par l'utilisateur, soit une phase ultérieure qui peut se dérouler avec ou sans traces des opérations effectuées précédemment. Ce dernier cas est logiquement le plus difficile puisque le système doit au préalable comparer l'ontologie après modification avec son état initial de façon à retrouver les évolutions subies par chaque artefact de l'ontologie et pouvoir ensuite corriger toute situation d'inconsistance et synchroniser toutes les constructions se servant de l'ontologie avec sa version actuelle. Dans cette situation, l'approche choisie se fonde généralement sur des méthodes de fusion d'ontologies. Si les travaux de [Stojanovic *et al.*, 2002] privilégient une gestion des conséquences simultanée à chaque changement de l'ontologie, les recherches de [Luong, 2007] préfèrent envisager tous les cas de figure, arguant que dans le contexte de partage du Web Sémantique, il n'est pas rare que les ingénieurs maintenant l'ontologie ne soient pas les (seuls) utilisateurs de cette représentation.

Nous abordons maintenant de façon plus détaillée la contribution de [Luong, 2007], qui se préoccupe de la gestion des annotations sémantiques en réaction à une évolution ontologique. Pour l'auteur, une annotation sémantique prend la forme d'un triplet $< sujet, predicat, valeur >$ classique en RDF. Selon une démarche heuristique, Luong associe à chaque stratégie d'évolution proposée par [Stojanovic *et al.*, 2002] une ou plusieurs stratégies de correction des annotations. Par exemple, dans le cas d'une suppression d'un concept C, si la stratégie suivie pour les instances concernées consiste à les rattacher au père C_0 de C, Luong propose de ne garder pour C que les annotations le concernant et faisant intervenir des relations pour lesquelles C_0 fait partie du domaine ou co-domaine. L'ancien type des instances concernées (C) est alors remplacé par C_0. Dans le cas où une trace des modifications subies par l'ontologie n'est pas disponible, Luong procède en deux temps : il détecte les annotations inconsistantes par le fait qu'elles violent une ou plusieurs contraintes de consistance (définition dans l'ontologie des concepts et propriétés intervenant dans les annotations, compatibilité entre le type d'une instance et le domaine ou co-domaine de la propriété mise en jeu dans l'annotation). Dans un second temps, un ensemble d'heuristiques permet d'émettre des hypothèses plus ou moins fortes sur la nature de l'évolution ayant amené une inconsistance au niveau de certaines annotations sémantiques. L'approche de Luong nous paraît intéressante car elle évite de recourir à des techniques trop complexes du domaine de la fusion d'ontologies. Toutefois, nous nous interrogeons quant à l'exhaustivité annoncée par Luong de ses règles. Il nous semble en effet difficile d'envisager tous les cas de figure possibles pour amener une annotation dans un état inconsistant, d'autant plus que l'effet combiné de plusieurs changements pourrait potentiellement avoir les mêmes conséquences qu'un changement donné (ce qui accroît le risque de supprimer à tort certaines

annotations).

En guise de conclusion de cette partie, nous soulignons à nouveau le manque de littérature concernée par l'influence d'une évolution d'ontologie sur les annotations sémantiques associées. L'approche proposée par [Luong, 2007] nous a paru relativement attrayante par sa simplicité dans le cas où des traces indiquent précisément l'évolution suivie par l'ontologie. Dans le cas contraire, il nous faudrait plutôt nous pencher sur certaines techniques de la fusion d'ontologies. Puisque le contexte industriel nous le permet, nous opterons en 4.1.2.3 pour une solution consistant à propager systématiquement et immédiatement tout changement dans l'ontologie au niveau des annotations sémantiques. De cette façon, les index seront toujours synchronisés à l'état courant de l'ontologie.

2.2.4 Revue d'outils disponibles

2.2.4.1 Critères retenus pour la comparaison des outils

Pour conclure sur cette section, nous présentons quelques outils issus de travaux en relation avec la Recherche d'Informations Sémantique. Avant toute chose, il est nécessaire que nous donnions l'ensemble des critères que nous avons retenus afin de comparer les logiciels. Pour leur définition, nous nous sommes inspirés de l'état de l'art présenté dans [Uren *et al.*, 2006] et des dimensions d'analyse mentionnées dans [Maynard, 2005]. Nous avons séparé les critères en 4 groupes, les deux premiers correspondant aux entrées de tout SRI sémantique (i.e. documents et ontologie), les deux suivants aux deux étapes fondamentales dans ces systèmes (i.e. indexation sémantique et appariement sémantique).

Pour l'ontologie, nous nous préoccuperons de connaître les formats acceptés par chaque outil (Sont-ils propriétaires au logiciel ? Sont-ce des standards ?) ainsi que le choix de modélisation pour la relation entre termes et concepts. De plus, comme nous considérons une ontologie comme un artefact susceptible de changer selon l'évolution du domaine modélisé et la façon dont il est utilisé, il est capital de savoir si un outil comporte la fonctionnalité permettant d'éditer la structure et les entités de la RTO. Le cas échéant, comme nous avons pu le voir en 2.2.3, les annotations antérieures à une modification risquent de se trouver dans un état d'inconsistance avec la ressource ontologique. Un outil de RI sémantique permettant de la faire évoluer a donc tout intérêt à envisager un mécanisme de gestion approprié.

Au niveau des documents sur lesquels portera le processus de RI, il est intéressant de connaître les formats pris en charge par les outils, sachant que dans le contexte du Web Sémantique, les outils doivent soit prendre en compte les formats standards, soit proposer des traitements permettant de s'y ramener. Ensuite, tous les logiciels analysés ne travaillent pas sur la même échelle : alors que certains sont destinés à être utilisés sur des bases documentaires relativement restreintes (quelques milliers de textes), d'autres se spécialisent dans le traitement de larges quantités de données, voire ambitionnent de pouvoir être appliqués sur l'ensemble des textes du Web visible. Enfin, de même qu'avec l'ontologie, la modification d'une ressource textuelle peut faire apparaître des inconsistances dans les annotations correspondantes : la zone précédemment annotée aura pu disparaître, être décalée, et de nouvelles zones susceptibles d'être indexées ont pu être ajoutées.

Pour la suite des critères, nous avons préféré les dissocier selon qu'ils concernent l'annotation sémantique ou l'appariement sémantique entre la requête et les documents. En effet, nous avons pu constater qu'une bonne partie des outils que nous allons citer ne se préoccupe que de la première étape, laissant le soin à d'autres logiciels d'utiliser à leur guise les annotations créées. Pour le processus d'indexation sémantique, nous nous intéressons au degré d'automatisation de chaque outil (nous n'avons pas retenu les approches purement manuelles) ainsi qu'à la nature des méthodes employées (Instanciation de règles manuelles, Machine Learning ...). D'autre part, nous avons relevé pour chaque outil le mode et le format de stockage des annotations. De fait, une bonne interopérabilité entre différents logiciels exige que les entrées/sorties respectent un standard relativement pratiqué et que les résultats de chaque processus soit facilement consultable. C'est une des raisons pour lesquelles la plupart des logiciels se réclamant du Web Sémantique préfèrent stocker les annotations sur des serveurs adaptés.

Enfin, nous avons dégagé seulement deux critères pour la phase d'appariement entre une requête et un ensemble de documents : la forme de la requête et le modèle de RI sous-jacent. On notera qu'à l'inverse des logiciels d'annotation sémantique, nous n'avons pas retrouvé une aussi grande diversité d'outils axés vers l'appariement sémantique. On peut tenter d'expliquer ce fait par le besoin généralisé de maîtriser au préalable la phase préliminaire d'indexation.

Nous aurions pu rajouter quelques critères supplémentaires, comme l'ergonomie de l'outil, sa gestion des groupes d'utilisateurs (confidentialité), sa rapidité d'indexation sémantique ... Toutefois, nous avons souhaité nous concentrer dans un premier temps sur des points plus prioritaires et étroitement liés à la réalisation pratique de la tâche.

2.2.4.2 Comparaison des logiciels

Dans cette partie, nous rapportons, pour les trois premiers types de critères, le comportement d'outils soit orientés vers une application d'annotation sémantique (tab. 2.2), soit remplissant d'autres objectifs (tab. 2.3). Le tableau 2.4 rassemble tous les outils précédemment cités qui incorporent une solution pour l'appariement sémantique. Nous traiterons le cas des logiciels suivants : Melita [Ciravegna *et al.*, 2002], Armadillo [Ciravegna *et al.*, 2004], OntoMat [Handschuh *et al.*, 2002], C-PANKOW [Cimiano *et al.*, 2005], AeroSWARM [Corcho, 2006], SemTag [Dill *et al.*, 2003], KIM [Popov *et al.*, 2003], OntoPop [Amardeilh, 2007], H-TechSight [Stollberg *et al.*, 2004], SMORE [Kalyanpur *et al.*, 2005], SmartWeb [Wahlster, 2004], CoSWEM [Luong, 2007] et OntoSeek [Guarino *et al.*, 1999]. Pour les outils d'annotation sémantique, nous pouvons faire plusieurs remarques :

- la grande majorité d'entre eux se fonde sur des normes ou des standards, que ce soit pour la représentation des ontologies, des annotations ou les formats pris en charge pour les documents. Dans le contexte du Web Sémantique et des Web services, il est de fait devenu capital de pouvoir manipuler des logiciels interopérables.
- si certaines privilégient des besoins utilisateurs de haut niveau en fixant une ontologie pour l'annotation (e.g. KIM ou SemTag), il n'en va pas de même pour toutes les approches. De fait, afin de satisfaire des besoins spécifiques différents, d'autres laissent à l'utilisateur le soin de choisir l'ontologie adéquate pour son objectif (e.g. Melita, Arma-

dillo). Pour s'adapter aux exigences spécifiques d'une ontologie non sélectionnée au préalable, certains outils de ce type misent alors sur une interaction entre le système et l'utilisateur (e.g. OntoPop). Ceci leur garantit un bon compromis entre le temps d'intervention d'un humain et la précision des structures indexantes obtenues.

- la partie lexicale d'une ontologie peut être représentée différemment selon les outils considérés. On notera toutefois qu'une large proportion d'entre eux, à l'exception notable de SmartWeb, a tendance à la considérer comme nécessairement subordonnée aux concepts. Nous nous opposerons à cette position et aborderons ce problème plus en détail dans le chapitre suivant.
- parmi les différentes plate-formes d'annotation sémantique, l'intégration d'un éditeur d'ontologie n'est pas systématique. Soit les approches considèrent que les ontologies envisagées ne sont pas susceptibles d'évoluer, point difficilement défendable sur un domaine de technologie, soit elles préfèrent sous-traiter la tâche d'édition à un logiciel extérieur. Cette position nous paraît toutefois inopportune puisqu'il est capital de pouvoir gérer les modifications à apporter aux annotations à la suite d'une évolution d'une ontologie. Deux outils se distinguent sur ce point : OntoPop et CoSWEM. Le premier choisit de réannoter systématiquement tous les documents indexés par des concepts ayant subi des évolutions. Toutefois, un changement opéré dans une ontologie peut avoir des répercussions subtiles à la fois sur la représentation sémantique mais aussi sur certaines annotations. C'est pourquoi CoSWEM se propose de surveiller précisément chacune de ces évolutions pour en répercuter les conséquences de façon appropriée sur la base d'annotations. Comme nous le verrons dans les chapitres à venir, nous nous inspirerons de cette approche tout en proposant de gérer sur le moment les conséquences potentielles de toute modification.
- en ce qui concerne le problème similaire de gestion des conséquences d'un changement dans un document indexé, nous avons préféré considérer que notre base à indexer aurait tendance à être enrichie mais que les anciens textes ne seraient pas modifiés. Quand bien même ceci arriverait, nous pourrions suivre une heuristique simple : dès qu'un document est détecté comme ayant subi des modifications, il est automatiquement réindexé par l'outil.
- au niveau du format de stockage, il s'avère que la quasi-unanimité des logiciels propose un enregistrement des annotations externe aux documents. Ce choix nous semble judicieux car il permet de traiter les documents sur lesquels on n'a pas de droit de modification, il évite le parcours systématique d'un texte pour connaître ses index et enfin il laisse la possibilité d'indexer simultanément un même document selon différents points de vue.
- les techniques employées pour indexer un document se dissocient globalement en deux catégories. Une première consiste à retrouver dans les documents le lexique directement associé aux concepts. La seconde met en oeuvre des patrons lexico-syntaxico-sémantiques (i.e. pouvant faire intervenir des termes, des éléments de syntaxe et/ou des entités ontologiques). Si certaines plate-formes implémentent une combinaison fixe et figée de ces techniques, d'autres à base d'apprentissage se fondent sur des statistiques pour faire évoluer lexique et/ou patrons d'extraction selon un cercle vertueux (des termes du domaine permettent de découvrir des régularités, qui

donneront naissance à de nouveaux patrons, qui permettront eux-mêmes d'enrichir le lexique du domaine ...). Dans l'absolu, il est difficile d'opter pour une solution en dehors de tout contexte car elles ont des comportements opposés : dans le cas d'un ensemble peu fourni de documents, il est plus difficile de repérer et/ou d'appliquer des patrons d'extraction (la redondance peut être faible). A l'inverse, l'utilisation du lexique seul s'avère relativement robuste, mais elle présuppose de détenir des ressources lexicales riches et adaptées à l'objectif annotatif (et donc plus le nombre de textes à indexer sera faible, plus il sera facile de mettre en oeuvre cette technique).

En termes de critères d'appariement sémantique, s'il est difficile de se prononcer sur les modèles suivis par les différents outils, nous pouvons tout de même constater que les choix possibles d'interface de requête pour l'utilisateur sont relativement nombreux : de la requête par mots-clés à la requête par composition de graphe en passant par la requête en langue naturelle. Pour sa simplicité et sa bonne acceptabilité par le grand public dans des SRI classiques, nous essaierons d'exploiter au mieux le potentiel des mots-clés.

CRITERES	MnM / Melita	Armadillo	OntoMat	C-PANKOW	AeroSWARM	SemTag	KIM	OntoPop
ONTOLOGIE								
Type	Libre	Libre	Libre	Libre	SUMO ou OpenCyc	TAP	KIMO	Libre
Formats	RDF(S) DAML+OIL	RDF(S)	DAML+OIL OWL	KAON OI OWL	OWL	RDF(S)	RDF(S) OWL Light	XTM, OWL
Terme	label	label	label	label	instance	Non Connu	instance d'1 classe lex.	lexiques séparés
Edition	Non	Non	OntoEdit	KAON	Non	Non	Non	Oui
Consistance avec annot°	Non	Non	Non	Non	Non	Non	Non	Réannotation automatique
DOCUMENTS								
Formats	HTML texte libre	HTML	HTML (PDF)	HTML texte libre	HTML	HTML	HTML	HTML texte libre
Nb docs	Important	Important	Important	Web	Important	Web	Important	Important
Consistance avec annot°	Non	Non	Perspective	Non	Non	Non	Non	Non
ANNOTATION SEMANTIQUE								
Degré d'automatisation	Semi-auto	Auto	Manuel Semi-auto	Auto	Auto	Auto	Auto	Semi-auto
Approche	ML supervisé (Amilcare)	String matching + Amilcare	ML supervisé (Amilcare)	ML non supervisé (patrons)	Règles	Règles	Règles	Règles + lexique
Stockage	Séparé et interne	Séparé	Séparé ou interne	Séparé	Séparé (serveur)	Séparé (serveur)	Séparé et interne	Séparé (serveur)
Formats	SGML/XML	RDF	RDF, OWL, F-Logic	RDF	RDF	RDF(S)	RDF(S)	RDF

Tableau 2.2 — Outils sémantiques à vocation annotative

CRITERES	H-TechSight	SMORE	SmartWeb	CoSWEM	OntoSeek
Objectif de l'outil	Veille techno + évo onto semi-auto	Création intuitive d'ontologie	SRI sémantique	Gestion d'évolution d'un SRI sémantique	SRI sémantique
ONTOLOGIE					
Type	Ontologie de domaine	NP [13]	Fixe (projet)	Libre	Sensus
Formats	DAML+OIL RDF	OWL	RDF(S), OWL	RDF(S)	NP
Terme	instance	(pré)concept	instance de classe ling.	NP	synset
Edition	Oui	Oui	Non	Oui	NP
Consistance avec annot°	Non	Non	Non	Heuristiques de propag°	NP
DOCUMENTS					
Formats	HTML	HTML	HTML, XML	NP (amont du système)	HTML
Nb docs	Limité	Limité	Web	Limité	Limité
Consistance avec annot°	Non	Edition docs sans gestion	Non	Perspective	Non
ANNOTATION SEMANTIQUE					
Degré d'automat°	Auto	Manuel Semi-auto	Auto	NP (entrée du système)	Auto
Approche	Règles JAPE	NP	ML non supervisé	NP	Comparaison lexique
Stockage	Séparé	Séparé (serveur)	Séparé	Séparé (serveur)	Séparé (serveur)
Formats	GATE	RDF	RDF	RDF	BD

Tableau 2.3 — Outils sémantiques autres

CRITERES	KIM	SmartWeb	CoSWEM	OntoSeek
RECHERCHE D'INFORMATIONS SEMANTIQUE				
Format requête	Mots-clés concepts instances	LN	IHM CORESE	Mots-clés

Tableau 2.4 — Outils de Recherche d'Informations Sémantique

[13] NP : Non Pertinent

2.3 Bilan

Dans ce chapitre, nous avons commencé par aborder en 2.1 la thématique de RI dans sa globalité, pour nous focaliser rapidement sur deux sous-processus essentiels, à savoir l'indexation et le calcul de pertinence d'un document vis-à-vis d'une requête. L'indexation consiste à extraire une représentation synthétique et caractéristique du contenu d'un document en vue de le retrouver ultérieurement. Pour mener à bien cette tâche, il est possible de repérer les principaux termes d'un document par l'application de plusieurs techniques de TALN et d'en mesurer l'importance par l'analyse de données statistiques (e.g. TF-IDF). Dans un second temps, selon les termes caractéristiques retenus, un document sera jugé plus ou moins pertinent pour une requête. Cette étape de comparaison dépend du modèle de recherche choisi, qui s'avère généralement de type booléen, vectoriel ou probabiliste.

Dans nos travaux, nous préférons nous situer dans un paradigme de RI sémantique, plutôt que de faire appel à des techniques s'appuyant en grande partie sur des mesures statistiques. En effet, celles-ci s'avèrent peu efficaces dans notre contexte d'étude : nous nous intéressons à mettre en œuvre un processus de RI sur des **petites collections de documents relativement concis et limités à un domaine spécifique**. Du fait de la portée réduite des connaissances abordées dans ces textes, nous examinons en 2.2 une approche de RI se fondant, non plus sur un ensemble de termes, mais sur une représentation explicite des connaissances présentes dans les documents.

Dans un tel cadre de recherche, nous considérons que les annotations issues d'un texte correspondent à l'expression d'objets du domaine éventuellement en relation. Concrètement, l'étape d'indexation sémantique se fonde sur une RTO du domaine et consiste alors à associer au texte traité un (ou plusieurs) réseau(x) d'instances de concepts reliées entre elles par des relations sémantiques. A travers des techniques de TALN, il est possible d'exploiter l'occurrence d'un terme de la RTO dans un document pour en déduire la présence d'une instance du concept désigné. Un des principaux avantages de la RI sémantique s'affiche au cours du calcul de pertinence entre requête et document : le calcul peut s'appuyer de façon fiable sur la structure de la RTO d'une part, et sur les informations disponibles dans les réseaux sémantiques issus de l'indexation de la requête et du document d'autre part. En 2.2.2, nous avons ainsi réalisé un état des lieux sur les mesures existantes de similarité entre deux concepts, entre deux instances de concepts et entre deux réseaux d'instances.

L'utilisation d'une ressource extérieure aux documents entraîne néanmoins des problèmes d'un genre nouveau : tout d'abord, les annotations sémantiques sont uniquement valides et interprétables tant que la RTO qui a été utilisée pendant l'indexation garde un état stable. Si elle évolue (e.g. suite à un progrès technologique dans le domaine modélisé), il se peut que l'interprétation sémantique de certains documents soit faussée. Comme nous l'avons vu en 2.2.3, la plupart des approches cherchant à résoudre ce problème théorique se sont heurtées à sa difficulté et ont préféré, de façon plus pragmatique, s'appuyer sur un ensemble d'heuristiques. Si la question reste entière, nous n'aborderons que peu ce point par la suite, et ce afin de pouvoir nous concentrer sur d'autres problématiques de recherche (à nos yeux) plus capitales :

– Quel formalisme adopter pour une RTO destinée à la RI ? (*cf 3.2*)

– Comment optimiser le processus de maintenance de cette RTO ? (*cf 4.1*)
– Quelle mesure de similarité choisir pour comparer deux réseaux d'instances ? (*cf 4.2*)

Enfin, nous avons pu constater en 2.2.4, à travers une revue de plusieurs outils liés à la RI sémantique, qu'il n'existait pas de logiciel qui puisse opérer les étapes d'indexation et de comparaison telles que nous les concevons. Cette contrainte nous amènera par la suite à implémenter nos contributions théoriques au sein du prototype TextViz (*cf 5.1*).

Deuxième partie

Contribution

3 Contribution au processus de construction ontologique

Dans ce chapitre, nous allons aborder nos contributions théoriques relatives au processus de modélisation d'une ontologie. Notre premier apport à l'Ingénierie des Connaissances consiste à montrer l'importance de la prise en compte de certains paramètres qui orientent de façon plus ou moins implicite la construction ontologique. Nous traitons en 3.1 d'une problématique identique à [Aussenac-Gilles *et al.*, 2002] ou [Bourigault *et al.*, 2004], que nous approfondissons en analysant en détail et par l'exemple les influences de la tâche (diagnostic), du domaine (industrie automobile) et de l'application (recherche d'informations). Par ailleurs, l'étape de formalisation s'avère fondamentale dans la modélisation d'une ontologie : selon le langage choisi, il sera plus ou moins aisé (voire même parfois impossible !) d'exprimer formellement certaines connaissances. C'est pourquoi, dans notre contexte de RI sémantique largement fondée sur la partie lexicale d'une RTO, nous nous sommes intéressé en 3.2 à la création d'un méta-modèle exprimé dans le langage standard OWL (de façon à garantir un certain degré d'interopérabilité) et capable de manipuler séparément un terme et un concept, afin de disposer d'une représentation terminologique plus riche (avec notamment les usages en contexte de chaque terme) que les approches classiques.

3.1 Tâche, domaine et application : influences sur le processus de modélisation de connaissances

Cette section a fait l'objet d'une publication : [Reymonet et al.*, 2006].*

Nous comptons montrer, par une analyse a posteriori du processus de modélisation d'une ontologie adaptée au projet MODE, comment certains facteurs influencent le processus de construction de RTO et son contenu. Dans un premier temps, nous définissons ces facteurs, à savoir les notions de tâche, domaine et d'application, en particulier dans le contexte de notre étude. À travers des exemples issus de notre expérience dans le domaine du diagnostic automobile, nous nous intéressons ensuite aux conséquences de ces paramètres sur le processus de modélisation d'une RTO. Pour notre analyse, nous avons suivi une méthode de construction de RTO proche de Terminae [Aussenac-Gilles *et al.*, 2003a], partant de l'analyse de textes, mais en accordant une place plus importante aux connaissances détenues par les experts du domaine. Nous ne détaillerons qu'ultérieurement la phase de construction de

l'ontologie de domaine (cf 5.1.1.3). Seuls nous intéressent dans cette section les paramètres orientant le processus de modélisation.

3.1.1 Définition des paramètres d'influence

La notion d'**application** renvoie à la prise en compte, au sein du logiciel qui sera développé, des besoins de l'utilisateur et plus généralement des raisons ayant motivé le processus de modélisation. Ce paramètre primordial influence de façon implicite certains choix faits en fonction de la tâche et du domaine. Dans notre cas d'étude, l'application correspond à la recherche d'information dans une base d'expériences d'une part, à la mise en collaboration de plusieurs méthodes de raisonnement d'autre part.

Le concept de **tâche** est à prendre au sens classique utilisé dans CommonKADS [Schreiber et de Hoog, 1999]. La tâche définit les buts que doit réaliser le système ainsi que la ou les méthodes de résolution mises en œuvre pour les atteindre. Pour le projet MODE, la tâche correspond au diagnostic automobile. Le module OBIR n'effectue pas une résolution de problème mais une recherche d'information dans ces fiches. Le modèle de la tâche n'a donc pas besoin d'être développé pour décrire le raisonnement au delà de ce qui est explicité par la structure des fiches de la base d'expériences (décrite plus précisément en 5.1.1.1). On peut voir un modèle simplifié de la tâche sur la figure 3.1 : à partir d'un symptôme (révélé par un indicateur ou un comportement anormal signalé par le garagiste et recherché dans le champ " constatation " des fiches), il s'agit d'identifier un élément défaillant (mentionné dans le champ " diagnostic ").

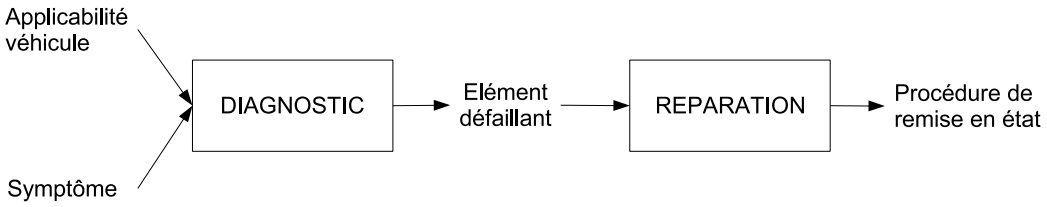

Figure 3.1 — Modèle de la tâche du garagiste

L'élément défaillant est, dans le meilleur des cas, un composant (si le diagnostic a complètement abouti), au pire une (ou plusieurs) prestation(s) suspecte(s)[1]. Le processus de réparation n'est pas géré directement par les modules de MODE, cette étape est laissée en partie à la charge du garagiste, qui peut toutefois s'aider des schémas de montage disponibles sur l'outil de diagnostic.

Enfin, la notion de **domaine** correspond à la sphère de connaissances que l'on cherche à modéliser. Dans le cadre de notre construction, il s'agit de certains savoirs liés à l'automobile et détenus par un garagiste. Le domaine est étroitement lié à la tâche : le modèle doit couvrir uniquement les connaissances que le garagiste utilise pour réaliser un diagnostic.

[1]Une prestation véhicule est un service rendu à l'utilisateur (comme la climatisation, l'essuyage de la lunette arrière ...) par le biais d'un système de composants.

3.1.2 Choix préliminaires

3.1.2.1 Méthode de construction

Comme le souligne [Bourigault *et al.*, 2004], il serait idéal que la personne en charge de la modélisation détienne un certain nombre de compétences sur le domaine, en ingénierie des connaissances, en linguistique et en informatique. Pratiquement, c'est en fonction de ses propres compétences, des ressources disponibles et des besoins applicatifs que l'analyste désigné (qui pourra éventuellement faire appel à des spécialistes complémentaires) adapte sa méthode de travail pour construire une RTO.

Dans le projet MODE, la méthode utilisée se rapproche fortement de celle employée dans [Aussenac-Gilles *et al.*, 2003b] : comme les connaissances de l'analyste dans le domaine du diagnostic automobile étaient faibles, il a fallu compenser en utilisant au mieux les connaissances présentes dans une sélection de textes, les connaissances de spécialistes du diagnostic automobile, et enfin les connaissances élémentaires que nous avons en tant qu'utilisateurs de véhicules pouvant tomber en panne. L'utilisation combinée d'un outil d'analyse syntaxique (Syntex [Bourigault *et al.*, 2005]) et d'un logiciel d'analyse distributionnelle (Upery [Bourigault, 2002]) nous a permis d'étudier les formes sous lesquelles apparaissaient les concepts. Mais l'organisation hiérarchique des classes conceptuelles, implicite dans les textes, a été fournie par un expert.

3.1.2.2 Constitution du corpus

Pour la définition du corpus, nous nous situons dans l'acception de [Condamines, 2003], selon laquelle un corpus est " *une collection de textes [...] constituée à partir de critères linguistiques ou extra-linguistiques pour évaluer une hypothèse linguistique ou pour répondre à un besoin applicatif* ".

En vue de la construction d'une RTO, le critère primordial pour constituer un corpus est la prise en compte de l'application pour laquelle est bâtie la RTO. Dans notre cas, comme un des objectifs consiste à rechercher des informations dans une base d'expériences, nous avons sélectionné l'ensemble des fiches de réparation comme corpus.

L'influence de la nature de l'application se traduit par la prise en compte de l'utilisateur. En effet, les fiches retenues comme corpus de départ sont rédigées par des experts du diagnostic automobile mais devront être comparées à des requêtes posées par des garagistes. Il est donc indispensable de vérifier la bonne adéquation entre les concepts et la terminologie utilisés par chacun des deux groupes. Dans notre cas, une étude d'ergonomie menée auprès de concessionnaires a permis de montrer que les deux groupes manipulaient des termes et des concepts très proches. Il n'a donc pas été nécessaire de modifier le corpus pour adapter le vocabulaire aux futurs utilisateurs.

L'indexation sémantique d'une base de fiches évolutive nous a confronté à un autre problème intéressant : il nous fallait savoir si la RTO resterait adaptée aux nouveaux textes à indexer. Dans un domaine technologique comme celui de l'automobile, de nouveaux documents viennent régulièrement enrichir la base de fiches initiale. Le domaine couvert par le corpus évolue donc avec l'apparition de nouveaux concepts et de nouveaux termes relatifs à

de nouveaux symptômes, à l'intégration de technologies de pointe ou à la mise sur le marché de nouveaux types de véhicules. Nous avons donc dû concevoir des mécanismes de prise en compte de l'évolutivité du corpus pour le processus d'indexation de la base d'expériences. La conception d'un procédé de maintenance d'ontologie est abordée de façon détaillée par la suite en 4.1.

Dans le but d'adapter la RTO aux autres modes de raisonnement, nous avons pensé accroître le corpus avec des textes portant sur des connaissances différentes de celles disponibles dans les fiches de réparation. En ceci, les documents de conception semblaient une piste intéressante. Cependant, du fait de leur structure non textuelle (sous forme de schémas) et de leur focalisation sur les composants (peu utiles pour nos besoins), ceux-ci se sont avérés inadéquats.

Enfin, si les fiches de réparation semblaient suffire pour construire la RTO en question, les principes de modélisation d'une ontologie exigent d'avoir accès à des connaissances définitoires. Comme la base d'expériences s'appuie sur des connaissances opératoires, elle permet la construction d'un modèle de la tâche et du domaine, mais pas l'élaboration d'une ontologie. Ne disposant pas de documents appropriés (e.g. documents de formation au métier de garagiste), nous avons décidé d'utiliser un second type de ressource, à savoir un expert du domaine. Celui-ci nous permettrait de plus de valider par ses connaissances la structure ontologique bâtie à partir du corpus.

3.1.3 Structuration de l'ontologie

3.1.3.1 Concepts centraux et rôles de l'ontologie

Une fois le corpus d'étude constitué, le choix des concepts essentiels à modéliser peut se faire sur la base de l'usage des termes en corpus, et à l'aide de connaissances a priori sur le domaine et la tâche. Dans le cas de notre étude, la particularité du corpus n'est pas liée aux termes utilisés, mais à la structuration des fiches de réparation, qui reflète la démarche du garagiste. Les champs de la fiche directement utilisés dans l'application, " constatation " et " diagnostic " correspondent l'un à la description des *symptômes*, l'autre à celle de *composants*, de *tests d'état*, d'*hypothèses de panne* (cause possible d'un symptôme) et de *normes* si l'on reprend le vocabulaire de modélisation " à la KADS " du diagnostic [Schreiber *et al.*, 1993]. La structure de la fiche comporte en plus la notion de *réparation* dans le champ " remède après-vente ". Selon le modèle de la tâche associée au module OBIR (*fig. 3.1*), seuls nous intéressent les rôles de *symptôme*[2] (en entrée) et de *prestation* (en sortie). En effet, nous adoptons le point de vue d'un garagiste en phase de diagnostic de panne. Nous avons choisi d'intégrer ces concepts dans l'ontologie selon une modélisation discutée dans la partie suivante.

Dans le cadre d'une stratégie descendante de construction, ces rôles nous ont permis de tracer les grandes lignes de l'ontologie à réaliser et de définir les types de concepts du domaine à identifier. De plus, les fiches étant très structurées, chacune des parties fait référence à des concepts de types particuliers : les champs " constatation " et " diagnostic " ont ainsi été exploités en priorité pour trouver des concepts caractérisant les *symptômes* et les *prestations*.

[2]Nous définissons ici plus formellement cette entité comme une observation non conforme à un modèle de fonctionnement nominal.

De plus, le choix des concepts a été guidé par la nécessité de disposer d'un ensemble d'index judicieux pour la recherche de fiches, capables d'isoler chaque type de panne. Dans cette optique, la notion de symptôme était un concept primordial, tant pour son pouvoir discriminant sur les différentes sortes de panne que pour sa fréquence d'utilisation dans le diagnostic automobile. Les concepts ont donc été identifiés en réponse aux contraintes imposées par la tâche et l'application ciblées.

3.1.3.2 Modélisation d'un concept

Au delà du choix des concepts et des rôles, la tâche oriente la manière de les décrire. En effet, la notion de symptôme est un des rôles principaux dans un raisonnement de diagnostic. Dans la plupart des méthodes de résolution applicable au diagnostic [Schreiber et de Hoog, 1999], le symptôme doit permettre, en référence à un modèle de fonctionnement de l'objet à diagnostiquer, de repérer un écart par rapport au fonctionnement attendu et d'identifier un composant en panne. Ici, la représentation de la notion de symptôme doit donc aiguiller vers des dysfonctionnements visibles du véhicule ou de parties de véhicule, à savoir des prestations. Du fait de la forte structuration des fiches, on s'attend à trouver les symptômes dans le champ " constatation ". Les termes identifiés dans ce champ grâce à l'extracteur de terme (Syntex et Upery) renvoient à des concepts spécifiques de type problème, **prestation** et *contexte* :

- " la **climatisation** ne fonctionne pas *au ralenti* "
- " allumage du témoin **ABS** "
- " *en accélération*, **le moteur** fume noir "

La notion de symptôme est représentée en faisant appel à ces concepts. Il est difficile de trouver un critère ontologique de différenciation qui organiserait les symptômes : toute différence entre deux symptômes renvoie à la comparaison des prestations et/ou des problèmes associés. Nous avons donc modélisé le rôle de symptôme sous la forme de concept défini (au sens de la logique de description). Formellement, nous le définissons comme un problème affectant une prestation dans un certain contexte (*fig. 3.2*). Ainsi, les concepts définis nous semblent adaptés à la représentation d'un rôle comme symptôme.

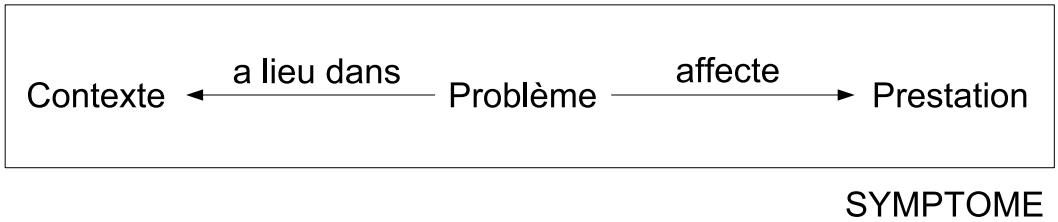

Figure 3.2 — Représentation du symptôme

3.1.3.3 Organisation hiérarchique des concepts

L'application pour laquelle est développée la RTO influence évidemment la structuration des concepts entre eux. Nous soulignons ici un impact de l'application moins évident, à

savoir celui sur le degré de décomposition de l'ontologie. En effet, il faudra plus ou moins différencier certains concepts d'un point de vue hypéronymique, de manière à servir au mieux les besoins applicatifs.

Pour que la RTO assure la collaboration de plusieurs méthodes de raisonnement au sein du projet MODE, nous avons dû détailler avec précision les différents niveaux de prestations véhicule. L'objectif poursuivi était double : fournir une vision du domaine proche de celle d'un réparateur en garage (et rendre l'utilisation de l'ontologie plus "intuitive"), mais aussi avoir la possibilité d'isoler certains sous-systèmes suspects au cours d'un processus de diagnostic à base de modèles (MBR). Pour cela, nous nous sommes inspirés des approches [Chittaro *et al.*, 1993, Kitamura et Mizoguchi, 2004], pour lesquelles le véhicule est considéré comme un ensemble de systèmes réalisant des fonctions pour le conducteur ou les passagers. Dans ce cadre, la prestation telle qu'elle a été définie en 3.1.1 correspond à une macro-fonction.

Le module MBR du projet MODE essaie d'établir un lien entre les connaissances comportementales et structurelles sur un système d'une part (*comment fonctionne le système ?*) et les connaissances téléologiques d'autre part (*quel est le but du système ?*). Plus concrètement, ce module doit faire l'association entre des équations de fonctionnement d'un sous-système du véhicule et la prestation que celui-ci remplit pour le client. Pour cela, le module MBR doit proposer au garagiste une série de tests dans le but de parvenir à un diagnostic de la panne. Il faut donc que l'ontologie construite propose une structure hypéronymique des prestations suffisamment détaillée pour qu'on puisse les distinguer selon leur mode d'activation (essuyage avant par commande manuelle / essuyage avant automatique). Nous avons mis en place cette structure en nous appuyant sur des hiérarchies disponibles auprès de constructeurs automobiles et sur les prestations mentionnées dans les fiches.

A l'inverse, l'indexation conceptuelle des fiches ne nécessite pas une modélisation poussée des composants de fonctionnement qui entrent dans la réalisation de ces prestations. En effet, la recherche au sein des fiches ne se fait pas sur la base des pannes identifiées sur des composants mais sur celle des problèmes constatés sur des prestations. La structure de cette partie de l'ontologie s'en est trouvée simplifiée. Toutefois, si une fonctionnalité supplémentaire concernant les composants devait être ajoutée (e.g. associer un schéma électrique à un composant), il faudrait alors utiliser des critères de différenciation supplémentaires pour détailler à un degré de précision satisfaisant la sous-ontologie des composants.

3.1.4 Choix du langage de formalisation

Selon les contraintes, une ontologie peut être représentée de façon plus ou moins formelle [Gomez-Pérez *et al.*, 2004]. Il faut donc bien analyser les besoins pour lesquels est construite cette ressource afin de choisir un langage de description adéquat.

Dans notre projet, nous souhaitons que l'ontologie permette d'optimiser la recherche d'information en raisonnant sur les relations entre concepts ou leurs propriétés. En effet, en faisant des inférences sur certains objets du domaine, en l'occurrence ici sur les symptômes, une requête peut être reformulée et renvoyer à la recherche d'un ensemble plus large ou plus pertinent de concepts. Par exemple, on sait qu'une " fumée noire à l'échappement "

est symptomatique d'une " surconsommation de carburant ". Lorsque le garagiste spécifie un symptôme de surconsommation, il serait intéressant de lui demander s'il constate des fumées noires, ou de rechercher a priori des fiches traitant de l'un des deux symptômes. Ceci élargirait le champ de la prospection effectuée par le module OBIR dans les fiches.

De plus, les choix de modélisation faits préalablement peuvent imposer certaines contraintes supplémentaires sur le langage à utiliser. Dans le cadre de notre expérience, il est nécessaire de disposer d'un langage permettant de représenter les restrictions sur les relations sémantiques : du fait de notre choix de représenter un symptôme sous la forme d'un problème affectant une prestation (dans un certain contexte), nous devons pouvoir modéliser qu'un problème spécifique peut éventuellement n'affecter que certaines prestations (e.g. une fuite peut uniquement se déclarer sur le circuit de freinage, la motorisation, la climatisation ou une suspension hydraulique).

Pour ces deux raisons essentielles, nous avons choisi de manipuler l'ontologie résultante sous le format OWL-DL. En effet, parmi les trois principaux sous-langages de OWL disponibles, OWL-DL est le meilleur compromis entre expressivité et calculabilité [3] : OWL-Full possède une bonne expressivité mais n'est pas calculable tandis qu'OWL-Light, bien que calculable, ne comporte qu'un faible nombre de primitives. La première propriété que nous cherchons à vérifier nous contraint à abandonner OWL-Full (du moins pour un temps, voir 3.2.3) ; la seconde finit de nous orienter vers OWL-DL, puisque, à la différence d'OWL-Light, celui-ci permet de représenter une restriction de propriété :

```
<owl:Class rdf:ID="Engorgement">
  <rdfs:subClassOf rdf:resource="#Problème"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="affecte"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:ID="Motorisation"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Les restrictions de propriétés s'avèrent essentielles dans notre ontologie du diagnostic automobile car elles permettent d'exprimer le caractère spécifique de certains problèmes : tandis que certains peuvent a priori affecter n'importe quel type de prestation (e.g. `Absence_de_fonctionnement` ou `Fonctionnement_altéré`), d'autres problèmes ne peuvent survenir que sur un sous-ensemble précis des prestations (e.g. `Engorgement`, qui ne peut affecter que la `Motorisation`). Cette distinction démontre son utilité au cours du processus d'indexation sémantique, car elle évite notamment qu'une instance de problème spécifique ne soit incorrectement associée à une instance de prestation incompatible.

[3] Un langage ontologique est qualifié de calculable si, à partir d'une proposition dans ce formalisme, un raisonneur logique est capable de conclure en un temps fini sur sa véracité.

3.1.5 Impacts sur la terminologie

3.1.5.1 Spécificité terminologique

Comme nous nous plaçons dans un paradigme de construction de RTO à partir de textes, nous partons de l'hypothèse selon laquelle une partie importante des termes présents dans le corpus sont spécifiques du domaine concerné. Le tableau 3.1 nous conforte dans cette idée : on constate que parmi les 200 termes les plus fréquents, au moins 70% peuvent être reliés à des classes sémantiques spécifiques au domaine du diagnostic automobile (à savoir composant, contexte et prestation).

Une autre observation sur le corpus nous permet de mesurer l'influence du domaine sur la spécificité de la terminologie. En effet, si l'on s'intéresse à la profondeur moyenne à laquelle on trouvera le concept désigné par un terme donné (i.e. nombre de noeuds concepts traversés pour parvenir à lui depuis le haut de l'arborescence hypéronymique), on mesure une profondeur moyenne de 4, avec peu de variations selon la sous-ontologie à laquelle appartient le concept (*tableau 3.1*). Plus intéressant encore, 87% des termes pris en compte correspondent à une profondeur dans l'ontologie strictement supérieure à 3. Ces résultats pourraient s'interpréter en termes de technicité du domaine : dans les niveaux supérieurs de la RTO, les critères de séparation entre différents concepts proviendraient plus d'un besoin de classement que de réelles occurrences des termes correspondants dans le corpus.

Classe sémantique	Prop°	Exemples de termes	Prof_{moy} ds RTO
composant	31%	sonde de température, calculateur d'injection	4,08
contexte	20%	moteur coupé, régime moteur	non dispo [4]
problème	20%	bruit de claquement, mauvais fonctionnement	3,91
prestation	16%	feux de détresse, démarrage moteur	3,92
autre	13%	...	-

Tableau 3.1 — Répartition par classe sémantique pour les 200 termes les plus fréquents du corpus

3.1.5.2 Précision de la terminologie

Le domaine détermine également les termes à intégrer dans la RTO, et ce en fonction de la spécificité terminologique du corpus. En effet, indexer un corpus de langue générale supposerait de disposer d'une ressource très large et générique dont l'utilisation peut nécessiter une phase de désambiguïsation (due à la polysémie des termes). Au contraire, lorsque l'on se trouve sur un domaine particulier, il est indispensable de faire référence à une RTO spécialisée d'un domaine et plus précise : de nombreux termes polysémiques en l'absence de tout

[4] La structuration de cette partie de la RTO n'était pas terminée au moment des observations statistiques.

contexte prennent une sémantique plus précise sur un domaine particulier. Par exemple, le verbe intransitif " patiner " possède deux sens principaux dans le Trésor de la Langue Française (TLF)[5]. Le premier correspond à l'action d'évoluer sur la glace ou le bitume à l'aide de patins tandis que le second se rapporte à un dérapage par manque d'adhérence au sol. Ces définitions se distinguent notamment par la nature du sujet (personne ou objet). En examinant le corpus, nous constatons que le verbe " patiner " est utilisé uniquement avec des objets du domaine automobile : " une courroie patine ", " la poulie patine ", " le véhicule patine ", " l'embrayage patine "[6]. Pour notre RTO, ce verbe a donc un sens univoque (celui de la seconde définition) et ne renvoie qu'à un seul concept.

Dans un même ordre d'idée, afin de représenter fidèlement les usages linguistiques d'une communauté, le modèle d'une RTO peut aller à l'encontre de conceptions de sens commun. Ainsi, le terme " témoin " défini entre autres par le TLF comme une " *chose qui permet de constater, de vérifier* " est par essence un hypéronyme de " voyant " (" *lampe s'allumant pour attirer l'attention sur un danger, un dysfonctionnement ou pour indiquer qu'un appareil fonctionne* "). Toutefois, l'observation des expansions de ces termes en corpus nous a révélé qu'ils étaient utilisés de manière équivalente : environ 55% de termes spécifiant " témoin " (e.g. " témoin de charge batterie ", " témoin de climatisation ") ont un équivalent avec " voyant " (" voyant de charge batterie ", " voyant de climatisation "). Ceci nous a conduit à regrouper les deux termes comme des synonymes désignant un même concept (dont la définition est celle d'un voyant).

3.1.5.3 Association des termes aux concepts

Le nombre et la variété de termes associés aux concepts dépendent étroitement des besoins applicatifs pour lesquels la RTO est bâtie. Si par exemple l'ontologie que nous avons construite était exploitée comme un réseau de causalité entre symptômes, elle aurait une composante terminologique beaucoup moins importante. Dans le cas de l'indexation conceptuelle, l'association (automatique) de concepts à des fragments de textes requiert de caractériser les formes linguistiques exprimant les concepts, c'est-à-dire les termes les désignant ou bien les contextes dans lesquels ils sont présents. Etant donné que les fiches sont peu rédigées, nous avons choisi de nous appuyer sur les termes pour identifier la présence de concepts. De ce fait, la RTO doit comporter le plus de termes possibles associés aux concepts. L'importance de la partie lexicale d'une RTO pour notre application ainsi que le manque d'expressivité en termes lexicaux des formalismes ontologiques existants nous ont amené à envisager un méta-modèle original capable de manipuler séparément les notions de terme et de concept. Cette proposition sera abordée plus longuement dans la section suivante.

[5] `http ://atilf.atilf.fr/`

[6] Nous savons que ce sont des objets du domaine soit par les résultats préalables de l'étude du corpus, soit par le recours à un expert.

3.1.6 Bilan

Entre l'application cible, la tâche et le domaine, il est difficile de faire clairement la part des influences sur le processus de modélisation et le contenu du modèle construit. L'application cible (dans notre cas la recherche d'information dans des textes) englobe en partie le fait que le modèle reflète les connaissances d'un domaine mises en œuvre pour réaliser une certaine tâche (le diagnostic de panne). La particularité de notre application est justement qu'elle n'automatise pas la tâche en effectuant une résolution de problème, mais en réalisant une recherche d'information. De notre expérience, il ressort que l'application et le besoin de raisonner sur les connaissances modélisées déterminent le processus de modélisation (choix des modes d'analyse et des logiciels de TAL) et le degré de formalisation des connaissances. Pour leur part, la tâche et le domaine caractérisent le contenu de l'ontologie (choix des concepts, manière de les définir, niveau de détail).

La notion de rôle (au sens de classe conceptuelle utilisée dans le raisonnement) tient ici une place charnière : à un rôle, sont associés des types de connaissances à définir en tant que concepts du domaine. Nous avons choisi de placer les rôles utilisés pour le diagnostic au sein de l'ontologie, qui couvre ainsi le domaine des connaissances du spécialiste en diagnostic automobile, suffisantes pour retrouver des cas analogues. Dans l'ontologie, les rôles sont représentés comme des concepts de haut niveau (i.e. les prestations), lorsqu'ils mettent en jeu un ou plusieurs concepts qui ne jouent pas d'autre rôle (i.e. les composants), ou comme des concepts définis (i.e. les symptômes) quand ils impliquent la mise en relation de plusieurs concepts. Toutefois, il s'agit là d'un choix pragmatique dont il faudrait évaluer la généralisation dans d'autres contextes analogues, où le système réalise une recherche d'information à propos d'une tâche.

Ainsi, la définition des concepts rend compte d'un point de vue particulier sur le domaine. Même si l'ontologie à construire a pour vocation d'indexer des documents, ce projet confirme que le vocabulaire et les connaissances présents dans ces documents ne suffisent pas toujours pour obtenir des définitions. Celles-ci se trouvent davantage dans des documents de type pédagogique ou de formation, ou bien elles sont à recueillir auprès d'experts. De même, comme l'application utilisant l'ontologie vise la recherche d'information, la composante terminologique associée à l'ontologie doit être d'autant plus riche. Ceci confirme le besoin d'adapter la terminologie à l'application et au domaine à couvrir.

Au delà de ces énoncés, il serait intéressant de localiser explicitement ces influences dans une méthode comme TERMINAE. Il s'agit d'une première perspective à notre travail. De plus, la notion de RTO telle que nous l'avons utilisée renvoie plus à un modèle conceptuel d'un domaine qu'à une ontologie au sens où le seul principe ontologique appliqué est la différenciation. Une approche complémentaire serait de déterminer la manière de prendre en compte l'application ciblée pour construire une ontologie selon des méthodes plus contraintes comme OntoSpec [Kassel, 2005] ou OntoClean [Guarino et Welty, 2004].

3.2 Formalisation d'une Ressource Termino-Ontologique en OWL

Cette section est le fruit de réflexions en partie publiées dans [Reymonet et al.*, 2007a] et [Reymonet* et al.*, 2007b].*

Dans la section précédente, nous avons conduit une analyse a posteriori sur le processus de construction d'une RTO et nous avons pu illustrer par un cas d'étude l'influence que peuvent exercer les besoins applicatifs sur le déroulement de cette étape de modélisation. Nous avons notamment constaté la nécessité de disposer d'une composante terminologique riche en vue de l'utilisation de la RTO correspondante dans une application de RI sémantique. Une fois le modèle théorique établi, nous avons été confronté au choix du langage dans lequel formaliser celui-ci.

Comme nous l'avons vu en 1.4, plusieurs formalismes permettent de représenter un artefact terminologique et/ou ontologique. Afin de représenter une RTO, nous avons choisi de nous intéresser au langage OWL car il comporte plusieurs avantages : tout d'abord, choisir un standard relativement populaire dans la communauté de l'IC garantit de disposer dès le départ de plusieurs outils de gestion compatibles et de développer par la suite des applications avec une bonne interopérabilité ; en outre, OWL s'avère suffisamment expressif pour que l'on puisse conduire des raisonnements logiques à partir d'une RTO fondée sur ce formalisme[7]. Ce choix de langage entraîne une réflexion nécessaire liée à la place de la composante terminologique : comme nous le démontrons en 3.2.1, les différentes approches de la littérature ne permettent pas de représenter de façon satisfaisante la notion de terme en OWL. C'est pourquoi, dans les parties 3.2.2 et 3.2.3, nous nous appliquerons à mettre en place un méta-modèle de RTO en OWL qui réponde à nos besoins.

3.2.1 Limites des formats termino-ontologiques actuels sous OWL

Dans cette sous-section, nous décrivons et analysons quelques modèles permettant de représenter conjointement les parties ontologique et terminologique d'une RTO dans le standard OWL. Nous verrons notamment en quoi ceux-ci pourraient être améliorés, ce qui nous amènera dans la sous-section suivante à une proposition originale de modèle.

3.2.1.1 La modélisation classique du terme en OWL

En OWL, les éléments de base d'une ontologie sont matérialisés de la façon suivante :
- les concepts de l'ontologie sous forme de `owl:Class`,
- les attributs de concepts sous forme de `owl:DatatypeProperty`,
- les relations entre concepts sous forme de `owl:ObjectProperty`.

OWL a été élaboré dans l'idée de servir à l'indexation de ressources sur le Web. Il permet donc de représenter le lexique sous la forme duquel un concept (respectivement une instance de concept) pourra apparaître dans un document ou dans l'interaction avec l'utilisateur. Pour modéliser les termes désignant ce concept (resp. cette instance), OWL associe à la

[7] Nous rappelons ici qu'une des motivations à envisager un processus de RI fondé sur une RTO consiste à pouvoir déduire, à partir de connaissances explicitement mentionnées dans un document ou une requête, des connaissances implicites, formalisées dans l'ontologie.

classe (resp. instance) correspondante une (ou plusieurs) chaîne(s) de caractères au moyen d'une propriété d'annotation, `rdfs:label`.

Plusieurs problèmes découlent de ce choix de modélisation. Tout d'abord, un terme ainsi représenté n'a pas d'existence en tant que tel. Par conséquent, on ne peut pas lui associer directement de propriétés qui lui sont pourtant intimement liées (comme ses contextes d'usage dans les textes), il faut passer par le concept (resp. l'instance) qu'il désigne. Il est alors entre autres impossible de respecter le méta-modèle à vocation terminologique de TMF qui prône la dissociation des informations conceptuelles et lexicales. En outre, le formalisme OWL ne prévoit pas de représenter les occurrences des termes associés aux concepts. Cette notion est pourtant cruciale dans plusieurs cas de figure : au cours de la phase de construction ou de maintenance d'une RTO, la capacité de consulter les occurrences d'un terme dans leur contexte d'usage permet au cogniticien de mieux appréhender sa sémantique plus ou moins implicite en vue de l'associer (ou le réassocier) à un concept donné. De même, dans le cas d'une tâche d'indexation sémantique, il serait utile que le méta-modèle de l'ontologie utilisée permette d'exprimer certains phénomènes linguistiques comme l'anaphore [8] : pour l'appariement entre une requête et des textes, on pourrait alors pondérer différemment un texte contenant deux instances d'un même concept et un texte contenant deux occurrences d'une seule instance du même concept.

3.2.1.2 L'emploi de propriétés d'annotations structurées : la méthode Terminae

Le logiciel Terminae est un support à la méthode semi-manuelle [9] de construction d'ontologies orientée vers les textes [Szulman *et al.*, 2002]. Pour assurer une traçabilité des concepts vers les textes qui les évoquent, Terminae permet de gérer une terminologie du domaine, chaque terme donnant lieu à une fiche terminologique.

Dans un souci de standardisation des formats de représentation, il a été envisagé d'utiliser OWL pour représenter de façon détaillée les termes. Les concepteurs de Terminae ont souligné les insuffisances d'OWL à cet égard et ont alors proposé d'enrichir le modèle (sans extension de la syntaxe OWL) afin que des paramètres des termes, comme leurs synonymes et leurs occurrences dans les textes, puissent être pris en compte [Szulman et Biébow, 2004]. A cet effet, les auteurs utilisent la structure de propriété d'annotation d'OWL (`owl:AnnotationProperty`) qui permet d'ajouter des informations particulières à une classe. On peut voir ci-dessous comment la représentation conjointe d'un concept et d'un terme se traduit alors en OWL selon Terminae. Le choix de modélisation consiste à rajouter autant de propriétés d'annotation à chaque concept que d'attributs de terme nécessaires : synonyme, occurrence, catégorie syntaxique ...

```
<owl:AnnotationProperty rdf:about="&terminae;term"/>
<owl:AnnotationProperty rdf:about="&terminae;synonym"/>
```

[8] On peut définir l'anaphore comme le procédé grammatical assurant une reprise sémantique d'un précédent segment textuel (appelé antécédent) par un mot ou un syntagme.

[9] Nous préférons établir une distinction entre semi-automatique et semi-manuelle dans le sens où Terminae est un outil reposant fortement sur le cogniticien qui modélise l'ontologie, seuls quelques pré-traitements étant automatisés.

```
<owl:AnnotationProperty rdf:about="&terminae;occurrence"/>
[...]
<owl:Class rdf:ID="Code_défaut">
  <rdfs:subClassOf rdf:resource="#Problème"/>
  <terminae:term>code défaut</terminae:term>
  <terminae:synonym>CD</terminae:synonym>
  <terminae:occurrence>
    Le CD 198 apparaît à chaud.
  </terminae:occurrence>
</owl:Class>
```

Cette structuration permet de stocker un plus grand nombre d'informations sur chacun des termes que l'utilisation de la seule propriété `rdfs:label`. Nous avons d'ailleurs retenu dans un premier temps cette solution pour la construction et le stockage d'une RTO du diagnostic automobile. Toutefois, un méta-modèle de ce type rend impossible la manipulation du terme indépendamment du concept associé : on ne peut par exemple accéder aux occurrences d'un terme particulier. En outre, à la différence des propriétés d'objet et des propriétés de type de données, les propriétés d'annotation ne sont pas prises en compte par un raisonneur OWL. Il sera donc impossible de conduire des inférences logiques sur les liens associant termes et concepts, sauf à modifier le comportement de ces raisonneurs.

3.2.1.3 L'assimilation du terme à une instance de classe

Le terme, instance de concept Une autre approche, commune en extraction d'information, consiste à considérer le terme comme instance du concept auquel il est associé. GATE, plateforme incontournable du domaine, adopte ce choix de modélisation [Bontcheva *et al.*, 2004]. Cet environnement de développement fournit à la fois un modèle pour systèmes de traitement de langage naturel, une interface de programmation (API) et un environnement graphique permettant d'effectuer des traitements linguistiques sur des corpus de textes. S'il ne manipule pas une ontologie directement en OWL, GATE dispose d'une fonctionnalité d'import / export sous ce format. Il permet notamment de reconnaître dans les textes des listes d'instances (appelées *gazetteers*), de définir puis de projeter des patrons d'extraction grâce auxquels des expressions linguistiques sont associées à des instances de concepts (cette étape est réalisée par l'application de règles JAPE). C'est ainsi que les expressions extraites accèdent au statut de terme.

Ce choix de considérer le terme comme une instance de concept n'est pas sans poser certains problèmes théoriques et pratiques. En effet, si l'on revient aux définitions de base de l'ingénierie des connaissances, un concept est une représentation mentale destinée à regrouper un ensemble d'objets (les instances) partageant des traits communs identifiables [Kassel, 1999]. Assimiler un terme à une instance de concept (i.e. à un objet) ne respecte pas la différence entre symbole référent telle qu'établie dans le triangle sémiotique [Ogden et Richards, 1923]. D'un point de vue pragmatique, une telle modélisation oblige une résolution immédiate de toute ambiguïté sémantique : il est impossible de représenter l'occurrence d'un terme sans présumer de son sens. Pourtant, il peut être intéressant de relever la présence d'un terme sans désambiguïser son sens, en attendant de trouver

de nouveaux indices (via la co-occurrence, par exemple) qui permettront de déterminer la sémantique locale de son occurrence.

Le terme, instance de catégorie syntaxique On trouve dans [Liang *et al.*, 2006] une approche comparable réalisée dans le cadre d'un programme de recherche de la FAO (Food and Agriculture Organization). L'objectif du méta-modèle proposé consiste à créer une ontologie multilingue du domaine agricole à partir du thésaurus Agrovoc. Le contexte multilingue a poussé les auteurs à un effort particulier pour distinguer un terme d'un concept. Le méta-modèle, formalisé en OWL, propose de séparer les concepts de leur(s) lexicalisation(s) selon une partition binaire des classes OWL. L'article choisit alors de représenter les termes comme instances de la classe correspondant à leur catégorie syntaxique.

Un modèle de ce type va plus loin que le précédent : au lieu de représenter le lien entre un concept et un terme par une simple relation d'instanciation, il utilise la propriété d'objet spécifique `has_lexicalization` pour les relier. L'existence d'un terme ne présuppose alors plus l'existence préalable d'un concept auquel on pourrait l'associer. Toutefois, cette solution n'est pas exempte de défauts : de fait, il est techniquement impossible en OWL-DL de créer un lien entre une classe et un individu. Les auteurs sont donc obligés de créer un individu artificiel pour chaque concept afin de le connecter au(x) terme(s) adéquats via la relation `has_lexicalization`. Cet artifice permet de rendre le modèle proposé fonctionnel d'un point de vue pratique mais ne garantit en aucun cas sa correction d'un point de vue théorique : la manifestation linguistique d'une classe d'objet (i.e. d'un ensemble d'objets partageant certaines propriétés en commun) ne devrait pas être reliée à un objet particulier. De plus, la décision de différencier un terme en fonction de sa catégorie syntaxique est discutable : cette modélisation oblige à conduire une analyse syntaxique (ce qui n'est pas toujours possible ou souhaité) et elle n'envisage pas le cas des catégories syntaxiques cycliques (e.g. un syntagme nominal peut inclure plusieurs syntagmes nominaux) puisqu'elle se restreint aux catégories de base comme nom commun ou verbe.

3.2.2 Proposition de méta-modèle en OWL-DL

Les différents modèles existants sont donc insuffisants à représenter correctement la partie terminologique associée à une ontologie. Leurs limites nous ont permis de dégager quelques principes sur lesquels baser notre modèle. Tout d'abord, il est nécessaire de matérialiser la notion de terme de manière à pouvoir la manier aussi aisément qu'un concept. Ceci permettra notamment de pouvoir modéliser un nombre quelconque d'informations relatives à l'ancrage des concepts dans la langue. De plus, on a constaté qu'établir un lien de classe à instance entre les deux notions ne permet pas une reproduction fidèle de certains phénomènes linguistiques. Nous exposons ci-après une manière de stocker les termes en OWL-DL, sans qu'il y ait besoin d'étendre la syntaxe du langage ontologique. Nous verrons ensuite comment nous proposons de relier terme et concept dans le nouveau modèle. Un méta-modèle idéal mettrait ces deux notions sur un même méta-niveau, mais ceci rendrait alors indécidable toute ontologie s'en inspirant [Vrandecic *et al.*, 2006]. Nous avons choisi une approche sous-optimale mais fonctionnelle et préservant la décidabilité. Nous en verrons en 3.2.2.3 les principaux avantages et inconvénients, et nous finirons cette sous-section

par la description d'une solution alternative se fondant sur OWL Full.

3.2.2.1 La représentation du terme

L'idée principale qui nous guide pendant le processus de méta-modélisation concerne la nature des artefacts nécessaires à la prise en compte de phénomènes linguistiques dans un texte : le méta-modèle doit permettre de représenter à la fois les termes et leurs occurrences dans les textes. En effet, un terme polysémique est - par définition - ambigu et l'on ne peut trouver son sens qu'en analysant son contexte d'usage.

Nous commençons par réifier le terme en le représentant sous forme de `owl:Class`. Le plus haut niveau d'abstraction de l'ontologie permet alors de faire la distinction entre un objet de type concept, représenté par une sous-classe de `DomainThing`), et un objet de type terme, représenté par une sous-classe de `Terme`. Ce choix de modélisation s'avère pratique car il nous permet d'assimiler une occurrence de terme à une instance de sous-classe de `Terme`.

De façon à permettre une approche multilingue des RTO dans laquelle plusieurs lexiques de langues différentes seraient connectés à une même ontologie [10], nous classons les termes selon leur langue d'origine. Enfin, comme OWL n'utilise pas l'hypothèse de nom unique, nous déclarons toutes les classes de l'arbre terminologique comme étant mutuellement disjointes. Ainsi qu'on peut le voir sur la figure 3.3 [11], le modèle d'ontologie que nous proposons enrichit celui de OWL en séparant l'ontologie proprement dite (sous la classe OWL `DomainThing`) d'une hiérarchie terminologique de profondeur maximale 3 et dont les feuilles représentent les termes.

Il est intéressant de noter que dans la norme ISO 12620 (sur laquelle s'appuie le langage terminologique TMF), un terme abrégé est considéré comme distinct de sa forme complète. Afin de ne pas multiplier le nombre de termes à manipuler, nous préférons rassembler les variantes (abréviation, omission de mot) sous un même terme comme dans [Aussenac-Gilles, 1999]. Dans le cadre de notre étude, nous avons choisi de ne représenter comme attribut du terme que sa localisation dans le corpus (un identifiant de texte et un identifiant de position). Toutefois, comme le terme est réifié dans notre modèle, il est aisé de lui rajouter de nouvelles propriétés (e.g. catégorie syntaxique, morphologie) sous forme de `owl:DatatypeProperty`.

3.2.2.2 La modélisation des liens terme-concept

Une fois la structure des termes établie dans OWL, analysons comment les relier aux concepts. En OWL, la structure de donnée adéquate pour relier deux classes est la propriété d'objet (`owl:ObjectProperty`). Comme une propriété de ce type est orientée (elle a une classe domaine et une classe codomaine), il nous faut envisager les deux cas possibles et voir

[10] Si l'on modélise un domaine relativement technique, on peut aisément envisager qu'une ontologie puisse être construite indépendamment des différentes langues natives des membres de la communauté associée (car les mêmes notions concrètes seront manipulées).

[11] On remarquera que le lien entre terme et concept n'est pas encore dirigé, cette discussion sera abordée en 3.2.2.2.

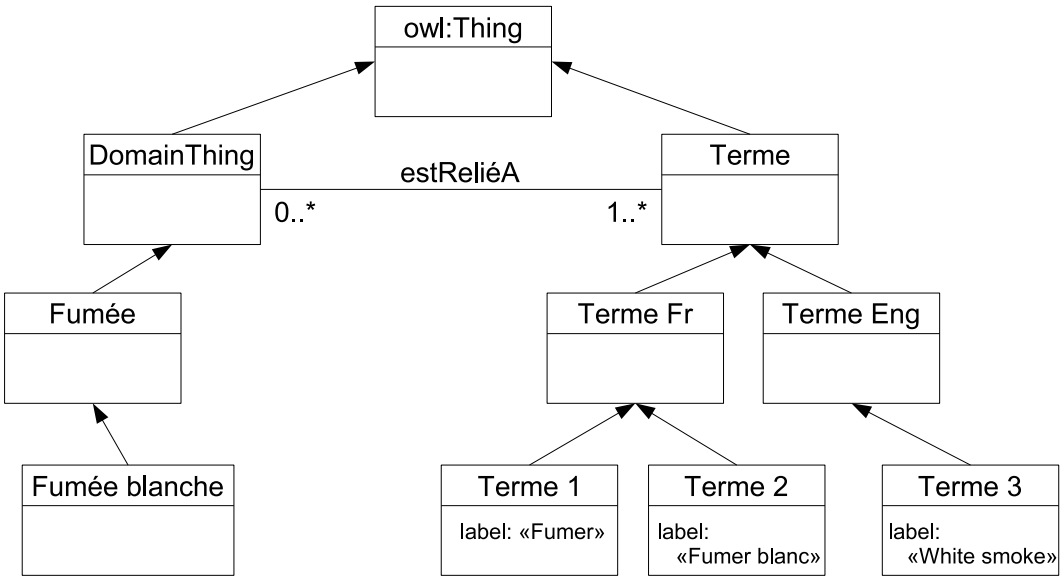

Figure 3.3 — Représentation simplifiée du modèle

lequel des deux est le plus fidèle à la conceptualisation souhaitée et/ou le plus facilement représentable en OWL. Pour la suite, on rappellera qu'en OWL, lorsqu'une propriété d'objet P a pour domaine un concept A et pour codomaine B, alors tout fils de A héritera de P. Lors de cet héritage, chacun des fils peut restreindre sur son domaine le codomaine de P à l'un des fils de B.

Considérons d'abord le cas d'une propriété *dénotéPar* ayant pour domaine un concept C et pour codomaine l'union de plusieurs classes de type `Terme`. Si l'on prend un fils de C, celui-ci sera nécessairement dénoté par au plus les mêmes termes que son père (du fait du mécanisme d'héritage des propriétés en OWL, mentionné plus haut). D'un point de vue conceptuel, ce résultat n'est pas satisfaisant puisque l'on souhaite pouvoir associer des termes différents à deux concepts reliés hiérarchiquement. On pourrait imaginer d'associer aux termes dénotant C tous les termes dénotant ses fils mais un autre problème survient alors : on est incapable de connaître directement les termes dénotant C et uniquement lui (si ce n'est en parcourant systématiquement l'ensemble de ses fils).

Plaçons nous maintenant dans la situation inverse où terme et concept sont reliés par une propriété *dénote* dirigée du terme vers le concept. On constate un premier avantage à cette modélisation : comme seules les feuilles de l'arbre terminologique sont des termes, il n'y aura pas de problème d'héritage de la propriété *dénote*. Du fait de la propriété d'héritage des classes en OWL, un tel choix entraîne qu'un terme dénotant un concept C dénote aussi les fils de C. Ce phénomène renvoie à la figure linguistique d'anaphore infidèle : l'occurrence d'un terme dénotant un concept C_0 peut faire indirectement référence à une instance d'un concept C_1, fils de C_0. Par exemple, dans la phrase "*Le compteur de vitesse ne s'allume pas au démarrage*", on repère une occurrence du terme "*compteur de vitesse*" qui dénote le concept *Tachymètre*. Toutefois, on peut déduire grâce au contexte (i.e. l'absence d'*allumage*) et à une connaissance suffisante du domaine que l'instance de terme désigne en fait une instance d'un concept plus précis, à savoir *Tachymètre digital*. Nous avons donc choisi de retenir cette

solution, illustrée sur la figure 3.4.

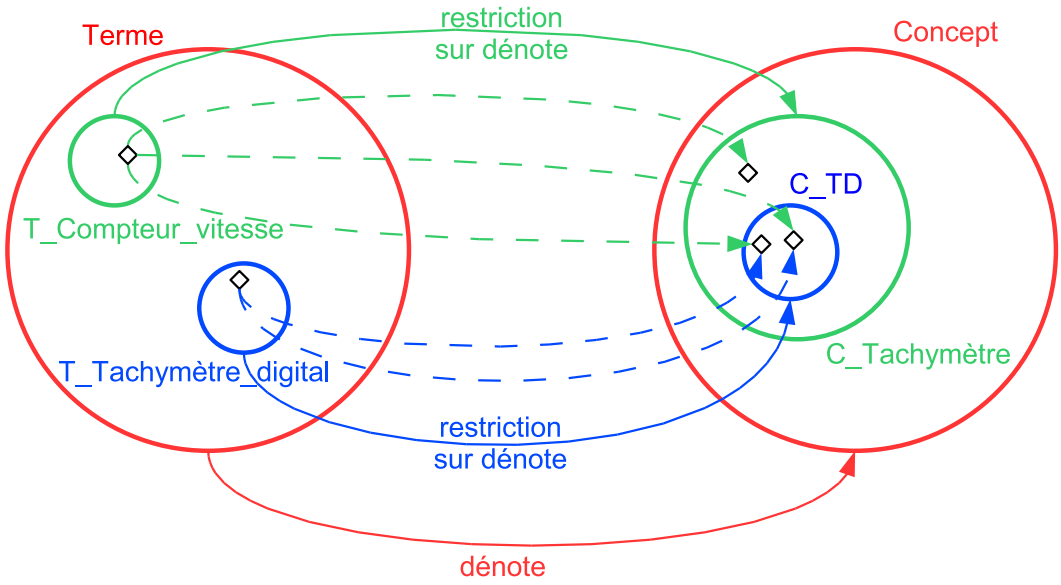

Figure 3.4 — Lien Terme-Concept

La figure 3.5 résume le fonctionnement de notre méta-modèle sur un exemple simple : l'occurrence du terme "Fumée blanche" (`Terme33`) dans le document `Texte57` est représentée par l'instance de classe `inst3`. Pour symboliser que cette occurrence se réfère à une fumée blanche particulière, elle est associée à l'instance adéquate `fb1`.

3.2.2.3 Avantages et limites du méta-modèle

En choisissant de ne pas utiliser OWL Full et de garder un langage décidable, nous avons délibérément accepté de ne pas fonder notre modèle de RTO sur de réels mécanismes de méta-modélisation (e.g. méta-concept), non disponibles en OWL-DL. Par conséquent, nos choix architecturaux ont amené à un modèle opérationnel mais théoriquement bancal. Nous examinons dans cette partie les bénéfices et les difficultés liés à l'utilisation de notre modèle.

Du fait de la réification de la notion de terme, le modèle de RTO proposé a l'avantage de permettre une manipulation directe des termes, complètement indépendante du mode de représentation des concepts. En outre, la structure choisie pour modéliser un terme rend possible de régler le degré de précision de la représentation : autant de propriétés que souhaitées peuvent être ajoutées à la structure d'un terme (e.g. catégorie syntaxique, morphologie, usage en contexte ...) par l'emploi de la structure de propriété de type de donnée (`owl:DatatypeProperty`). Malheureusement, notre méta-modèle mélange des propriétés de niveaux épistémologiques différents : il est incapable de faire la distinction entre les propriétés de classe (e.g. langue, label) et les propriétés d'instance (identifiant du texte dans lequel l'occurrence a été retrouvée, sa position relative dans ce texte ...). Cette caractéristique entraîne un problème de duplication de certaines données : ainsi, la propriété *langue* pour un terme français est restreinte par un axiome de type `owl:hasValue` (de façon à

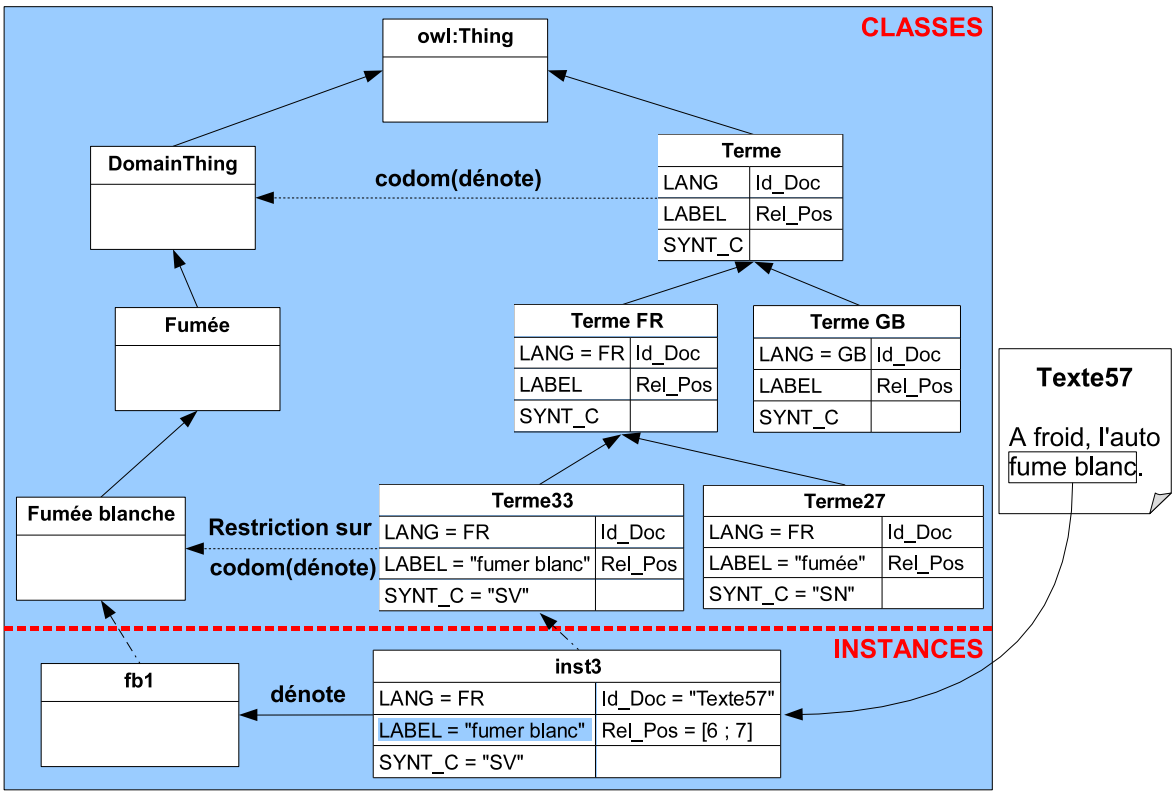

Figure 3.5 — Méta-modèle proposé

s'assurer qu'aucune instance du terme ne puisse prendre une valeur de langue différente) de telle sorte que chaque occurrence / instance de ce terme doit stocker la même information.

Un apport indéniable du modèle que nous proposons réside dans sa capacité à représenter des phénomènes linguistiques comme la polysémie ou l'anaphore : sur la figure 3.3, on peut voir que l'occurrence du terme "fumée blanche" dans le texte 57 dénote "fb1", représentant une fumée blanche spécifique qui aura pu être mentionnée au préalable au travers d'une autre occurrence de terme. Un problème potentiel pourrait découler de cette fonctionnalité : notre méta-modèle crée jusqu'à deux fois plus d'instances que d'occurrences de termes retrouvées. Cette spécificité reste acceptable puisqu'en termes de passage à l'échelle, le nombre d'instances manipulées reste du même ordre de grandeur que si seules étaient stockées les instances de concepts (et non les occurrences de terme). Par ailleurs, un inconvénient avéré de notre approche réside dans notre hypothèse implicite selon laquelle une occurrence de terme se rapporte nécessairement à un objet du domaine. La validité de cette supposition n'est pas systématique et l'on peut facilement trouver des contre-exemples : dans la phrase "*La voiture est un moyen de transport répandu*", l'occurrence de "Voiture" ne renvoie pas à un objet spécifique (i.e. une instance de la classe `Voiture`), mais à une classe d'objets (i.e. le concept lui-même). Selon le degré de spécificité du domaine et la nature des textes utilisés en combinaison avec l'ontologie, ce phénomène peut a priori s'avérer handicapant. Concrètement, nous avons pu constater que ce genre de phrases énonçant des vérités générales est très rare dans des documents à finalité non définitoire d'un domaine technique.

Le méta-modèle que nous proposons comporte également une limite dans la nature du lexique que l'on peut associer aux concepts. En effet, à travers la notion de terme, nous n'avons pas envisagé la représentation d'entités nommées. Pour rappel, une entité nommée (EN) correspond à un groupe de mots désignant de façon univoque une entité du domaine ; parmi les différents types possibles, on peut rencontrer les noms propres (e.g. "*Zinedine Zidane*"), mais aussi certains syntagmes nominaux (e.g. "*Université de Toulouse*"). A la différence des termes que nous envisageons, toute EN dénote non un concept mais une instance de concept : ainsi dans une ontologie du football, l'EN "Zinedine Zidane" fait référence à l'instance du concept `Footballeur` identifiable par son numéro 10 en équipe de France. Quel que soit le contexte textuel autour de l'occurrence d'une EN donnée, cette occurrence désigne systématiquement une seule et même instance de l'ontologie. On voit donc par leurs usages que la nature même d'une EN la différencie de celle d'un terme tel que nous l'envisageons. Même si la notion d'EN nous paraît constituer un champs d'étude à part entière et que nous choisissons de ne pas la traiter plus en détail, nous pouvons évoquer une solution simple permettant la représentation d'EN dans notre méta-modèle : pour symboliser l'association d'une EN à une instance particulière, nous pouvons restreindre le co-domaine de la relation de dénotation sur cette EN à la classe dont l'extension se réduit à la seule instance adéquate. Pour cela, il suffit de créer en OWL une classe énumérée (par définition anonyme) dont la seule instance correspond à celle dénotant de l'EN. En reprenant l'exemple précédent, on a alors en OWL :

```
<owl:Class rdf:ID="T_ZZ">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Terme_Fr"/>
        <owl:Restriction>
          <owl:onProperty>
            <owl:FunctionalProperty rdf:about="#texte"/>
          </owl:onProperty>
          <owl:hasValue rdf:datatype="&XMLS;string">
            Zinédine Zidane
          </owl:hasValue>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#dénote"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:oneOf rdf:parseType="Collection">
            <footballeur rdf:ID="footballeur_29">
              <numero rdf:datatype="&XMLS;int">10</numero>
              <equipe rdf:datatype="&XMLS;string">France</equipe>
            </footballeur>
          </owl:oneOf>
```

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

Reste à soulever un dernier point relatif à la nature ontologique possible des artefacts désignés par un terme. En effet, certains termes peuvent correspondre à la lexicalisation d'une relation sémantique. Avec la solution que nous avons présentée, nous sommes dans l'impossibilité de représenter ce phénomène sans une profonde modification du modèle : la syntaxe OWL-DL ne dispose pas de primitives permettant d'associer une classe (dans notre cas un terme) à une propriété d'objet ou de type de donnée.

3.2.3 Solution alternative non fondée sur OWL-DL

Dans cette dernière sous-section, nous évoquons une approche de la littérature ainsi qu'une solution s'en inspirant partiellement dans le but de résoudre les problèmes soulevés par notre méta-modèle en OWL-DL, à savoir :

- l'absence de distinction entre les propriétés de classe et d'instance,
- l'impossibilité d'associer une occurrence de terme à un concept (et non à une instance de concept)
- l'impossibilité pour un terme de désigner une relation sémantique

L'approche que nous allons décrire ainsi que la nouvelle solution que nous allons proposer se fondent toutes deux sur le langage OWL Full avec son défaut inhérent d'indécidabilité. Nous verrons ensuite que certains travaux parallèles essaient de définir un langage à mi-chemin entre OWL-DL et OWL Full et qui permette de garantir la décidabilité d'une ontologie dérivée tout en proposant des primitives de méta-modélisation [12].

Le méta-modèle LingInfo a été proposé dans [Buitelaar *et al.*, 2006b] et [Buitelaar *et al.*, 2006a] pour le projet SmartWeb (déjà mentionné en 2.2.4.2). Partant du constat que la partie lexicale d'une RTO est classiquement trop pauvrement représentée, les auteurs ont cherché à mettre au point un modèle compatible avec un format standard (à savoir RDFS et OWL Full) et capable de rendre compte d'informations lexicales telles que la langue, la décomposition morpho-syntaxique ou le contexte d'usage d'un terme. Pour ce faire, les auteurs utilisent le mécanisme de méta-classe : ils créent `feat:ClassWithLingInfo`, une sous-classe de `rdfs:Class`[13] avec pour particularité de contenir un ensemble `lf:LingInfo` de traits linguistiques (cf plus haut). De cette façon, chaque concept de l'ontologie (qui sera instance de `feat:ClassWithLingInfo`) pourra se voir dénoté par plusieurs termes dans plusieurs langues. Soulignons au passage que Buitelaar et ses collègues font eux aussi dans ces travaux l'hypothèse qu'une RTO multilingue peut s'appuyer sur une seule ontologie et plusieurs terminologies de nationalités différentes. Plus intéressant encore, en procédant de même avec les propriétés

[12] Précisons toutefois qu'à ce jour, nous n'avons pu trouver de travaux utilisant directement ce type de formalismes et se préoccupant précisément de la représentation des termes dans une ontologie.

[13] En effet, dans ce langage, le type d'une classe s'avère être aussi une classe, ce qui permet d'envisager des architectures sur plusieurs niveaux d'instanciation.

(de type attribut ou relation entre concepts), les travaux en question permettent de leur associer une partie lexicale, ce qui s'avérait impossible dans notre proposition. Par contre, le méta-modèle LingInfo n'aborde pas le problème des occurrences de terme, il se limite à exprimer qu'il existe un lien entre un concept et un terme. Il ne se préoccupe ni de savoir si dans un certain contexte d'utilisation, un terme dénote un concept plutôt qu'un autre (pas de désambiguïsation), ni si ce terme par son occurrence fait référence à un universel (i.e. un concept) ou un particulier (i.e. une instance de concept).

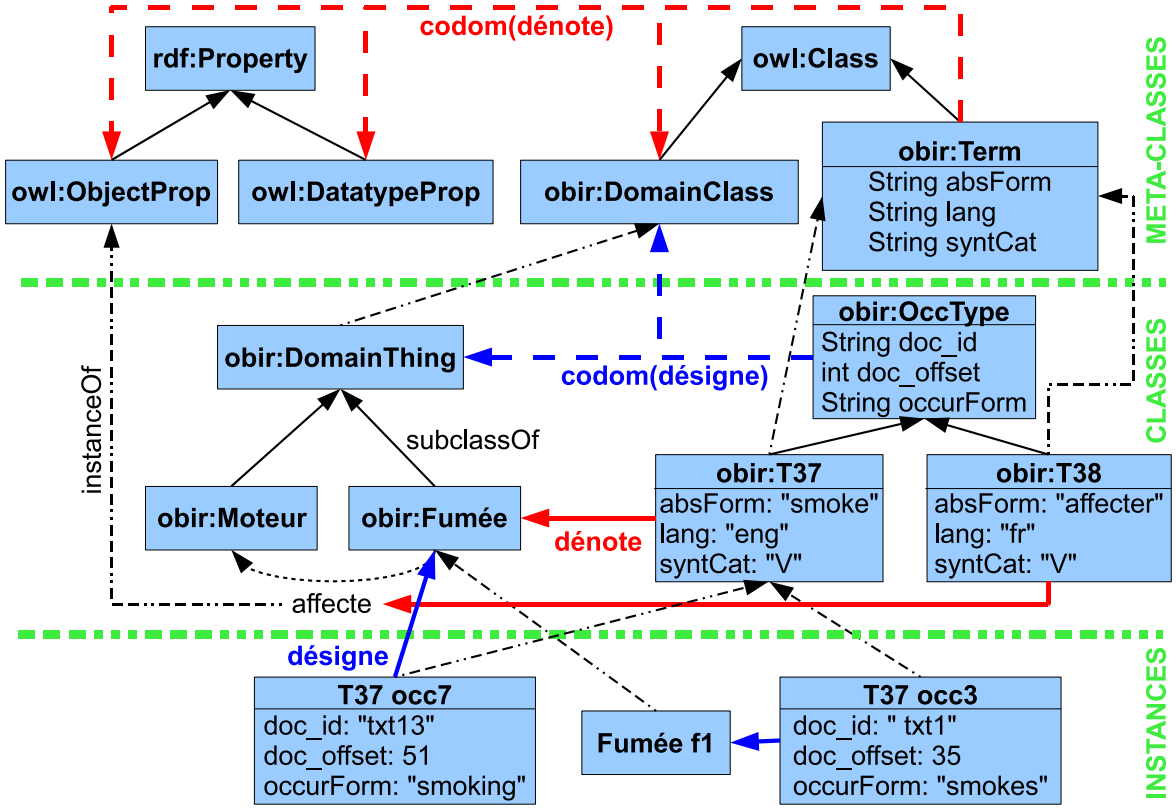

Figure 3.6 — Proposition de méta-modèle en OWL Full

Tout en nous inspirant du méta-modèle LingInfo, nous avons alors modifié notre proposition initiale de façon à en obtenir une version certes non décidable mais plus riche en pratique et théoriquement plus acceptable. La première étape consiste à créer deux méta-classes `obir:DomainClass` et `obir:Term` qui permettront au niveau des classes de séparer les entités conceptuelles des terminologiques. Dans le but de gérer plus aisément le lien existant d'une part entre un terme et un concept (jusqu'alors représenté artificiellement par une restriction de la relation de dénotation) et d'autre part entre une occurrence de terme et une instance de concept (ou un concept), nous séparons les deux types d'information en une relation `dénote` pour le premier type et une relation `désigne` pour le second. Nous fixons alors `obir:Term` pour domaine de la relation `dénote`; pour le co-domaine, nous le définissons comme l'union des méta-classes `obir:DomainClass` (*un terme peut dénoter un concept ...*), `owl:ObjectProperty` (*... mais aussi une relation sémantique ...*) et `DatatypeProperty` (*... voire un attribut de concept*). Pour la relation `désigne`,

nous choisissons comme co-domaine l'union de la classe `obir:DomainThing` (correspondant au concept le plus générique) et de la méta-classe `obir:DomainClass` de telle sorte qu'une occurrence de terme pourra désigner soit une instance de concept (cas des particuliers), soit un concept (cas des universaux). Pour domaine de `désigne`, nous créons la classe `obir:OccType` de façon à ce que tout terme possède, par héritage, des propriétés caractéristiques à son occurrence dans un ensemble de textes : identifiant du texte, position relative dans le document, forme sous laquelle le terme est répéré. On notera que la classe `obir:OccType` n'est pas une instance de `obir:Term` car ce n'est pas un terme : il n'instancie aucune des propriétés caractéristiques d'un terme (forme canonique, langue, catégorie syntaxique . . .). On obtient alors le méta-modèle représenté en figure 3.6.

Si l'on revient aux principales limites de notre méta-modèle dans sa version "compatible OWL-DL", on constate que celles-ci ont été levées dans la nouvelle version :

- il y a bien séparation entre les propriétés inhérentes à un terme et celles liées à une de ses occurrences,
- il est désormais possible de prendre en compte des phrases à portée universelle (e.g. "*Un plongeur doit respecter les paliers de décompression*") étant donné le co-domaine de définition de la relation `désigne`,
- le méta-modèle permet d'associer un terme donné à une relation sémantique ou à un attribut de concept.

Le principal inconvénient lié à notre nouveau méta-modèle réside alors dans la nécessaire indécidabilité d'une ontologie qui se fonderait dessus. Dans le cadre de nos travaux, nous verrons notamment en 4.1.2.2 que cette caractéristique n'est pas souhaitable car nous conduisons des inférences sur toute RTO manipulée, dans le but de vérifier sa cohérence. En se plaçant d'un point de vue plus global, il faut néanmoins relativiser l'importance de cet obstacle d'indécidabilité : certains travaux n'utilisent les ontologies que pour stocker des informations et ne cherchent pas à les exploiter au cours d'un processus de raisonnement. En outre, plusieurs approches qui mènent des inférences sur des ontologies non décidables mettent en avant que par définition, l'indécidabilité d'une ontologie n'empêche pas de conduire des raisonnements à partir de cette ressource, elle se borne à entraîner une absence de garantie théorique qu'un raisonnement arrive à une conclusion en temps fini. Ces approches illustrent qu'en pratique, il est possible, sous certaines conditions, de raisonner avec une ontologie indécidable sans constater de blocage des raisonneurs. Enfin, nous avons récemment pris connaissance de certaines recherches visant à mettre en place un langage de méta-modélisation décidable fondé sur OWL-DL et que nous allons maintenant aborder.

Comme le prouve formellement [Motik, 2005], l'indécidabilité en OWL Full est causée par le positionnement des primitives de méta-modélisation au même niveau que les instances et les classes. De façon à éviter une quelconque ambiguïté quant au niveau auquel interpréter une "méta-primitive", l'approche de [Pan et Horrocks, 2006] introduit avec OWL FA une architecture sur plusieurs couches comportant chacune leurs propres primitives de construction :

- le niveau **instance** (niveau 0) permet de représenter les objets du domaine modélisé,
- le niveau **ontologique** (niveau 1) permet de modéliser les concepts usuels du domaine,
- le niveau **langagier** (niveau 2) permet de créer des méta-ressources (concept ou propriété) comparables aux primitives du langage

– le niveau **méta-langagier** (niveau 3) permet de typer et de comparer les méta-ressources du niveau inférieur

Les primitives du langage sont semblables à celles disponibles en OWL, sauf qu'elles sont indicées par le niveau auquel elles appartiennent. Si on prend par exemple la définition du méta-concept `obir:Term` dans notre méta-modèle en OWL Full, celle-ci s'exprime alors en OWL FA de la façon suivante :

```
<fa:Class2 rdf:about="&obir;Term">
  <rdfs:subClassOf3 rdf:resource="&fa;Class2"/>
</fa:Class2>
```

L'avantage de ce langage ontologique est clair : en désambiguïsant systématiquement la couche dans laquelle est défini un axiome, OWL FA permet de garantir la décidabilité du langage [14]. D'un point de vue théorique, il faudrait que OWL FA dispose d'une infinité de couches pour atteindre une expressivité aussi importante que celle d'OWL Full. Cependant, les auteurs indiquent d'expérience qu'une architecture à 4 niveaux suffit à la plupart des cas d'utilisation usuels. Etant donné la relative nouveauté de OWL FA, aucun raisonneur n'a encore été adapté afin de manipuler sa sémantique. Du fait de sa simplicité et de la décidabilité inhérente, nous sommes tout de même persuadés que ce format est appelé à se répandre, d'autant plus qu'il gère aussi facilement toutes les primitives de OWL, y compris les propriétés de type de donnée (ce dont ne sont pas capables les sémantiques proposées par [Motik, 2005]).

[14] Pour plus de précisions, consulter la preuve formelle fournie dans [Pan et Horrocks, 2006].

4 Conception d'une plate-forme de recherche sémantique

Dans ce chapitre, nous présentons nos réflexions liées aux différentes étapes inhérentes à un processus de RI sémantique. Dans un premier temps, nous nous focalisons sur l'élaboration d'un cycle de maintenance de la RTO (section 4.1). En effet, comme nous avons pu le souligner précédemment en 1.3.4 et en 2.2.3, une RTO modélisant un domaine et/ou une tâche spécifique(s) doit être envisagée comme un artefact dynamique : les objets modélisés évoluent dans le temps, certains peuvent apparaître, d'autres à l'inverse peuvent sortir du champ d'intérêt de l'utilisateur. Ce dernier point est en relation avec la problématique des influences sur le processus de modélisation que nous avons abordée en 3.1 ; les besoins applicatifs peuvent de fait fluctuer temporellement, auquel cas leur évolution peut rendre nécessaire une phase d'adaptation de la RTO. De façon logique, nous décrivons ensuite une approche pragmatique destinée à gérer les conséquences d'une modification terminologique et/ou ontologique sur les annotations sémantiques.

Dans une deuxième section, nous nous penchons plus en détail sur les processus d'indexation et d'interrogation sémantiques. Nous expliquons notamment comment nous utilisons le méta-modèle de RTO proposé au chapitre précédent pour découvrir la trace de concepts dans un ensemble de documents. Nous nous intéressons ensuite à l'appariement sémantique entre cette même collection de textes et une requête exprimée en langue naturelle. Nous montrons à cette occasion que le processus de RI peut s'avérer bien plus qu'un simple moyen de repérer des informations décorrélées, pour peu que les méthodes d'appariement sémantique ne se bornent pas à une simple comparaison deux à deux des concepts retrouvés dans la requête d'une part et dans les documents indexés d'autre part.

4.1 Définition d'un cycle de maintenance supervisée de RTO

Avant d'exposer notre vision du cycle de vie d'une RTO destinée à une tâche de RI, nous jugeons utile de signaler au lecteur que parmi l'ensemble des étapes que nous allons mentionner, nous n'avons pu accorder une même importance à l'étude de chacune d'entre elles. Par conséquent, l'implémentation du prototype TextViz que nous proposons au chapitre 5 est amenée à évoluer au gré de nos réflexions futures sur cette problématique [1]. En effet,

[1] Nous pensons notamment à la partie traitant de la gestion des conséquences d'une évolution de RTO sur les annotations sémantiques, qui est actuellement l'objet du projet Dynamo, financé par l'ANR.

à travers nos recherches, ne pouvant approfondir de façon originale toutes les phases du cycle de maintenance (du fait de l'ampleur du problème et des contraintes temporelles liées à toute thèse), nous avons préféré étudier celles qui nous semblaient peu développées dans la littérature pour les intégrer dans un cadre commun : la construction et la maintenance d'une RTO pour une RI sémantique. Dans cette section, nous décrivons le cercle vertueux de maintenance et d'indexation tel que nous le concevons dans ce contexte de RI. On peut en voir une représentation synthétique sur la figure 4.1.

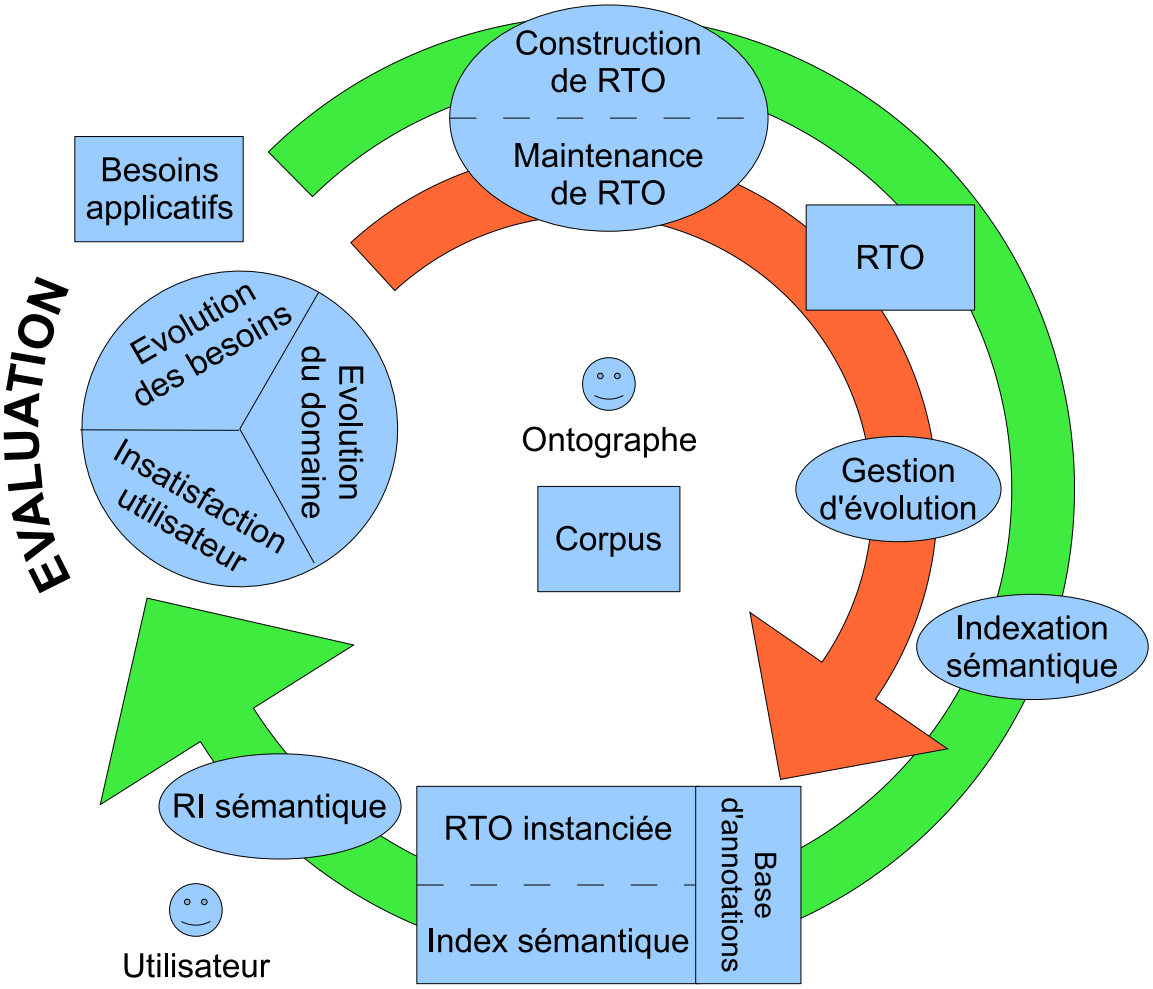

Figure 4.1 — Cycle de vie théorique d'une RTO pour une RI sémantique

4.1.1 Phase de construction de RTO appliquée en RI

Conformément à nos convictions quant à l'influence de l'application sur le processus de modélisation, la phase de construction de la RTO nécessite de connaître au préalable les besoins de l'utilisateur de façon à ce que le système se fonde ensuite sur une ressource adéquate et puisse apporter une aide pertinente pour la tâche d'interrogation. Par définition, les besoins de l'utilisateur sont intimement liés à la nature des éléments qu'il sera amené à manipuler lors d'une phase de RI ultérieure. On voit alors apparaître une situation d'inter-

dépendance : la phase de modélisation se fonde sur les besoins applicatifs de l'utilisateur, qui ne sont exprimables - dans le cas d'une application de RI sémantique - qu'en fonction des concepts manipulés. Pour remédier à ce problème, il est possible d'intégrer comme sources de connaissance au processus de construction de RTO les documents à indexer d'une part, un (ou plusieurs) expert(s) du domaine [2] d'autre part.

La méthode de modélisation que nous choisissons de suivre est une approche interactive de construction de RTO à partir de textes, sujet que nous avons précédemment abordé en 1.3.3.2. Cette méthode, présentée dans [Aussenac-Gilles *et al.*, 2008], est issue de la volonté d'exploiter au mieux un ensemble de documents relatifs au domaine à modéliser. Elle met notamment en jeu des outils de traitement automatique du langage sur le corpus de documents qui permettent à l'ontographe [3] de dégager des indices en faveur de la création de certains concepts et/ou relations. Tenants de l'hypothèse selon laquelle tout corpus textuel n'est qu'une source partielle de connaissances, les auteurs proposent alors de recourir à un spécialiste du domaine à modéliser, uniquement lorsque nécessaire (car son intervention s'avère souvent coûteuse).

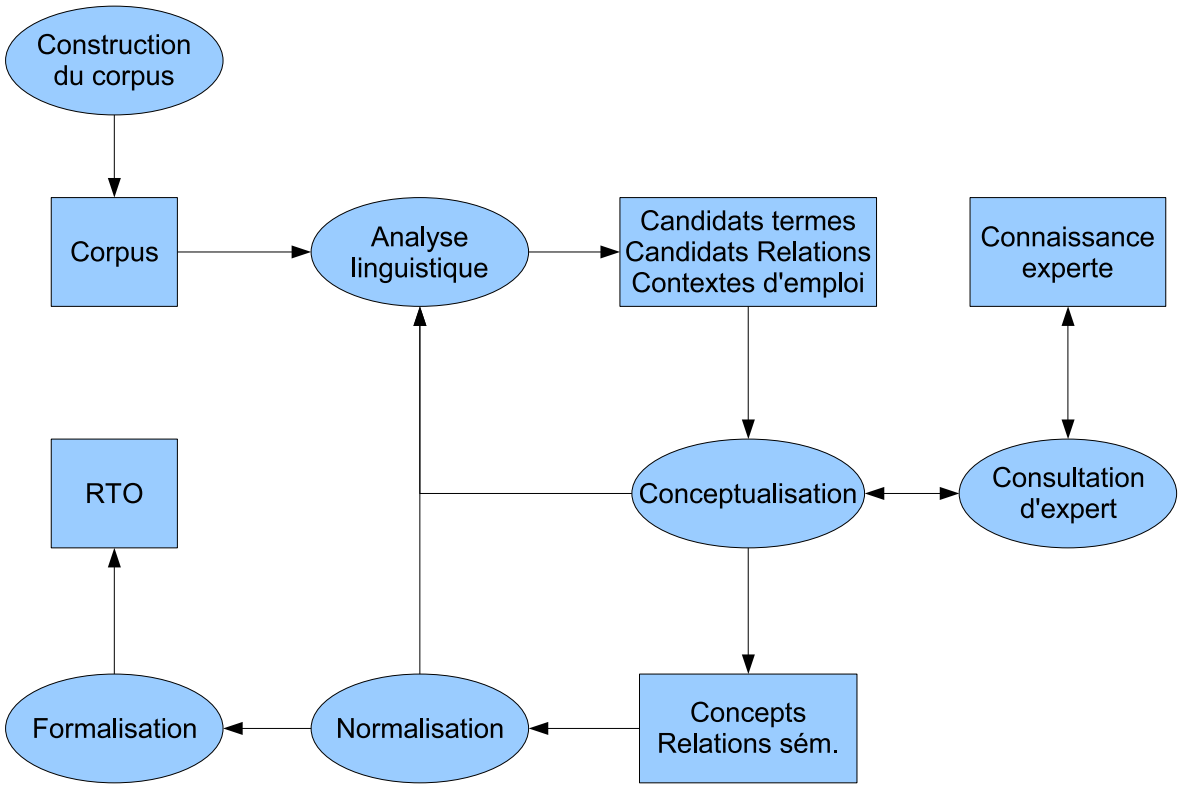

Figure 4.2 — Méthode de construction de RTO selon [Aussenac-Gilles *et al.*, 2008]

Comme on peut le voir sur la figure 4.2, la méthode de construction présuppose la création préalable d'un corpus de documents représentatifs du domaine. Dans un contexte de

[2] Nous rappelons que du fait de leurs trop grandes différences, nous étudions uniquement les moteurs de recherche sémantique spécifiques à un domaine, et non les moteurs génériques.

[3] On utilisera ce terme pour désigner l'ingénieur de la connaissance responsable de la construction de la ressource via l'analyse des résultats des outils d'analyse et des entretiens avec le(s) spécialiste(s) du domaine.

RI sémantique, il semble naturel que ce corpus inclue au moins les textes qui sont amenés à être indexés. La phase d'analyse linguistique, entièrement automatique, peut faire appel à plusieurs outils de type extracteur de termes ou de relations (voir 1.3.3.1) et permet d'obtenir des candidats terme et/ou relation (i.e. des groupes de mots ou des patrons lexico-syntaxiques réapparaissant régulièrement et témoins potentiels de la présence d'un concept ou d'une relation sémantique) ainsi que des informations fréquentielles et contextuelles associées. A l'aide de ces indications et en accord avec le point de vue du (des) spécialiste(s), l'ontographe peut alors entreprendre le processus de conceptualisation afin de créer progressivement les concepts du domaine et les relier entre eux par les relations sémantiques appropriées. Les heuristiques de construction de la RTO peuvent suivre trois axes : une démarche ascendante rassemble les concepts entre eux selon les points communs qu'ils partagent, une démarche descendante les organise selon des critères de différenciation (là encore dépendant de l'application) et une démarche centrifuge part des concepts centraux (généralement les plus simples à trouver) pour découvrir les concepts voisins à travers les relations transverses (i.e. non taxonomiques). Pour converger plus rapidement vers une RTO en adéquation avec les besoins applicatifs, la méthode présentée par Aussenac-Gilles et ses collègues préconise d'utiliser les trois types de démarches sans pour autant contraindre l'ontographe à respecter un ordre d'application. L'étape suivante, la normalisation, permet de vérifier que la ressource construite respecte certains principes théoriques (voir 1.3.1) et de la corriger si besoin est. Les étapes d'analyse linguistique, de normalisation sémantique et de formalisation peuvent s'appliquer cycliquement, tant qu'un résultat satisfaisant n'est pas atteint. Enfin la dernière phase, la formalisation, consiste à stocker la RTO résultante sous un format donné (e.g. RDF, SKOS, OWL ...), à choisir selon son degré d'expressivité et, de façon plus générale, des besoins applicatifs.

Nous souhaitons ici souligner deux points importants dans cette méthode de construction. Tout d'abord, celle-ci a pour avantage d'envisager la tâche de modélisation comme un processus à double sens : l'ontographe doit non seulement effectuer un travail de dépouillement des données (extraites de l'analyse des documents du corpus ou d'entretiens avec un expert) pour enrichir le modèle, mais il doit aussi s'assurer que la RTO respecte certains critères de bonne structuration, issus de besoins de modélisation spécifiques. Or, il est important, pour un outil de construction, de proposer une navigation multidirectionnelle entre la RTO, les besoins de modélisation et le corpus. Comme il est difficile, dans un contexte générique, de représenter explicitement les besoins applicatifs motivant la construction d'une RTO, la méthodologie proposée par Aussenac et ses collègues n'approfondit pas la problématique de représentation et d'utilisation de ces besoins. Ces constatations nous amènent au deuxième point que nous souhaitons aborder à propos de cette méthode de construction : bien qu'elle suive une approche cyclique, la méthode ne détaille pas explicitement les conditions à remplir par le modèle pour qu'il puisse être considéré comme terminé. Dans la prochaine sous-section, nous allons présenter une méthode de construction et/ou maintenance [4] de RTO pour une tâche d'indexation sémantique ; cette méthode se fonde notamment sur une représentation explicite et une utilisation des besoins applicatifs en terme d'entités termino-ontologiques reconnues au cours de l'étape d'indexation sémantique.

[4] La méthode est principalement axée sur le processus de maintenance, mais peut être facilement adaptée à la méthode cyclique de construction de RTO proposée dans [Aussenac-Gilles *et al.*, 2003a].

4.1.2 Maintenance de RTO par indexation sémantique

Le principe de la maintenance de RTO telle que nous l'envisageons dans un contexte de RI sémantique repose en partie sur l'hypothèse que la RTO est utilisée pour modéliser les entités et les assertions devant être repérées au cours du processus d'indexation. Par rapport à l'étape de construction, la phase de maintenance d'une RTO a en effet l'avantage de pouvoir se fonder sur des besoins explicites en information de l'utilisateur : certaines entités manipulées ont déjà été définies pendant la construction de la RTO et utilisées pendant l'indexation sémantique. Le processus cyclique de maintenance que nous proposons peut se résumer en trois phases :

- évaluation des besoins en maintenance pour une RTO selon des critères définis au préalable, applicables à chaque document du corpus et liés aux résultats d'indexation attendus,
- maintenance semi-automatique de la RTO en fonction des critères non satisfaits et indexation sémantique simultanée,
- gestion des impacts d'évolution de la RTO sur les annotations sémantiques

Comme on le verra plus en détail par la suite, faire évoluer la RTO de façon à satisfaire certains critères pour un document peut avoir des conséquences imprévues sur la satisfaction des critères pour d'autres documents du corpus. C'est pourquoi nous envisageons le processus global de maintenance comme une répétition de ces trois étapes tant que la proportion globale de critères vérifiés sur l'ensemble des documents du corpus n'aura pas atteint un seuil suffisant. Nous faisons ci-après une description plus fine de chacune des trois étapes mentionnées.

4.1.2.1 Evaluation des besoins en maintenance pour une RTO

Dans le cadre d'une utilisation dans un processus de RI sémantique, nous allons évaluer la qualité d'une RTO par son adéquation à l'ensemble des documents sur lesquels s'opèrera le processus de recherche. Ce choix reflète bien un des principaux arguments développés au chapitre 3, selon lequel le processus de modélisation de RTO (et par conséquent, celui de maintenance) est implicitement influencé par les besoins applicatifs. Dans notre contexte d'étude, le besoin essentiel réside dans une bonne indexation sémantique des textes de la base de recherche. En nous fondant explicitement sur l'évaluation des résultats d'indexation, nous acquérons la capacité de mieux contrôler leur influence sur la modification de RTO. Nous rappelons que, dans le cadre de nos travaux, nous représentons les annotations sémantiques d'un document comme un (ou plusieurs) réseau(x) d'instances de concepts reliées par des relations sémantiques. Par exemple, pour l'ontologie que nous avons construite pour le diagnostic automobile, une fiche de réparation est indexée par au moins un réseau d'instances de taille minimale 3 (une instance de type `Symptôme`, une de type `Problème`, une de type `Prestation`[5])

Selon une démarche similaire à [Hernandez, 2005], nous définissons plusieurs critères destinés à évaluer l'adéquation de la RTO aux textes de la base de recherche. L'objectif que nous poursuivons reste néanmoins légèrement différent de cette approche : nous ne cher-

[5]Pour plus de précisions sur ce modèle, voir en 5.1.1.3.

chons pas à déterminer la RTO (parmi plusieurs) la plus adaptée à la tâche d'indexation sémantique, mais à trouver des indices capables de montrer la nécessité (et la façon) de faire évoluer la RTO pour améliorer le processus d'annotation. En ce sens, nos travaux se rapprochent plus de ceux de [Simon *et al.*, 2003]. Présentons maintenant les différents critères que nous avons pu définir au cours de nos recherches.

Un premier groupe d'indices est constitué de critères obligatoires chargés de vérifier qu'aucune contrainte définie dans l'ontologie n'est violée par la base de faits inférée à la suite du processus d'indexation sémantique. Ainsi par exemple, si une relation sémantique est définie entre deux concepts avec une cardinalité minimale de 1 et que la phase d'indexation sémantique permet de découvrir l'existence d'une instance du concept origine dans un certain document, celui-ci devra nécessairement inclure au moins une instance du concept destination sous peine d'être considéré comme un indice de la nécessité de réviser la RTO. De façon plus générale, l'application de cet ensemble de critères correspond en fait à une vérification de complétude de la base d'instances ou de cohérence entre la base d'instances d'une part et la partie conceptuelle de la RTO d'autre part.

Le second ensemble d'indices que nous définissons correspond à des critères facultatifs applicables sur un document qui peuvent être sélectionnés et ajustés par l'ontographe. Parmi eux, on peut notamment citer :

- un **nombre minimal de termes et/ou de concepts** à retrouver dans un document ou une de ses sous-structures[6] permet d'éviter qu'un document soit sous-exploité par l'étape de recherche sémantique du fait d'une indexation trop pauvre.
- définir une **couverture minimale** de l'étape d'indexation (valeur comprise entre 0 et 1 et correspondant au nombre de mots reconnus sur le nombre de mots pleins[7]) sert aussi à assurer que le document en question a été suffisamment exploité au cours de l'indexation sémantique. Selon la nature des textes analysés, la couverture minimale moyenne sera plus ou moins élevée : ainsi, dans le cas de documents fortement informatifs et au style concis, la valeur seuil de ce critère sera bien plus grande que pour des documents plus généraux.
- une **liste de concepts** censés apparaître au moins une fois (voire un nombre donné de fois) dans chaque document. On retrouvera notamment ce critère pour des documents organisés selon une même structure liée au domaine et/ou à la tâche modélisé(e|s). Par exemple, dans notre cas d'étude (voir la description de la base d'expériences en 5.1.1.1), une fiche de réparation se décompose en plusieurs champs pour lesquels on peut facilement émettre des hypothèses quant au contenu sémantique global : on peut s'attendre à retrouver au moins un symptôme dans le champ " constatation ", au moins un composant fautif dans le champ " diagnostic ", et au moins un modèle de véhicule dans le champ " applicabilité ".

Par rapport à ce second groupe de critères, on pourra remarquer qu'ils se fondent partiellement sur un présupposé non anodin, à savoir que tous les documents mis en jeu au cours de la phase de RI sémantique (indexation et recherche) doivent être suffisamment homogènes

[6]Un nombre croissant de documents possède une structure commune qui pourrait être utilisée pour des traitements différenciés : les pages Web, par exemple, comportent toutes au moins un titre et un corps de texte.

[7]Les mots vides ne sont pas pris en compte dans le calcul car le processus d'indexation n'en tient lui-même pas compte dans l'index.

en longueur (i.e. en nombre de mots utilisés) et en structure pour que les mêmes critères puissent s'appliquer à chacun d'eux.

4.1.2.2 Processus simultané de maintenance de RTO et d'indexation sémantique

Après avoir mis en place un ensemble de critères potentiels évaluant la qualité de la RTO en fonction des résultats d'indexation sémantique sur chaque document, voyons maintenant comment utiliser ces critères en interaction avec l'utilisateur pour maintenir la RTO. Plaçons nous dans une situation qui mène potentiellement à une révision de la ressource, à savoir l'ajout de plusieurs nouveaux documents à la base de recherche. Le système possède donc comme entrées (boîtes colorées sur la figure 4.3) la RTO dans son état courant et la base de recherche, composée de documents indexés et de documents non indexés.

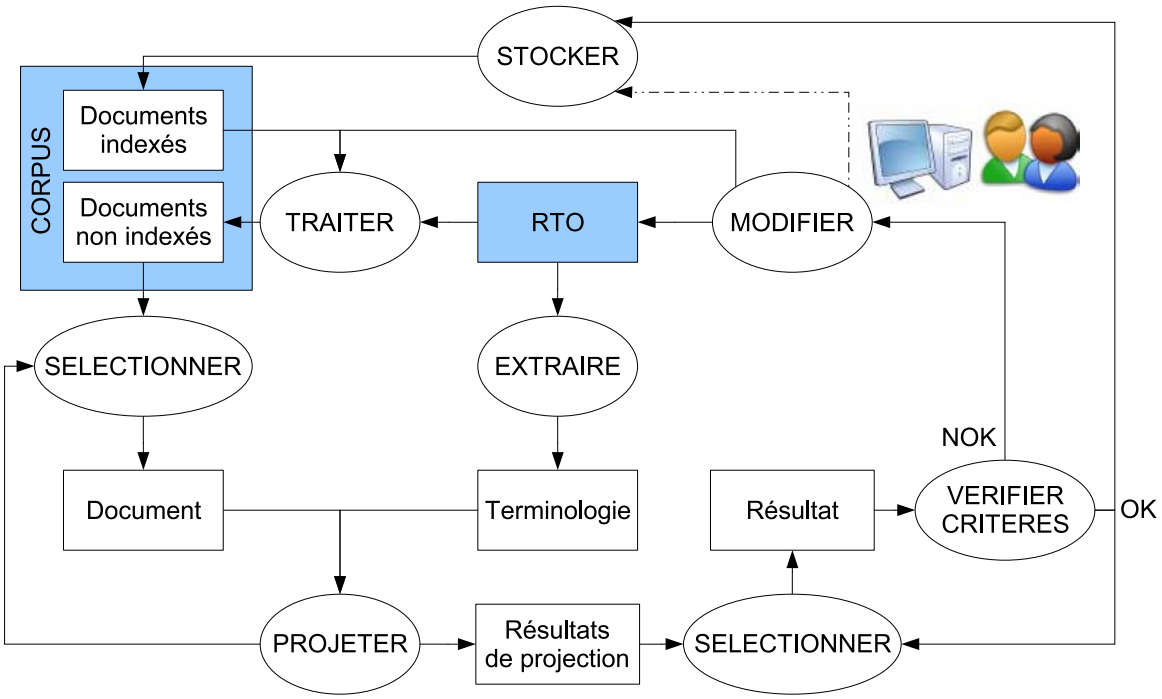

Figure 4.3 — Méthode de maintenance de RTO en RI sémantique

Projection de la RTO sur la base de recherche La première étape consiste à procéder à l'indexation sémantique des documents rajoutés. Pour cela, nous nous fondons à la fois sur la structure du méta-modèle en OWL-DL proposé au chapitre 3 et sur des mécanismes de RI classique. En effet, pour tester progressivement l'utilité de notre méta-modèle, nous avons préféré dans un premier temps limiter son emploi à un processus d'indexation sémantique simple, i.e. destiné uniquement à repérer des manifestations linguistiques de concepts (et non de relations sémantiques ou d'entités nommées). Dans ce cadre, l'indexation sémantique peut facilement se ramener à des techniques de RI classique : pour connaître l'ensemble des concepts mis à contribution dans un document, il suffit (au sens logique) d'y rechercher tout terme dénotant un concept de la RTO. Cette tâche revient alors à indexer

de façon classique les documents de la base de recherche (voir en 2.1.3) puis à rechercher systématiquement dans tous ces textes le label de chaque terme de la RTO.

De façon plus détaillée pour le processus d'indexation classique, nous avons fait le choix de ne pas conduire d'analyse syntaxique de façon à limiter les étapes dépendant de la langue utilisée dans le corpus [8]. Les seules tâches dépendantes de la langue dans notre méthode correspondent à l'emploi d'un anti-dictionnaire qui supprime les mots vides de l'index ainsi que d'un algorithme de radicalisation adapté pour le français de celui proposé par [Porter, 1980]. L'étape suivante consiste à rechercher les occurrences de termes de la RTO dans l'index. Pour cela, nous nous sommes fondé sur un modèle de recherche booléen [van Rijsbergen, 1979]. En effet, nous cherchions à favoriser une bonne précision et qu'un bon rappel est garanti par le protocole même (tous les termes de la RTO sont testés sur chaque document de la base de recherche). Nous avons adapté le modèle booléen simple en rajoutant la contrainte selon laquelle une occurrence de terme n'est considérée comme présente dans un document que si les mots pleins la composant sont tous repérés dans une fenêtre d'un certain nombre de mots (dépendant de la longueur moyenne des textes de la base de recherche). En outre, si deux termes différents T_1 et T_2 possèdent des occurrences "en conflit" (i.e. une occurrence de T_1 correspond à une zone du document incluse dans une zone plus large détectée comme une occurrence de T_2), nous choisissons, à la manière de [Baziz, 2005], de ne garder que l'occurrence de terme qui est la plus longue.

A l'issue de cette phase de projection de la terminologie sur les documents, nous devons alors déterminer quels concepts sont mis en jeu dans chaque document en fonction des termes qui l'indexent. Comme un terme peut potentiellement désigner plusieurs concepts (phénomène de polysémie), il est a priori nécessaire d'envisager une phase de désambiguïsation. Concrètement, ce phénomène est relativement rare dans un domaine spécifique, ce qui nous a amené à diriger nos efforts sur d'autres questions plus problématiques. Pour plus de précisions quant à cette phase, nous conseillons au lecteur de se référer à [Baziz, 2005], qui développe une méthode permettant de désambiguïser le sens d'un terme en fonction des autres termes retrouvés dans le même document. Une fois levée toute ambiguïté, nous devons, en accord avec notre méta-modèle de RTO (cf fig. 3.5), déterminer l'instance à laquelle rattacher chaque occurrence de terme. En effet, une occurrence de terme peut éventuellement dénoter une instance de concept déjà citée auparavant dans le document [9] (phénomène d'anaphore), tout comme elle peut faire référence à une instance de concept n'existant pas encore dans la base de faits (auquel cas, il convient de la créer). Pour plus de simplicité et comme les textes de notre corpus d'application s'avéraient très concis, nous avons fait l'hypothèse que, pour chaque document, il n'existe au plus qu'une seule instance d'un concept donné. Nous avons donc choisi de créer une nouvelle instance d'un concept uniquement si aucune autre instance de ce même concept n'avait déjà été associée à ce document ; dans le cas contraire, nous associons l'occurrence de terme trouvée à l'instance existante du concept adéquat.

Au niveau de la base d'instances, nous obtenons donc un ensemble d'instances de termes

[8] De cette façon, l'adaptation de notre méthode globale à une autre langue devrait en théorie s'avérer plus simple.

[9] Comme nous n'abordons pas l'indexation sémantique dans son aspect "reconnaissance d'entités nommées", nous faisons ici la supposition qu'une instance de concept aura uniquement le document comme portée.

(i.e. leurs occurrences) ainsi que les instances de concepts qui leur correspondent. A la fin de cette étape, les instances de concepts ne sont pas encore mises en relation (e.g. deux instances reliées par une relation sémantique). Cette tâche s'apparente plus à la vérification de contraintes de nature ontologique et sera donc menée pendant la phase suivante.

Application des critères d'adéquation entre RTO et documents La seconde étape du processus simultané de maintenance de RTO et d'indexation sémantique consiste à vérifier automatiquement que les résultats d'indexation respectent, pour chaque document, les critères choisis au préalable par l'ontographe (parmi un nombre minimal d'annotations, une couverture minimale du texte et un ensemble de concepts à retrouver dans chaque document). Lorsqu'aucun critère n'est violé pour un texte, celui-ci est automatiquement considéré comme indexé, tandis que l'algorithme passe à la vérification des critères pour le document suivant.

Lorsqu'un ou plusieurs critères ne sont pas satisfaits, le système peut réagir de deux manières :

- soit il dispose d'assez d'informations pour modifier automatiquement les annotations sémantiques. Par exemple, il se peut qu'une contrainte spécifie que toute instance d'un concept donné doit être associée à une instance d'un second concept via une relation sémantique spécifique. Si deux instances adéquates existent, alors le système peut spontanément créer le lien entre celles-ci. Une autre illustration consiste en la spécialisation du type d'une instance : plaçons nous dans la situation où la phase d'indexation a repéré une instance de concept appartenant au domaine de définition d'une relation rel_{sim} à cardinalité minimale non nulle ; si une instance de concept est retrouvée dans son voisinage, et que le concept mis en jeu est un hypéronyme du codomaine de définition de rel_{sim}, le système peut de lui-même spécialiser le type de l'instance vers le codomaine de la relation, ce qui permet par la suite la création automatique du lien entre les deux instances.
- soit le système ne peut arriver seul à corriger les annotations, auquel cas il doit faire appel à l'ontographe durant une phase de dialogue. Le système lui fournit le texte indexé ainsi que les critères qu'il n'a pu satisfaire. L'ontographe choisit alors d'enrichir ou de modifier la RTO afin que le document puisse être indexé de façon plus satisfaisante au cours du cycle d'indexation suivant. Eventuellement, comme le seuil de certains critères est évalué empiriquement, l'ontographe peut décider de considérer comme correctement indexé un document violant certaines contraintes.

Suite aux modifications apportées à la RTO par le système ou l'ontographe, les documents à l'origine de cette évolution doivent être réindexés et les valeurs de critères associées réévaluées. Toutefois, comme nous l'avons déjà mentionné en 2.2.3, certains documents auparavant considérés comme indexés peuvent aussi, selon les annotations qu'ils contiennent, nécessiter de subir le même traitement. Nous décrivons plus en détail en 4.1.2.3 l'algorithme heuristique que nous avons élaboré pour déterminer quels documents doivent être réintroduits dans le cycle de maintenance en fonction des évolutions de la RTO.

Une fois obtenue la liste de documents potentiellement affectés par la modification de la RTO, le processus consiste à les soumettre à une nouvelle itération du cycle simultané de maintenance de RTO et d'indexation des documents, ceci jusqu'à ce que tous les documents

soient considérés comme correctement indexés par la ressource, qui aura alors atteint un état stable. D'un point de vue théorique, il paraît clair qu'il est impossible de prouver la convergence vers cet état au bout d'un nombre (fini ou pas) d'itérations du cycle. En effet, ce phénomène n'est garanti concrètement que par une homogénéité suffisante des notions abordées dans l'ensemble des documents à indexer. Comme nous l'avons déjà signalé auparavant, nous avons fondé notre démarche sur l'hypothèse préalable que les documents de la base de recherche traitent de problématiques appartenant au même domaine [10] et ont un contenu très stable. Nous considérons de ce fait que la convergence théorique de l'état de la base d'annotations et de celui de la RTO sont assurés. Nous pourrons en voir une illustration pratique ultérieurement.

Bilan de l'approche Pour conclure sur la méthode que nous proposons, nous en donnons son principal avantage pour une application de type recherche sémantique dans une base documentaire : ce processus cyclique permet de faire interagir deux phases jusqu'alors distinctes dans la littérature, à savoir l'indexation sémantique des documents et la maintenance de la RTO. Les deux tâches peuvent alors s'échanger des informations utiles et s'entraider pour la réalisation de leurs objectifs respectifs. De plus, en permettant aux tâches de se recouvrir, le processus global constitue théoriquement un gain de temps pour l'ontographe qui n'intervient que pour une part modérée dans le cycle. Du point de vue des perspectives, en nous inspirant de l'approche décrite dans [Stojanovic *et al.*, 2002], nous pensons ajouter une fonctionnalité supplémentaire au processus, à savoir la possibilité pour l'ontographe de voir les conséquences sur les annotations existantes des modifications qu'il s'apprête à faire sur la RTO de telle sorte qu'il puisse plus facilement contrôler le bénéfice (ou l'effet négatif) de ses actions. De même, la méthode que nous suivons pour déterminer les documents à réindexer suite à une modification de RTO est fondée sur un ensemble d'heuristiques et reste largement perfectible. Nous avons l'intention de nous atteler à cette problématique dans la suite de nos travaux.

4.1.2.3 Gestion des impacts d'évolution de RTO sur les annotations sémantiques

En premier lieu, il convient de définir quelles opérations nous souhaitons rendre possibles dans le cadre d'une modification de RTO, et de savoir pour chacune d'entre elles si elle correspond à une opération élémentaire ou composite (cf 2.2.3). Parmi l'ensemble des objets de la RTO qui peuvent subir des évolutions, nous distinguons trois types :

- les entités ontologiques, à savoir les concepts, leurs instances, les relations sémantiques ainsi que les axiomes
- les objets terminologiques, correspondant aux termes et à leurs instances (i.e. leurs occurrences, selon notre méta-modèle)
- les liens de dénotation entre un terme et un (ou plusieurs) concepts

Sur ces différents types, trois grandes classes d'actions sont possibles : l'ajout d'une nouvelle entité, la suppression, ou la modification. Signalons dès à présent que la modification d'un terme n'est pas envisagée comme une action réalisable par le système. En effet, un terme est

[10] Dit de façon plus rigoureuse, seules les informations à repérer dans un but ultérieur de recherche sémantique doivent concerner un même domaine et/ou une même tâche.

défini par ses attributs initiaux (label, langue, catégorie syntaxique ...) qui ne peuvent pas évoluer au cours du temps. Pour le reste, nous envisageons donc les opérations suivantes :

- l'**ajout de terme**, qui consiste à créer un nouveau terme dans la RTO et à le faire dénoter un concept existant ou pas (auquel cas il faut le rajouter au préalable)
- la **suppression de terme**, qui entraîne nécessairement la suppression de ses occurrences ainsi que du (des) lien(s) de dénotation le liant à un (des) concept(s)
- l'**ajout d'un lien de dénotation** d'un terme vers un concept (tous deux préexistants)
- la **modification d'un lien de dénotation**, qui consiste à changer le concept destination (le lien est par définition lié au terme origine, celui-ci ne peut donc être remplacé) ; cette opération peut être assimilée à une suppression du lien suivie d'un ajout
- la **suppression d'un lien de dénotation** suite à laquelle le terme et concept originellement reliés continuent à exister indépendamment de la disparition de la relation
- l'**ajout de concept**, qui peut comprendre une phase d'association à un ou plusieurs termes existants [11]
- la **modification de concept**, opération de portée relativement vaste (modification d'attributs, déplacement dans la hiérarchie taxonomique ...) et que nous ne détaillerons pas plus dans un premier temps
- la **suppression de concept**, envisagée ici comme n'influant pas sur l'existence d'autres concepts (ainsi, nous rattacherons tous les hyponymes du concept supprimé à son hypéronyme [12]), mais qui entraîne la disparition de tous les liens de dénotation pointant vers le concept
- les opérations d'**ajout/modification/suppression sur les autres entités ontologiques** (instances, relations sémantiques et axiomes)

On remarque que l'ajout et la suppression de terme ou de concept sont des opérations composites car elles intègrent un certain nombre d'actions élémentaires (les autres cas cités). Nous envisageons donc une gestion spécifique des conséquences de ces opérations d'évolution. Comme nous l'avons déjà mentionné, nous avons préféré privilégier une approche heuristique pour la politique de gestion des annotations suite à une évolution terminologique et/ou ontologique. Le fait d'envisager un environnement intégrant l'ensemble des étapes du cycle global de maintenance nous permet de traiter les conséquences des modifications de RTO sur la base d'annotations au fur et à mesure de leur apparition (ce qui évite notamment de devoir tenir un journal des événements survenus sur la RTO). Nous résumons dans le tableau 4.1 notre algorithme d'évolution des annotations pour chacune des opérations de modification de RTO envisagées.

Succinctement et pour permettre au lecteur une meilleure appréhension de nos heuristiques, nous pouvons en faire une synthèse en quelques points principaux :

- la phase de "réindexation" ne se produit généralement qu'à une seule occasion (l'ajout de terme) et consiste en la projection du nouveau terme (et seulement lui) sur l'ensemble des documents, de façon à minimiser le temps de traitement,
- si un concept ou un lien de dénotation est créé en relation avec un terme existant (qui a donc déjà été projeté sur l'ensemble des documents), il est inutile de réindexer les

[11] Cet ajout de lien n'est pas systématique, le concept ajouté peut en effet jouer un rôle structurant et ne posséder aucune manifestation linguistique propre.

[12] Dans nos travaux, nous ne nous sommes pas penchés sur les ontologies à héritage taxonomique multiple.

Objet de RTO	Opération	Répercussions sur les annotations
Terme	Ajout	– Projection du terme sur tous les documents – Ajout des documents indexés avec le terme à la liste de vérification des critères
	Suppression	– Suppression des occurrences du terme et des instances de concept dénotées uniquement par une (partie) de celles-ci – Ajout des documents indexés avec le terme à la liste de vérification des critères
Lien de dénotation	Ajout	∀ document indexé par le terme origine du lien : – Ajout d'une instance du concept destination (si aucune n'existe déjà) – Ajout du lien entre la ou les occurrences du terme et l'instance de concept – Ajout du document à la liste de vérification des critères
	Suppression	– Suppression des instances du concept destination uniquement dénotées par des occurrences du terme origine – Ajout des documents concernés à la liste de vérification
Concept	Ajout	Si le concept est associé à un ou plusieurs termes : – Ajout d'une instance du concept à tout document indexé par un des termes associés – Ajout du document à la liste de vérification
	Modification	Ajout de tout document indexé avec à la liste de vérification des critères
	Suppression	– Suppression de toute instance de concept – Ajout des documents précédemment indexés avec à la liste de vérification
Autres entités ontologiques	Ajout Modification Suppression	– Si l'entité modifiée participe à la définition d'un critère d'évaluation, réindexation de tous les documents – Sinon simple vérification de la cohérence globale de la partie ontologique

Tableau 4.1 — Gestion des annotations sémantiques en fonction des modifications de RTO

documents dans lesquels le terme est présent, il suffit de créer une instance du concept adapté reliée aux documents correspondants (ou simplement le lien entre occurrence et instance)
– les critères d'évaluation sont recalculés pour tout document qui ne respectait pas tous les critères au cours du cycle précédent ou dont les annotations ont changé

Le cas d'une modification sur une entité ontologique qui n'est pas un concept est assez particulier. A priori, deux cas de figure se dégagent : soit l'entité participe directement à la définition d'un ou plusieurs critères d'évaluation de la RTO, soit elle en est totalement indépendante. La seconde situation est la plus simple, dans le sens où la modification opérée n'impactera en rien sur l'évaluation ultérieure des critères : il suffit alors de vérifier la cohérence globale de l'ontologie et de réévaluer les critères des documents dont l'index sémantique a évolué. Le premier cas est plus délicat : la façon dont évolue l'entité peut avoir des conséquences imprévues pour l'évaluation des critères sur l'ensemble des documents. Par exemple, si un critère d'évaluation de la RTO spécifie qu'un document doit contenir un certain nombre d'instances d'un concept particulier et que l'on déplace ce concept dans la hiérarchie taxonomique (avec rattachement des hyponymes à l'ancien hypéronyme), certains documents qui sont indexés par les anciens hyponymes du concept en question peuvent désormais violer une contrainte qui était auparavant respectée[13]. Comme les conséquences d'une telle modification semblent hasardeuses et nous paraissent difficilement prévisibles, nous avons préféré émettre une heuristique prudente consistant à réindexer l'ensemble des documents.

Nous pouvons enfin aborder un dernier point relatif aux modifications manuelles des annotations sémantiques. En effet, si certaines annotations sémantiques peuvent être automatiquement produites par l'algorithme d'indexation grâce aux indications contenues dans les critères d'évaluation de la RTO, d'autres annotations restent à la discrétion de l'ontographe. Toutefois, en cas de défaillance des heuristiques, l'ontographe peut souhaiter corriger manuellement certains résultats produits automatiquement. Il peut aussi se fonder sur les annotations automatiques pour rajouter de l'information sémantique (en reliant deux instances de concept par une relation donnée, par exemple). Un problème se pose alors : lorsque le système réindexe un document dont les annotations ont été manuellement modifiées, comment doit-il se comporter face à celles-ci ? Doit-il les ignorer ? Les faire évoluer pour qu'elles restent cohérentes avec les nouvelles annotations automatiques ? De quelle façon ? Actuellement, cette question reste en suspens dans nos travaux. Nous avons décidé de façon temporaire de suivre une heuristique simple : si au cours d'un cycle d'indexation, on retrouve les mêmes annotations sémantiques qui ont été manuellement modifiées au cours du cycle précédent, alors les modifications manuelles sont conservées ; dans le cas contraire, les annotations manuelles sont perdues car le système recalcule toute les annotations sémantiques de ce document. Une perspective à nos travaux consisterait à affiner cette heuristique de façon à ce que le système soit capable de faire coexister plus efficacement les deux types d'annotation sémantique (i.e. automatique et manuelle).

[13] Fort logiquement, l'inverse est totalement possible.

4.2 Appariement sémantique entre une requête en langage naturel et une base documentaire indexée

Après nous être intéressé à la mise en place d'une synergie entre les étapes de construction/maintenance et d'indexation sémantique, nous nous focalisons dans cette deuxième partie sur l'exploitation des index sémantiques au cours d'un processus de RI. Nous supposons les documents annotés à l'issue du processus décrit en 4.1, les annotations étant représentées selon le méta-modèle présenté en 3.2.2. Dans cette partie, nous commençons par détailler le procédé voisin d'indexation sémantique d'une requête exprimée par un utilisateur en langue naturelle. Nous abordons ensuite la problématique d'évaluation de la proximité sémantique entre cette requête et un document tout deux préalablement indexés. Comme nous n'apportons pas de contribution théorique à la phase de restitution à l'utilisateur des documents pertinents pour une requête, nous n'en ferons aucune description dans cette section.

4.2.1 Traitement semi-automatique de la requête

Dans le contexte d'un moteur de recherche sémantique et malgré un objectif similaire, les techniques mises en œuvre pour annoter sémantiquement les requêtes de l'utilisateur ne peuvent être strictement identiques à celles prévues pour l'indexation des documents. En effet, la phase d'interaction décrite en 2.1.2 doit être adaptée à l'opérateur humain : dans le cas d'un document, le système dialogue avec l'ontographe qui a une connaissance approfondie de la structure de la RTO (puisqu'il en est le co-créateur) et des critères évaluant l'adéquation de cette ressource aux textes à indexer (c'est lui qui les définit au cours de la construction ou au début de la maintenance) ; dans le cas d'une requête, celle-ci est formulée dans une phase ultérieure par un simple utilisateur, qui possède un certain nombre de connaissances sur la tâche et/ou le domaine en question, sans pour autant avoir forcément accès à la RTO utilisée par le moteur de recherche.

D'un point de vue ergonomique, un système de RI sémantique idéal doit proposer à l'opérateur, dans un temps acceptable, des résultats au moins aussi bons que ceux d'un système classique [14], tout en suivant une démarche intelligible qui minimise autant que possible la part de travail à effectuer par l'utilisateur. Au niveau de la saisie des requêtes, différents types d'interfaces ont été envisagés dans la littérature :

- l'emploi de **langages dédiés à l'expression de requêtes** comme SPARQL, recommandé par le W3C [Prud'hommeaux et Seaborne, 2008] pour l'interrogation d'entrepôts RDF, ou comme le formalisme de requête de Corese [Corby *et al.*, 2004] adapté au formalisme des graphes conceptuels [Sowa, 1984]
- la formulation de requêtes par **construction de graphes**, comme dans [Guarino *et al.*, 1999]
- la saisie de requête **à base de formulaires** [15], solution envisageable à la condition ex-

[14] L'évaluation des résultats doit prendre en compte à la fois des critères absolus liées à la qualité intrinsèque de l'outil et des critères relatifs liées au ressenti des utilisateurs.

[15] Un formulaire peut contraindre de plusieurs manières l'expression d'une requête : dans sa forme globale, dans le type de concepts et/ou de relations employées. Dans le même ordre d'idée, les champs du formulaire

presse de connaître au préalable la nature précise des besoins de l'utilisateur (projet PICSEL, [Rousset et Reynaud, 2004])
- l'expression de la requête par une **question en langage naturel**, telle que la conçoit [Tran *et al.*, 2007], permet de proposer - au prix d'importants traitements automatiques de langue naturelle - un système permettant à l'utilisateur de formuler des besoins de nature précise et ne requérant de sa part aucune compétence ou connaissance particulière quant au modèle sémantique en arrière-plan
- l'utilisation d'un **langage libre** (voir la définition en 2.1.2.1) est une approche relativement populaire, comme tendent à le prouver les travaux de [Guha *et al.*, 2003], [Rocha *et al.*, 2004] ou [Lei *et al.*, 2006]. Ce type d'interfaces a l'avantage de ne pas nécessiter d'investissement cognitif de la part des utilisateurs qui ont l'habitude de manipuler des moteurs de recherche classiques comme Google ou Yahoo. A l'inverse d'une rédaction en langue naturelle, il convient bien à des situations où les besoins en information sont mal définis (car ceux-ci rendent alors difficile toute formulation exacte).

On remarquera que le choix d'une interface est nécessairement un compromis entre le degré de compétence de l'utilisateur à employer un langage de requête particulier et la complexité des traitements de formalisation des requêtes : plus la syntaxe du langage de requête utilisé est flexible, plus grand est le risque de mal interpréter la requête (ambiguïté, implicite ...), ce qui incite à la mise en place de mécanismes interprétatifs efficaces et exhaustifs.

Ne souhaitant pas présumer des compétences informatiques des utilisateurs, nous avons décidé de proposer à l'utilisateur une interface de base minimale, identique à celle d'un moteur de recherche classique dont il a l'habitude, de façon à garantir une bonne acceptabilité de l'outil. Nous n'avons pas retenu la solution à base de langue naturelle parce que nous avons jugé les traitements inhérents à cette approche trop longs (en termes de temps de réponse du système), trop contraignants [16] et que nous avons pu constater sur un échantillon de personnes que celles-ci ne prennent pas forcément le temps de rédiger des requêtes grammaticalement correctes et/ou commettent des fautes d'orthographe.

L'utilisateur formule donc librement sa requête dans notre interface en juxtaposant un ensemble de mots-clés lui paraissant importants. La requête est alors indexée durant une phase de projection de RTO (similaire à celle décrite en 4.1.2.2), suivie par l'application d'un ensemble d'heuristiques créant les liens nécessaires (i.e. de cardinalité minimale non nulle pour une paire de concepts donnée) entre instances. A l'issue de cette étape, deux situations peuvent se présenter :
- si les instances de terme et de concept liées à la requête ne créent aucune incohérence dans la RTO, le système considère que la requête a été correctement indexée et passe donc à la phase d'appariement avec les documents.
- dans le cas contraire, un problème est détecté et un message d'erreur est retourné afin d'amorcer un dialogue à but correctif avec l'utilisateur.

Dans les deux situations, il est souhaitable que l'utilisateur puisse connaître l'interprétation que fait le système de sa requête. En effet, le respect de la cohérence de la RTO n'est pas une condition suffisante pour s'assurer que le système interprète correctement la requête.

peuvent être libres ou contraints par une liste de valeurs possibles

[16] Notamment dans la perspective d'une recherche sémantique multilingue, il nous a semblé malvenu de nous reposer sur des techniques spécifiques à la grammaire d'une langue.

En visualisant cette interprétation, l'utilisateur pourra ainsi juger de sa qualité et - si besoin est - interagir avec le système pour la corriger.

Pour présenter son interprétation à l'utilisateur, le moteur de recherche fait appel à la base d'annotations correspondant à l'ensemble des documents indexés. Pour chaque groupe d'instances reliées dans la requête, il recherche les groupes de termes associés les plus représentés dans tous les documents. Par exemple, dans le domaine des vertébrés marins, l'utilisateur peut chercher des pages relatives au régime alimentaire de l'orque (fig. 4.4). S'il entre les mots "*alimentation orque*", le système va retrouver la trace des concepts `Orque` et `Régime_alimentaire` reliés par la relation `mange` et il va reformuler la requête en "*régime épaulard*", si tant est que cette paire de termes synonymes s'avère la plus utilisée [17] (et donc potentiellement la plus consensuelle). On voit ici encore que la méthode que nous proposons a pour avantage d'exploiter les liens explicites de dénotation entre termes et concepts de notre méta-modèle.

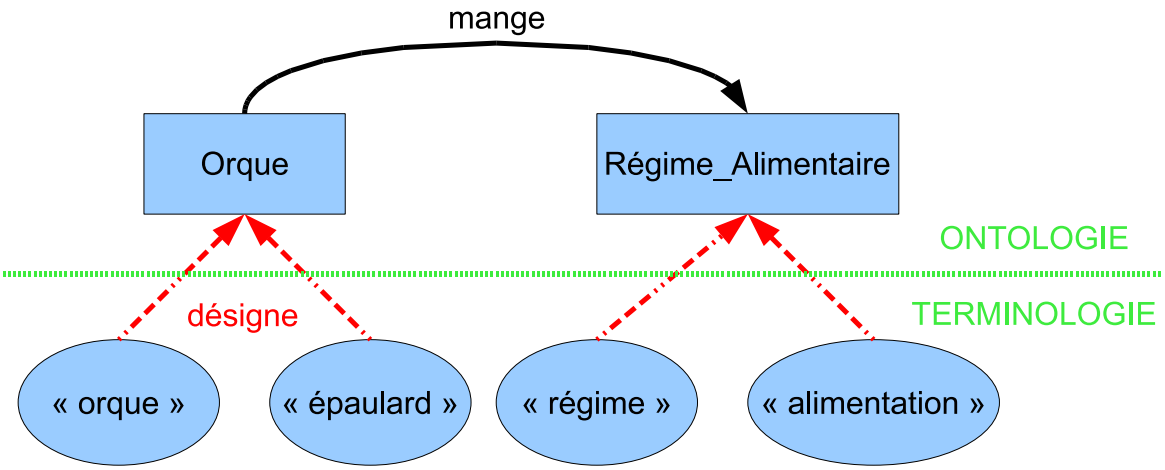

Figure 4.4 — Exemple d'utilisation de la synonymie pour la reformulation de requête

Si un problème a été détecté pendant la phase d'indexation sémantique ou si l'utilisateur n'est pas satisfait de la manière dont sa requête a été interprétée, le système doit envisager une étape corrective. Celle-ci doit être totalement efficace et obtenir directement un résultat satisfaisant pour l'opérateur sans avoir à être répétée (car cela pourrait provoquer à long terme une lassitude de l'utilisateur et un abandon de l'outil). Selon la structure des requêtes posées à l'outil, deux méthodes de correction sont prévues :

– Si toutes les requêtes reposent sur une structure similaire (i.e. sont systématiquement composées des mêmes types de concepts reliés entre eux par les mêmes relations), le système peut gérer la correction de l'interprétation des requêtes avec un simple formulaire qui permet d'abord de sélectionner dans une liste déroulante un (ou plusieurs) concept(s) parmi les hyponymes d'un concept apparaissant nécessairement dans la requête [18] ; une fois les concepts sélectionnés, le système crée les instances adéquates,

[17] Si la formulation originelle de la requête correspond déjà aux termes les plus fréquents, le système reformulera avec les seconds termes les plus utilisés.

[18] Naturellement, tout concept est symbolisé dans la liste par le terme le dénotant le plus fréquemment dans la base de recherche.

tente de les relier automatiquement selon les mêmes heuristiques déjà utilisées pendant la phase automatique et il présente les résultats éventuels à l'utilisateur. Celui-ci peut alors corriger les relations et/ou en ajouter certaines via une interface à base de menus déroulant regroupés en triplets (pour le concept origine, la relation sémantique et le concept destination).

– Si les requêtes n'ont aucune unité de structure, il peut être plus intéressant d'employer une interface adaptée pour un accès direct à l'ontologie sous-jacente. Plutôt que tenter de corriger l'interprétation de la requête, le système bascule alors dans une phase de navigation dans l'ontologie, au cours de laquelle il exploite les concepts sélectionnés (et leurs relations mutuelles éventuelles) pour réduire progressivement l'ensemble des documents présentés (initialement constitué de toute la base de recherche) à ceux pertinents pour l'utilisateur. Nous n'abordons pas plus cette problématique qui relève du domaine de l'exploration d'informations et sur laquelle nous n'apportons aucune réelle contribution. Le lecteur intéressé pourra néanmoins se référer à [Fluit *et al.*, 2003], [Mutton et Golbeck, 2003] ou [Mothe *et al.*, 2003].

De façon à améliorer les phases automatiques ultérieures d'indexation sémantique des requêtes, le moteur de recherche peut, à la manière de [Cimiano *et al.*, 2007], stocker sur un serveur centralisé toute requête dont l'interprétation a été jugée défaillante par un utilisateur ainsi que les annotations sémantiques finalement retenues. De cette façon, durant l'étape de maintenance suivante [19], l'ontographe pourra étudier les décalages entre le comportement réel du système et celui attendu par les utilisateurs afin d'en déduire un ensemble d'évolutions possibles pour la RTO : alignement du lexique de la RTO à celui de l'utilisateur, mise à jour des liens entre termes et concepts, ajout de concepts plus spécifiques ...

4.2.2 Comparaison sémantique de réseaux d'instances conceptuelles

Intéressons-nous à présent à la seconde phase d'un processus de recherche sémantique : après la saisie de la requête par l'utilisateur et son interprétation par le système, celui-ci doit évaluer sa proximité avec chacun des documents de la base de recherche, en fonction des annotations sémantiques relevées et de la structure ontologique de la RTO. Nous avons présenté en 2.2.2.3 plusieurs approches destinées à comparer deux instances de concepts (voire deux réseaux d'instances) et en avons analysé les principales limites. Nous nous proposons ici de mettre en place une mesure qui intègre plusieurs aspects de chacune de ces méthodes, tout en gérant de façon innovante certains points supplémentaires.

Si nous reprenons de façon synthétique les avantages et inconvénients des approches de la littérature, nous pouvons en extraire plusieurs propriétés valables pour la mesure que nous cherchons à mettre en place (voir fig. 4.5) :

– cette mesure doit pouvoir prendre en charge des requêtes et des documents indexés par **plusieurs instances de concepts** qui sont potentiellement **reliées entre elles** par des relations sémantiques ;

[19] Le lancement d'une phase de maintenance peut être décidé arbitrairement par l'ontographe ou déclenché automatiquement suite à une insatisfaction trop grande des utilisateurs, une indexation sémantique défectueuse (lors de l'ajout de nouveaux documents à la base de recherche) ou du fait d'une évolution des besoins applicatifs (cf fig. 4.1).

- cette mesure doit permettre la **comparaison d'instances de concepts différents** (si celle-ci s'avère pertinente, naturellement) ;
- cette mesure doit **prendre en compte explicitement** les différentes informations potentiellement communes associées aux deux instances comparées, à savoir leurs **attributs**, les **relations sémantiques** dont elles sont sujets, ainsi que les **autres instances** auxquelles elles peuvent être **reliées** ;
- notre mesure doit garantir un certain degré de liberté au niveau de l'appariement structurel, i.e. **pouvoir retenir** comme pertinents des documents ne comportant **pas exactement la même organisation structurelle** de relations sémantiques entre instances que la requête ;
- nous devons convenir de la **pertinence relative**, pour une requête donnée, d'un document dont les annotations sémantiques possèdent une structure différente de celles de la requête, par rapport à un document dont la **structure relationnelle** des annotations correspond parfaitement.

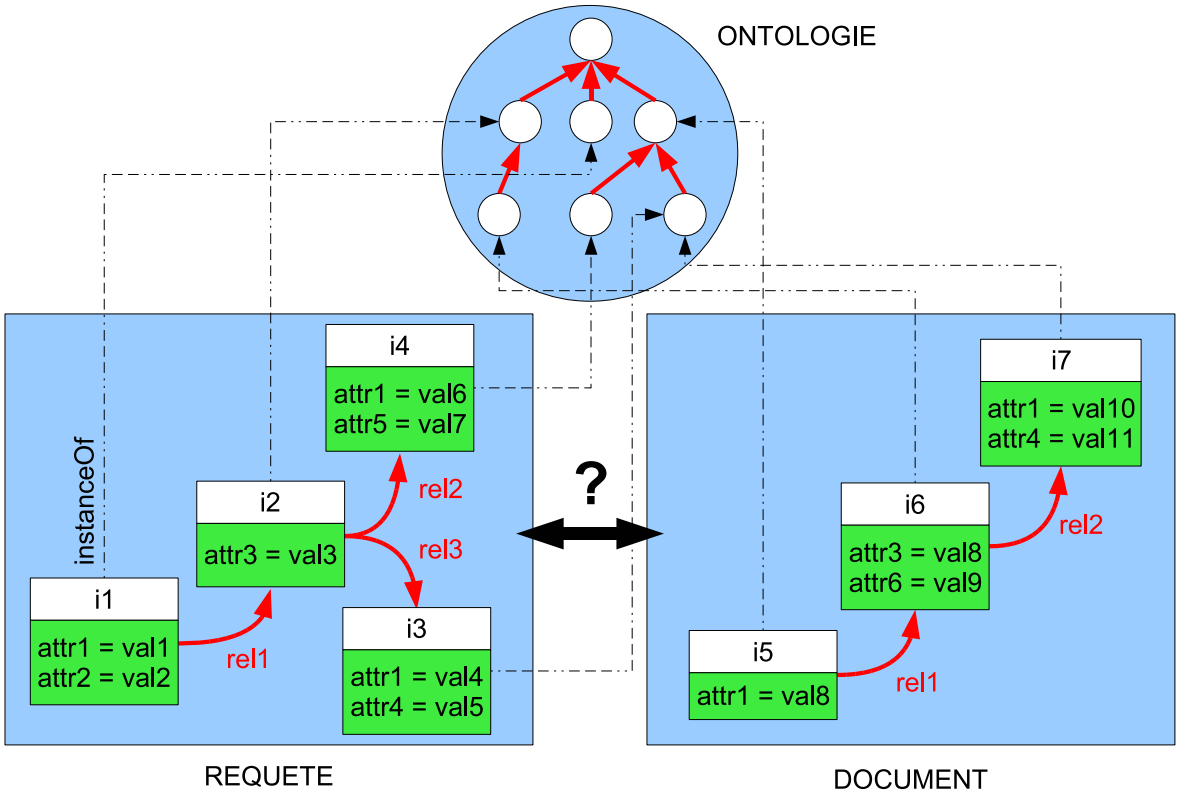

Figure 4.5 — Illustration du problème d'appariement sémantique

Par la suite, nous nous intéressons plus particulièrement aux deux derniers points soulevés. Comme l'illustre la figure 4.5, le calcul d'une proximité sémantique entre une requête et un document nécessite de savoir apparier les "bonnes" paires d'instances (i.e. maximisant le résultat final). Cette phase préliminaire n'est pas obligatoire, puisqu'il est toujours possible de tester toutes les combinaisons de paires d'instances possibles ({($i1$,$i5$) ; ($i2$,$i6$) ; ($i3$,$i7$)} ou {($i1$,$i5$) ; ($i2$,$i6$) ; ($i4$,$i7$)} ou ...) et de prendre la meilleure de toutes. Comme ce problème se ramène à la recherche des plus grands sous-graphes partiels isomorphes entre le graphe de

la requête et celui du document [Bisson, 2000], il fait partie de la classe des problèmes NP complets. Après avoir introduit la notion de comparabilité pour le calcul de similarité entre deux entités ontologique, nous proposons quelques heuristiques capables d'aider à la tâche d'appariement sémantique.

4.2.2.1 Comparabilité de deux concepts

Au sein d'une ontologie, même si plusieurs mesures de similarité conceptuelle se fondent sur la forme du réseau taxonomique (voire des relations transverses entre les deux concepts comparés), il semble peu raisonnable de disposer d'une seule mesure permettant de comparer tous les concepts. En effet, selon les critères de différenciation choisis à chaque niveau de la taxonomie pour distinguer plusieurs sous-concepts, il est parfois déraisonnable de chercher à comparer deux concepts issus de branches taxonomiques différentes. Ce phénomène est d'autant plus pertinent que l'hyperonyme le plus spécifique commun au deux concepts se trouve haut dans la hiérarchie : par exemple, si on considère une ontologie des aliments, il est plus difficile de comparer une poire (aliment végétal) à un steak (aliment animal) plutôt qu'à une pomme. Ceci s'explique par le plus faible nombre de critères de rapprochement entre les deux entités. Comme nous l'affirmons en 3.1.3.3, ces critères dépendent des besoins applicatifs. Il peut donc être utile de définir en parallèle à l'ontologie un paramètre booléen de comparabilité qui repère les sous-branches taxonomiques qui possèdent une homogénéité faisant sens pour le calcul de similarité entre deux concepts d'une même branche. Ainsi, dans le cas d'une ontologie alimentaire, on pourrait restreindre l'évaluation de la similarité conceptuelle sur la branche d'aliment animal d'une part et sur la branche d'aliment végétal d'autre part (i.e. qu'on ne pourra comparer entre eux deux aliments uniquement s'ils sont tous deux de type animal ou de type végétal).

A partir de cette hypothèse, on voit apparaître la possibilité de définir une similarité conceptuelle locale à chacune des différentes branches comparables : ainsi, on pourra utiliser une mesure de type [Lin, 1998] sur les aliments d'origine animale, tandis qu'on emploiera une mesure de type [Jiang et Conrath, 1997] pour les aliments végétaux. Cette propriété devient d'autant plus intéressante lorsque les mesures de similarités locales se fondent sur des heuristiques (découpage plus efficace et plus logique en autant de similarités que de sous-ensembles comparables).

Nous concluons sur cette notion par une remarque sur les cas où la similarité entre deux concepts est nulle. De fait, ce cas de figure est différent de celui de deux concepts non comparables : les deux concepts sont comparables selon certains critères mais ils sont en tout point opposés. Par exemple, sur notre domaine du diagnostic automobile, nous avons modélisé un symptôme dans l'ontologie comme un problème affectant une prestation ; il n'y aurait aucun sens à comparer un problème avec une prestation, mais au sein de la taxonomie des prestations, nous pouvons constater que certaines d'entre elles ont une similarité nulle (e.g. motorisation et climatisation) car elles jouent des rôles fonctionnels trop différents pour servir à rapprocher deux symptômes concernant chacun l'une des deux. Par la suite, l'avantage de représenter explicitement la non-comparabilité entre une prestation et un problème permet de réduire la complexité du processus d'appariement sémantique en ne testant pas les combinaisons entre paires d'instances de type $(i_{probleme}, i_{prestation})$.

4.2.2.2 Heuristiques pour l'appariement sémantique

En restant sur une analogie entre annotations sémantiques et graphes orientés, nous considérons qu'une requête (ou un document) peut être indexée par un ou plusieurs graphes d'instances de concepts de taille variable. Comme nous cherchons à prendre en compte explicitement dans le calcul final les relations sémantiques existant entre les instances, nous fondons nos heuristiques sur la forme et la nature des liens du réseau d'instances de la requête.

Nous reprenons tout d'abord de [Zhong *et al.*, 2002] l'idée, à l'issue de la phase d'indexation sémantique de la requête, de savoir isoler une ou plusieurs entrées importantes du graphe associé. Intuitivement, ces instances de concepts centrales correspondent à l'information sémantique devant être retrouvée en priorité dans les documents. Soit l'utilisateur est capable d'indiquer ces instances de lui-même (par exemple en surlignant les mots qu'il juge essentiels), soit le système est utilisé pour un besoin assez bien circonscrit pour savoir a priori quel type d'information sémantique est systématiquement recherché : dans notre contexte d'étude, comme nous élaborons un système recherchant des documents de réparation en fonction des symptômes constatés, nous en déduisons que l'information centrale dans la requête est représentée par toutes les instances de symptôme trouvées dans la requête.

Suite à l'introduction de la notion d'entrée de graphe, plusieurs cas de figure peuvent se présenter : un graphe donné de la requête peut comporter aucune, une seule, ou plusieurs instances centrales de concept. Au sein d'une même requête (considérée comme une conjonction de graphes mutuellement indépendants), on cherchera à apparier en priorité les graphes incluant une ou plusieurs instances centrales, puis par ordre décroissant de taille (i.e. nombre d'instances / nœuds). L'heuristique d'appariement sémantique d'un graphe de requête donné dépend du nombre d'instances centrales incluses :

- si le graphe de requête ne comporte **aucune instance centrale**, on favorisera l'appariement de relations obligatoires. Nous définissons une **relation obligatoire** comme une relation qui comporte une contrainte de cardinalité minimale sur tout ou partie de son domaine de définition (e.g. la relation d'affectation dans le cas d'un symptôme de panne formalisé par un problème affectant une ou plusieurs fonctionnalités d'un véhicule). A l'inverse, une **relation facultative** n'est pas contrainte en cardinalité (e.g. tout téléphone portable peut capter plusieurs réseaux d'opérateur, mais la cardinalité de cette propriété n'est pas garantie en toute circonstance). Dans le cas d'un graphe sans instance centrale, nous commençons donc par traiter les chaînes de relations obligatoires par ordre décroissant de longueur en repérant, au sein de chacune, la plus longue sous-chaîne appariable à un sous-graphe du document comparé. En effet, en cas de détection d'une paire (i_1^{req}, i_2^{doc}) d'instances de concepts compatibles (i.e. dont les types sont inclus dans le domaine de définition de la relation obligatoire rel_{oblig} qu'ils partagent), la contrainte de cardinalité minimale sur ce type de relation assure l'existence d'au moins une instance i_4^{doc} à apparier avec i_3^{req} car reliée à i_2^{doc} par rel_{oblig} de la même façon que i_3^{req} est reliée à i_1^{req} (cf fig. 4.6).
- si le graphe de requête possède **une unique instance centrale** i_0, on commencera par aligner i_0 à i_1, l'instance de concept du document la plus proche en fonction d'une

similarité locale Sim_{loc} abordée plus loin. L'algorithme suit dans ce cas une heuristique consistant à considérer en priorité pour le calcul de $Max_k[Sim_{loc}(i_0, i_k)]$ toute instance compatible avec i_0 vis-à-vis d'un maximum de relations obligatoires. Indirectement, du fait de l'héritage des restrictions de propriétés, cette instance aura également comme caractéristique un type proche de celui de i_0 dans la taxonomie.

– si le graphe de requête étudié comporte **plusieurs entrées centrales**, nous cherchons à retrouver dans le document un graphe compatible avec le sous-graphe de la requête joignant les différentes instances centrales (en priorité via des relations obligatoires ainsi qu'en minimisant le chemin parcouru).

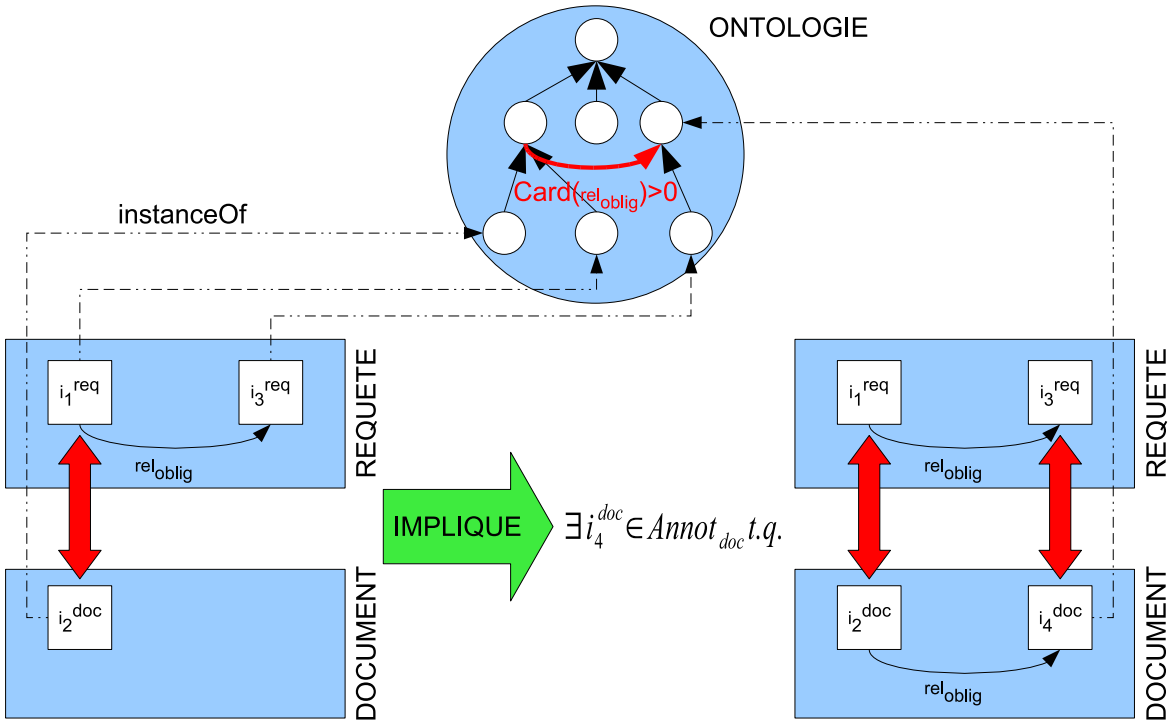

Figure 4.6 — Illustration de l'heuristique des relations obligatoires

4.2.2.3 Description de la similarité locale entre instances

Avant de pouvoir calculer une proximité globale entre une requête et un document, nous définissons une mesure de similarité locale Sim_{loc} entre deux instances de concepts. En accord avec la notion de comparabilité introduite plus haut, cette similarité locale, comprise entre 0 et 1, n'a de sens qu'entre instances de deux concepts comparables. De façon arbitraire, on peut tout de même lui attribuer la valeur -1 si les concepts mis en jeu n'appartiennent pas à deux branches taxonomiques comparables.

Pour comparer deux instances, nous avons la volonté d'utiliser un maximum de sources d'information susceptibles de les rapprocher ou de les éloigner, à savoir :

– la **proximité entre les deux concepts instanciés** Sim_{cpt} peut être prise en compte de différentes manières, comme en témoignent les mesures de la littérature [Wu et Palmer, 1994], [Jiang et Conrath, 1997] ou [Lin, 1998] ;

- la **comparaison des valeurs d'attributs communs** aux deux instances, Sim_{attr}, peut être mise plus ou moins facilement en œuvre selon le type de données manipulé : s'il existe une relation d'ordre total sur celui-ci (e.g. domaine des réels, des entiers), la similarité correspond au complémentaire à 1 d'une distance normalisée [20], sinon on peut adopter la mesure associant une similarité maximale sur l'attribut en cas de valeurs égales, nulle dans le cas contraire ;
- l'**influence relative des paires d'instances reliées** à la paire comparée, Sim_{rel}, correspond grossièrement à la moyenne des similarités locales entre les paires reliées et peut donc être calculée de façon récursive.

La similarité locale telle que nous la définissons est alors une combinaison linéaire de ces trois similarités :

$$Sim_{loc}(i_1, i_2) = \alpha * Sim_{cpt}(i_1, i_2) + \beta * Sim_{attr}(i_1, i_2) + \gamma * Sim_{rel}(i_1, i_2)$$

avec $\alpha + \beta + \gamma = 1$

Comme de nombreuses études ont déjà été menées sur la définition d'une similarité conceptuelle (cf 2.2.2.2), nous nous intéressons dans cette partie à définir plus précisément la similarité relationnelle Sim_{rel}. Pour cela, nous reprenons un exemple similaire à celui avec lequel nous critiquions le parti-pris de [Bisson, 1993] : dans le cadre de nos recherches sur le domaine du diagnostic automobile, nous modélisons un symptôme comme la présence nécessaire d'un problème sur une prestation du véhicule, avec éventuellement un ou plusieurs contextes d'environnement (e.g. moteur chaud, véhicule à l'arrêt, en accélération ...). La comparabilité n'est valable qu'entre deux concepts issus tous deux de la même hiérarchie taxonomique des symptômes, ou [21] de celle des problèmes, ou de celle des prestations, ou encore de certaines sous-hiérarchies incluses dans la représentation des contextes [22]. Puisque l'objectif applicatif consiste à trouver des fiches de réparation à partir de la description d'un ou plusieurs symptômes, le schéma global d'une requête met l'accent sur les concepts de type symptôme. Toute instance de ce type correspond donc dans la requête à un nœud central. Comme on peut le voir sur la figure 4.7, il existe quatre relations sémantiques différentes, toutes obligatoires (trait continu) à l'exception du lien `survient_dans` (trait pointillé).

Nous souhaitons aller dans le même sens que [Bisson, 1993] qui considère que pour comparer deux instances, il faut tenir compte de la similarité des instances qui leur sont reliées. Ainsi, dans le contexte du diagnostic automobile, deux problèmes donnés seront d'autant plus proches qu'ils affectent des prestations relativement similaires dans des contextes comparables relativement semblables. Nous définissons alors la similarité relationnelle entre deux instances i_1 et i_2 comme la combinaison linéaire pondérée (selon l'importance relative du lien suivi) des similarités locales entre les paires d'instances comparables reliées via une même relation sémantique (ou avec la relation sémantique dans le document spécialisation de celle dans la requête) à la paire (i_1, i_2) et telles qu'elles maximisent la similarité relationnelle. En effet, si le document à comparer à la requête mentionne un problème affectant deux

[20] On peut normaliser la distance entre deux valeurs d'attribut en la divisant par exemple par le plus grand écart de valeurs existant sur cet attribut dans la base d'instances.

[21] Il s'agit naturellement d'un "ou" exclusif.

[22] En effet, il n'est pas pertinent de comparer deux contextes comme *véhicule en accélération* et *moteur chaud*.

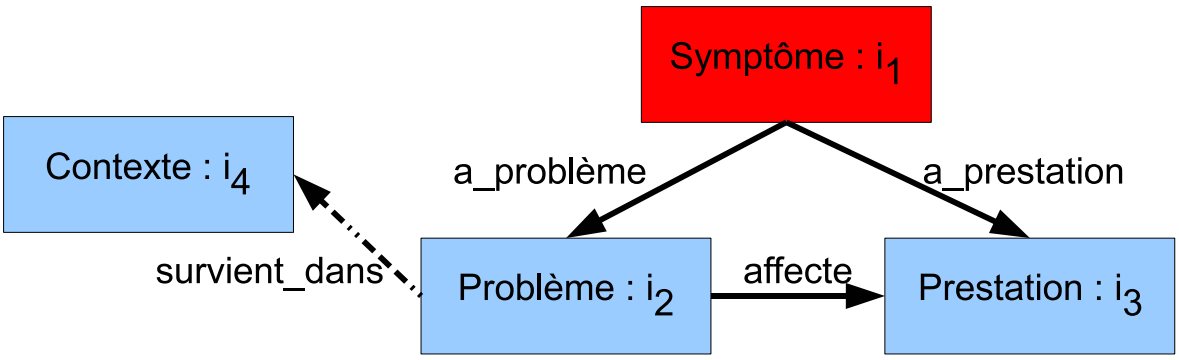

Figure 4.7 — Représentation graphique d'un symptôme automobile

prestations, nous considérerons les deux possibilités d'appariement pour ne garder que celle qui maximise la valeur de $Sim_{rel}(i_{pb}^{req}, i_{pb}^{doc})$.

Un phénomène supplémentaire doit toutefois être pris en compte. Dans le cas de relation facultatives (e.g. la présence d'un contexte dans le document qui soit comparable avec celui mentionné dans la requête), nous souhaitons que le comportement suivant du système soit reproductible : lorsque l'utilisateur mentionne un problème affectant une prestation dans un contexte de moteur chaud, nous souhaitons récupérer en priorité des documents relatifs au même symptôme, puis les documents traitant d'un problème et une prestation identiques sans mention d'information quant à la température du moteur. En dernier lieu, le système peut retourner à l'utilisateur les documents relatifs aux mêmes problème et prestation, mais avec un moteur froid. La méthode proposée par [Bisson, 1993] ne respecte pas cet ordonnancement puisqu'elle fait systématiquement apparaître les documents possédant une relation facultative avant ceux qui ne la possède pas, et ce quelle que soit la similarité entre les instances issues de la relation facultative.

De façon plus générale, nous constatons que si nous voulons suivre une méthode proche de celle de Bisson avec garantie sur l'ordre de similarité relationnelle décrit plus haut, il nous faut augmenter artificiellement Sim_{rel} entre, d'une part, une instance de requête i_1 reliée à une instance i_2 par une relation sémantique rel_{fac} et, d'autre part, une instance de document i_3 non liée à rel_{fac}. Pour ce faire, nous considérons que cette situation est équivalente au calcul de $Sim_{rel}(i_1, i_3)$ avec i_3 reliée à une instance artificielle i_{artif} via rel_{fac}. Cette instance artificielle n'a pas d'existence réelle, mais constitue un "point neutre" : toute instance plus proche (en termes de similarité conceptuelle et en attribut) de i_2 que i_{artif} améliore le score de $Sim_{rel}(i_1, i_3)$ tandis que toute instance plus éloignée le réduira. Pour calculer $Sim_{rel}(i_1, i_3)$ dans le cas problématique, nous nous ramenons donc au cas avec i_{artif}, que nous résolvons en faisant la moyenne des similarités conceptuelle et en attributs entre i_3 et toute instance de concept pointée par rel_{fac} dans la base d'instances [23]. La figure 4.8 résume notre proposition. On notera que chaque distance représentée correspond au complément à 1 de la similarité résultant de la combinaison linéaire de Sim_{cpt} et Sim_{attr}. De plus, les flèches symbolisent

[23] En toute rigueur, il faudrait également faire intervenir les similarités relationnelles entre i_3 et chacune des autres instances mentionnées. Pour éviter une trop grande complexité des calculs de similarité relationnelle, nous ne calculons pas cette partie récursive dans le cas de relations facultatives.

toutes une relation de type rel_{fac}.

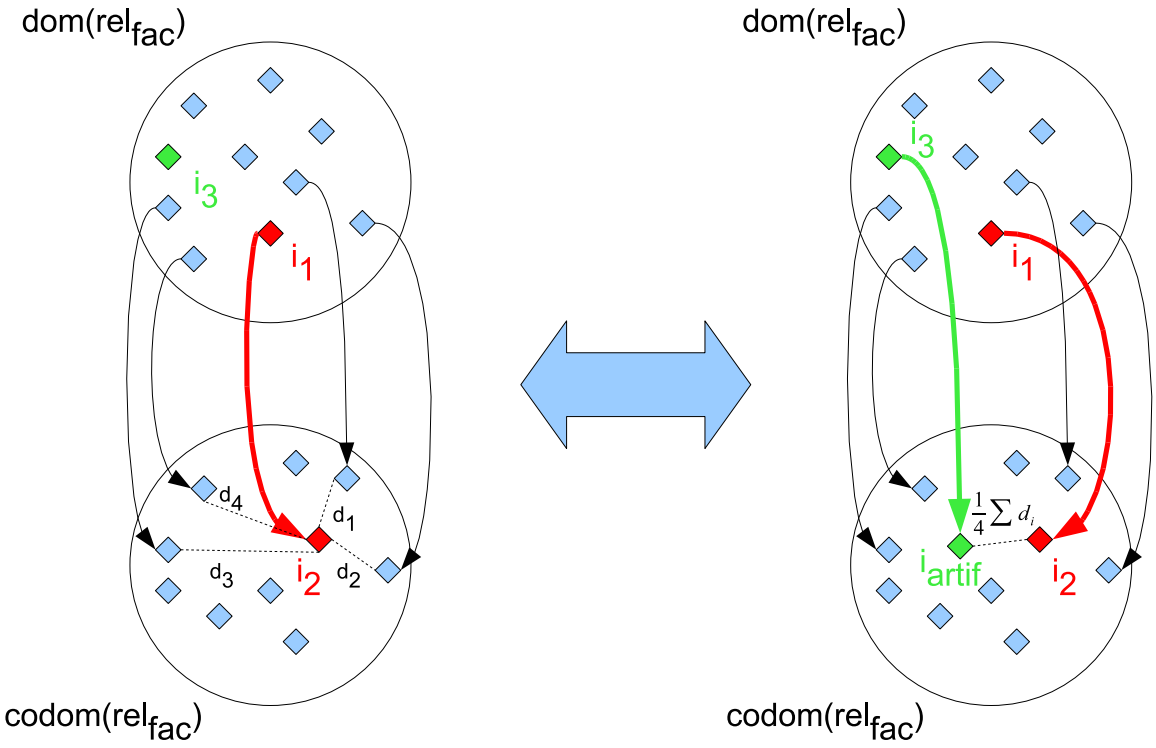

Figure 4.8 — Illustration d'introduction d'instance artificielle dans le calcul de $Sim_{rel}(i_1, i_3)$

Par rapport à l'approche de [Zhong *et al.*, 2002] que nous avons critiquée précédemment, notre démarche, pour une requête contenant une relation facultative rel_{fac} pointant vers une instance i_0 d'un concept C_0, permet de ne pas systématiquement privilégier les documents ne contenant pas rel_{fac} à ceux contenant rel_{fac} pointant vers une instance d'un concept différent de C_0. Pour démontrer notre propos, plaçons-nous dans la situation où C_0 possède trois frères taxonomiques inclus dans le codomaine de définition de rel_{fac}, que chacun comporte autant d'instances pointées par rel_{fac} et que la similarité de deux instances ne dépend que de la nature de leur type, avec :

$$Sim_{loc}(i_0, i_1) = Sim_{cpt}(C_0, C_1) = 0.7$$

$$Sim_{loc}(i_0, i_2) = Sim_{cpt}(C_0, C_2) = 0.25$$

$$Sim_{loc}(i_0, i_3) = Sim_{cpt}(C_0, C_3) = 0.05$$

Dans cette configuration (voir fig. 4.9), le point neutre correspond à une similarité de :

$$Sim_{loc}(i_0, i_{artif}) = \frac{1}{4} * (1 + 0.7 + 0.25 + 0.05) = 0.5$$

Il s'ensuit de ce calcul que par rapport à la requête envisagée plus haut, nous privilégierons les documents comportant rel_{fac} pointant vers C_1 aux documents sans mention de rel_{fac}, eux mêmes préférés aux documents incluant une instance de rel_{fac} dirigée vers C_2 ou C_3.

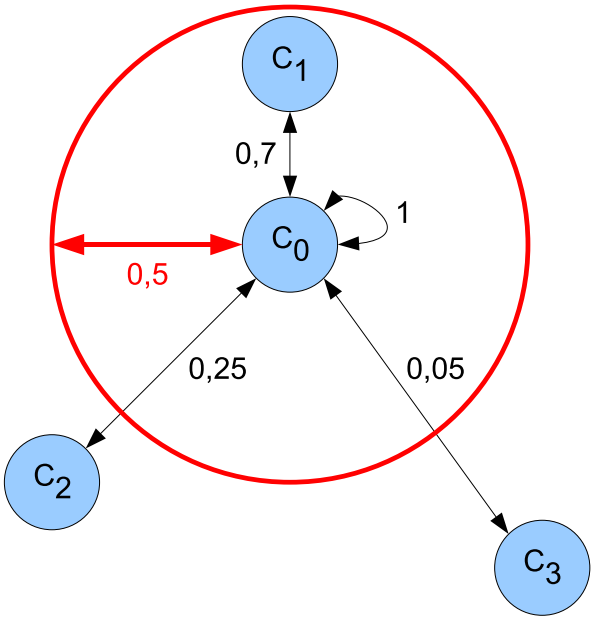

Figure 4.9 — Exemple de calcul du point neutre pour les concepts codomaines d'une relation facultative

4.2.2.4 Définition d'une mesure d'appariement spécifique

Etant données les heuristiques d'appariement de graphe présentées ainsi que la mesure de similarité locale entre deux instances, nous pouvons alors définir une mesure de proximité globale $Prox_{tot}(req, doc)$ entre le graphe de la requête[24] et celui représentant le document comparé. Comme la structure du graphe de requête est déjà prise en compte à la fois dans l'appariement (on part des instances centrales pour aligner le reste des instances à celles du document via les relations origine et destination de celles-ci) et dans le calcul de similarité locale, il est inutile de la prendre en compte autrement dans le calcul final de la proximité totale des graphes. Nous décidons donc de mesurer $Prox_{tot}(req, doc)$ en calculant l'ensemble des similarités locales en rapport avec les nœuds centraux :

- si le graphe de requête contient une ou plusieurs instances centrales, elles sont toutes prises en compte dans le calcul global au niveau des différentes similarités locales possibles selon les appariements,
- dans le cas où le graphe ne contient aucune instance centrale, on utilise tout nœud origine ou destination d'une relation obligatoire.

Une fois les différentes configurations d'appariement explorées, $Prox_{tot}(req, doc)$ correspond alors à la plus grande des moyennes calculées.

Cependant, nous introduisons une contrainte supplémentaire, à savoir un seuil numérique lié aux instances centrales de la requête. Si la similarité locale entre chaque instance centrale de la requête et l'instance de concept la plus proche présente dans le document ne dépasse pas ce seuil, nous considérons que l'instance centrale en question n'a pas été

[24] Si la requête contient plusieurs graphes indépendants, le calcul est le même sur chacun d'entre eux, et $Prox_{tot}(req, doc)$ correspond à la moyenne des proximités maximales pour chaque graphe, pondérées par la taille de chaque graphe.

retrouvée dans le document, et qu'il est donc inutile de retourner le document comparé à l'utilisateur.

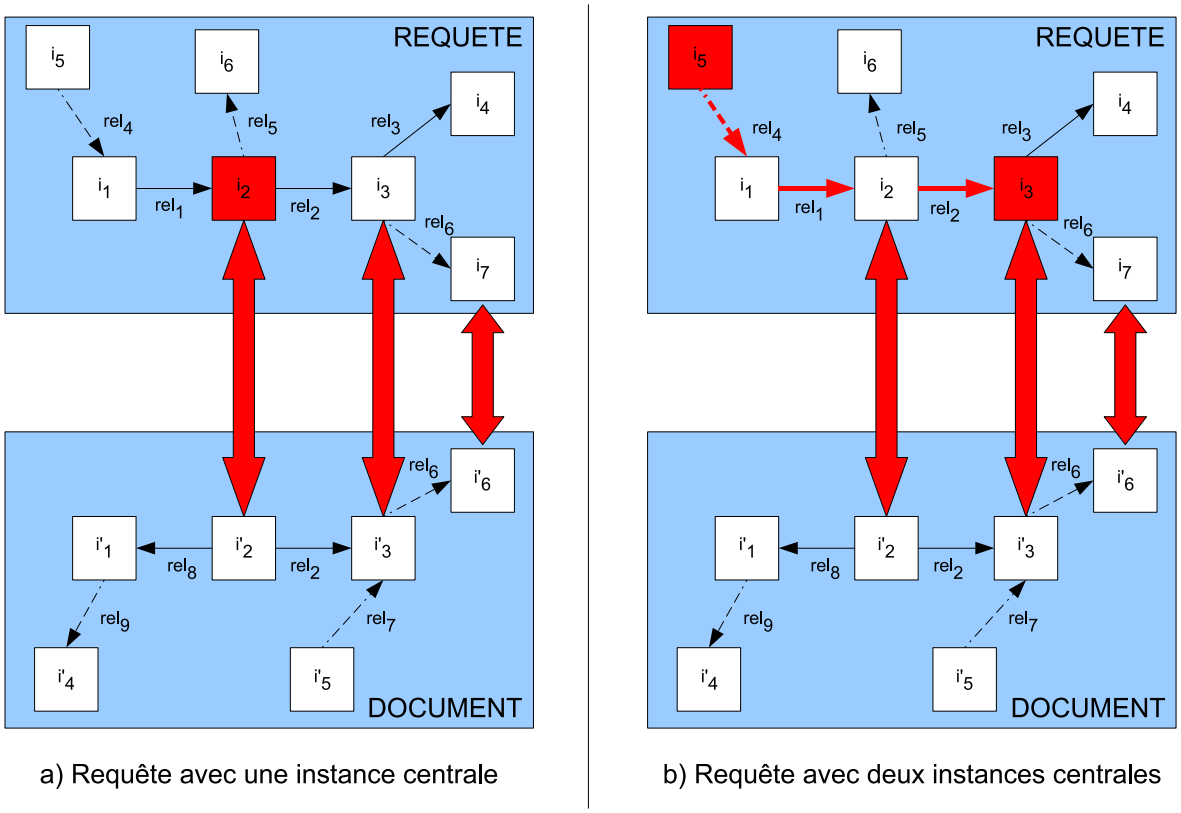

Figure 4.10 — Illustration du calcul de proximité totale

Deux exemples sont donnés sur la figure 4.10. Nous signalons dès le départ que toutes les instances appariées sont nécessairement comparables. Dans le premier exemple, avec la configuration d'appariement décrite et i_2 en nœud central, nous obtenons, pour trois nœuds à apparier (i_2, i_3 et i_4), deux similarités locales $Sim_{loc}(i_2, i'_2)$ et $Sim_{loc}(i_3, i'_3)$. En effet, l'appariement (i_7, i'_6) n'est pas issu d'une relation obligatoire (rel_6) mais il est indirectement pris en compte dans les calculs de $Sim_{loc}(i_2, i'_2)$ et $Sim_{loc}(i_3, i'_3)$ (voir plus haut). Comme i_4 n'est pas apparié, ce phénomène baissera la valeur finale de proximité totale à travers la valeur de $Sim_{loc}(i_3, i'_3)$. On a donc la formule finale suivante :

$$Prox_{tot}(req, doc) = Sim_{loc}(i_2, i'_2)$$

Notons que si aucune instance centrale n'était indiquée, nous obtiendrions une proximité totale moyennée sur quatre instances (de i_1 à i_4) :

$$Prox_{tot}(req, doc) = \frac{1}{4} * [Sim_{loc}(i_2, i'_2) + Sim_{loc}(i_3, i'_3)]$$

Pour le second exemple, la proximité globale se calcule à partir des meilleurs appariements de similarité locale de i_3 et i_5. L'absence, dans cette configuration, d'un appariement pour le nœud i_5 permet de déduire, en amont des calculs de similarité locale, que le document ne semble pas pertinent pour la requête. Si aucune configuration d'appariement ne

permet de trouver de correspondance simultanée pour i_3 et i_5 dans le document (ou que les similarités locales sont inférieures au seuil fixé), le document comparé sera définitivement abandonné vis-à-vis de la requête donnée.

4.3 Bilan

Dans ce chapitre, nous nous sommes appliqué à mettre en place un processus complet de RI sémantique capable de gérer une étape de maintenance de la RTO sur laquelle se fondent les étapes d'indexation et de formulation de requêtes. Dans nos travaux, l'ensemble des phases du cycle d'utilisation de la RTO pour la RI ne sont pas toutes abordées avec le même degré de granularité. En effet, nous avons cherché en priorité à en fournir un enchaînement cohérent, quitte à ce que le mode de fonctionnement envisagé reste relativement simple pour certaines étapes.

Nous avons toutefois voulu approfondir plusieurs points qui ne nous semblaient pas traités de façon satisfaisante dans la littérature. Dans un premier temps, nous nous sommes intéressé aux moyens qui permettraient d'indiquer à l'ontographe quand (et - dans une certaine limite - comment) réaliser une opération de maintenance sur une RTO utilisée dans un processus de RI. Nous avons à cette occasion émis et développé l'idée d'utiliser des critères prédéfinis par l'utilisateur pour mesurer automatiquement la qualité des annotations produites et ainsi repérer les limites de la RTO dans sa version courante. Les critères présentés mettent notamment en jeu les notions de termes et de concepts retrouvés et exploitent les liens de dénotation existant entre eux, ce qui nous a permis de réutiliser la principale contribution du chapitre 3, à savoir un méta-modèle de RTO en OWL. Concernant l'impact que peuvent avoir les modifications de la RTO sur les annotations produites, nous avons déjà eu en 2.2.3 un aperçu de l'intérêt et de la profondeur d'un tel sujet. Par choix, nous avons préféré résoudre le problème en créant plusieurs heuristiques selon l'opération de modification et la nature de l'entité ontologique modifiée.

Nous avons ensuite abordé la problématique d'appariement sémantique entre une requête et un document. A la différence de la plupart des approches de la littérature, nous envisageons une structure relativement riche pour les annotations sémantiques : chaque document est indexé par un (ou plusieurs) groupe(s) d'instances de concepts reliées entre elles par des relations sémantiques transverses. Dans ce paradigme, il devient vite difficile d'estimer numériquement la similarité sémantique entre requête et document : quelles instances apparier ? Quelles informations à leur sujet prendre en compte dans les calculs ? Par rapport aux quelques travaux existants, nous introduisons deux notions supplémentaires, la comparabilité de deux concepts et le caractère nécessaire / facultatif d'une relation. Ceci nous permet de réduire fortement le nombre et la complexité des calculs de similarité sémantique entre deux instances de concepts. Au niveau de cette mesure locale, nous nous proposons, par un calcul récursif, de prendre en compte un maximum d'informations sur les instances : le type des concepts, les valeurs d'attributs en commun, mais aussi la nature des instances auxquelles chacune d'elle est reliée. A notre connaissance, aucun travail de recherche n'avait pour l'instant proposé d'utiliser autant d'informations dans un calcul de similarité entre instances.

Dans le chapitre suivant, nous implémenterons l'ensemble de nos contributions théoriques au sein d'un seul et unique prototype : TextViz. A travers son utilisation dans le cadre d'un protocole d'évaluation à définir, nous pourrons ainsi apprécier la qualité et l'intérêt pratique de nos apports.

Troisième partie

Réalisations et Evaluation

5 Le projet OBIR

Dans ce chapitre, nous allons présenter une mise en pratique et une évaluation des contributions théoriques des chapitres 3 et 4. Nous mettons particulièrement l'accent sur les points suivants :

- le cycle en parallèle de maintenance de RTO / indexation sémantique,
- la mesure d'appariement sémantique entre une requête exprimée en langue naturelle et un document textuel,
- l'utilisation conjointe du méta-modèle en OWL-DL pour la formalisation explicite des différents éléments d'une RTO,

Du fait de l'ampleur de chacune des tâches et de contraintes temporelles, nous avons dû adapter nos contributions spécifiquement au domaine visé, à savoir le diagnostic automobile. Pour une mise à l'épreuve optimale de nos approches, il aurait alors fallu tester nos apports théoriques sur différents domaines avec pour seul point commun un objectif applicatif de RI. Nous nous sommes pour l'instant cantonné au domaine du diagnostic automobile, jugeant que le critère capital consistait en la satisfaction des utilisateurs en termes d'efficacité : facilité d'utilisation, meilleurs résultats sans effort supplémentaire par rapport à une approche de RI classique ... Toutefois, nous envisageons également d'évaluer nos contributions sur des domaines différents dans le cadre de travaux ultérieurs liés au projet Dynamo [1].

Dans ce chapitre, nous décrivons en premier lieu l'implémentation de nos différentes contributions théoriques dans le cadre du projet OBIR (section 5.1). Nous nous attachons ensuite à mettre en place et à appliquer un protocole d'évaluation relatif à nos apports scientifiques (section 5.2).

5.1 Implémentation

Nous décrivons dans cette section les deux principales composantes de notre système, à savoir Textviz, l'outil qui assure à la fois la maintenance supervisée de RTO par évaluation des résultats d'indexation sémantique et le calcul de cette indexation, ainsi que l'outil de recherche sémantique. Nous commençons par exposer comment nous avons construit une RTO adaptée au contexte industriel à partir des documents de la base de recherche.

[1] `http://www.irit.fr/dynamo`

5.1.1 Construction d'une RTO du diagnostic automobile

Après avoir décrit la base d'expériences ultérieurement sollicitée pendant la phase de recherche sémantique, nous détaillons le processus que nous avons suivi pour construire une RTO du domaine en se fondant sur ces documents.

5.1.1.1 La base d'expériences

Origines et évolution L'ensemble des documents dans lesquels sont recherchées des informations est constitué de fiches d'incidents qui rapportent des pannes survenant sur des modèles particuliers de véhicules (applicabilité du cas) et leur associent des informations liées au diagnostic et à la réparation de la panne décrite. On pourra nommer cet ensemble *base d'expériences* ou encore *base de recherche*. Parmi les documents à disposition, ceux-ci peuvent provenir de deux sources différentes :

- un premier groupe émane d'un constructeur automobile, qui les distribue aux concessionnaires de son réseau international,
- le second groupe fait partie d'une base d'expériences vendue par une société de service spécialiste du diagnostic automobile à des réseaux indépendants de garagistes ; celle-ci ne se limite donc pas à une seule marque de véhicules, contrairement aux documents obtenus auprès du constructeur.

Ces fiches sont dans les deux cas des synthèses d'experts en diagnostic automobile à partir d'analyses de cas réels rencontrés en garage. Pour le premier groupe, nous disposons également de la traduction de chaque document en différentes langues (anglais, allemand, italien, espagnol, russe ...). Comme le montre la figure 5.1, l'intégralité des fiches du constructeur est traduite dans toutes les langues, ce qui rend le traitement simultané des différentes langues superflu. Cette base multilingue ouvre toutefois certaines perspectives supplémentaires quant à l'utilisation de notre méta-modèle de RTO : comme nous avons explicitement séparé les informations conceptuelles des informations terminologiques, il semble concevable, sur un domaine à modéliser bien délimité, de se fonder sur une même représentation ontologique sur laquelle viennent se greffer différents lexiques selon la langue des documents à indexer. Pour une contribution relative à cette problématique, nous conseillons au lecteur de se référer à [Roussey *et al.*, 2006].

En termes de quantité, nous disposons actuellement d'environ 800 fiches du constructeur (700 réellement distinctes [2]) et 4700 fiches indépendantes (1400 cas distincts [3]).

Cette base est amenée à évoluer régulièrement afin de prendre en compte et de diffuser les nouveaux problèmes que rencontre chaque garagiste. Actuellement, le rythme moyen de mise à jour de la base d'expériences chez Actia est de l'ordre d'une cinquantaine de fiches supplémentaires (i.e. un accroissement de la base de l'ordre de 1%) tous les 4 mois. Intuitivement, la faible proportion de documents rajoutés, ainsi que la relative fréquence de ces ajouts, ne paraît pas poser de problèmes pour l'approche de maintenance de RTO envisagée puisqu'en aussi peu de temps et avec aussi peu de rajouts, la RTO aura tendance

[2] Certaines fiches sont dupliquées lorsque le problème survient sur plusieurs modèles de véhicules.

[3] Le nombre de duplications est dans ce cas très important car la base fait souvent référence à des symptômes génériques affectant plusieurs modèles.

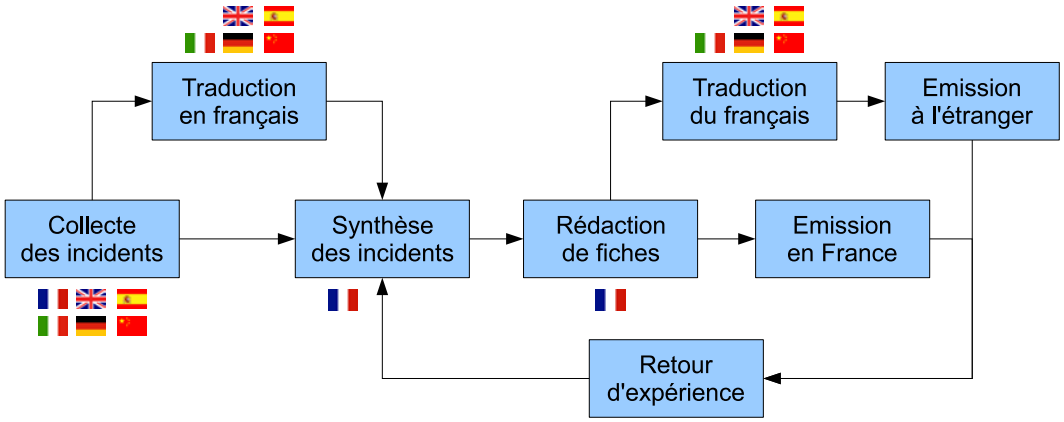

Figure 5.1 — Origine des documents de la base d'expériences du constructeur automobile

à rester pertinente dans sa globalité pour la tâche et le domaine modélisés (i.e. pas de remise en cause de la structure fondamentale) et à ne nécessiter que quelques modifications afin d'indexer correctement les nouveaux documents. De plus, le faible nombre de documents disponibles à chaque mise à jour rend difficile toute approche fondée sur des techniques statistiques (dont notre approche s'affranchit).

APPLICABILITE
Citronault Pipo

SYMPTOME
En roulant le moteur manque de puissance en accélération et en vitesse de pointe.

DIAGNOSTIC
Pas de défaut mémorisé dans le calculateur d'injection. Après avoir contrôlé que le débit d'air dans le débitmètre est trop faible, vérifier la connexion du débitmètre.

SOLUTION
Démontage et remontage du connecteur du débitmètre. Procédure habituelle.

Figure 5.2 — Exemple de fiche de réparation

Caractéristiques notables Les documents de la base de recherche sont tous formatés selon une structure fort similaire avec 4 champs principaux :

- l'applicabilité (i.e. le type de véhicule concerné),
- le symptôme client, qui décrit le(s) problème(s) rencontré(s) par l'utilisateur et/ou le garagiste,
- le diagnostic du problème, qui correspond aux causes du comportement anormal

constaté,
– une description de la solution, pouvant inclure des informations sur sa mise en œuvre

On peut voir sur la figure 5.2 une illustration de la structure ainsi que du contenu typiques d'un document de la base de recherche. Outre une structure extrêmement contrainte et clairement orientée selon une tâche de diagnostic, l'ensemble des documents de la base de recherche partage plusieurs points communs d'ordre linguistique. En effet, nous avons pu constater que les fiches de réparation étaient rédigées de façon concise, avec une proportion importante de phrases nominales : sur un échantillon de 110 documents sélectionnés aléatoirement (environ 5% de la base de recherche), nous avons observé que le nombre de mots dans chaque texte était relativement faible, puisque compris entre 15 et 165 [4] (54 mots en moyenne, écart-type de 25 mots) ; de même, pour un nombre moyen de 4,4 phrases par document, nous avons relevé une proportion de phrases nominales de l'ordre de 33%. Pour résumer, nous avons été amenés à travailler sur un ensemble de **faible taille** de documents **très succincts** et **structurés de façon homogène**.

5.1.1.2 Tour d'horizon des ontologies liées à l'automobile

Afin d'obtenir une RTO adaptée à nos besoins de RI, nous avons commencé par analyser les ontologies automobiles déjà disponibles dans la littérature, dont nous avons pu malheureusement constater la rareté. Nous supposons que ce phénomène est lié au fait que les modèles de connaissances automobiles sont généralement développés pour tout ou partie grâce à des investissements de la part des constructeurs automobiles. Dans ce contexte, toute ontologie du domaine revêt sans nul doute un intérêt stratégique qui la rend trop sensible pour être divulguée.

A travers MARIA (Model for Automotive Repair Information Applications), les travaux de [Bryan et Wright, 2005] s'intéressent à la modélisation de connaissances liées à la tâche de diagnostic automobile. Ces recherches, conduites dans le cadre du projet MYCAREVENT [5], cherchent à mettre en place un ensemble minimal de concepts nécessaires pour représenter une session de diagnostic sur lesquels peuvent s'appuyer différents projets pour construire leur propre ontologie adaptée à leurs besoins applicatifs. Comme on peut le voir sur la figure 5.3, MARIA se limite dans sa portée à modéliser des informations pour la réparation d'un véhicule, et ne se préoccupe pas des modèles physiques sous-jacents : le modèle représente les caractéristiques statiques des objets du domaine, pas leur comportement.

Si l'on revient aux besoins spécifiques à notre projet, les concepts modélisés par MARIA qui pourraient nous être utiles sont au nombre de 3 : comme chaque fiche de la base de recherche comporte à la fois le(s) problème(s) constaté(s) (i.e. **symptômes**) sur un type (i.e. **applicabilité**) de véhicule donné et la manière de les résoudre (i.e. **réparation**), il nous est inutile de modéliser dans la RTO les tests (et donc par extension leurs résultats) permettant de diagnostiquer une panne donnée. De même, nous n'aurons pas besoin de représenter l'applicabilité d'un véhicule car la sélection des fiches applicables à une certaine si-

[4] Comme son contenu devrait être à terme exprimé dans un formalisme spécifique, le champ d'applicabilité n'a pas été inclus dans le décompte.

[5] `http://www.mycarevent.com`

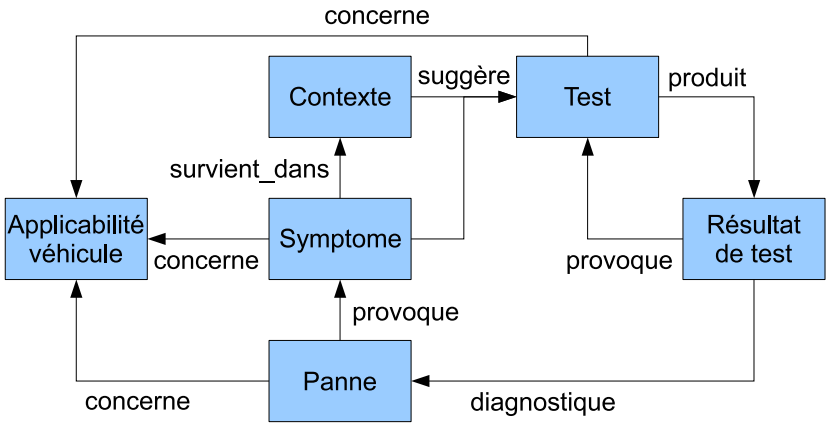

Figure 5.3 — Représentation simplifiée du modèle conceptuel MARIA

tuation [6] sera faite par un module spécifique préalable au moteur de recherche sémantique. Dans l'implémentation possible des symptômes proposée par [Bryan et Wright, 2005], les auteurs séparent les symptômes selon qu'ils proviennent d'une observation physique ou de codes défauts (Diagnostic Troubleshooting Code) signalés par les calculateurs électroniques. Comme ces codes défauts sont présents sous forme d'identifiants dans les fiches de la base d'expériences étudiée, nous préférons ne pas les modéliser dans notre RTO et les traiter plus directement avec un module spécifique. Plus bas encore dans la hiérarchie des symptômes proposée, nous remarquons que ceux-ci sont différenciés selon un critère relatif au sous-système dans lequel ils apparaissent : problèmes de démarrage, de freinage, de direction ... Un découpage de la sorte ne nous semble pas convenable car il mêle des informations liées au problème constaté (e.g. absence de fonctionnement totale, partielle, vibrations, bruits) et d'autres relatives à la fonctionnalité touchée sur le véhicule (e.g. motorisation, climatisation, freinage). Ceci aboutit à des duplications de concepts superflues dans la hiérarchie des symptômes : absence de fonctionnement de la motorisation, absence de fonctionnement de la climatisation ... Nous avons donc ici un premier indice quant au besoin de considérer dans notre RTO le symptôme comme un concept défini par (au moins) deux concepts primitifs qui seraient de type `Problème` et `Fonctionnalité`.

Une seconde approche de modélisation est abordée avec Samovar (Système d'Analyse et de MOdélisation des Validations des Automobiles Renault) dans [Golebiowska, 2002]. L'ontologie décrite sert de base à une aide au processus de validation d'un véhicule. En effet, le cycle de développement d'un produit automobile peut être décomposé en plusieurs sous-cycles itératifs enchaînant conception, implémentation et validation, ponctués par la production de versions successives de maquettes et/ou prototypes. L'étape de validation permet de s'assurer de la conformité du produit vis-à-vis du cahier des charges. Golebiowska envisage d'apporter une aide aux concepteurs et techniciens de Renault en utilisant l'ontologie comme un moyen d'accès simplifié aux nombreux documents relatifs aux problèmes antérieurs de validation. Comme on peut le constater, son objectif applicatif s'avère très

[6] Il n'est pas pertinent de retourner une fiche-incident aux symptômes totalement similaires à ceux exprimés par le garagiste si le véhicule en panne ne correspond pas à l'applicabilité mentionnée dans la fiche : rien n'assure que la réparation associée fonctionne dans les deux cas.

proche du nôtre. L'auteur met d'ailleurs en place un mode d'accès aux documents de sa base de recherche par l'utilisation conjointe de son modèle des validations automobiles et du moteur de recherche sémantique Corese (que nous avons décrit en 2.2.2.3). Après avoir construit un modèle de la tâche de validation, Golebiowska isole quatre composantes principales de l'ontologie :

- une **pièce** correspond à un composant physique du système étudié, et peut connaître un certain nombre de problèmes dont elle peut être à l'origine (elle peut aussi être la cause d'un problème sur une autre pièce) ;
- un **problème**, dans un contexte de validation, décrit un écart entre les comportements attendu et constaté pour un composant donné ;
- une **prestation** représente un service rendu au client ;
- un **projet** est défini par le protocole de test suivi, le groupe de personnes intervenant dans le cycle de développement du produit, la sous-partie de la chaîne de montage concernée par la réalisation de la maquette et/ou du prototype et la configuration (i.e. les caractéristiques propres) du prototype.

Nous constatons que nous nous trouvons dans une situation de modélisation comparable mais pas identique à celle de Samovar. En effet, s'il semble acquis que les deux approches doivent modéliser des problèmes, ceux-ci ne sont pas situés au même niveau de granularité : dans le cas des validations, Golebiowska cherche à représenter tout comportement non nominal des pièces constitutives du produit testé ; dans nos travaux, nous souhaitons pouvoir modéliser les symptômes de pannes observables pour une prestation de véhicule (si aucun écart au comportement nominal des prestations n'est constaté, le client n'a que peu de raisons d'amener son véhicule au garage). Les problèmes tels que nous devons les modéliser ne peuvent donc être ni identiques ni découpés selon les mêmes critères que ceux de Samovar. De plus, comme les prestations jouent pour nous le même rôle central que les pièces chez Golebiowska, notre sous-ontologie des prestations a de fortes chances d'être bien plus importante que celle des pièces. A cet égard, la taxonomie des prestations utilisée pour Samovar ne nous satisfait pas : elle est construite sur un seul niveau et suit le découpage organisationnel des équipes de production chez Renault. Dans notre cas, modéliser les prestations selon un tel critère de différenciation aboutirait à la création de prestations au comportement difficilement évaluable par de simples observations : un automobiliste aurait du mal à formuler un jugement de satisfaction sur une prestation d'aérodynamisme ou de dépollution du moteur (car il ne perçoit que vaguement à quoi correspond un comportement nominal pour ces fonctionnalités). Enfin, comme nous n'avons qu'un besoin secondaire des pièces (ce qui nous intéresse lorsqu'une pièce est mentionnée, c'est la prestation qu'elle contribue à réaliser), il semble raisonnable que la sous-ontologie correspondante ne soit pas aussi détaillée que celle pour Samovar (qui comporte plus de 800 concepts uniquement pour le sous-système du poste de conduite, dont celui de *vis* ou d'*écrou*). Nous retiendrons toutefois l'idée d'associer un composant avec ses sous-composants à l'aide de relations méronymiques (i.e. `partie_de`). En effet, dans le cadre d'un raisonnement de diagnostic, cette propriété permet, à partir d'un composant suspect, de connaître une liste plus précise des composants à tester (jusqu'à isoler le(s) composant fautif(s)).

5.1.1.3 Modélisation du domaine

Suite aux réflexions présentées en 3.1.3 et en 5.1.1.2, nous structurons les connaissances pour le diagnostic automobile en plusieurs sous-arbres, liés entre eux par des relations sémantiques. Pour chacun, les concepts racines sont les suivants (voir fig. 5.4) :

- un **symptôme** est défini (condition nécessaire et suffisante) par un **problème** et une **prestation** du véhicule. La présence explicite de ce concept n'a été décidée que relativement tard au cours du processus de structuration et elle est liée à la nécessité de représenter la constatation éventuelle d'un symptôme absent (e.g. "*Le moteur manque de performances sans allumage du témoin diagnostic*").
- un **problème** (e.g. *blocage, claquement, surconsommation*) affecte une ou plusieurs **prestations** (e.g. *climatisation, motorisation, freinage*) et peut survenir dans certains **contextes** (e.g. *en accélération, à chaud, en mode automatique*).
- une **prestation** correspond à une fonctionnalité rendue à l'usager par son véhicule.
- un **composant** est une pièce ou un ensemble de pièces mécaniques, électriques et/ou électroniques ; il participe à la réalisation d'une **prestation**.

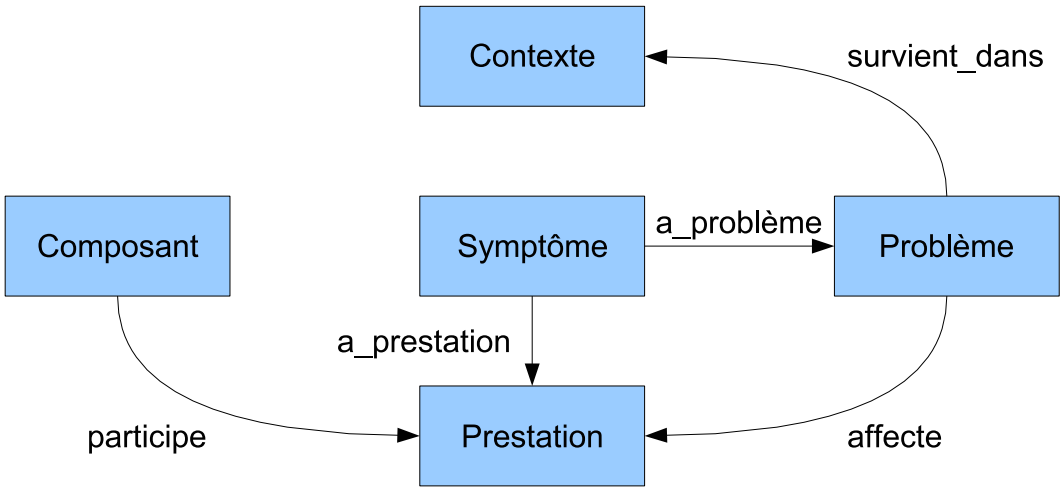

Figure 5.4 — Partie supérieure de l'ontologie résultante

En fin d'étape de structuration, l'ontologie résultante comporte environ 340 concepts différents :

- 80 problèmes, de profondeur moyenne 2.7 dans la taxonomie et de profondeur maximale 5,
- 110 prestations, de profondeur moyenne 3.4 et de profondeur maximale 6,
- 75 composants, de profondeur moyenne 3.3 et de profondeur maximale 5,
- 75 contextes, de profondeur moyenne 2.6 et de profondeur maximale 3.

Nous précisons qu'à ce jour, les quatre sous-arbres taxonomiques ne sont placés sous aucun concept générique issu d'ontologies comme DOLCE [Gangemi *et al.*, 2002] ou SUMO [Niles et Pease, 2001]. En effet, le rapprochement a posteriori d'une ontologie aussi spécifique que la nôtre avec des modèles de connaissances aussi générales ne présentait pour nous aucun avantage évident. Toutefois, nous n'excluons pas la possibilité de revenir ultérieurement sur cette position et d'envisager, si un besoin applicatif le suggérait, de posi-

tionner les concepts de notre ontologie du diagnostic automobile par rapport à ces modèles de connaissances génériques.

Pour l'ensemble des concepts des sous-arbres décrits par la suite, nous ne leur avons associé aucune propriété. En effet, une première tentative a mis en évidence qu'il était impossible de trouver des propriétés pertinentes (car partagées par un nombre suffisant de concepts) pour la comparaison de deux instances de concepts de la même sous-ontologie. Nous nous appuierons donc sur la forme de la taxonomie et sur les relations sémantiques dont chaque instance est origine pour déterminer la similarité sémantique de deux entités. Ici encore, notre position n'est pas définitive et selon les réorganisations futures que chaque sous-ontologie subira, nous pourrons être amenés à distinguer un certain nombre de propriétés communes à deux concepts.

Sous-ontologie des problèmes Comme on peut le voir sur la figure 5.5, un problème pour le diagnostic automobile peut correspondre à une classe parmi trois principales :

- une observation de symptôme liée à un sens humain particulier (e.g. un bruit, une odeur, une vibration, un signal visuel),
- une observation de panne sans mention du moyen de perception lié (e.g. absence de fonctionnement, arrêt impossible, sur-régime),
- un mode dégradé, qui correspond à un fonctionnement d'une prestation du véhicule selon un comportement très simplifié et dû à une panne grave.

Parmi les défaillances constatées, on distingue les absences totales de fonctionnement des fonctionnements non nominaux (i.e. différents du comportement de base du sous-système incriminé).

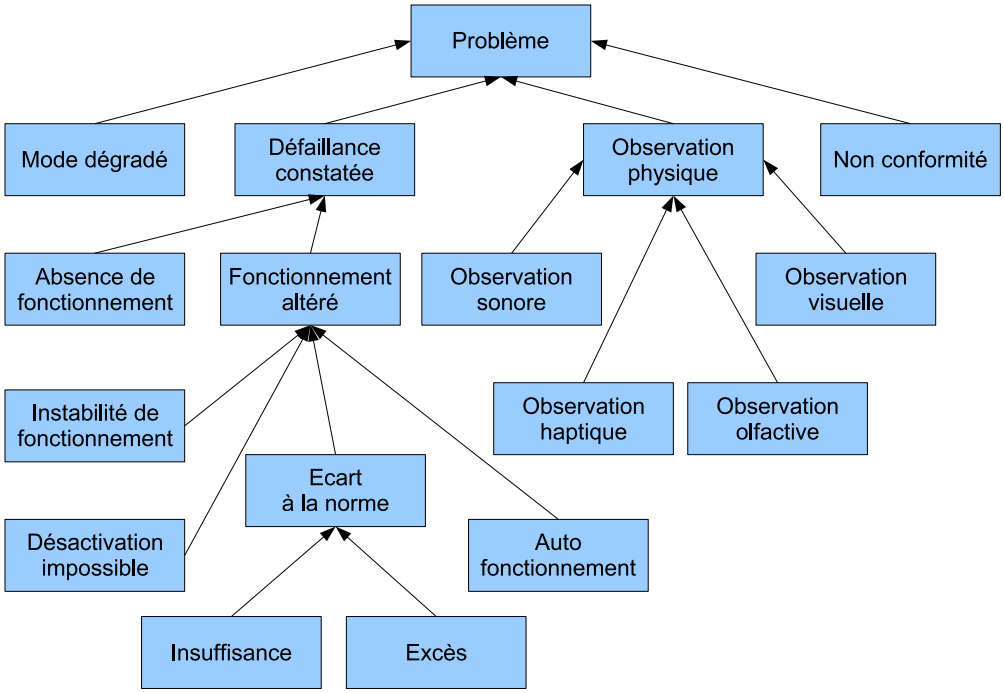

Figure 5.5 — Partie supérieure de l'ontologie des problèmes

Sous-ontologie des prestations Pour l'ensemble des prestations, nous avons construit le squelette schématisé sur la figure 5.6 (profondeur limitée à 3 sur le schéma pour une meilleure lisibilité). Le découpage obtenu correspond dans sa majorité à celui disponible dans plusieurs documentations techniques relatives à la prise en charge après-vente d'un véhicule. Le critère de différenciation initial porte notamment sur la localisation du service rendu à l'utilisateur et le degré de criticité d'une panne survenant sur celui-ci : les prestations liées au châssis sont les plus critiques car une panne survenant sur l'une d'entre elles pourrait rendre le pilotage du véhicule très dangereux ; toute prestation de motorisation est également importante car si elle venait à s'arrêter de fonctionner, cela engendrerait une impossibilité de conduire ; enfin, les prestations habitacle correspondent à des services optionnels dont le fonctionnement est moins sujet à caution. On retrouve parfois ce découpage au niveau même des réseaux de données entre calculateurs électroniques d'une automobile, avec trois réseaux mutuellement indépendants avec des débits maximaux différents.

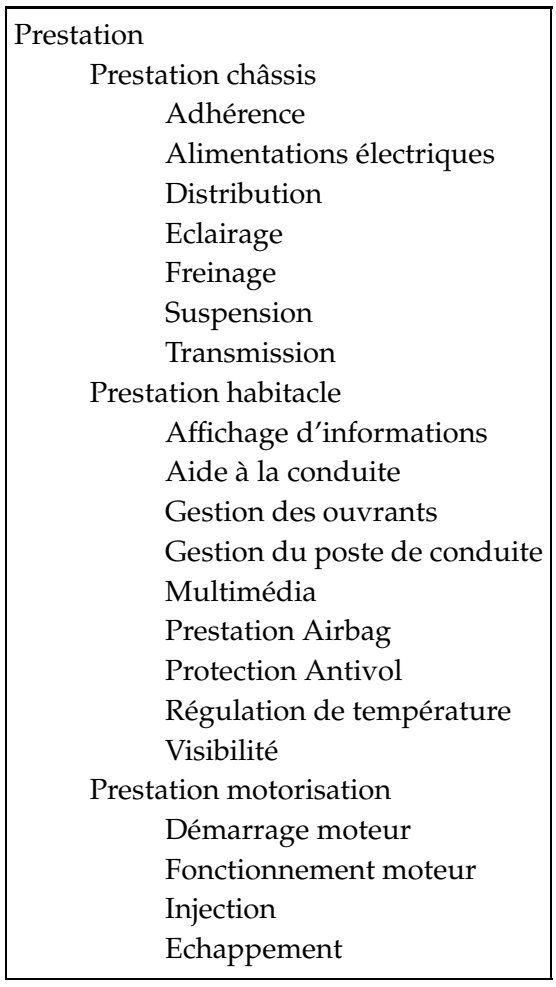

Figure 5.6 — Partie supérieure de l'ontologie des prestations

Sous-ontologie des composants Comme le montre la figure 5.7, nous avons séparé les composants selon qu'ils participent ou pas à la réalisation d'une fonction de contrôle d'une ou plusieurs prestations. En effet, les composants de ce type peuvent signaler des problèmes

sur les services surveillés mais ils peuvent aussi être eux-mêmes victimes d'une panne : par exemple, un témoin lumineux peut signaler à tort un problème sur une prestation, auquel cas c'est son comportement qui est symptômatique d'une panne, et non celui de la prestation surveillée. Ces composants de contrôle peuvent récupérer (*composant de mesure*), intégrer (*calculateur électronique*) ou afficher (*indicateur*) des données relatives au fonctionnement des services surveillés. Pour les composants de fonctionnement, nous avons appliqué un critère de différenciation lié à leur complexité, i.e. s'ils sont composés (*composant complexe*) ou pas (*composant remplaçable*) de composants plus petits susceptibles d'être remplacés par le garagiste en cas de dysfonctionnement. Nous ne nous sommes pas intéressés aux composants de granularité inférieure car dans le cadre d'un diagnostic automobile, il est inutile de repérer précisément une pièce fautive si son mauvais fonctionnement implique le changement systématique d'un composant plus gros dont elle fait partie.

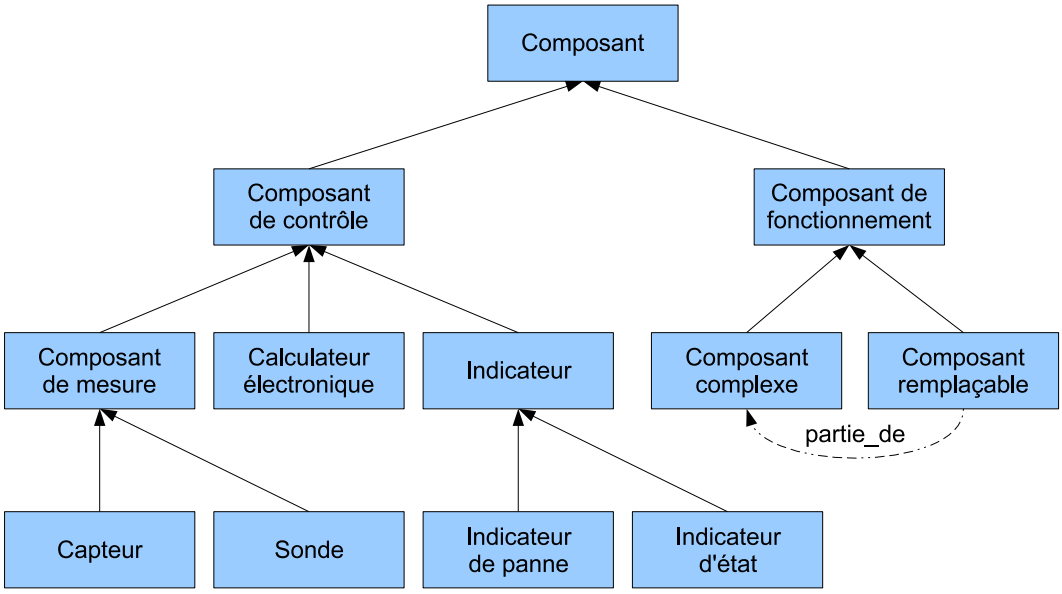

Figure 5.7 — Partie supérieure de l'ontologie des composants

Sous-ontologie des contextes On peut distinguer deux grands types de contexte pour un symptôme : un contexte quantificateur renseigne sur la fréquence d'apparition du symptôme (rare, intermittent, régulier, systématique), son importance (faible, partielle, forte, totale) et sa rapidité d'apparition (lent, modéré, rapide), tandis qu'un contexte matériel décrit un état global du véhicule (charge, température du moteur, phase de progression du véhicule, mode de fonctionnement d'une prestation . . .). Comme nous l'avons expliqué en 4.2.2, les contextes ne sont pas tous comparables entre eux : sur la figure 5.8, aucun des concepts mentionnés ne sont comparables entre eux, il est nécessaire de descendre plus bas dans la taxonomie pour trouver des concepts traitant d'informations rapprochables, comme deux indications de température, ou deux régimes de fonctionnement.

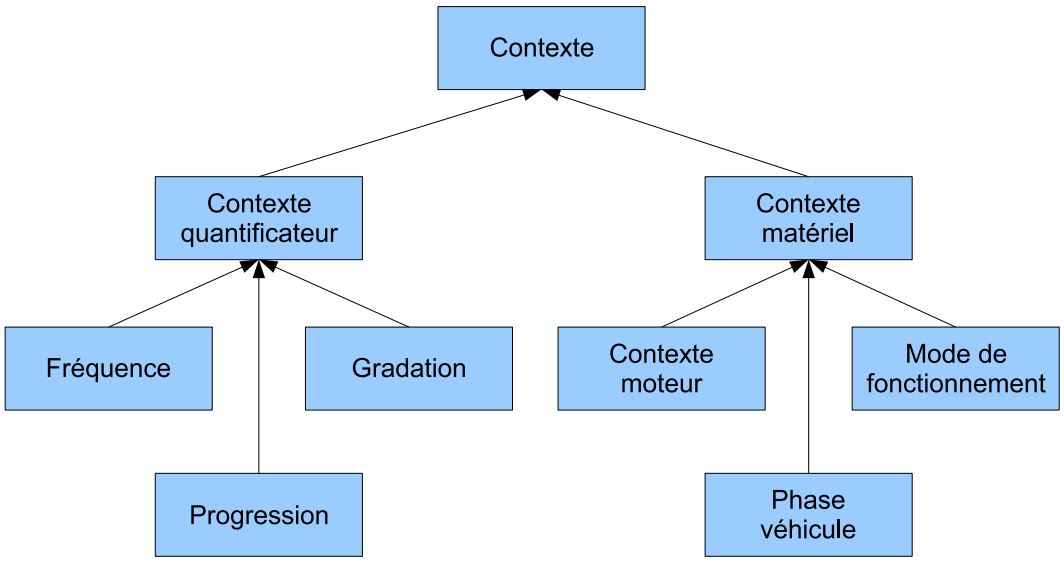

Figure 5.8 — Partie supérieure de l'ontologie des contextes

5.1.2 Cycle de maintenance de RTO du diagnostic automobile par l'indexation sémantique

5.1.2.1 Présentation de l'éditeur d'ontologies Protégé

Afin d'implémenter le système de maintenance ontologique que nous présentons en 4.1, nous avons cherché un environnement capable à la fois d'éditer une ontologie et de l'utiliser dans une tâche d'indexation semi-automatique. Nous avons arrêté notre choix sur Protégé-OWL pour sa bonne ergonomie, son architecture ouverte (facilement extensible) et sa large diffusion dans la communauté d'IC.

Protégé-OWL est une extension de l'API (Application Programming Interface, ou interface de programmation applicative) Protégé[7] pour manipuler le format OWL. Elle permet notamment d'importer des ontologies en OWL stockées dans un fichier ou une base de données, de visualiser, d'éditer classes et propriétés OWL, de communiquer avec des raisonneurs logiques ou de peupler l'ontologie d'instances trouvées dans des documents. Pour son interface utilisateur, le logiciel se présente sous la forme de plusieurs onglets permettant l'accès à différents types d'information : on peut citer entre autres les onglets *OWLClasses* pour la gestion de concepts (cf fig. 5.9), *Properties* pour la gestion des attributs et relations sémantiques (cf fig. 5.10) et *Individuals* pour la gestion des instances de concept (cf fig. 5.11).

Pour plus de détails sur cet outil, le lecteur peut se référer à [Knublauch *et al.*, 2004].

5.1.2.2 Lucene, une boîte à outils pour la RI

Lucene[8] est une API libre écrite en Java qui permet de construire des moteurs classiques et performants de recherche d'informations au sein de documents textuels. En premier lieu,

[7] `http://protege.stanford.edu/`
[8] `http://lucene.apache.org`

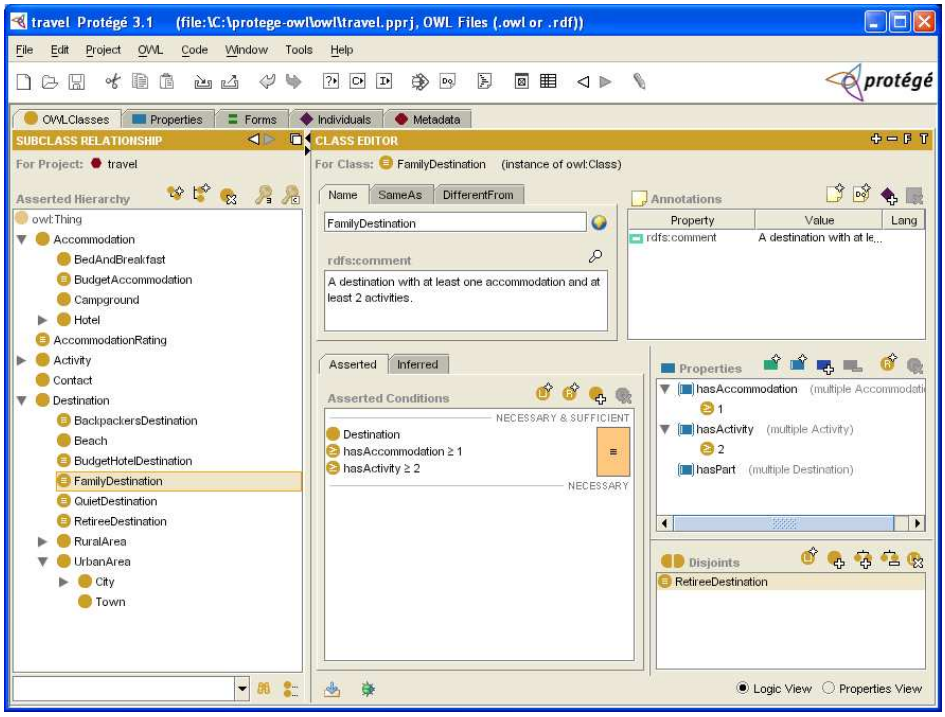

Figure 5.9 — Copie d'écran de l'onglet "OWLClasses" de Protégé-OWL

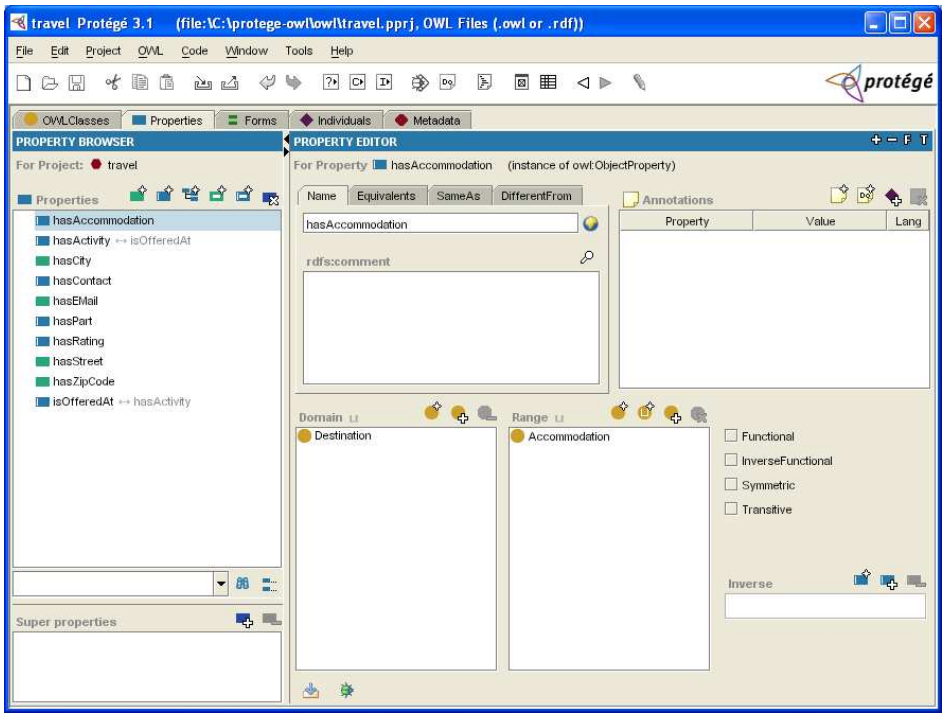

Figure 5.10 — Copie d'écran de l'onglet "Properties" de Protégé-OWL

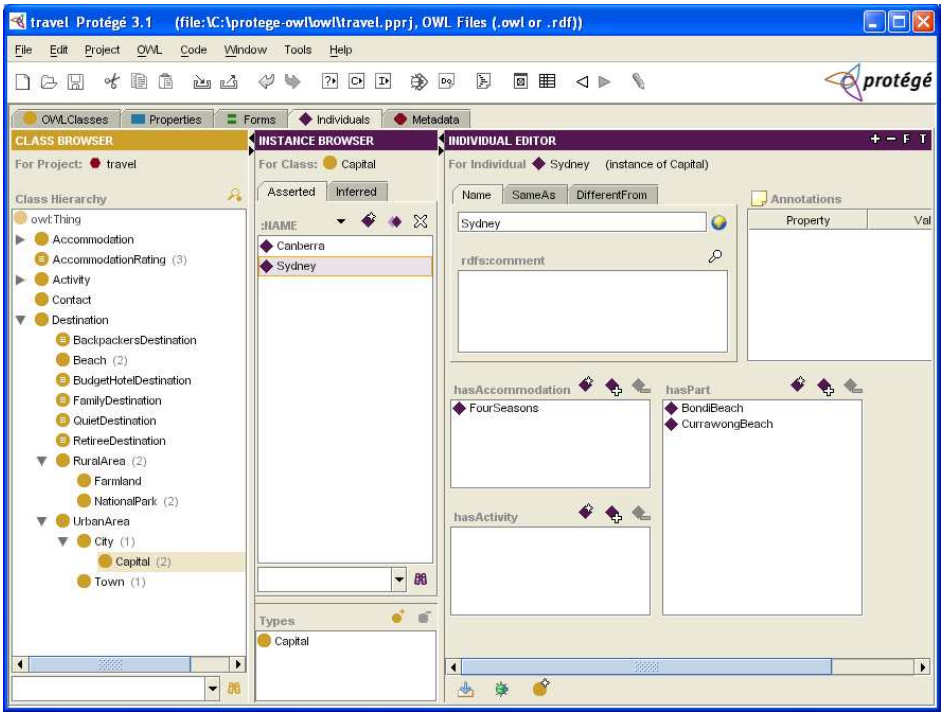

Figure 5.11 — Copie d'écran de l'onglet "Individuals" de Protégé-OWL

Lucene met à disposition des méthodes génériques d'analyse lexicale (à adapter en fonction des besoins applicatifs et de la langue employée) pour l'indexation d'un ensemble de documents textuels (potentiellement structurés selon différents champs) : découpage en lexèmes, filtrage par anti-dictionnaire, radicalisation ...

Une seconde partie de l'outil est centrée sur la tâche de recherche dans les textes et fournit notamment à l'utilisateur la possibilité d'exploiter différents types de requêtes (cf tab. 5.1). Comme Lucene stocke dans son index la position de chaque terme d'un document indexé, des fonctionnalités comme le surlignage de mots-clés peuvent être implémentées en aval. Enfin, si il n'est pas satisfait par le module générique de tri des résultats (fondé sur la mesure TF-IDF), un utilisateur de Lucene peut choisir de définir ses propres méthodes.

Pour une description précise des capacités de Lucene, se reporter à l'ouvrage [Hatcher et Gospodnetic, 2004a].

5.1.2.3 Description des fonctionnalités de TextViz

Nous détaillons dans cette partie les principales caractéristiques de TextViz, notre extension à Protégé-OWL pour la maintenance de RTO par l'indexation sémantique. Pour cela, nous nous fondons sur l'ensemble des étapes mentionnées dans le schéma global 4.3. Outre une simple description des fonctionnalités de l'outil, nous nous intéressons ici aux changements intervenus dans la méthode générique proposée en 4.1.2.2 pour l'adapter au domaine spécifique du diagnostic automobile. Nous montrons également pourquoi et comment certaines étapes du processus de maintenance ont été gérées de façon différente que prévue.

Type de requête	Exemples de requête	Exemples de segments textuels retrouvés
booléen	chat AND chien ≡ +chat +chien	"le chien pourchassa le chat"
	argent OR beurre	"mettre du beurre dans les épinards"
	drapeau NOT France ≡ +drapeau -France	"le drapeau du club flottait fièrement"
avec joker	auto*	"automobile", "automatique"
	?arc	"arc", "parc", "marc"
flou	élever~	"élève", "enlever", "laver"
par intervalle	publi_year :[2005 TO 2007[	tout document publié en 2005 ou 2006
par fenêtre de mots	"professeur école"~10	"le professeur se rend pour la première fois à son école d'affectation"

Tableau 5.1 — Différents types de requête disponibles dans Lucene

Technique de projection de la RTO sur les documents Les annotations sémantiques sont créées à partir de la détection des termes dénotant les concepts de la RTO. Dans un souci de simplicité et d'efficacité, nous avons construit ce processus à partir d'un moteur de RI classique, via l'utilisation de Lucene :

1. Le label de chaque terme de la RTO est traité par le module d'analyse lexicale de Lucene : ce module découpe le label en une liste de différents mots, retire ceux qui apparaissent dans un anti-dictionnaire classique (prépositions, articles, conjonctions . . .), et radicalise les autres selon un algorithme similaire à [Porter, 1980] adapté pour le français. Les entités restantes sont alors combinées dans une requête de type "fenêtre de mots" (cf tab. 5.1) de taille paramétrable.
2. Chaque requête obtenue à partir du label d'un terme est soumise au moteur de recherche de Lucene qui restitue une liste de documents la contenant, avec pour condition expresse que tous les "mots" de la requête doivent être retrouvés dans l'ordre et dans la fenêtre de mots spécifiée.
3. S'ensuit la phase de créations des annotations sémantiques à proprement parler : chaque occurrence de terme trouvée (représentée dans notre méta-modèle par une instance de classe) doit être associée à une instance de concept adaptée. Se posent alors les problèmes de désambiguïsation sémantique et d'anaphore. Pour l'instant, et ce malgré l'absence de terme polysémique sur notre domaine, TextViz est prévu pour créer automatiquement autant de liens de dénotation que de sens possibles pour le terme, à la charge de l'ontographe de choisir par la suite l'interprétation adéquate. Concernant le problème d'anaphore, nous nous fondons sur la concision des documents dans notre étude (cf 5.1.1.1) et ne créons d'instance de concept que dans le cas où aucune du même type n'existe déjà sur le document en cours d'indexation.

Description des critères et de l'étape de vérification Parmi les critères, nous faisons en 4.1.2.1 la distinction entre génériques et spécifiques :

– d'un point de vue indépendant du domaine d'application, il est nécessaire que l'évolution de la RTO (dont sa base d'instances) n'entraîne pas d'incohérence globale ;
– selon les besoins applicatifs et le domaine modélisé, certains critères particuliers doivent être définis.

Pour la vérification systématique de la cohérence d'une ontologie, il existe plusieurs raisonneurs logiques comme RacerPro [9] ou Pellet [10] qui peuvent s'interfacer directement avec Protégé-OWL. Toutefois, les échanges d'informations entre les deux parties respectent le formalisme DIG [Turhan *et al.*, 2006] qui, dans sa version 1.1 (i.e. celle intégrée dans Protégé-OWL 3.3.1), n'avait pas une expressivité comparable à celle de OWL-DL. Comme nous utilisons dans notre méta-modèle certaines constructions non traduisibles en DIG 1.1 (e.g. les restrictions de valeurs d'attribut), nous n'avons pu à ce jour mettre en place l'étape de vérification de la cohérence de l'ontologie. Cette limite n'est que temporaire puisque DIG 2.0 remédie au problème, nous pourrons donc intégrer dans notre extension la vérification de cohérence dès que Protégé-OWL supportera cette version.

Au niveau des critères spécifiques à OBIR, nous en dénombrons deux : tout d'abord, un document indexé doit contenir au moins un symptôme, ce qui, étant donné les contraintes de cardinalités minimales sur les relations `a_problème`, `a_prestation` et `affecte` (cf fig. 4.7 et 5.4), impose la présence d'au moins un problème et une prestation tous deux compatibles (i.e. que la prestation soit incluse dans le codomaine de la relation `affecte` pour le problème donné). En parallèle, comme le champ symptôme d'une fiche de la base d'expériences contient le plus d'informations aidant à la découverte d'une solution acceptable, nous cherchons à y reconnaître un maximum de termes. C'est pourquoi nous instaurons un seuil relativement élevé (0.8) pour la couverture minimale de ce champ.

L'étape de vérification des critères (et notamment celui concernant la détection minimale d'un symptôme) s'avère insuffisante pour orienter le processus de création des symptômes [11] de façon assez précise. En effet, si un document contient plus d'un problème et d'une prestation, les critères ne permettent pas de conclure quant à la combinaison correcte pour obtenir une représentation sémantique fidèle des symptômes présents. Nous avons donc développé un ensemble d'heuristiques permettant en sortie de proposer des symptômes à l'ontographe afin de l'aider dans cette tâche.

Nous pouvons résumer en quelques points cet enchaînement d'heuristiques, décrit plus en détail dans l'algorithme 5.1 :

1. **Création des instances de prestation implicites** (lignes 1-5) : certaines prestations sont implicitement présentes dans un document à travers la mention d'un ou plusieurs composants participant sans équivoque à la réalisation de celles-ci. En créant pour chaque prestation une annotation sémantique associée, cette étape les rend explicites.

2. **Distinction entre problèmes spécifiques et génériques** (lignes 6-9) : nous distinguons problèmes génériques, qui peuvent affecter n'importe quelle prestation véhicule, et problèmes spécifiques, affectant un nombre limité d'entre elles.

[9] http ://www.racer-systems.com/
[10] http ://pellet.owldl.com/
[11] Dans notre application, un symptôme n'a de manifestation linguistique qu'à travers celle du problème et de la prestation associés, ainsi que des contextes éventuels.

3. **Gestion des problèmes spécifiques** (lignes 10-26) : nous calculons pour chaque problème spécifique la prestation compatible la plus proche à partir de l'éloignement des termes associés dans le document. Nous créons ensuite le lien `affecte` entre les deux instances. S'il n'existe aucune prestation compatible et que le problème traité ne peut en affecter qu'une, nous créons une instance artificielle de cette prestation (en supposant qu'elle est implicite).
4. **Gestion des problèmes génériques** (lignes 27-28) : nous utilisons là encore l'éloignement dans le document pour constituer des paires problème-prestation susceptibles de constituer chacune un symptôme. De façon synthétique, nous calculons toutes les combinaisons possibles et gardons celle qui minimise l'éloignement total.
5. **Gestion des contextes** (lignes 29-32) : nous relions tout contexte avec le problème le plus proche.
6. **Création des instances de symptômes** (lignes 33-37) : nous créons enfin une instance de symptôme (et les liens adéquats) pour chaque problème affectant une prestation. Si une expression négative (e.g. "*ne [...] pas*", "*aucun*", "*sans*" ...) est retrouvée dans le voisinage immédiat du terme associé au problème courant, l'instance de symptôme créée correspondra à une absence d'observation du symptôme. Ceci nous permet par la suite de représenter correctement des exemples du type : "*le moteur tousse sans allumage du témoin.*".

Comme nous utilisons des heuristiques afin d'enrichir automatiquement les annotations sémantiques, il existe une probabilité non nulle que certaines informations issues de cet enrichissement soient incorrectes. Dans notre cas particulier, les passages à indexer sémantiquement sont très concis et nous souhaitons privilégier une annotation aussi riche et précise que possible. C'est pourquoi nous avons pris la décision de ne pas stocker automatiquement un document qui satisferait pourtant tous les critères définis, mais de le présenter à l'ontographe afin qu'il valide les annotations issues des heuristiques (notamment tous les symptômes). D'un point de vue plus général, ce choix dépend fortement de la longueur du texte à indexer (plus le texte est long, plus l'utilisateur passe de temps à vérifier les annotations) et du degré de précision souhaité : si une approche préfère privilégier le rappel du processus d'indexation sémantique, il est envisageable de sauter la phase de correction manuelle et de directement stocker un document répondant aux critères de satisfaction exprimés au préalable.

Navigation dans les résultats de projection De façon à assurer à l'ontographe un parcours des résultats d'indexation simple et agréable, nous avons mis en place dans TextViz un certain nombres de fonctionnalités (cf fig. 5.12) :

- La liste des documents à parcourir pour vérification manuelle est visible dans le quart supérieur gauche, ainsi que des informations sur la couverture de l'indexation ou sur les types de concepts retrouvés directement dans chaque document. Le bouton contextuel "*To indexed files*" permet d'afficher, à cette place, la liste des documents indexés (i.e. respectant les critères d'évaluation prédéfinis et/ou validés par l'utilisateur) avec les mêmes types d'information.
- Le contenu du document sélectionné dans la liste est affiché dans le quart inférieur gauche, avec un surlignage des termes de la RTO reconnus au cours de l'indexation.

Algorithme 5.1 — Heuristique de création des symptômes

pour $i_{comp} \in E_{comp} \mid Codom(participe, type(i_{comp}))\ instanceOf\ Named\_Class$ **faire**
  si $\nexists i_{prest} \in E_{prest} | (type(i_{prest}) = Codom(participe, type(i_{comp})))$ **alors**
    créer i_{prest} et l'ajouter à E_{prest}
  fin
fin

pour $i_{pb} \in E_{pb}$ **faire**
  si $Codom(affecte, type(i_{pb})) \neq Prestation$ **alors** ajouter i_{pb} à E_{pb_specif}
  sinon ajouter i_{pb} à E_{pb_gen}
fin

pour $i_{pb} \in E_{pb\_specif}$ **faire**
  trouver
  $i_{pr0} \in E_{prest} \mid [taxo(C_{pr0}, Codom(affecte, type(i_{pb})))] \wedge [dist_{geo}(i_{pb}, i_{pr0}) = Min_{prest}(dist_{geo}(i_{pb}, i_{prest}))]$
  si $i_{pr0} \neq null$ **alors**
    créer lien $affecte$ entre i_{pb} et i_{pr0}
    si $(Codom(affecte, type(i_{pb})) \subset C_{pr0})$ **alors**
      $type(i_{pr0}) = Codom(affecte, type(i_{pb}))$
    fin
    ajouter i_{pb} à E_{pb_Relie}
  fin
fin
pour $i_{pb} \in (E_{pb\_specif} \setminus E_{pb\_Relie})$ **faire**
  si $Codom(affecte, type(i_{pb}))\ instanceOf\ Named\_Class$ **alors**
    créer $i_{prest\_artif} \mid type(i_{prest\_artif}) = Codom(affecte, type(i_{pb}))$
    créer lien $affecte$ entre i_{pb} et i_{prest_artif}
    ajouter i_{pb} à E_{pb_Relie}
  fin
fin

pour $(i_{pb}, i_{pr}) \in E_{pb\_gen} \times E_{prest}$ **faire** calculer et stocker $dist_{geo}(i_{pb}, i_{pr0})$
créer lien $affecte$ entre $i_{pb}^{k[j0]}$ et $i_{prest}^{k[j0]}$ vérifiant
$\sum_{k[j0]} dist_{geo}(i_{pb}^{k[j0]}, i_{prest}^{k[j0]}) = Min_j[\sum_{k[j]} dist_{geo}(i_{pb}^{k[j]}, i_{prest}^{k[j]})]$

pour $i_{ctxt} \in E_{ctxt}$ **faire**
  trouver $i_{pb0} \in E_{pb} \mid dist_{geo}(i_{pb0}, i_{ctxt}) = Min_{pb}[dist_{geo}(i_{pb}, i_{ctxt})]$
  créer lien $survient\_dans$ entre i_{pb0} et i_{ctxt}
fin

pour $i_{pb} \in E_{pb\_relie}$ **faire**
  si $\exists$ *negation mitoyenne* **alors** créer $i_{sympt} \mid type(i_{sympt}) = Symptome\_absent$ à partir de i_pb
  sinon créer $i_{sympt} \mid type(i_{sympt}) = Symptome\_present$ à partir de i_{pb}
  ajouter i_{sympt} à E_{sympt}
fin

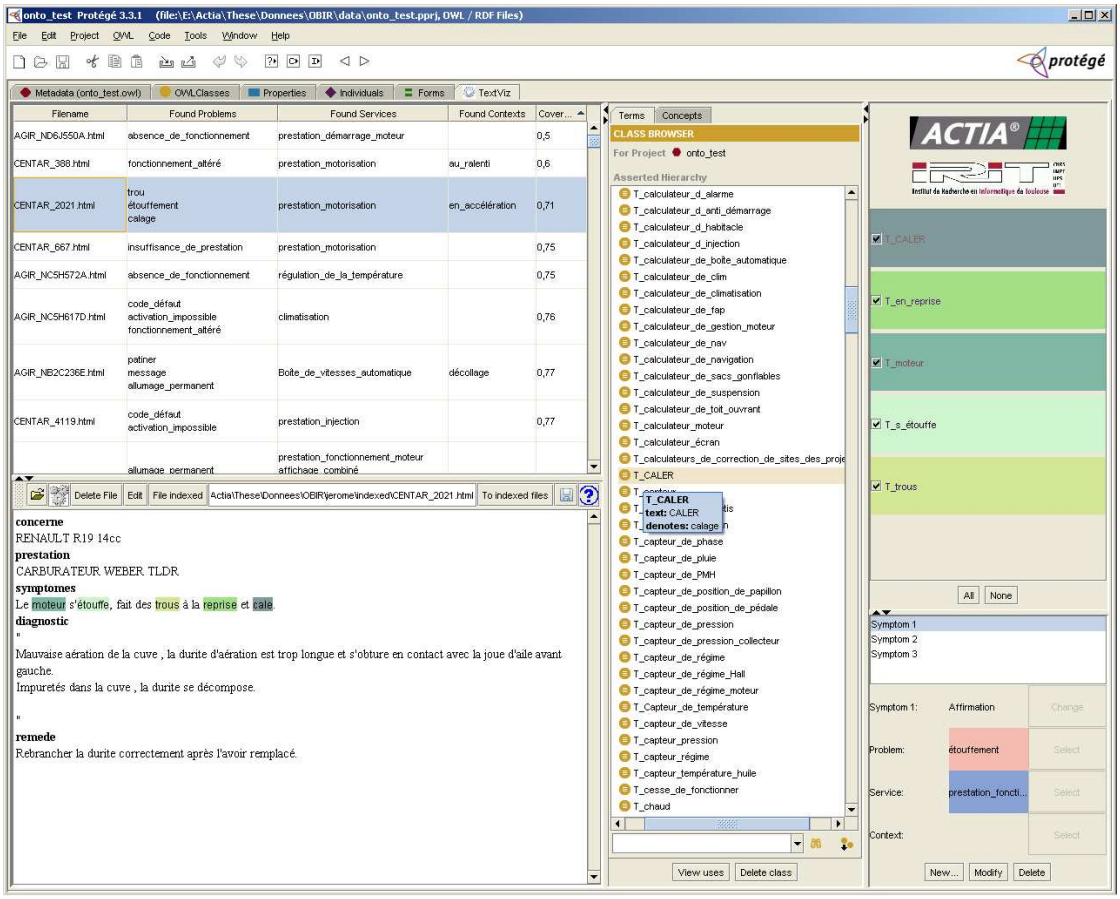

Figure 5.12 — Copie de l'écran principal de TextViz

Le code couleur correspondant est indiqué à droite et permet de sélectionner quelles instances surligner dans le texte (par défaut, elles le sont toutes).

– dans le coin inférieur droit, se situe la zone d'édition réservée aux concepts devant être instanciés un certain nombre de fois dans chaque document, en l'occurrence, dans notre contexte, les instances de symptôme. Pour le document en cours de vérification, l'ontographe peut ainsi, selon les propositions qui lui sont faites, ajouter, modifier ou supprimer un symptôme associé.
– Au centre de l'écran, on trouve une représentation arborescente de la partie ontologique ou terminologique de la RTO, selon l'onglet sélectionné (*Terms* | *Concepts*). Lorsque l'ontographe clique sur un terme surligné dans le document, le terme (ou le concept, selon l'onglet actif) correspondant est sélectionné dans l'arbre. Une fois un terme choisi dans l'arbre, l'ontographe peut, en cliquant sur le bouton "*View uses*", visualiser la liste des documents indexés avec ce terme (ou ce concept [12]). Cette fonctionnalité peut notamment lui être utile pour décider de faire dénoter au terme sélectionné un concept plus spécifique que le concept désigné initialement (si il a pu observer une hétérogénéité dans les contextes d'utilisation de ce terme).

[12] Dans ce cas, sont recensés tous les documents indexés par le concept ou l'un de ses hyponymes.

Fonctionnalités d'édition de RTO Pour modifier des éléments de la RTO, l'ontographe a deux possibilités : soit il utilise les nombreuses fonctionnalités des onglets de base de Protégé-OWL (i.e. `OWLClasses`, `Properties`, ou `Individuals`), soit il se sert de certaines interfaces développées spécialement pour TextViz (cf fig. 5.13) : pour le moment, il peut ajouter un terme (par sélection des mots dans le document) ou un concept (menu contextuel de l'arborescence), les modifier par un double-clic dans l'arbre, ou encore les supprimer avec le bouton "*Delete class*". La suppression d'une instance reconnue dans le document se fait par clic droit sur le groupe de mots surligné. L'ontographe peut en outre visualiser les termes (respectivement les concepts) associés à un concept (resp. un terme) via une bulle d'aide apparaissant au passage du curseur sur un élément de l'arborescence (cf bulle pour `T_caler` sur la fig. 5.12).

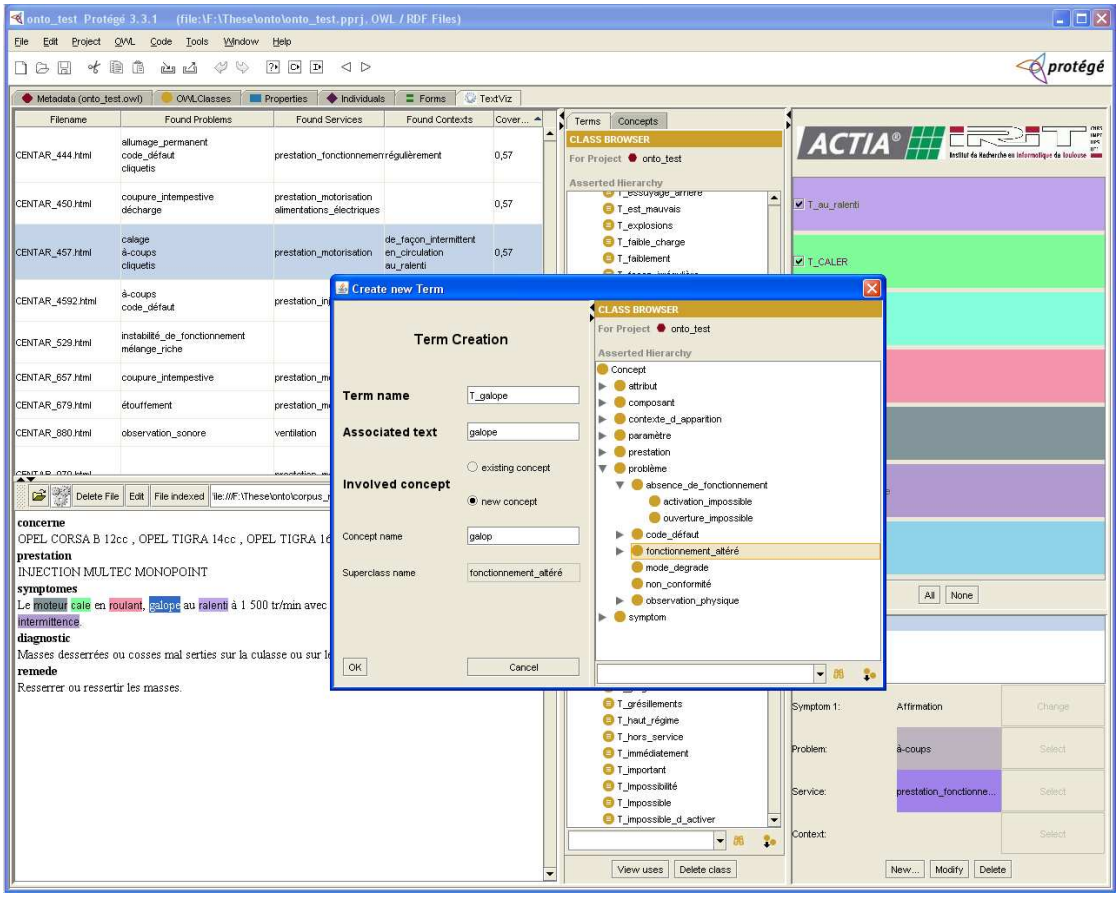

Figure 5.13 — Ajout d'un nouveau terme dans la RTO avec TextViz

Gestion des annotations sémantiques après modification de la RTO Contrairement à ce que nous prévoyions en 4.1.2.3, nous n'avons pas implémenté une gestion uniforme des conséquences d'une opération de modification de la RTO sur les annotations sémantiques : si les traitements sont déclenchés systématiquement après certaines modifications (comme la suppression d'un terme ou d'un concept), d'autres peuvent l'être manuellement, après plusieurs autres évolutions (e.g. l'ajout de plusieurs nouveaux termes). Ce choix a pour

avantage d'éviter un trop grand nombre de reprojections sur les documents en cours de validation : la projection de la terminologie sur chaque document est rapide mais, en cas de répétitions trop fréquentes, elle risque de devenir un fardeau pour l'utilisateur. En permettant à l'ontographe de contrôler le lancement du processus, nous le laissons libre de choisir entre une reprojection systématique pour tout ajout de terme à la RTO et une reprojection unique après un certain nombre d'ajouts, qui s'avèrent plus nombreux en début du cycle de maintenance.

Ces écarts à la méthode théorique proviennent à la base de deux facteurs : d'abord, nous ne disposons pas encore avec Protégé-OWL de la technologie nécessaire pour vérifier la cohérence de l'ontologie (voir plus haut le passage sur la définition des critères pour le diagnostic automobile), ensuite nous présentons systématiquement à l'ontographe tout document pour validation (comme nous l'avons déjà expliqué, nous souhaitons éviter, sur notre domaine, toute imperfection dans l'indexation). Cela nous a donc amené à repenser la gestion des annotations suite à une évolution de RTO.

Nous avons suivi une heuristique relativement claire : si l'ensemble des annotations sémantiques d'un document validé se trouve réduit (e.g. à cause d'une suppression de concept ou de terme), alors le document est reclassé comme à confirmer. Dans le cas contraire, le document n'est ni réindexé ni soumis aux critères de bonne annotation.

Stockage et format de la RTO Pour enregistrer ou charger un projet de maintenance (i.e. une RTO et la base de recherche), nous avons choisi de séparer la partie instanciée du reste de la RTO. En effet, les instances de terme ou de concept sont spécifiques aux documents de la base, alors que les concepts et les termes de la RTO peuvent être réutilisés pour indexer une autre base de recherche. En séparant la partie instance via le mécanisme d'importation de OWL (grâce à la syntaxe `<owl:import>`, une ontologie peut importer les classes et/ou instances d'une autre ontologie), nous rendons plus facile toute utilisation de la RTO en parallèle.

Nous souhaitons également aborder ici le problème du format informatique de la RTO : nous avons jusqu'à présent utilisé la sauvegarde de la RTO dans un fichier local en fin de session, et chargé en mémoire, via l'API de Protégé-OWL, l'ensemble de l'ontologie (classes et instances) en début de session suivante. Ce parti-pris soulève néanmoins plusieurs questions relatives au partage de la RTO et au passage à l'échelle de l'outil. De façon à résoudre ces deux problèmes, nous avons essayé d'utiliser la possibilité donnée par Protégé-OWL de stocker une ontologie dans une base de données et d'y accéder à distance (sans chargement de la totalité en mémoire). A ce jour, nous avons malheureusement constaté que si la sauvegarde et les modifications de l'ontologie directement dans une base de données fonctionne, les temps d'accès à celle-ci, au moyen des primitives de l'API de Protégé-OWL, sont bien trop importants pour constituer une alternative viable. Par la suite, nous comptons nous intéresser plus en détail à ces problématiques et, si aucune amélioration n'est constatée dans les prochaines versions de Protégé-OWL, nous chercherons à réimplémenter le processus de dialogue entre l'outil et une base de données distante.

5.1.3 Calcul de la similarité entre symptômes

Dans cette partie, nous reprenons en l'adaptant la contribution théorique développée en 4.2.2 sur le calcul de proximité sémantique entre deux groupes d'instances de concept reliées entre elles.

5.1.3.1 Comparabilité et appariement sémantique

Pour la notion de **comparabilité** de concepts, étant donnée la forme que nous avons retenue pour la définition d'un symptôme (cf fig. 4.7), nous considérons comparables deux concepts uniquement s'ils sont identiques ou tous deux hyponymes d'un même concept de type `Symptôme`, `Problème` ou `Prestation`, ou encore, si ce sont des contextes, que leur hypéronyme commun le plus spécifique soit de profondeur taxonomique minimale 3 (voir en 5.1.1.3). On peut faire plusieurs remarques ou précisions à la suite de cette définition :

- nous n'effectuons aucune comparaison entre composants car, comme expliqué plus haut, leur présence n'est utilisée que pour rendre explicites dans les annotations sémantiques les prestations réalisées par leur truchement
- nous n'autorisons que dans un cas spécifique la comparaison entre deux symptômes dont l'un correspond à une constatation réelle tandis que l'autre correspond à une absence d'observation d'un symptôme. En effet, il est en règle générale illogique de vouloir comparer deux entités qui n'appartiennent pas à la même sous-ontologie. Nous ne permettons de comparer un symptôme et une absence d'un symptôme que si leurs caractéristiques sont identiques (i.e. problème, prestation et contextes identiques) : dans cette situation, la similarité est nulle et permet ainsi, pour un symptôme donné dans la requête, de déclasser les documents mentionnant sa non observation.
- tous les contextes ne sont pas comparables, il semble par exemple difficile de comparer un contexte lié à une température avec un contexte lié à la vitesse du véhicule [13].

Au niveau des **heuristiques d'appariement**, la structure d'un symptôme (un et un seul problème, une et une seule prestation, un ou plusieurs contextes potentiels) rend l'appariement des structures relativement simple. Il suffit donc de commencer à apparier les deux graphes via leur nœud central, à savoir les instances de symptômes. En suivant les relations obligatoires `a_problème` et `a_prestation`, on obtient immédiatement l'appariement des problèmes et des prestations. Enfin, en suivant (si elle existe) la relation `survient_dans` à partir de l'instance de problème de la requête, on peut apparier chaque contexte du symptôme de la requête avec le contexte compatible le plus proche mentionné dans le document (s'il en existe un).

5.1.3.2 Définition d'une proximité sémantique de symptômes

Nous abordons maintenant le calcul de $Sim_{loc}(i_1, i_2)$, i.e. la similarité locale entre deux instances de concepts. Comme il a déjà été dit, cette valeur se fonde sur trois types de données différents :

[13] Il serait toutefois intéressant d'étudier la possibilité d'ajout de liens transverses de causalité, permettant de savoir, par exemple, qu'un véhicule en circulation est rarement au point mort.

– la ressemblance des concepts dont les instances sont issues (similarité conceptuelle),
– la similarité des instances en fonction de leurs caractéristiques communes, i.e. leurs valeurs d'attributs en commun (similarité en attributs),
– la comparaison des instances auxquelles chaque instance est liée par une relation sémantique (similarité relationnelle).

Concernant la similarité en attributs, nous n'avons pas pu tester son efficacité par la pratique car la modélisation que nous avons faite du domaine du diagnostic automobile n'a entraîné la création d'aucun attribut de concept. En effet, nous avons implémenté une solution qui décrit les caractéristiques définitoires d'un concept à travers les critères de différenciation de l'arbre taxonomique et les relations sémantiques qui le relient à d'autres concepts de l'ontogie (cf fig. 5.4).

Similarité conceptuelle Nous avons implémenté les deux mesures classiques de [Wu et Palmer, 1994] et [Lin, 1998], dans le but d'étudier leur efficacité respective au sein de la similarité locale. Ce faisant, nous avons pu constater un avantage supplémentaire à notre méta-modèle de RTO : la mesure proposée par Lin, de même que [Resnik, 1995], se fonde sur la notion de contenu d'information d'un concept, fonction du nombre d'occurrences en corpus des termes liés à ce concept (voir en 2.2.2.2). La réification de la notion de terme et la représentation explicite du lien de dénotation entre termes et concepts dans notre méta-modèle rend l'implémentation de la mesure de Lin relativement aisée. Les deux types de mesures implémentées sont utilisés pour la comparaison de concepts de type `Problème` ou `Prestation`. Nous rajoutons toutefois une contrainte dans le cas de deux prestations : si les deux prestations comparées ne sont pas reliées via une ou plusieurs relations d'hypéronymie, leur similarité conceptuelle sera nulle. Nous avons mis en place cette obligation pour traduire le fait qu'il est inutile d'essayer de rapprocher deux symptômes qui ciblent deux prestations sans lien direct : les réparations associées seraient certainement fortement différentes, ce qui est contraire à notre objectif initial de proposer à un garagiste une méthode de réparation en fonction des symptômes rencontrés.

La similarité conceptuelle entre deux contextes compatibles est calculée de façon différente. En effet, parmi la plupart des sous-groupes de contextes comparables, la structure taxonomique ne permet pas de distinguer deux concepts plus proches que les autres, car ils sont tous hyponymes directs du même concept (même profondeur). Pourtant, certains de ces groupes de contexte possèdent une relation d'ordre total qui doit permettre de calculer de façon plus intuitive une valeur de similarité selon la nature des concepts comparés : par exemple, le contexte de faible vitesse semble plus proche de celui de vitesse moyenne que de celui de grande vitesse. Nous avons décidé de représenter explicitement la relation d'ordre sous forme d'une relation sémantique transverse `inférieur_à` entre les différents contextes comparables. De cette façon, on peut calculer une distance entre deux concepts : on compte le nombre de relations `inférieur_à` séparant les deux concepts, puis on ramène cette valeur entre 0 et 1 en la divisant par le nombre de relations `inférieur_à` séparant le plus petit concept du plus grand du groupe. En prenant le complément à 1, on obtient alors une valeur plus intuitive de la similarité conceptuelle de ces contextes. La figure 5.14 permet d'illustrer notre propos : comme le sous-arbre taxonomique de racine C_0 comporte 6 relations `inférieur_à`, on peut visualiser la situation comme un segment découpé en 6 al-

lant du plus petit concept (i.e. C_4) au plus grand (i.e. C_{10}). Les concepts qui, comme C_1,C_2 et C_3, possèdent des hyponymes ordonnés entre eux, sont traités différemment. En effet, nous considérons que leur similarité avec leurs hyponymes est maximale et que leur similarité avec un autre concept est calculée selon la formule :

$$Sim_{cpt}(C_1, C_2) = Max_{(i,j)}(Sim_{cpt}(hypo_i(C_1), hypo_j(C_2)))$$

Nous pouvons ainsi calculer la similarité conceptuelle entre deux contextes comparables :

$$Sim_{cpt}(C_7, C_8) = 1 - \frac{1}{6} = \frac{5}{6}$$

$$Sim_{cpt}(C_5, C_9) = 1 - \frac{4}{6} = \frac{1}{3}$$

$$Sim_{cpt}(C_1, C_9) = Max_i(Sim_{cpt}(hypo_i(C_1), C_9)) = Sim_{cpt}(C_6, C_9) = \frac{1}{2}$$

On peut remarquer que, contrairement à des mesures fondées sur la structure taxonomique, la stratégie développée ci-dessus ne tient pas compte, pour la similarité de deux concepts de même profondeur, de la profondeur de leur hypéronyme commun le plus spécifique :

$$Sim_{cpt}(C_5, C_6) = Sim_{cpt}(C_6, C_7)$$

Ce phénomène ne nous dérange pas car nous considérons que la prise en compte de l'ordre des relations `inférieur_à` prime sur celle de la taxonomie. Si nous souhaitions corriger ce phénomène, nous pourrions toutefois pondérer chaque relation `inférieur_à` selon sa profondeur taxonomique, de façon à augmenter la distance entre deux concepts reliés selon qu'ils possèdent ou non des hyponymes :

$$Sim_{cpt}(C_1, C_2) = 1 - \frac{\sum_{(i,j) \in path(C_1,C_2)}(weight_{inferieur\_a}(depth(C_i))}{\sum_{(i,j) \in longest\_path}(weight_{inferieur\_a}(depth(C_i))}$$

Similarité relationnelle Nous avons produit une implémentation possible de la méthode proposée en 4.2.2 pour prendre en compte la similarité relationnelle entre deux instances. Nous soulignons le fait que cette implémentation est spécifique à notre contexte d'utilisation et que tenter de la généraliser amènerait de nombreuses questions dépassant largement le cadre de notre étude. Afin d'expliciter notre implémentation, nous définissons d'abord $R^{rel_j}_{inst_i}$ comme l'ensemble des instances pointées par $inst_i$ via une relation de type rel_j, et $R^{rel_j}_{all}$ comme l'ensemble des instances pointées par une relation de type rel_j. Notre formule de similarité relationnelle entre deux instances i_1 et i_2 est alors de la forme suivante :

$$Sim_{rel}(i_1, i_2) = \mu * \frac{\sum_{rel_j}^{oblig} w(rel_j) * \sum_{i_{req}}^{R^{rel_j}_{i_1}} Max_{i_{doc} \in R^{rel_j}_{i_2}}[Sim_{loc}(i_{req}, i_{doc})]}{\sum_{rel_j}^{oblig} w(rel_j) * Card(R^{rel_j}_{i_1})}$$
$$+ (1 - \mu) * \frac{\sum_{rel_j}^{facult} w(rel_j) * Prox_{facult}(i_1, rel_j)}{\sum_{rel_j}^{facult} w(rel_j) * Card(R^{rel_j}_{i_1})}$$

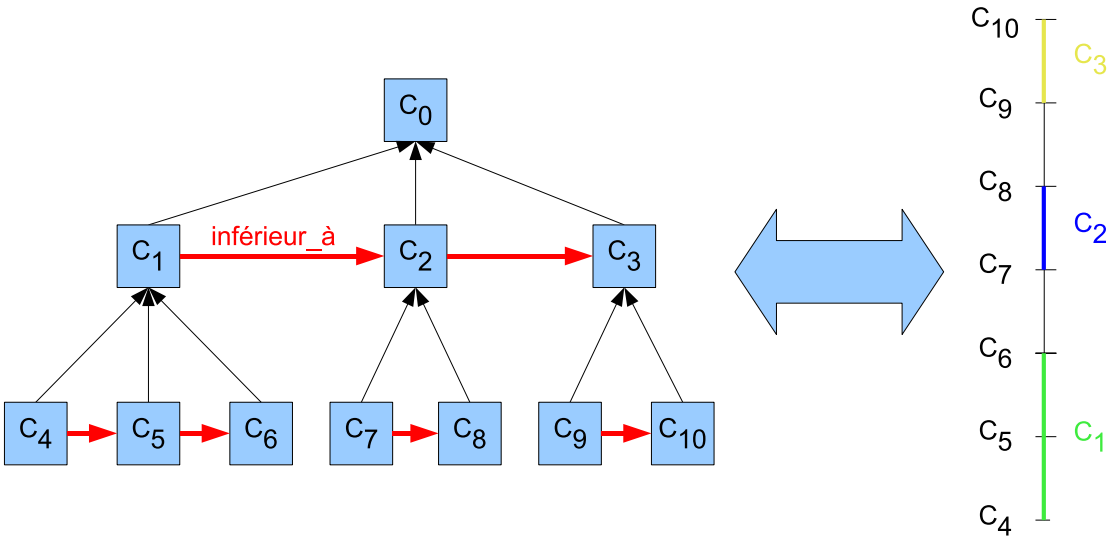

Figure 5.14 — Illustration de l'ajout de relations d'ordre pour le calcul de similarité conceptuelle entre contextes

$$Prox_{facult}(i_1, rel_j) = \begin{cases} \sum_{i_{req}}^{R_{i_1}^{rel_j}} Max_{i_{doc} \in R_{i_2}^{rel_j}} [Sim_{fixe}(i_{req}, i_{doc})] & \text{si } R_{i_2}^{rel_j} \neq \emptyset, \\ \\ \frac{1}{card(R_{all}^{rel_j})} * \sum_{i_{req}}^{R_{i_1}^{rel_j}} \sum_{i_m}^{R_{all}^{rel_j}} Sim_{fixe}(i_{req}, i_m) & \text{sinon.} \end{cases}$$

$$Sim_{loc}(a,b) = \alpha * Sim_{cpt}(a,b) + \beta * Sim_{attr}(a,b) + \gamma * Sim_{rel}(a,b) \text{ avec } \alpha + \beta + \gamma = 1$$

$$Sim_{fixe}(a,b) = \frac{1}{\alpha + \beta} * [\alpha * Sim_{cpt}(a,b) + \beta * Sim_{attr}(a,b)]$$

On constate tout d'abord que la similarité relationnelle est composée de deux parties, correspondant aux relations obligatoires d'une part, facultatives de l'autre. Les deux parties sont relativement semblables, à la différence près que la première fait intervenir la similarité locale des paires d'instances reliées à la paire comparée (ce qui la rend récursive), tandis que la seconde s'affranchit de la partie relationnelle sur les paires reliées. Nous justifions cette décision de façon intuitive : une relation obligatoire relie des informations plus difficilement dissociables qu'une relation facultative, ce qui rend plus intéressant dans le premier cas de connaître précisément la valeur de similarité des paires d'instances reliées. On peut considérer le calcul avec Sim_{fixe} dans le cas d'une relation facultative comme une approximation permettant d'éviter la complexité d'un calcul récursif. La seconde remarque concerne la définition de $Prox_{facult}(i_1, rel_j)$: par définition, la relation rel_j n'est pas forcément instanciée dans le document, ce qui amène à distinguer deux situations. Si rel_j est instanciée dans le document, le calcul suit la forme de celui défini en cas de relation obligatoire ; sinon, nous faisons la moyenne de toutes les similarités fixes entre les instances reliées par rel_j à i_1 et toutes les instances pointées par rel_j dans la base d'instances, ce qui revient à comparer les instances de $R_{i_1}^{rel_j}$ à une instance moyenne (cf 4.2.2).

Pour le calcul final de proximité entre la requête et le document, comme le nœud central d'un graphe de requête correspond à une instance de la classe `Symptôme`, nous avons en théorie :

$$Prox_{tot}(req, doc) = \frac{\sum_{i^{req}_{sympt}} Max_{i^{doc}_{sympt}}[\alpha * Sim_{cpt}(i^{req}_{sympt}, i^{doc}_{sympt}) + \gamma * Sim_{rel}(i^{req}_{sympt}, i^{doc}_{sympt})]}{Card(\{i^{req}_{sympt} \mid type(i^{req}_{sympt})\ instanceOf\ Symptome\})}$$

avec $\alpha + \gamma = 1$ (comme nous n'utilisons pas la similarité en attributs, nous prenons $\beta = 0$)

Toutefois, nous sommes amenés à gérer le cas particulier de la comparabilité de deux symptômes : comme nous l'expliquions plus haut, les seuls concepts possibles pour un symptôme correspondent soit à une observation de panne, soit à l'absence d'une observation donnée. La similarité conceptuelle entre deux instances de symptômes est donc de la forme "booléenne" (i.e. 0 ou 1). En gardant la formule donnée de calcul de similarité entre ces deux entités, nous ne traduisons pas la réalité, à savoir que deux symptômes sont absolument dissemblables si l'un désigne une observation constatée et l'autre la constatation d'une absence de symptôme. A l'inverse, la formule proposée aurait tendance à augmenter artificiellement de α la valeur de similarité de deux symptômes constatés ou non constatés. Si l'ordre de retour des documents ne semble pas menacé par ce phénomène (puisque α sera ajouté systématiquement pour deux symptômes comparables), les scores finaux risquent de troubler l'appréciation des résultats de l'utilisateur. Nous adaptons donc la formule de la façon suivante :

$$Prox_{tot}(req, doc) = \frac{\sum_{i^{req}_{sympt}} Max_{i^{doc}_{sympt}}[Sim_{cpt}(i^{req}_{sympt}, i^{doc}_{sympt}) * Sim_{rel}(i^{req}_{sympt}, i^{doc}_{sympt})]}{Card(\{i^{req}_{sympt} \mid type(i^{req}_{sympt})\ instanceOf\ Symptome\})}$$

5.1.3.3 Interface graphique

Bien qu'à terme, la phase d'exploitation de la RTO par recherche sémantique est amenée à être séparée de l'outil de maintenance, nous avons, pour des raisons de simplicité, implémenté l'ensemble des opérations au sein d'une même interface. Ainsi, il est possible d'accéder à la fenêtre de recherche par le bouton illustré d'un point d'interrogation (cf fig. 5.12). Celle-ci comporte des informations de différents types, comme on peut le voir sur la figure 5.15 :

- la partie gauche permet la saisie de la requête et le paramétrage de plusieurs coefficients relatifs à la mesure de similarité,
- la partie droite permet à l'utilisateur de visualiser l'interprétation sémantique de sa requête par le système [14] (e.g. sur la copie d'écran, deux symptômes ont été reconnus),
- la partie inférieure permet de visualiser la liste, ordonnée par valeurs décroissantes de similarité, des documents pertinents par rapport à la requête [15] ; un double-clic sur une ligne de résultat permet d'afficher le document correspondant.

[14] A ce jour, aucune aide à la reformulation n'est proposée. En cas de traduction erronée des symptômes de la part du système, l'utilisateur doit reformuler sa requête jusqu'à obtenir une interprétation satisfaisante.

[15] Bien que prévu, le tri par applicabilité de véhicule, évoqué en 5.1.1.1, n'a pas encore été mis en place.

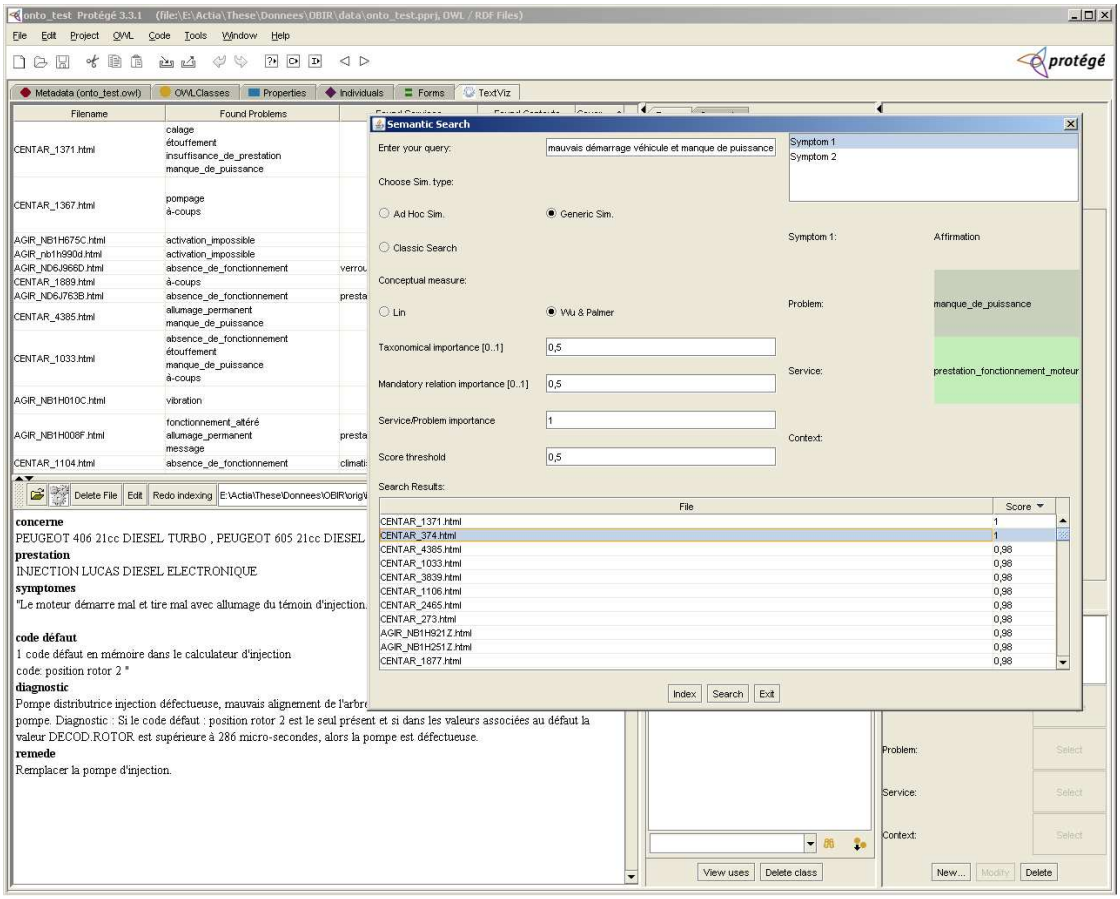

Figure 5.15 — Interface de recherche dans TextViz

5.2 Evaluation du système

5.2.1 De la difficulté d'évaluer

Comme nous l'avons expliqué dans la section précédente, nos contributions théoriques ont été implémentées au sein d'une même extension de Protégé-OWL qui remplit deux types de fonctionnalités pour deux catégories différentes d'utilisateurs :

- la première partie de l'outil permet à un ontographe (éventuellement accompagné d'un expert du domaine modélisé) d'indexer sémantiquement de nouveaux documents et de modifier la RTO courante en accord avec la valeur des critères qu'il a définis au préalable pour évaluer l'adéquation entre la RTO et les textes annotés ;
- la seconde partie de l'outil permet à un utilisateur non spécialiste de l'IC de saisir en langage libre un ou plusieurs symptômes constatés sur un véhicule, de visualiser l'interprétation sémantique qu'en fait le système et de consulter une liste de fiches de réparation pertinentes par rapport au(x) symptôme(s).

L'ensemble de nos apports théoriques se situe donc dans les domaines scientifiques de l'IC et de la RI, voire certains en transcendent parfois leur frontière, notamment le cycle de maintenance de RTO par évaluation automatique des résultats d'indexation sémantique. On peut

néanmoins considérer que nos contributions relatives à l'IC se retrouvent plutôt dans l'outil de maintenance de RTO tandis que nos contributions touchant à la RI se situent dans le module d'interrogation sémantique.

Au niveau de l'IC, l'évaluation des résultats est une tâche ardue : comme l'explique [Aussenac-Gilles, 2005], il est délicat de trouver des critères de validité d'une approche et il n'existe aucune métrique consensuelle, au contraire de certains domaines comme la RI. Le problème vient notamment de la difficulté intrinsèque à dissocier les différents artefacts à évaluer : méthode de construction de l'ontologie, techniques employées, ontologie résultante . . . Ensuite, se pose la question de la généricité d'une évaluation à partir de la construction d'une ontologie spécifique à une tâche et/ou un domaine. En effet, certaines réflexions comme [Bachimont, 2004] considèrent que l'étude d'un cas ne valide pas une technique ou une méthode d'IC, elle permet simplement de découvrir de nouveaux angles de critiques et de nouvelles possibilités d'adaptation à un contexte spécifique.

Pour la RI, le problème rencontré est légèrement différent. De fait, si notre approche n'était constituée que d'un enchaînement de méthodes automatiques permettant d'indexer des documents et d'émettre des requêtes sur ceux-ci, nous serions dans un cas typique de la RI et disposerions indiscutablement de moyens standards pour évaluer sa pertinence (notamment la comparaison à une référence idéale, appelée "gold standard"). Cependant, l'approche que nous proposons s'avère bien différente de la plupart des outils classiques de RI : pour les processus d'indexation et d'interrogation, nous faisons intervenir les différentes entités d'une ontologie, qui est une source de connaissances extérieure et supplémentaire aux documents indexés. Même si elle se fonde en partie sur les mêmes documents que ceux à indexer, la méthode d'obtention de cette ontologie fait obligatoirement appel à un expert du domaine. Comme un outil de RI sémantique utilise plus de connaissances qu'un outil classique, l'évaluation par comparaison peut donner lieu à critiques : de meilleurs résultats n'impliquent pas que cet outil sémantique soit "meilleur", il nécessite un travail en amont (construction de l'ontologie et vérification de l'indexation sémantique), il est limité à un domaine (un outil classique est généralement générique) . . .

Néanmoins, de façon cohérente avec les motivations pratiques qui ont initialement motivé nos recherches, nous cherchons dans un premier temps à évaluer l'intérêt de nos contributions en termes de qualité de réponse au besoin des utilisateurs. Au cours de l'évaluation de nos apports, nous chercherons donc à prouver l'intérêt d'utiliser une ontologie dans un processus de RI, tout en montrant que les coûts supplémentaires liés à une telle approche restent raisonnables. Pour cela et malgré les critiques potentielles liées à notre choix, nous comparons les résultats de notre étape d'interrogation sémantique à ceux d'un outil de RI traditionnelle. Pour mesurer les coûts engendrés par l'usage d'une ontologie, nous focaliserons notre attention sur des critères liés à la quantité de travail à fournir et au temps nécessaire pour l'indexation sémantique de nouveaux documents du domaine. Même si cette étude de cas ne peut se suffire en elle-même pour évaluer nos contributions, elle permettra néanmoins de mettre en exergue certaines de leurs qualités et défauts et constituera une première base de réflexion pour d'éventuelles modifications, et pourra être suivie de nouveaux tests sur des domaines différents.

5.2.2 Evaluation de la partie maintenance de RTO / indexation sémantique

5.2.2.1 Protocole expérimental

L'objectif principal que nous recherchons dans cette partie est de prouver la viabilité d'une approche de RI sémantique vis-à-vis des surcoûts engendrés par les étapes nécessaires d'indexation sémantique et de maintenance de RTO. Pour cela, nous nous focalisons sur deux propriétés fondamentales de TextViz, à savoir le **temps nécessaire pour accomplir ces deux tâches** (celles-ci étant réalisées en parallèle dans notre approche) et la **convergence de la RTO vers un état satisfaisant** et stable vis-à-vis de l'ensemble des documents indexés.

La première étape de notre processus d'analyse de TextViz consiste à sélectionner les documents qui seront ajoutés à la base d'expériences existante. Etant donné le laps de temps s'étant écoulé entre la dernière opération de maintenance de la RTO à partir de la base de documents et celle qui sert de base à notre analyse, nous disposons d'un nombre important de documents non indexés. Cette réalité risque d'entraîner des différences de comportement du système entre le cas d'étude et le cas typique (i.e. ajout de 50 fiches à la base d'expériences tous les 4 mois) : le nombre de termes ou de concepts à ajouter sera a priori bien plus important que dans un cas habituel. Parmi l'ensemble des documents non indexés, nous n'en gardons que 50, tous sélectionnés de façon aléatoire.

Notre évaluation se déroule ensuite selon un scénario relativement simple : après avoir choisi les documents à rajouter à la base indexée, nous lançons le processus de maintenance / indexation. Pendant cette activité de mise à jour, nous effectuons un relevé périodique d'un certain nombre de données. La surveillance de ces éléments doit permettre de visualiser l'évolution au cours du temps de la RTO et des annotations, de façon à constater (ou non) la convergence progressive des deux processus vers un état de la RTO adapté aux nouveaux documents : ainsi, l'**ajout de nouveaux termes et/ou concepts** est théoriquement plus important au début de l'activité, permettant une bonne amélioration initiale de la **couverture** [16] **moyenne des documents** tandis que les ajouts doivent peu à peu diminuer au fur et à mesure que la RTO est modifiée pour mieux indexer les nouveaux documents. Voici les éléments dont nous surveillerons l'évolution :

- la proportion de documents avec une couverture très mauvaise (0-20%), mauvaise (20-40%), moyenne (40-60%), bonne (60-80%) ou excellente,
- le nombre de termes,
- le nombre de concepts,
- le nombre de documents restant à indexer.

Concernant les algorithmes heuristiques permettant de créer les instances de symptômes, même s'ils s'avèrent spécifiques à notre RTO et au domaine modélisé, nous évaluerons néanmoins leur efficacité en mesurant la **proportion moyenne de propositions correctes** et le **nombre moyen de modifications** sur les annotations nécessaires pour les corriger.

Afin de pouvoir mieux interpréter l'évolution des valeurs surveillées, nous essayons en tant qu'ontographe de nous conformer autant que possible à une stratégie de mise à jour : au début de l'activité, nous analysons les annotations sémantiques produites sur les do-

[16] Nous rappelons que la couverture d'un document correspond à la proportion de mots plein indexés.

cuments faiblement couverts dans le but de dégager rapidement un ensemble conséquent de termes et/ou concepts manquant à l'ontologie. Une fois cette phase suffisamment avancée (découverte moins fréquente de nouveaux éléments ou intégration plus délicate), nous concentrons nos efforts sur les documents les mieux couverts et modifions ou validons les annotations sémantiques issues des heuristiques de création automatique (à savoir, dans notre cas d'étude, les symptômes). Il ne devrait alors ne nous rester à examiner que les documents les plus problématiques vis-à-vis de la RTO dans son état courant. Logiquement, nous nous attendons à ce que le temps de traitement de ces derniers documents non validés soit sensiblement plus élevé que les précédentes mesures.

5.2.2.2 Résultats

Suite à la mise en pratique du protocole d'évaluation de l'outil de maintenance de RTO et d'indexation sémantique, nous obtenons les résultats synthétisés dans le tableau 5.2.

Temps écoulé (min)	**Documents selon leur couverture**					**Docs validés**	**Annotations manuelles**	**Termes ajoutés**	**Concepts ajoutés**
	0 à 20%	**21 à 40%**	**41 à 60%**	**61 à 80%**	**81 à 100%**				
0	4	11	13	14	8	0	0	0	0
10	1	12	14	15	2	6	0	9	1
20	0	10	17	15	2	6	0	8	5
30	0	5	19	18	2	6	0	6	5
40	0	4	19	12	1	14	4	5	1
50	0	4	17	4	2	23	5	4	4
60	0	4	12	1	0	33	7	1	1
70	0	2	3	1	0	44	7	2	0
80	0	0	0	0	0	50	5	4	3

Tableau 5.2 — Données temporelles liées à l'ajout de 50 documents à la base indexée

La première constatation que nous pouvons faire concerne le temps total de mise à jour de la RTO et d'intégration des nouveaux documents à la base de textes indexés. La durée de 80 minutes nous paraît raisonnable étant donnée la tâche accomplie : 50 documents annotés sémantiquement, 39 termes et 20 concepts ajoutés à la RTO. Cette mesure de temps est toutefois soumise à caution : comme l'opération a été réalisée par nos soins et que nous avions une connaissance approfondie de la structure de la RTO, nous avons sans doute passé moins de temps que ne l'aurait fait une personne non familière de la ressource. A notre décharge, le cas que nous avons étudié possédait plus de "risques" que la situation typique dans laquelle sera utilisé l'outil : du fait du temps écoulé entre la construction de la RTO et sa mise à jour, nous avons sans doute été amené à enrichir la RTO de façon plus importante que si nous avions eu l'occasion de le faire aussi fréquemment que prévu par la suite. Pour résumer, dans le cadre de notre étude, une durée de maintenance de RTO de l'ordre de 1h30 / 2h paraît d'autant plus envisageable que cette opération n'aurait lieu qu'une fois tous les quatre mois.

Au niveau de l'ajout de termes, nous pouvons observer une évolution relativement conforme à nos attentes : le nombre de termes intégrés à la RTO, assez élevé au départ, diminue progressivement pour atteindre des valeurs faibles. La légère recrudescence en fin de traitement peut s'expliquer par le fait que les derniers documents à être validés correspondent au "ventre mou" du critère de couverture (i.e. aux alentours de 40%) : la couverture mitigée peut correspondre indifférement à un document insuffisamment indexé (amenant la création de termes, voire de concepts) ou à un document comportant des informations superflues (ou non modélisables, voir plus loin). Quant à lui, le nombre de nouveaux concepts ne suit pas une loi d'évolution monotone. Nous pensons que, pour un document donné, ce nombre est fortement dépendant des nouveaux termes rencontrés : si le terme est simplement synonyme d'un terme préexistant, il sera relié par la relation de dénotation au concept adéquat ; dans le cas contraire, il entraîne la création d'un nouveau concept, voire de plusieurs [17].

A travers l'analyse de l'évolution des différentes familles de documents en fonction de leur couverture par l'algorithme d'indexation sémantique, nous pouvons faire plusieurs remarques :

- les catégories extrêmes (i.e. entre 0 et 20% et entre 81 et 100%) tendent assez rapidement vers 0,
- les autres catégories adoptent une forme croissante puis décroissante, avec un pic d'autant plus tardif que la catégorie est proche d'une couverture de 50%

Ces observations peuvent s'expliquer assez simplement : un document très mal indexé est examiné en priorité par l'ontographe car il peut aider à découvrir de nouveaux termes (voire concepts) utiles pour l'ensemble des documents en cours d'indexation. A l'opposé, un document avec une couverture très élevée pourra être validé rapidement car il est peu susceptible de bénéficier de la découverte de nouveaux termes. Restent alors à traiter les documents à couverture moyenne. Il est difficile de dégager une loi générique quant à leur évolution, puisque celle-ci dépend fortement de leur contenu sémantique (vis-à-vis de celui de la RTO) et des choix faits par l'ontographe. Pour notre part, nous avons préféré valider en priorité les documents dont les annotations étaient les plus exhaustives et les moins sujettes à controverses. On peut observer notre choix dans l'évolution du nombre de documents validés : au fur et à mesure que le temps passait et la couverture moyenne augmentait (du fait de l'ajout de nouveaux termes), nous avons pu valider un nombre croissant de documents, jusqu'aux 10 dernières minutes. A partir de ce moment, il ne nous restait plus que les documents posant des problèmes de modélisation et pour lesquels nous avons recouru à un expert. Le choix de ne s'adresser à l'expert qu'en fin du cycle de maintenance a ainsi permis de minimiser son temps d'intervention.

En ce qui concerne l'efficacité de nos heuristiques d'annotation (spécifiques à la tâche et/ou domaine modélisés), nous avons dû effectuer une trentaine de modifications (suppression d'un symptôme incorrect, ajout d'un ou plusieurs contextes à un symptôme, ajout d'un symptôme ...) pour parvenir à des annotations sémantiques correctes sur l'ensemble des documents à indexer. Rapportée au nombre de symptômes traités (de l'ordre d'une cen-

[17] Par exemple, la mention d'un symptôme apparaissant par temps humide nous a amené à créer deux concepts : un contexte `Condition_meteo` et un hyponyme `Temps_pluvieux`, dénoté par le terme "*sous la pluie*".

taine pour les 50 documents), cette valeur nous semble relativement peu élevée. Il nous faudra toutefois examiner cette observation à la lumière des résultats du module de RI sémantique, qui utilise les mêmes heuristiques pour interpréter les requêtes du garagiste.

A la fin du cycle de maintenance proposé, nous avons réussi à obtenir une indexation satisfaisante pour l'ensemble de documents à intégrer dans la base de recherche. Nous signalons toutefois que dans certains cas, nous avons décidé de ne pas modéliser l'intégralité du sens contenu dans le champ indexé : de fait, certains documents nous ont permis de constater que la structure de notre RTO ne permet pas de représenter certains phénomènes temporels comme la succession (un symptôme qui survient systématiquement après un autre) ou la simultanéité. Nous avons préféré dans un premier temps indexer "partiellement" les documents correspondants et attendre de détenir assez d'exemples significatifs avant de mettre en place les mécanismes d'indexation capables de gérer ce genre de situation.

Pour résumer cette évaluation, nous avons pu vérifier, dans notre contexte applicatif, que notre outil permettait d'obtenir des temps acceptables en industrie (quelques heures par trimestre) pour la maintenance de la RTO en parallèle avec l'indexation sémantique de nouveaux documents. Toutefois, même si notre solution semble convenir dans notre cadre applicatif, il reste nécessaire, pour juger de la généricité de l'approche, de renouveler l'évaluation dans des conditions différentes. Selon nous, les paramètres à tester en priorité (auxquels notre approche semble la plus sensible) s'apparentent à la taille et la fréquence d'évolution de la base de recherche, ainsi qu'à la taille moyenne des documents à indexer. En effet, plus les documents à intégrer à la base de recherche seront nombreux, fréquents et longs, plus les temps de validation risquent d'augmenter.

5.2.3 Evaluation de la partie interrogation sémantique

Comme nous l'avons déclaré en 5.2.1, nous comparons les résultats de notre système avec ceux obtenus par un moteur de recherche n'exploitant aucune RTO et fondé sur une indexation par mots simples (appelé, par abus de langage, moteur de recherche classique). Après avoir décrit comment nous mettons en œuvre cette comparaison, nous exposons et commentons un ensemble de résultats.

5.2.3.1 Protocole expérimental

Considérations théoriques D'un point de vue théorique, nous cherchons à évaluer deux propriétés de notre système d'interrogation :

- la qualité de l'indexation sémantique de la requête,
- la pertinence des résultats de recherche.

Du fait de l'absence de contribution théorique réelle sur le sujet, nous ne nous intéressons que partiellement à la première caractéristique dans le cadre de l'évaluation : en effet, l'étape d'indexation est effectuée automatiquement par un ensemble d'heuristiques spécifiques à l'ontologie du diagnostic automobile que nous avons construite. Pour chaque requête traitée, nous nous bornerons donc à attribuer à son interprétation par le système une valeur parmi les cinq suivantes :

- *parfaite* : le système n'a fait aucune d'erreur d'interprétation et a traduit l'ensemble de la requête sous forme d'entités conceptuelles,
- *bonne* : le système n'a fait aucune d'erreur d'interprétation mais n'a pas réussi à traduire certains éléments mineurs,
- *moyenne* : le système a mal interprété certains éléments, mais une partie de l'interprétation reste directement exploitable,
- *médiocre* : plusieurs modifications sont nécessaires pour parvenir à une interprétation correcte,
- *nulle* : le système a fait un lourd contresens ou n'a pas réussi à interpréter la requête.

A travers la proportion de requêtes mal interprétées, cette évaluation, malgré son aspect arbitraire, nous permettra néanmoins de juger de l'urgence à étudier la problématique d'aide à la reformulation.

La pertinence des résultats de recherche s'avère bien plus importante pour nous car elle nous permettra de quantifier, par l'étude d'un cas, l'efficacité relative sur un domaine précis d'une RI sémantique par rapport à des techniques de RI classique. Pour comparer les deux types d'outil, nous avons choisi de nous fonder sur deux mesures complémentaires largement acceptées en RI, à savoir le rappel et la précision. Le rappel mesure la proportion de documents pertinents qui sont retrouvés par le système tandis que la précision évalue la proportion de documents retrouvés par le système qui sont pertinents vis-à-vis de la requête. Par rapport au tableau 5.3, les formules de ces deux mesures sont les suivantes :

$$Rappel = \frac{a}{a+c}$$

$$Precision = \frac{a}{a+b}$$

Nombre de documents	pertinents	non pertinents
retrouvés	a	b
non retrouvés	c	d

Tableau 5.3 — Exemple de valeurs pour la définition du rappel et de la précision

Afin d'obtenir une vue de l'efficacité globale de notre approche, nous combinons en proportions égales rappel et précision dans la F-mesure :

$$F - Mesure = \frac{2 * Rappel * Precision}{Rappel + Precision}$$

Le calcul de ces valeurs implique de connaître au préalable quels documents sont pertinents pour chaque requête testée. Comme nous effectuons notre évaluation sur un domaine spécifique (le diagnostic automobile) et sur un ensemble de documents donné (un sous-ensemble de la base d'expériences), nous n'avons pas pu recourir à des corpus et des requêtes pour lesquels une référence manuelle était déjà disponible (e.g. pour des campagnes d'évaluation de TREC). Nous avons donc dû, pour chaque requête testée, faire examiner

par un expert du domaine la pertinence de chaque document présent dans notre corpus. Afin de respecter certaines contraintes temporelles, nous avons décidé de nous limiter à un sous-ensemble de la base d'expériences (350 documents sélectionnés aléatoirement).

Si les mesures de rappel et de précision permettent de comparer l'efficacité de notre module de recherche à celle d'un moteur classique, elles ne prennent pas en compte l'ordre dans lequel sont retournés les documents. Cette information a d'autant plus d'importance pour notre approche que la RI sémantique s'appuie sur des modélisations de connaissances fines et permet des rapprochements d'entités plus ou moins similaires. Nous nous heurtons toutefois à la subjectivité de cette information : la tâche d'ordonnancement d'un ensemble de documents en fonction de leur pertinence à une requête s'avère d'autant plus difficile pour un opérateur humain que le nombre de documents est important.

Moteur de recherche classique Dans un premier temps, nous avons cherché à comparer notre approche de RI sémantique à un moteur de recherche classique préexistant, comme Google Desktop [18] ou Copernic Desktop Search [19]. Toutefois, l'absence de certaines fonctionnalités (recherche sur un répertoire ciblée pour le premier, recherche approchée pour le second) ainsi que l'impossibilité de connaître ou contrôler le processus de recherche de ces outils nous ont amené à réviser notre position.

Etant données la simplicité et l'expressivité de Lucene (auquel nous avons recours pour la projection de terme sur le corpus de documents), nous avons décidé de l'utiliser comme point de départ pour réaliser un moteur de recherche simple. Pour plus de facilité d'usage, nous avons intégré de façon transparente cette fonctionnalité au sein de notre interface d'interrogation. Nous avons limité l'indexation d'une fiche à celle de son champ symptôme afin que les deux approches de recherche (classique et sémantique) se fondent strictement sur les mêmes portions de documents. La phase d'interrogation repose sur des primitives fondamentales de Lucene : la requête en langage naturel subit les mêmes traitements (découpage en mots, suppression de mots vides, radicalisation) que ceux opérés en 4.1.2.2 sur les documents pour la projection de la RTO ; une fois formalisée, la requête est alors soumise à une primitive de recherche dont le score de pertinence pour chaque document fait appel à la notion de TF-IDF (définie en 1.3.3.1) sur chaque mot radicalisé de la requête [20].

Mise en place technique D'un point de vue pratique, nous avons dû commencer par trouver un ensemble pertinent de requêtes symbolisant des symptômes de panne. En effet, par manque de temps et de contacts, nous n'avons pu mettre en œuvre la solution la plus logique, qui consistait à faire manipuler notre prototype d'interrogation de la base d'expérience par un ou plusieurs garagistes. Plutôt que de créer de toute pièce des symptômes (problème de partialité), nous avons alors préféré former les requêtes à partir des champs symptôme d'un sous-ensemble aléatoire des fiches (sémantiquement indexées ou pas) de la base d'expériences. Nous n'avons évidemment gardé que des symptômes pour lesquels

[18] `http://desktop.google.com/fr/`

[19] `http://www.copernic.com/fr/products/desktop-search/index.html`

[20] Pour plus de précisions sur la procédure de recherche de Lucene, se référer à [Hatcher et Gospodnetic, 2004b].

nous constations un nombre non nul de documents pertinents dans la sous-partie de la base d'expérience indexée où nous avions restreint les recherches (300 documents).

Pour chaque requête, le protocole d'évaluation s'est alors déroulé de la façon suivante sur notre prototype :

1. soumission de la requête au moteur de recherche sémantique,
2. vérification de l'interprétation sémantique de la requête et correction si nécessaire,
3. récupération de la liste ordonnée par pertinence décroissante des documents indexés

Nous avons systématiquement chronométré l'enchaînement de ces trois étapes dans le but d'évaluer le temps moyen nécessaire à la réalisation de la tâche. En effet, sur la plupart des moteurs de recherche classique, ce temps est très court, de l'ordre de la seconde. Intuitivement, le temps passé pour une recherche avec notre outil de recherche sera supérieur, puisqu'en plus de saisir sa requête, l'utilisateur doit vérifier son interprétation et la corriger si nécessaire. Plus le temps passé à ces tâches sera réduit, mieux l'outil sera perçu et accepté par ses usagers.

Au niveau de l'étape de vérification de l'interprétation de la requête, nous avons jugé de la qualité de la traduction en lui attribuant un qualificatif parmi cinq possibles (cf plus haut). Les documents pertinents que nous prenons en compte dans l'étape suivante sont issus du processus de recherche à partir de l'interprétation correcte, et non de la première interprétation du système. En effet, si nous ne modifiions pas la traduction pour la rendre acceptable, les résultats du processus d'interprétation influenceraient directement les performances du processus de recherche sémantique. Si ce phénomène semble logique, nous avons préféré décorréler les deux évaluations afin d'identifier avec plus de précision l'origine des lacunes potentielles de notre prototype.

Une fois obtenues les deux listes de documents pertinents pour notre outil et pour le moteur de recherche classique, nous les avons alors comparées à notre référence manuelle et avons ainsi obtenu les valeurs de rappel et de précision pour chacune des approches.

5.2.3.2 Résultats

Comme nous l'avons expliqué en 5.2.3.1, nous avons conduit l'évaluation de notre outil de RI sémantique en deux parties : nous avons d'abord estimé l'efficacité de nos heuristiques d'indexation des symptômes des requêtes pour nous concentrer ensuite sur l'étude comparée entre un moteur de recherche classique et notre outil des valeurs de rappel et de précision obtenues.

Interprétation de requête Le tableau 5.4 fait état des résultats du processus d'indexation sémantique de chacune des 10 requêtes soumises au système. On peut constater que les heuristiques de création de symptômes semblent efficaces, puisque dans 8 cas sur 10, l'interprétation sémantique de la requête est considérée comme bonne ou parfaite. Cette observation est à tempérer par deux remarques : tout d'abord, les requêtes sont issues de documents de même nature que ceux déjà indexés, ce qui implique que les vocabulaires employés dans les deux cas sont relativement homogènes. Dans cette optique, il serait intéressant de conduire

une étude sur le terrain pour évaluer le degré de compatibilité entre les lexiques de deux communautés légèrement différentes (ingénieurs experts du diagnostic électronique d'une part, garagistes d'autre part). Une deuxième limite à une mauvaise interprétation de la requête concerne les résultats du système : si notre système n'est pas capable d'interpréter le symptôme décrit par le garagiste (même après reformulation), il ne pourra lui retourner aucun document pertinent. Comme le suggère [Baziz, 2005], nous envisageons de coupler notre outil à un moteur classique afin de pouvoir traiter des cas de figure semblables.

N° Req	Requête originale	Qualité de l'interprétation sémantique initiale	Traduction pour la RI classique
1	calage moteur en décélération lors du retour au régime de ralenti	bonne	calage décélération ralenti
2	manque de puissance moteur avec allumage du voyant diagnostic, manque de brio au démarrage	moyenne	manque puissance allumage diagnostic démarrage
3	démarreur long lors du premier démarrage à froid	nulle	démarrage froid difficile
4	témoin diagnostic moteur, allumé fixe, moteur tournant	bonne	diagnostic moteur allumé tournant
5	témoin diag allumé en permanence avec message sur EMF : anomalie anti-pollution	parfaite	diag allumé permanence anomalie anti-pollution
6	La batterie se décharge en quelques heures	bonne	batterie décharge
7	Ventilation du chauffage hors service	bonne	ventilation chauffage hors service
8	Témoin diag s'allume par intermittence	parfaite	diag allumé intermittent
9	Le moteur fait des à-coups, cale au ralenti et le témoin d'injection s'allume	parfaite	à-coups cale ralenti injection allumé
10	le moteur fonctionne avec un mélange très riche par intermittence	parfaite	mélange riche intermittent

Tableau 5.4 — Données relatives aux requêtes soumises aux systèmes de RI

Au niveau du temps nécessaire entre la saisie d'une requête et l'obtention de résultats, nous obtenons une durée moyenne de l'ordre de 25 secondes. Dans tous les cas, la plupart du temps est consacrée à la vérification de l'interprétation et à la reformulation éventuelle. Une fois ces deux étapes terminées, l'opération de calcul des résultats est quasi-instantanée (tout au plus 1 seconde). Si les résultats semblent acceptables, nous pensons qu'il existe ici

encore un biais dans l'évaluation : nous connaissions la structure de la RTO et savions quels éléments pouvaient (ou pas) être interprétés par le système, de même que nous savions comment reformuler une requête afin qu'elle soit comprise par le système (pas de phénomène de "tâtonnement"). Après une étude sur le terrain, nous serons plus à même de juger de la pertinence de nous préoccuper des problématiques de reformulation de requête (avec les pistes évoquées en 4.2.1) et de navigation dans un modèle de connaissances (dans un but de sélection de concepts).

Interrogation classique VS sémantique Comme nous l'avons précédemment annoncé, nous nous sommes préoccupé des mesures de précision (notées P) et de rappel (notées R) pour chacune des approches comparées, et ce dans différentes conditions. En effet, plutôt que de fixer arbitrairement un critère qui, à partir du score d'un document ou de sa position de retour, le juge pertinent ou pas, nous avons mis en place les deux types possibles et avons traité séparément quelques critères de seuillage différents (notés "s" par la suite), ainsi qu'un critère fixant le nombre de documents à retourner. Pour chaque requête, nous avons alors pu observer les valeurs résumées dans le tableau 5.5.

N° Req	**RI classique**						**RI sémantique**							
	$s \geq 0.23$		$s \geq 0.29$		10^{ers} *docs*		$s \geq 0.93$		$s \geq 0.96$		$s \geq 0.98$		10^{ers} *docs*	
	R	P	R	P	R	P	R	P	R	P	R	P	R	P
1	1,00	0,25	0,00	0,00	1,00	0,20	1,00	0,03	1,00	0,04	1,00	1,00	1,00	0,20
2	0,68	0,32	0,68	0,42	0,42	0,73	1,00	0,59	0,95	1,00	0,95	1,00	0,84	1,00
3	1,00	0,71	0,59	0,63	0,59	0,63	0,88	0,38	0,88	0,38	0,65	1,00	0,65	1,00
4	0,77	0,45	0,39	0,41	0,16	0,45	1,00	0,82	0,03	1,00	0,03	1,00	1,00	0,82
5	0,75	0,60	0,50	0,67	1,00	0,40	0,75	1,00	0,75	1,00	0,50	1,00	1,00	0,29
6	0,60	1,00	0,40	1,00	1,00	0,83	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,38
7	0,20	1,00	0,20	1,00	1,00	0,38	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,36
8	1,00	0,40	1,00	0,43	1,00	0,75	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,43
9	0,75	0,40	0,63	0,63	0,75	0,40	1,00	0,26	1,00	0,62	0,88	1,00	1,00	0,57
10	0,23	0,71	0,23	0,83	0,23	0,24	0,23	0,56	0,23	1,00	0,23	1,00	0,23	0,31
Moy.	0,70	0,58	0,46	0,60	0,71	0,50	0,89	0,66	0,78	0,80	0,72	1,00	0,87	0,54
F-Mes.	0,64		0,52		0,59		0,76		0,79		0,84		0,66	

Tableau 5.5 — Résultats comparés des deux approches de RI

De façon synthétique, nous constatons que l'approche que nous avons adoptée obtient de meilleurs résultats à la fois en terme de rappel et de précision qu'une approche par RI classique : la meilleure valeur de rappel (respectivement de précision) pour la RI classique reste inférieure à la moins bonne valeur de rappel (resp. de précision, à l'exception de la seule prise en compte des 10 premiers documents) pour la RI sémantique. En moyenne, nous observons un gain de rappel et de précision de l'ordre de 20% avec notre solution. Avec les résultats obtenus, on peut remarquer le phénomène bien connu de compromis nécessaire entre rappel et précision : selon la valeur de seuil retenue, le rappel diminue lorsque la précision augmente (et vice versa). On notera toutefois que contrairement à l'approche par

RI classique, la solution de ne retenir pour la RI sémantique que les 10 premiers documents comme pertinents est la moins performante en termes de F-Mesure. Ce phénomène illustre que les scores calculés de proximité sémantique sont plus pertinents car ils correspondent à une notion de similarité plus intuitive (plus "sémantique") que celle qui se limite au lexique de la requête et des documents.

D'après les valeurs de précision et de F-mesure, les meilleurs résultats produits par notre solution correspondent à un seuil fixé à $0,98$. Ce seuil paraît très élevé et pourrait remettre en question le gain théoriquement apporté par l'appariement de symptômes similaires mais non identiques. Nous pensons toutefois que ce phénomène peut s'expliquer par la nature des documents indexés : nous avons pu constater qu'une grande partie des documents de la base d'expérience concernent la prestation motorisation. De ce fait, de nombreux documents obtiennent des scores élevés de similarité symptomatique vis-à-vis d'une requête à propos d'un problème sur le moteur, puisqu'ils ne sont distinguables que par la similarité des problèmes associés et de leurs contextes compatibles éventuels. Cette situation est sans doute amenée à s'améliorer lorsque nous prendrons en compte l'applicabilité (i.e. le type de véhicule concerné par la panne) dans le module de recherche : une requête donnée ne concernera que certains modèles de véhicule, ce qui limitera le nombre de documents potentiellement intéressants. Notre outil pourrait alors bénéficier d'un "bruit" moins important entre deux documents jugés pertinents, ce qui permettrait une bonne précision à des seuils inférieurs.

Si nous nous intéressons aux résultats locaux à chaque requête, nous pouvons distinguer deux requêtes particulièrement délicates à traiter, à savoir la première et la dernière. Les problèmes liés à la première requête proviennent du fait que l'annotateur spécialiste du domaine n'a jugé pertinents que 2 documents sur les 350, et ce malgré la présence de documents mentionnant simultanément des problèmes et des prestations proches de la requête. Après une discussion a posteriori avec celui-ci, nous avons pu constater que celui-ci a été fortement influencé par la présence simultanée des contextes de décélération et de ralenti (voir tab 5.4). Cette observation nous amène à repenser l'importance accordée initialement aux contextes d'une panne dans le calcul de similarité de symptômes : jusqu'à maintenant, nous avons considéré l'influence d'un contexte de façon indirecte, à travers la similarité de deux problèmes (cf 5.1.3.2) ; nous pensons désormais modéliser un contexte comme relatif au symptôme (au même titre qu'un problème ou une prestation) et non au problème (cf fig. 5.4), ce qui permettrait de le prendre en compte plus directement dans le calcul de proximité sémantique entre deux symptômes. D'un point de vue théorique, ce changement nous semble également justifié : un contexte d'apparition d'une panne est tout autant relié au problème caractéristique du symptôme qu'à la prestation affectée. Pour sa part, la requête n°10 entraîne systématiquement de mauvais résultats, que ce soit avec une approche de RI classique ou sémantique. Après analyse des documents pertinents sélectionnés par l'annotateur de référence, nous avons découvert que celui-ci s'était fondé sur ses propres connaissances du domaine pour rapprocher deux symptômes dans l'absolu non similaires, à savoir une surconsommation de carburant et l'observation de fumée noire à l'échappement. Cette situation illustre l'intérêt que nous aurions à représenter au sein de notre ontologie les relations de causalité entre symptômes. Ces connaissances expertes permettraient ainsi d'améliorer l'efficacité de notre outil de RI sémantique dans des situations semblables.

Nous devons toutefois émettre certaines réserves concernant la procédure d'évaluation suivie. Tout d'abord, il est difficile de tirer des conclusions définitives à partir des résultats obtenus car le nombre de requêtes utilisées pour analyser le comportement des deux moteurs de recherche est largement insuffisant. Une première perspective consisterait donc à étendre le nombre de paires $(requete, documents\_pertinents)$ disponibles : il nous faudrait envisager une phase manuelle de détermination de la pertinence de chaque document vis-à-vis de plusieurs nouvelles requêtes. Ensuite, nous avons pleinement conscience qu'il peut paraître complètement arbitraire de comparer deux moteurs de recherche que nous avons tous deux conçus. Comme ces deux moteurs sont fondés sur les mêmes techniques de TALN, la comparaison ne s'avère cependant pas inutile : elle nous permet d'évaluer la plus-value apportée au SRI uniquement à travers l'utilisation d'une RTO. Nous reconnaissons néanmoins qu'il serait intéressant de confronter notre moteur sémantique à un outil de recherche de la littérature pour lequel on disposerait de résultats tangibles dans le cadre de certains grands standards d'évaluation pour la RI (e.g. la campagne TREC).

Pour résumer les résultats (partiels) de cette seconde évaluation, notre contribution semble apporter, dans notre contexte applicatif, des gains non négligeables (en termes de rappel et de précision) face à une approche de RI n'exploitant pas de RTO. Malgré son caractère provisoire, l'évaluation nous a également permis de repérer une amélioration possible de la façon de représenter un symptôme, ainsi que l'intérêt de prendre en compte des connaissances plus riches que celles présentes jusqu'alors dans la RTO. Une fois ces modifications opérées, nous nous proposons d'effectuer, au sein d'un ou plusieurs garages, une étude de terrain qui permettra d'apprécier plus finement les qualités et limites de notre approche.

Conclusion et Perspectives

Avec cette partie, nous atteignons la fin de ce manuscrit. Nous récapitulons ici les motivations théoriques et pratiques qui nous ont poussé à étudier en parallèle des problématiques issues de l'IC et de la RI. Nous résumons ensuite les contributions scientifiques que nous avons pu apporter à ces deux domaines, ainsi que leur mise en pratique au sein d'un outil de maintenance de RTO et de RI sémantique destiné à l'industrie du diagnostic automobile. Nous restons pleinement conscient de la somme de travail restant à accomplir suite à nos recherches, tant au niveau des problèmes conceptuels soulevés (et non abordés faute de temps) que de l'évaluation de nos apports théoriques ou de la suite leur implémentation à travers l'extension TextViz. C'est pourquoi nous concluons ce manuscrit en évoquant les principales perspectives liées à nos travaux.

Synthèse

Motivations

D'un point de vue strictement chronologique, plusieurs raisons pratiques nous ont poussé à mener nos études à la croisée de deux domaines scientifiques. En effet, dans le cadre du projet MODE, nous étions initialement chargés de deux tâches, à savoir la réalisation d'un outil de RI dans un ensemble de fiches de réparation et la modélisation d'une ontologie capable de jouer le rôle d'intermédiaire cognitif entre plusieurs modules de diagnostic automobile. Comme nous l'avons expliqué en introduction, les modules de diagnostic, en parallèle de nos travaux sur le sous-projet OBIR, ont aussi constitué des champs d'investigation scientifique à part entière de la part d'autres membres du laboratoire commun Autodiag. Nous avons donc préféré concentrer d'abord nos efforts personnels sur la tâche de RI et attendre un état plus avancé de MODE pour nous préoccuper de l'interaction des modules via une ontologie adaptée. Les réflexions préliminaires sur l'ontologie en tant que moyen d'échange d'informations nous auront permis d'envisager dès le départ l'incorporation d'un modèle de connaissances au sein d'un processus de RI classique.

De fait, si les opérations inhérentes au processus de RI sémantique possèdent un coût humain et logiciel non négligeable, elles détiennent également des avantages indéniables : comme deux textes peuvent avoir un sens proche sans pour autant utiliser les mêmes termes, un outil couplant RI sémantique et classique possède a priori un meilleur rappel que ses homologues ne recourant à aucun modèle de connaissances. De même, une onto-

logie permet d'exploiter des connaissances extérieures (ou implicites) à un document pour déterminer s'il est pertinent vis-à-vis d'une requête (le contraire étant aussi vrai). De plus, si la structure choisie pour les annotations sémantiques est assez riche, l'emploi d'une ontologie en RI permet d'écarter certains documents utilisant les mêmes termes que la requête, mais avec un sens global différent. Par exemple, si un garagiste s'occupe d'un véhicule dont la climatisation manque de puissance avec des coupures intempestives du moteur, il ne sera pas intéressé par une fiche traitant d'une motorisation pas assez puissante et d'une climatisation qui se bloque par moments. Ce phénomène met en lumière que l'usage de RI sémantique peut également augmenter la précision d'un moteur de recherche.

Par conséquent, nos recherches se sont alors orientées vers une double problématique : l'intégration de connaissances dans un processus classique de RI, et l'optimisation des coûts de construction / maintenance d'une ontologie de tâche et/ou de domaine adaptée à la RI.

Apports théoriques et pratiques

Une des premières contributions que nous avons pu réaliser est liée à la construction d'une ontologie du diagnostic automobile (cf 3.1). En effet, en accord avec les travaux de [Aussenac-Gilles *et al.*, 2002] ou [Bourigault *et al.*, 2004], nous défendons la position selon laquelle une ontologie de tâche et/ou de domaine est nécessairement orientée dans sa modélisation par l'application qui l'utilise. A travers notre étude de cas, nous prolongeons les précédentes études de la littérature et nous insistons sur l'importance de prendre en compte les besoins applicatifs pendant la phase de construction de l'ontologie. En cela, nous privilégions sciemment le critère d'utilisabilité à celui de réutilisabilité, puisque nous considérons qu'une ontologie ne peut pas forcément être directement reprise sur un même domaine. En outre, nous prouvons, avec notre exemple étudié, que la terminologie associée peut aussi être impactée par les besoins applicatifs : l'importance de la partie terminologique serait bien moindre pour une RTO non destinée au repérage systématique d'entités ontologiques dans des textes (e.g. une RTO pour l'organisation de rubriques d'un site du Web Sémantique).

Concernant le processus de construction ontologique, nous avons également pu constater certaines carences au niveau des formats ontologiques actuels : malgré son importance potentielle (cf méthode Terminae [Aussenac-Gilles *et al.*, 2008]), la notion de terme n'est généralement pas représentée indépendemment du concept qu'il dénote. C'est pourquoi nous nous sommes employé à proposer un formalisme qui réifie les termes et représente de façon explicite leur lien avec les concepts (cf 3.2). Nous avons choisi de construire ce méta-modèle en OWL-DL pour plusieurs raisons : tout d'abord, dans le contexte du Web Sémantique et de la popularité grandissante des ontologies informatiques, nous avons jugé nécessaire de nous fonder sur un langage standard qui permette une meilleure interopérabilité entre systèmes d'IC ; ensuite, la bonne expressivité de OWL-DL place peu de contraintes sur l'ingénieur en charge de la modélisation, tout en lui garantissant de pouvoir conduire des inférences sur les connaissances représentées (respect de la calculabilité). Toutefois, du fait de l'absence de primitives de méta-modélisation, ce choix de langage nous a obligé à adapter l'implémentation de notre méta-modèle. De façon à dépasser les principales limites de notre approche (i.e. mélange de propriétés de niveaux épistémologiques différents, impossibilité de repré-

senter une occurrence de terme à portée universelle, lexique non associable à des relations sémantiques), nous avons proposé (sans l'implémenter) un second méta-modèle en OWL-Full, dont le seul inconvénient s'avère lié à l'absence de garantie en OWL-Full quant à la calculabilité d'un raisonnement.

Dans un contexte d'étude aussi technique et évolutif que le diagnostic automobile, nous devions aborder la problématique de maintenance de RTO : comment savoir quand et comment le modèle de connaissances doit être modifié ? Comment aider l'ontographe dans sa tâche de révision ? Dans l'hypothèse où la RTO est utilisée par une application de RI, nous avons mis en place un processus cyclique permettant la collaboration, par leur mise en parallèle, des étapes de maintenance et d'indexation sémantique (cf 4.1). L'ontographe définit au préalable un ensemble de critères destinés à juger de l'adéquation entre la RTO et un ensemble de documents caractéristiques de la tâche et/ou du domaine modélisés. La vérification automatique de ces critères au cours de la phase d'indexation sémantique permet de déterminer si la RTO nécessite d'être maintenue. La présentation à l'ontographe des documents violant certaines des contraintes prédéfinies l'oriente vers les modifications essentielles à opérer sur la RTO. Ce processus profite aussi à la phase d'indexation puisqu'une fois correctement mise à jour en fonction des critères de bonne adéquation, la RTO garantira des documents bien indexés. Si cette approche de la maintenance de RTO peut paraître séduisante, elle ne reste applicable que sous plusieurs conditions : en premier lieu, la RTO doit comporter une composante lexicale assez importante de façon à pouvoir retrouver la trace des entités ontologiques dans un ensemble de textes caractéristiques ; ensuite, le corpus de documents exploités par la phase d'indexation sémantique doit être assez homogène de façon à ce que des critères de bonne adéquation entre ceux-ci et la RTO puissent être dégagés par l'ontographe.

Enfin, du fait du peu de littérature sur le sujet, nous nous sommes penché sur la question du calcul de similarité entre deux réseaux d'instances de concepts (cf 4.2). Chacune des quelques approches que nous avons pu rencontrer, même si elle comportait des idées intéressantes, ne correspondait pas totalement à notre conceptualisation : nous souhaitions que la mesure de similarité sémantique puisse comparer des réseaux d'instances de taille différente, que les entités appariées ne soient pas forcément instances du même concept, et que le calcul tienne compte de toutes les informations disponibles sur chaque instance (i.e. son type, ses attributs et ses voisins dans le réseau). C'est pourquoi nous avons mené plusieurs réflexions sur le sujet. Nous avons notamment introduit certaines notions utiles comme la comparabilité de deux concepts selon leur position dans la taxonomie, les concepts centraux à une requête ou encore les relations obligatoires (relation avec une contrainte de cardinalité minimale sur tout ou partie de son domaine de définition). Celles-ci permettent d'obtenir des heuristiques d'appariement relativement simples entre les deux réseaux d'instances comparés. La seconde partie de notre contribution s'est focalisée sur la gestion du calcul de similarité en cas d'appariement partiel des nœuds des réseaux comparés. Plus particulièrement, nous nous sommes intéressé au cas où un sous-graphe extrait de la requête ne trouve pas d'appariement dans le réseau d'annotations du document comparé. De façon à garantir une certaine logique dans l'ordre final de retour des documents par rapport à la requête, nous introduisons, pour chaque calcul sollicitant un nœud "manquant" au document comparé, une similarité entre l'instance de requête non appariée et une instance artificielle.

Perspectives

Approfondissement des implémentations

Pour plusieurs raisons pratiques (fonctionnalités absentes de logiciels tiers, outil pensé en priorité dans un contexte applicatif spécifique, manque de temps), nous avons dû adapter plusieurs parties de nos contributions pour en obtenir une implémentation fonctionnelle. Nous revenons ici sur certains points que nous jugeons importants d'approfondir ou de rendre plus conformes à nos apports théoriques.

Pour l'outil de maintenance de RTO et d'indexation sémantique destiné à un ontographe, nous envisageons les modifications suivantes :

- **Amélioration de la généricité** Nous avons pleinement conscience qu'en l'état, l'outil ne peut être facilement réutilisé avec une ontologie différente de la nôtre : les critères de bonne adéquation, de même que les heuristiques de création automatique d'annotations sémantiques, ne sont pas modifiables dans l'interface car codés "en dur" dans TextViz. Nous avons donc l'intention de travailler à l'abstraction des méthodes concernées, de façon à disposer par la suite d'un outil qui puisse parfaitement s'adapter à la tâche et/ou au domaine modélisés, ainsi qu'aux besoins de l'ontographe. Incidemment, ceci nous permettra d'évaluer l'intérêt de nos contributions sur des domaines différents du diagnostic automobile.
- **Vérification automatique de la cohérence ontologique** Cette fonctionnalité existe déjà dans Protégé-OWL, qui permet de faire appel de façon transparente à un raisonneur logique. Toutefois, pour laisser le choix du raisonneur à l'utilisateur, Protégé se fonde sur DIG, un langage ontologique d'interface qui n'est pour l'instant pas aussi expressif que OWL-DL. La version 2 de DIG, censée remédier au problème, est attendue par la communauté du Web Sémantique depuis déjà quelques temps[21]. A terme, nous serons peut-être amené à mettre en place une solution alternative (e.g. une couche de dialogue entre TextViz et un raisonneur spécifique).
- **Garantie du passage à l'échelle** Protégé propose deux modes de fonctionnements pour la manipulation d'ontologie : soit celle-ci est chargée directement en mémoire sous forme d'objets Java, soit elle est stockée dans une base de données et une interface permet de récupérer les informations nécessaires "à la volée". Si la seconde approche paraît en théorie la plus séduisante, nous nous sommes heurté en pratique à des problèmes de manque de réactivité de l'outil : par exemple, pour récupérer une valeur d'attribut d'instance, nous avons mesuré des temps jusqu'à mille fois supérieurs dans le second mode de fonctionnement. Si la version 4 de Protégé-OWL ne résout pas ce problème, nous chercherons à mettre en place une solution ad hoc qui consisterait à stocker les instances de concept dans une base de données et n'en charger temporairement en mémoire qu'une partie, lorsque nécessaire.

Au niveau de l'outil de RI, nous souhaitons par la suite approfondir certains points :

- **Sélection guidée des symptômes à rechercher** Jusqu'à présent, l'interprétation d'une requête est visible grâce à un formulaire adapté à la structure d'un symptôme (cf fig. 5.15). En cas de mauvaise interprétation, l'utilisateur ne peut pas la modifier di-

[21] `http://dig.cs.manchester.ac.uk/roadmap.html`

rectement, il est obligé de reformuler sa requête jusqu'à obtenir une interprétation satisfaisante. De façon à éviter toute lassitude de l'utilisateur, nous avons l'intention de rajouter une fonctionnalité de parcours taxonomique et de sélection de concept pour chacune des sous-ontologies intervenant dans la définition d'un symptôme.

- **Réglage optimal des paramètres de la mesure de similarité** Plusieurs paramètres interviennent dans le calcul de la proximité de deux symptômes (cf 5.1.3.2) : choix de la mesure de similarité conceptuelle, de son importance relative par rapport à la similarité relationnelle, de l'importance relative des relations obligatoires ... Pour les tests effectués dans le cadre de notre thèse, nous avons fixé une valeur par défaut à ces paramètres et nous ne les avons pas fait varier. Il serait toutefois utile pour l'application de RI de déterminer expérimentalement la combinaison de valeurs de ces paramètres qui entraîne les meilleurs résultats en termes de rappel et/ou de précision.
- **Evaluation de l'acceptabilité de l'outil** Au cours des évaluations que nous avons menées, nous n'avons pas eu le temps de faire tester l'outil par un utilisateur final, à savoir un garagiste. Il serait intéressant d'observer si le logiciel est perçu comme assez simple d'usage et si la phase de correction de l'indexation des requêtes n'est pas jugée trop fastidieuse. Cette évaluation en conditions réelles nous permettrait de plus de connaître (et, si besoin est, de remédier à) la variabilité terminologique entre un garagiste et un expert chargé de la rédaction des fiches de réparation.

Prolongement des contributions théoriques

Nous concluons sur les perspectives à nos travaux en donnant quelques axes d'étude potentiellement intéressants :

- **Evolution ontologique et dynamicité des annotations sémantiques** Au cours de nos travaux sur la mise en place d'une synergie entre maintenance d'ontologie et indexation sémantique, nous avons pu constater la difficulté de gérer des annotations sémantiques existantes en fonction des modifications apportées à l'ontologie. Etant donné l'ampleur du problème, nous avons préféré ne pas l'approfondir dans le cadre de notre thèse et avons opté pour une solution à base de simples heuristiques. Avec l'importance actuelle du Web Sémantique et des annotations sémantiques, ce sujet mérite néanmoins d'être plus amplement abordé. Le projet Dynamo [22] auquel nous participons a justement pour objectif d'étudier ce sujet de recherche par la collaboration d'outils d'IC et de Systèmes Multi-Agents Adaptatifs [Ottens, 2007].
- **Gestion de phénomènes linguistiques** Comme nous l'avons expliqué, le méta-modèle de RTO que nous proposons permet de représenter explicitement les termes et leurs liens de dénotation avec les concepts. Il devient alors possible de modéliser précisément les caractéristiques d'un terme (label, catégorie syntaxique, morphologie, étymologie ...) ainsi que certains phénomènes comme l'anaphore ou la polysémie. Nous n'avons toutefois que peu exploité ces possibilités dans le cadre de notre application. Nous pensons qu'il serait intéressant de proposer notre méta-modèle à d'autres utilisateurs. En effet, des travaux comme [Aussenac-Gilles *et al.*, 2008] se fondent sur une représentation relativement riche du terme pour construire une RTO et ils pourraient

[22] `http://www.irit.fr/dynamo`

tirer profit du cadre plus formel que propose notre méta-modèle. Réciproquement, une collaboration de ce type nous permettrait d'appliquer notre formalisme dans une situation différente de celle décrite dans nos recherches, ce qui pourrait nous donner des pistes pour l'améliorer.

– **Exploitation du raisonnement ontologique en RI** Une dernière perspective que nous avons jusqu'alors peu abordée concerne l'utilisation de l'ontologie comme source de connaissance extérieure au document indexé sémantiquement. En effet, nous nous sommes limités dans nos recherches à l'indexation et la comparaison d'entités (ou de groupes d'entités) explicitement mentionnées dans les requêtes et les documents. Il nous paraît prometteur d'intégrer au sein de la RTO des connaissances de type causal (e.g. la présence d'un symptôme entraîne l'observation d'un symptôme supplémentaire) et de les prendre en compte dans le processus de RI.

Bibliographie

[NIS, 2003] (2003). *Guidelines for the construction, format, and management of monolingual thesauri*. National Information Standards Organization. ANSI/NISO Z39.19-2003.

[Alfonseca et Manandhar, 2002] ALFONSECA, E. et MANANDHAR, S. (2002). Extending a lexical ontology by a combination of distributional semantics signatures. *In EKAW '02 : Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 1–7, London, UK. Springer-Verlag.

[Amardeilh, 2007] AMARDEILH, F. (2007). *Web Sémantique et Information linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle*. Thèse de doctorat, Université Paris X - Nanterre.

[Aubin et Hamon, 2006] AUBIN, S. et HAMON, T. (2006). Improving term extraction with terminological resources. *CoRR*, abs/cs/0609019.

[Aussenac-Gilles, 1999] AUSSENAC-GILLES, N. (1999). Gediterm : un logiciel pour gérer des Bases de Connaissances Terminologiques . *Terminologies nouvelles (revue internationale francophone)*, 19:111–123.

[Aussenac-Gilles, 2005] AUSSENAC-GILLES, N. (2005). Méthodes ascendantes pour l'Ingénierie des Connaissances. Mémoire d'Habilitation à Diriger des Recherches de l'Université Paul Sabatier - Toulouse III.

[Aussenac-Gilles *et al.*, 2006] AUSSENAC-GILLES, N., BAZIZ, M. et HERNANDEZ, N. (2006). Ontologies pour la recherche d'information : importance de la dimension terminologique. *In* MUSTAPHA EL HADI, W., éditeur : *Terminologie et accès à l'information spécialisée*, Techniques et traités des sciences et techniques de l'information, chapitre -, pages 211–234. Hermès, http ://www.editions-hermes.fr/.

[Aussenac-Gilles *et al.*, 2003a] AUSSENAC-GILLES, N., BIÉBOW, B. et SZULMAN, S. (2003a). D'une méthode à un guide pratique de modélisation de connaissances à partir de textes. *In* ROUSSELOT, F., éditeur : *Actes des 5e rencontres Terminologie et IA (TIA 2003)*, pages 41–53.

[Aussenac-Gilles *et al.*, 2003b] AUSSENAC-GILLES, N., BIÉBOW, B. et SZULMAN, S. (2003b). Modélisation du domaine par une méthode fondée sur l'analyse de corpus. *In* TEULIER, R., CHARLET, J. et TCHOUNIKINE, P., éditeurs : *Ingénierie des Connaissances*, pages 49–71. L'Harmattan.

[Aussenac-Gilles et Condamines, 2001] AUSSENAC-GILLES, N. et CONDAMINES, A. (2001). *Entre textes et ontologies formelles : les bases de connaissances terminologiques*, pages 153–177. Hermes.

[Aussenac-Gilles et Condamines, 2004] AUSSENAC-GILLES, N. et CONDAMINES, A. (2004). Documents électroniques et constitution de ressources terminologiques ou ontologiques. *In Revue I3*, volume 4, pages 75–93.

[Aussenac-Gilles *et al.*, 2002] AUSSENAC-GILLES, N., CONDAMINES, A. et SZULMAN, S. (2002). Prise en compte de l'application dans la constitution de produits terminologiques. *In Actes des 2e Assises Nationales du GDR I3*, pages 289–302. Cépaduès Editions.

[Aussenac-Gilles et Matta, 1994] AUSSENAC-GILLES, N. et MATTA, N. (1994). Making a method of problem solving explicit with macao. *Int. J. Hum.-Comput. Stud.*, 40(2):193–219.

[Aussenac-Gilles et Sörgel, 2005] AUSSENAC-GILLES, N. et SÖRGEL, D. (2005). Text analysis for ontology and terminology engineering. *In Applied Ontology*, volume 1, pages 35–46. IOS Press.

[Aussenac-Gilles *et al.*, 2008] AUSSENAC-GILLES, N., SZULMAN, S. et DESPRES, S. (2008). *The* TERMINAE *Method and Platform for Ontology Engineering from Texts*, pages 199–223. IOS Press.

[Bachimont, 2000] BACHIMONT, B. (2000). *Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en Ingénierie des Connaissances*. Eyrolles.

[Bachimont, 2004] BACHIMONT, B. (2004). Arts et sciences du numérique : Ingénierie des connaissances et critique de la raison computationnelle. Mémoire d'habilitation à diriger des recherches en Informatique de l'Université de Technologie de Compiègne.

[Baeza-Yates et Ribeiro-Neto, 1999] BAEZA-YATES, R. et RIBEIRO-NETO, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Baget *et al.*, 2004] BAGET, J., CANAUD, E., EUZÉNAT, J. et SAÏD-HACID, M. (2004). Les langages du Web Sémantique. *Revue I3*, Hors Série 2004.

[Baranyi *et al.*, 1998] BARANYI, P., GEDEON, T. et KOCZY, L. (1998). Intelligent information retrieval using fuzzy approach. *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, 2:1984–1989 vol.2.

[Baziz, 2005] BAZIZ, M. (2005). *Indexation conceptuelle guidée par ontologie pour la recherche d'information*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France.

[Baziz *et al.*, 2005] BAZIZ, M., BOUGHANEM, M. et AUSSENAC-GILLES, N. (2005). A Conceptual Indexing Approach based on Document Content Representation . *In* CRESTANI, F. et RUTHVEN, I., éditeurs : *CoLIS5 : Fifth International Conference on Conceptions of Libraries and Information Science*, pages 171–186. Springer-Verlag.

[Bechhofer *et al.*, 2002] BECHHOFER, S., CARR, L., GOBLE, C. A., KAMPA, S. et MILES-BOARD, T. (2002). The semantics of semantic annotation. *In On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1152–1167, London, UK. Springer-Verlag.

[Belew, 2000] BELEW, R. K. (2000). *Finding out about : a cognitive perspective on search engine technology and the WWW*. Cambridge University Press, New York, NY, USA.

[Berners-Lee, 1999] BERNERS-LEE, T. (1999). *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco.

[Bisson, 1993] BISSON, G. (1993). *KBG : Induction de Bases de Connaissances en Logique des Prédicats*. Thèse de doctorat, Université Paris XI - Orsay.

[Bisson, 2000] BISSON, G. (2000). La similarité : une notion symbolique/numérique. *In* MOULET et BRITO, éditeurs : *Apprentissage symbolique-numérique*, volume 2, pages 169–201. Cépaduès.

[Bontas *et al.*, 2005] BONTAS, E. P., MOCHOL, M. et TOLKSDORF, R. (2005). Case studies on ontology reuse. *In I-KNOW '05 : Proceedings of the 5tg International Conference on Knowledge Management*, Graz,Austria. http ://userpage.fu-berlin.de/ paslaru/papers/iknow05.pdf.

[Bontcheva *et al.*, 2004] BONTCHEVA, K., TABLAN, V., MAYNARD, D. et CUNNINGHAM, H. (2004). Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10(3/4):349–373.

[Bourigault, 2002] BOURIGAULT, D. (2002). Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. *In Actes de la 9e conférence annuelle sur le traitement automatique des langues (TALN 2002)*, pages 75–84, Nancy.

[Bourigault *et al.*, 2004] BOURIGAULT, D., AUSSENAC-GILLES, N. et CHARLET, J. (2004). Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas. *In* PIERREL, J.-M. et SLODZIAN, M., éditeurs : *Techniques informatiques et structuration de terminologies*, volume 18/1 de *Revue d'Intelligence Artificielle*, pages 87–110. Hermes Sciences.

[Bourigault et Charlet, 1999] BOURIGAULT, D. et CHARLET, J. (1999). Construction d'un index thématique de l'ingénierie des connaissances. *In Actes des 10èmes Journées Francophones de l'Ingénierie des Connaissances*, pages 107–118, Palaiseau.

[Bourigault *et al.*, 2005] BOURIGAULT, D., FABRE, C., FRÉROT, C., JACQUES, M.-P. et OZDOWSKA, S. (2005). Syntex, analyseur syntaxique de corpus. *In Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles*, Dourdan, France. http ://www.irit.fr/LN_IRIT/bourigault_101105.pdf.

[Broglio *et al.*, 1993] BROGLIO, J., CALLAN, J. P. et CROFT, W. B. (1993). Inquery system overview. *In Proceedings of a workshop on held at Fredericksburg, Virginia*, pages 47–67, Morristown, NJ, USA. Association for Computational Linguistics.

[Bruaux, 2007] BRUAUX, S. (2007). *Vers la construction centrée-ontologie de modèles de résolution de problèmes : la méthode OntoKADS*. Thèse de doctorat, Université de Picardie Jules Verne.

[Bryan et Wright, 2005] BRYAN, M. et WRIGHT, R. (2005). How can ontologies help repair your car ? *In Journal of the International SGML/XML Users Group*. http ://www.idealliance.org/proceedings/xtech05/papers/02-07-02/.

[Buitelaar *et al.*, 2006a] BUITELAAR, P., DECLERCK, T., FRANK, A., RACIOPPA, S., KIESEL, M., SINTEK, M., ENGEL, R., ROMANELLI, M., SONNTAG, D., LOOS, B., MICELLI, V., PORZEL, R. et CIMIANO, P. (2006a). Linginfo : Design and applications of a model for the integration of linguistic information in ontologies. *In Proc. of the OntoLex workshop in LREC'06*. http ://www.dfki.de/ paulb/OntoLex2006.pdf.

[Buitelaar *et al.*, 2006b] BUITELAAR, P., SINTEK, M. et KIESEL, M. (2006b). A multilingual/multimedia lexicon model for ontologies. *In Proceedings of the 3rd European Semantic Web Conference (ESWC)*, pages 502–513.

[Calabretto et Prié, 2004] CALABRETTO, S. et PRIÉ, Y. (2004). Indexation/recherche dans les corpus temporalisés. *"Les temps du document numérique", rapport final de l'action spécifique Fant-AS-STIC*.

[Callan, 1994] CALLAN, J. P. (1994). Passage-level evidence in document retrieval. *In SIGIR '94 : Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310, New York, NY, USA. Springer-Verlag New York, Inc.

[Caussanel *et al.*, 2002] CAUSSANEL, J., CAHIER, J.-P., ZACKLAD, M. et CHARLET, J. (2002). Les topic maps sont-ils un bon candidat pour l'ingénierie du web sémantique. *In Actes de la conférence Ingénierie des Connaissances (IC '02)*, pages 3–14, Rouen, France.

[Charlet, 2002] CHARLET, J. (2002). L'Ingénierie des Connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales. Mémoire d'habilitation à diriger des recherches en Informatique de l'université de Pierre et Marie Curie.

[Chen *et al.*, 1998] CHEN, H., ZHANG, Y. et HOUSTON, A. L. (1998). Semantic indexing and searching using a hopfield net. *Journal of Information Science*, 24(1):3–18.

[Chittaro *et al.*, 1993] CHITTARO, L., GUIDA, G., TASSO, C. et TOPPANO, E. (1993). Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems : a case study in diagnosis. *In Proc. of the IEEE Transactions on Systems, Man and Cybernetics*, volume 23.

[Cimiano, 2006] CIMIANO, P. (2006). *Ontology Learning and Population from Text : Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Cimiano *et al.*, 2007] CIMIANO, P., HAASE, P. et HEIZMANN, J. (2007). Porting natural language interfaces between domains : an experimental user study with the orakel system. *In IUI '07 : Proceedings of the 12th international conference on Intelligent user interfaces*, pages 180–189, New York, NY, USA. ACM.

[Cimiano *et al.*, 2005] CIMIANO, P., LADWIG, G. et STAAB, S. (2005). Gimme' the context : context-driven automatic semantic annotation with C-PANKOW. *In* ELLIS, A. et HAGINO, T., éditeurs : *WWW*, pages 332–341. ACM.

[Ciravegna *et al.*, 2004] CIRAVEGNA, F., CHAPMAN, S., DINGLI, A. et WILKS, Y. (2004). Learning to harvest information for the semantic web. *In Proc. of the First European Semantic Web Symposium*, pages 312–326.

[Ciravegna *et al.*, 2002] CIRAVEGNA, F., DINGLI, A., WILKS, Y. et PETRELLI, D. (2002). Timely and non-intrusive active document annotation via adaptive information extraction. *In Proceedings of ECAI Workshop on Semantic Authoring, Annotation and Knowledge Markup*, Lyon, France.

[Cleverdon, 1962] CLEVERDON, C. W. (1962). Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. Rapport technique, ASLIB-Cranfield Research Report.

[Condamines, 2003] CONDAMINES, A. (2003). Sémantique et Corpus spécialisés : Constitution de bases de connaissances terminologiques. Mémoire d'habilitation à diriger des recherches en Linguistique de l'Université Toulouse Le Mirail.

[Corby *et al.*, 2000] CORBY, O., DIENG, R. et HÉBERT, C. (2000). A conceptual graph model for w3c resource description framework. *In ICCS*, pages 468–482.

[Corby *et al.*, 2004] CORBY, O., DIENG-KUNTZ, R. et FARON-ZUCKER, C. (2004). Querying the semantic web with corese search engine. *In* de MÁNTARAS, R. L. et SAITTA, L., éditeurs : *ECAI*, pages 705–709. IOS Press.

[Corby *et al.*, 2005] CORBY, O., DIENG-KUNTZ, R., FARON-ZUCKER, C. et GANDON, F. (2005). Ontology-based approximate query processing for searching the semantic web with corese. Inria research report rr-5621, INRIA.

[Corcho, 2006] CORCHO, O. (2006). Ontology based document annotation : trends and open research problems. *International Journal of Metadata, Semantics and Ontologies*, 1(1):47 – 57.

[d'Aquin *et al.*, 2007] D'AQUIN, M., BALDASSARRE, C., GRIDINOC, L., SABOU, M., ANGELETOU, S. et MOTTA, E. (2007). Watson : Supporting next generation semantic web applications. *In Proc. of IADIS International Conference WWW/Internet 2007*, Vila Real, Portugal.

[Després et Szulman, 2008] DESPRÉS, S. et SZULMAN, S. (2008). Sémantique et réutilisation d'ontologie générique. *In Actes des 8èmes journées francophones "Extraction et Gestion des Connaissances" (EGC'08)*, pages 121–126.

[Dieng-Kuntz et Corby, 2005] DIENG-KUNTZ, R. et CORBY, O. (2005). Conceptual graphs for semantic web applications. *In ICCS*, pages 19–50.

[Dill *et al.*, 2003] DILL, S., EIRON, N., GIBSON, D., GRUHL, D., GUHA, R. V., JHINGRAN, A., KANUNGO, T., RAJAGOPALAN, S., TOMKINS, A., TOMLIN, J. A. et ZIEN, J. Y. (2003). SemTag and Seeker : Bootstrapping the Semantic Web via Automated Semantic Annotation. *In Proc. of the Twelfth International World Wide Web Conference (WWW2003)*, pages 178–186. ACM Press.

[Drouin, 2003] DROUIN, P. (2003). Acquisition des termes simples fondée sur les pivots lexicaux spécialisés. *In Actes de cinquièmes rencontres Terminologie et intelligence artificielle (TIA 2003)*, pages 183–186.

[Egghe, 1999] EGGHE, L. (1999). On the law of Zipf-Mandelbrot for multi-word phrases. *J. Am. Soc. Inf. Sci.*, 50(3):233–241.

[Enguehard et Pantera, 1994] ENGUEHARD, C. et PANTERA, L. (1994). Automatic natural acquisition of a terminology. *Journal of Quantitative Linguistics*, 2(1):27–32.

[Erdmann *et al.*, 2000] ERDMANN, M., MAEDCHE, A., SCHNURR, H.-P. et STAAB, S. (2000). From manual to semi-automatic semantic annotation : About ontology-based text annotation tools. *In* BUITELAAR, P. et HASIDA, K., éditeurs : *Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*.

[Faatz et Steinmetz, 2002] FAATZ, A. et STEINMETZ, R. (2002). Ontology enrichment with texts from the www. *In Proceedings of the 2nd Semantic Web Mining Workshop at ECMLI/PKDD*, Helsinki, Finland.

[Fernandez *et al.*, 2006] FERNANDEZ, M., CANTADOR, I. et CASTELLS, P. (2006). Core : A tool for collaborative ontology reuse and evaluation. *In EON '06 : Proceedings of the 4th Int. Workshop on Evaluation of Ontologies for the Web, at the 15th Int. World Wide Web Conference (WWW'06)*. http ://km.aifb.uni-karlsruhe.de/ws/eon2006/eon2006fernandezetal.pdf.

[Fluit *et al.*, 2003] FLUIT, C., SABOU, M. et van HARMELEN, F. (2003). *Ontology-based Visualisation : towards Semantic Web applications*, chapitre 3, pages 45–58. Birkhäuser.

[Fourel *et al.*, 1998] FOUREL, F., MULHEM, P. et BRUANDET, M.-F. (1998). A generic framework for structured document access. *In DEXA '98 : Proceedings of the 9th International Conference on Database and Expert Systems Applications*, pages 521–530, London, UK. Springer-Verlag.

[Furnas *et al.*, 1988] FURNAS, G. W., DEERWESTER, S., DUMAIS, S. T., LANDAUER, T. K., HARSHMAN, R. A., STREETER, L. A. et LOCHBAUM, K. E. (1988). Information retrieval using a singular value decomposition model of latent semantic structure. *In SIGIR '88 : Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 465–480, New York, NY, USA. ACM.

[Gaines et Shaw, 1980] GAINES, B. R. et SHAW, M. L. G. (1980). New directions in the analysis and interactive elicitation of personal construct systems. *International Journal of Man-Machine Studies*, 13(1):81–116.

[Gangemi *et al.*, 2002] GANGEMI, A., GUARINO, N., MASOLO, C., OLTRAMARI, A. et SCHNEIDER, L. (2002). Sweetening ontologies with dolce. *In EKAW '02 : Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management : Ontologies and the Semantic Web*, pages 166–181, London, UK. Springer-Verlag.

[Golebiowska, 2002] GOLEBIOWSKA, J. (2002). *Exploitation des ontologies pour la mémoire d'un projet véhicule*. Thèse de doctorat, Université de Nice-Sophia Antipolis.

[Golub et van Loan, 1983] GOLUB, G. H. et van LOAN, C. F. (1983). *Matrix Computations*. North Oxford Academic Publishing.

[Gomez-Pérez *et al.*, 2004] GOMEZ-PÉREZ, A., LOPEZ, M. F. et CORCHO, O. (2004). *Ontological Engineering : with examples from the area of Knowledge Management, e-Commerce and the Semantic Web*. Springer.

[Gros et Assadi, 1997] GROS, C. et ASSADI, H. (1997). Intégration de connaissances dans un système de consultation de documentation technique. *In Actes des 1eres Rencontres du Chapitre Français de l'ISKO*. Presses Universitaires du Septentrion.

[Gruber, 1993] GRUBER, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220.

[Gruber et Olsen, 1994] GRUBER, T. R. et OLSEN, G. R. (1994). An ontology for engineering mathematics. *In KR*, pages 258–269.

[Guarino, 1997] GUARINO, N. (1997). Understanding, Building and Using Ontologies. A commentary to "Using Explicit Ontologies in KBS Development". *International Journal of Human and Computer Studies*, 46:293–310.

[Guarino, 1998] GUARINO, N., éditeur (1998). *1st International Conference on Formal Ontology in Information Systems (FOIS '98)*, Trento, Italy. IOS Press.

[Guarino *et al.*, 1999] GUARINO, N., MASOLO, C. et VETERE, G. (1999). Ontoseek : Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80.

[Guarino et Welty, 2004] GUARINO, N. et WELTY, C. (2004). An overview of ontoclean. *In* STAAB, S. et STUDER, R., éditeurs : *Handbook on Ontologies*, pages 151–159. Springer, New York.

[Guha *et al.*, 2003] GUHA, R., MCCOOL, R. et MILLER, E. (2003). Semantic search. *In WWW '03 : Proceedings of the 12th international conference on World Wide Web*, pages 700–709, New York, NY, USA. ACM Press.

[Handschuh *et al.*, 2002] HANDSCHUH, S., STAAB, S. et CIRAVEGNA, F. (2002). S-cream - semi-automatic creation of metadata. *In* GÓMEZ-PÉREZ, A. et BENJAMINS, V. R., éditeurs : *EKAW*, volume 2473 de *Lecture Notes in Computer Science*, pages 358–372. Springer.

[Harris, 1968] HARRIS, Z. S. (1968). *Mathematical Structure of Language*. John Wiley and Sons.

[Hasling *et al.*, 1984] HASLING, D. W., CLANCEY, W. J. et RENNELS, G. (1984). Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies*, 20(1):3–19.

[Hatcher et Gospodnetic, 2004a] HATCHER, E. et GOSPODNETIC, O. (2004a). *Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA.

[Hatcher et Gospodnetic, 2004b] HATCHER, E. et GOSPODNETIC, O. (2004b). *Lucene in Action (In Action series)*, chapitre 3, pages 69–101. Manning Publications Co.

[Hayes-Roth et Jacobstein, 1994] HAYES-ROTH, F. et JACOBSTEIN, N. (1994). The state of knowledge-based systems. *Communications of the ACM*, 37(3):26–39.

[Hearst, 1992] HEARST, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. *In Proc. of the Fourteenth Conference on Computational Linguistics, Nantes, France*, pages 539–545, Morristown, NJ, USA. Association for Computational Linguistics.

[Hernandez, 2005] HERNANDEZ, N. (2005). *Ontologies de domaine pour la modélisation du contexte en Recherche d'information*. Thèse de doctorat, Université Paul Sabatier - Toulouse III.

[Hofmann, 1999] HOFMANN, T. (1999). Probabilistic latent semantic indexing. *In SIGIR '99 : Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA. ACM.

[Jiang et Conrath, 1997] JIANG, J. J. et CONRATH, D. W. (1997). Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. *In Proceedings of the International Conference on Research in Computational Linguistics (ROCLING)*. Taiwan.

[Kalyanpur *et al.*, 2005] KALYANPUR, A., HENDLER, J., PARSIA, B. et GOLBECK, J. (2005). Smore – semantic markup, ontology, and rdf editor. Available at : http ://www.mindswap.org/papers/SMORE.pdf.

[Kassel, 1999] KASSEL, G. (1999). PHYSICIAN is a role played by an object, whereas SIGN is a role played by a concept. *In Proc. of the IJCAI'99 Workshop on Ontologies and Problem-Solving Methods : Lessons Learned and Future Trends*, pages 61–69.

[Kassel, 2005] KASSEL, G. (2005). Integration of the DOLCE top-level ontology into the OntoSpec methodology. *CoRR*, abs/cs/0510050.

[Kavalec *et al.*, 2004] KAVALEC, M., MAEDCHE, A. et SVÁTEK, V. (2004). Discovery of lexical entries for non-taxonomic relations in ontology learning. *In* van EMDE BOAS, P., POKORNÝ, J., BIELIKOVÁ, M. et STULLER, J., éditeurs : *SOFSEM*, volume 2932 de *Lecture Notes in Computer Science*, pages 249–256. Springer.

[Kitamura et Mizoguchi, 2004] KITAMURA, Y. et MIZOGUCHI, R. (2004). Ontology-based functional-knowledge modeling methodology and its deployment. *In* MOTTA, E., SHADBOLT, N., STUTT, A. et GIBBINS, N., éditeurs : *EKAW*, volume 3257 de *Lecture Notes in Computer Science*, pages 99–115. Springer.

[Klinker *et al.*, 1991] KLINKER, G., BHOLA, C., DALLEMAGNE, G., MARQUES, D. et MCDERMOTT, J. (1991). Usable and reusable programming constructs. *Knowledge Acquisition*, 3(2):117–135.

[Knublauch *et al.*, 2004] KNUBLAUCH, H., FERGERSON, R. W., NOY, N. F. et MUSEN, M. A. (2004). The protégé owl plugin : An open development environment for semantic web applications. *In* MCILRAITH, S. A., PLEXOUSAKIS, D. et van HARMELEN, F., éditeurs : *International Semantic Web Conference*, volume 3298 de *Lecture Notes in Computer Science*, pages 229–243. Springer.

[Kozaki *et al.*, 2007] KOZAKI, K., SUNAGAWA, E., KITAMURA, Y. et MIZOGUCHI, R. (2007). Role Representation Model Using OWL and SWRL. *In Proc. of 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies of ECOOP*, Berlin.

[Lassila et McGuinness, 2001] LASSILA, O. et MCGUINNESS, D. L. (2001). The Role of Frame-based Representation on the Semantic Web. Rapport technique, Knowledge Systems Laboratory, Stanford, California.

[Lei *et al.*, 2006] LEI, Y., UREN, V. S. et MOTTA, E. (2006). SemSearch : A Search Engine for the Semantic Web. *In* STAAB, S. et SVÁTEK, V., éditeurs : *EKAW'06*, volume 4248 de *Lecture Notes in Computer Science*, pages 238–245, Podebrady, Czech Republic. Springer.

[Lewis, 1998] LEWIS, D. D. (1998). Naive (bayes) at forty : The independence assumption in information retrieval. *In ECML '98 : Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, London, UK. Springer-Verlag.

[Liang *et al.*, 2006] LIANG, A. C., LAUSER, B., SINI, M., KEIZER, J. et KATZ, S. (2006). From AGROVOC to the Agricultural Ontology Service / Concept Server : An OWL model for managing ontologies in the agricultural domain. *In Proc. of OWL : Experiences and Directions Workshop (OWLED)*.

[Lin, 1998] LIN, D. (1998). An information-theoretic definition of similarity. *In ICML '98 : Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Linster, 1992] LINSTER, M. (1992). *Knowledge acquisition based on explicit methods of problem–solving*. Thèse de doctorat, University of Kaiserslautern.

[Luhn, 1958] LUHN, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165.

[Luong, 2007] LUONG, P. H. (2007). *Gestion de l'évolution d'un Web Sémantique d'entreprise*. Thèse de doctorat, Ecole des Mines de Paris.

[Maedche, 2002] MAEDCHE, A. (2002). *Ontology learning for the Semantic Web*. Kluwer Academic Publisher.

[Mandala *et al.*, 1998] MANDALA, R., TAKENOBU, T. et HOZUMI, T. (1998). The use of wordnet in information retrieval. *In Procs of COLING/ACL-98 Workshop "Usage of WordNet in Natural Language Processing Systems"*.

[Marcus et McDermott, 1989] MARCUS, S. et MCDERMOTT, J. (1989). Salt : a knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence*, 39(1):1–37.

[Marshall, 1998] MARSHALL, C. C. (1998). Toward an ecology of hypertext annotation. *In HYPERTEXT '98 : Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 40–49, New York, NY, USA. ACM.

[Maynard, 2005] MAYNARD, D. (2005). Benchmarking ontology-based annotation tools for the semantic web. *In UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology"*, Nottingham, UK.

[Maynard *et al.*, 2007] MAYNARD, D., PETERS, W., D'AQUIN, M. et SABOU, M. (2007). Change management for metadata evolution. *In Proc. of International Workshop on Ontology Dynamics in ESWC '07*, pages 27–40.

[Maynard *et al.*, 2005] MAYNARD, D., YANKOVA, M., KOURAKIS, A. et KOKOSSIS, A. (2005). Ontology-based information extraction for market monitoring and technology watch. *In Proc. of ESWC Workshop "End User Apects of the Semantic Web"*. http ://gate.ac.uk/sale/eswc05/htechsight.pdf.

[Mazuel et Sabouret, 2007] MAZUEL, L. et SABOURET, N. (2007). Degré de relation sémantique dans une ontologie pour la commande en langue naturelle. *In 18es journées francophones d'ingénierie des connaissances*, pages 73–83, Grenoble, France.

[Mazuel et Sabouret, 2008] MAZUEL, L. et SABOURET, N. (2008). Protocole d'évaluation d'une mesure de degré de relation sémantique. *In Atelier sur les mesures sémantiques de la conférence EGC*, Toulouse, France. Cépaduès Editions.

[Mekki et Nazarenko, 2001] MEKKI, T. A. E. et NAZARENKO, A. (2001). Quel index pour le document électronique ? *In* MOJAHID, M. et VIRBEL, J., éditeurs : *Actes du Colloque International sur le Document Electronique (CIDE'01)*, pages 147–161, Toulouse, France. Europia.

[Meyer *et al.*, 1992] MEYER, I., SKUCE, D., BOWKER, L. et ECK, K. (1992). Towards a new generation of terminological resources : an experiment in building a terminological knowledge base. *In Proc. of 13th International Conference on Computational Linguistics*, pages 956–960.

[Morin, 1999] MORIN, E. (1999). Acquisition de patrons lexico-syntaxiques caractéristiques d'une relation sémantique. *In Revue Traitement Automatique des Langues (TAL)*, volume 40, pages 143–166.

[Mothe, 2000] MOTHE, J. (2000). Recherche et exploration d'informations - découverte de connaissances pour l'accès à l'information. Mémoire d'habilitation à diriger des recherches en Informatique de l'université Paul Sabatier de Toulouse.

[Mothe *et al.*, 2003] MOTHE, J., CHRISMENT, C., DOUSSET, B. et ALAUX, J. (2003). Doccube : multi-dimensional visualisation and exploration of large document sets. *J. Am. Soc. Inf. Sci. Technol.*, 54(7):650–659.

[Motik, 2005] MOTIK, B. (2005). On the properties of metamodeling in owl. *In* GIL, Y., MOTTA, E., BENJAMINS, V. R. et MUSEN, M. A., éditeurs : *International Semantic Web Conference*, volume 3729 de *Lecture Notes in Computer Science*, pages 548–562. Springer.

[Mutton et Golbeck, 2003] MUTTON, P. et GOLBECK, J. (16-18 July 2003). Visualization of semantic metadata and ontologies. *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 300–305.

[Niles et Pease, 2001] NILES, I. et PEASE, A. (2001). Towards a standard upper ontology. *In FOIS '01 : Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA. ACM.

[Noy et Klein, 2004] NOY, N. F. et KLEIN, M. (2004). Ontology evolution : Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440.

[Ogden et Richards, 1923] OGDEN, C. K. et RICHARDS, I. A. (1923). The meaning of meaning : A study of the influence of language upon thought and of the science of symbolism. *8th ed. 1923. Reprint New York : Harcourt Brace Jovanovich*.

[Ottens, 2007] OTTENS, K. (2007). *Un système multi-agent adaptatif pour la construction d'ontologies à partir de textes*. Thèse de doctorat, Université Paul Sabatier.

[Pan et Horrocks, 2006] PAN, J. Z. et HORROCKS, I. (2006). Owl fa : a metamodeling extension of owl dl. *In* CARR, L., ROURE, D. D., IYENGAR, A., GOBLE, C. A. et DAHLIN, M., éditeurs : *WWW*, pages 1065–1066. ACM.

[Pan *et al.*, 2006] PAN, J. Z., SERAFINI, L. et ZHAO, Y. (2006). Semantic import : An approach for partial ontology reuse. *In Proceedings of the ISWC 2006 Workshop on Modular Ontologies*.

[Pinto et Martins, 2001] PINTO, H. S. et MARTINS, J. P. (2001). A methodology for ontology integration. *In K-CAP '01 : Proceedings of the 1st international conference on Knowledge capture*, pages 131–138, New York, NY, USA. ACM.

[Popov *et al.*, 2003] POPOV, B., KIRYAKOV, A., KIRILOV, A., MANOV, D., OGNYANOFF, D. et GORANOV, M. (2003). KIM - Semantic Annotation Platform. *In* FENSEL, D., SYCARA, K. P. et MYLOPOULOS, J., éditeurs : *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 de *Lecture Notes in Computer Science*, pages 834–849. Springer.

[Porter, 1980] PORTER, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

[Prié, 2000] PRIÉ, Y. (2000). Sur la piste de l'indexation conceptuelle de documents. une approche par l'annotation. *Document Numérique*, 4(162):11–35.

[Prié et Garlatti, 2004] PRIÉ, Y. et GARLATTI, S. (2004). Méta-données et annotations dans le web sémantique. *Revue I3*, Hors-série.

[Prud'hommeaux et Seaborne, 2008] PRUD'HOMMEAUX, E. et SEABORNE, A. (2008). *SPARQL Query Language for RDF*. W3C. http ://www.w3.org/TR/rdf-sparql-query/.

[Rada *et al.*, 1989] RADA, R., MILI, H., BICKNELL, E. et BLETTNER, M. (Jan/Feb 1989). Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30.

[Rastier, 1995] RASTIER, F. (1995). Le terme : entre ontologie et linguistique. *In La Banque des Mots*. Conseil international de la langue française.

[Resnik, 1995] RESNIK, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *In IJCAI*, pages 448–453.

[Reymonet *et al.*, 2006] REYMONET, A., AUSSENAC-GILLES, N. et THOMAS, J. (2006). Tâche, domaine et application : influences sur le processus de modélisation de connaissances. *In* LEWKOWICZ, M., éditeur : *Actes des 17e journées francophones d'Ingénierie des Connaissances*, pages 71–80, Nantes, France. Université de Nantes.

[Reymonet *et al.*, 2007a] REYMONET, A., THOMAS, J. et AUSSENAC-GILLES, N. (2007a). Modélisation de Ressources Termino-Ontologiques en OWL. *In* TRICHET, F., éditeur : *Actes des 18e journées francophones d'ingénierie des connaissances*, pages 169–180, Grenoble, France. Cépaduès Editions. Prix AFIA meilleur article de la conférence.

[Reymonet *et al.*, 2007b] REYMONET, A., THOMAS, J. et AUSSENAC-GILLES, N. (2007b). Modelling Ontological and Terminological Resources in OWL DL. *In Proceedings of ISWC '07 workshop "From Text to Knowledge : The Lexicon/Ontology Interface" (OntoLex '07)*.

[Reynaud *et al.*, 1997] REYNAUD, C., AUSSENAC-GILLES, N., TCHOUNIKINE, P. et TRICHET, F. (1997). The notion of role in conceptual modeling. *In EKAW '97 : Proceedings of the 10th European Workshop on Knowledge Acquisition, Modeling and Management*, pages 221–236, London, UK. Springer-Verlag.

[Reynaud *et al.*, 2002] REYNAUD, C., LAUBLET, P. et CHARLET, J. (2002). Sur quelques aspects du web sémantique. *In Actes des deuxièmes Assises Nationales du GdR I3*, pages 59–78.

[Reynaud et Tort, 1997] REYNAUD, C. et TORT, F. (1997). Using explicit ontologies to create problem solving methods. *Int. J. Hum.-Comput. Stud.*, 46(2-3):339–364.

[Richardson et Smeaton, 1995] RICHARDSON, R. et SMEATON, A. (1995). Using wordnet in a knowledge-based approach to information retrieval. *In Proceedings of the BCS-IRSG Colloquium*.

[Robertson, 1977] ROBERTSON, S. E. (1977). The probability ranking principle in ir. *Readings in information retrieval*, pages 281–286.

[Robertson *et al.*, 1981] ROBERTSON, S. E., van RIJSBERGEN, C. J. et PORTER, M. F. (1981). Probabilistic models of indexing and searching. *In SIGIR '80 : Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56, Kent, UK, UK. Butterworth & Co.

[Robertson et Walker, 1997] ROBERTSON, S. E. et WALKER, S. (1997). On relevance weights with little relevance information. *In SIGIR '97 : Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 16–24, New York, NY, USA. ACM.

[Rocha *et al.*, 2004] ROCHA, C., SCHWABE, D. et de ARAGÃO, M. P. (2004). A hybrid approach for searching in the semantic web. *In Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 374–383.

[Rodriguez, 2000] RODRIGUEZ, M. A. (2000). *Assessing Semantic Similarity Among Spatial Entity Classes*. Thèse de doctorat, University of Maine, Orono, Maine 04469.

[Romary, 2001] ROMARY, L. (2001). An abstract model for the representation of multilingual terminological data : TMF - Terminological Markup Framework. *In Proc. of TAMA*.

[Rousset et Reynaud, 2004] ROUSSET, M.-C. et REYNAUD, C. (2004). Knowledge representation for information integration. *Information Systems*, 29(1):3–22.

[Roussey *et al.*, 2006] ROUSSEY, C., CALABRETTO, S., HARRATHI, F. et GAMMOUDI, M. M. (2006). Multilingual indexing based on ontologies. *In* GHODOUS, P., DIENG-KUNTZ, R. et LOUREIRO, G., éditeurs : *ISPE CE*, pages 418–425. IOS Press.

[Roux et Laublet, 1995] ROUX, B. L. et LAUBLET, P. (1995). Steps towards a unified approach of knowledge modelling. *In Proc. of the European Japanese Conference on Information Modeling and Knowledge Bases*. IOS Press.

[Royo *et al.*, 2005] ROYO, J. A., MENA, E., BERNAD, J. et ILLARRAMENDI, A. (2005). Searching the web : From keywords to semantic queries. *In ICITA '05 : Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2*, pages 244–249, Washington, DC, USA. IEEE Computer Society.

[Salton et Buckley, 1988] SALTON, G. et BUCKLEY, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

[Salton *et al.*, 1982] SALTON, G., FOX, E. A. et WU, H. (1982). Extended boolean information retrieval. Rapport technique, Ithaca, NY, USA.

[Salton et Lesk, 1965] SALTON, G. et LESK, M. E. (1965). The smart automatic document retrieval systems - an illustration. *Communications of the ACM*, 8(6):391–398.

[Salton *et al.*, 1975] SALTON, G., WONG, A. et YANG, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

[Sanderson, 2000] SANDERSON, M. (2000). Retrieving with good sense. *Inf. Retr.*, 2(1):49–69.

[Sanderson et Croft, 1999] SANDERSON, M. et CROFT, W. B. (1999). Deriving concept hierarchies from text. *In Research and Development in Information Retrieval*, pages 206–213.

[Schreiber *et al.*, 1993] SCHREIBER, A., WIELINGA, B. et BREUKER, J. (1993). *KADS : a Principled Approach to Knowledge Engineering*. Academic Press, London.

[Schreiber et de Hoog, 1999] SCHREIBER, G. et de HOOG, R. (1999). *Knowledge Engineering and Management : The CommonKADS Methodology*. MIT Press.

[Séguéla, 2001] SÉGUÉLA, P. (2001). *Construction de modèles de connaissance par analyse linguistiquee de relations lexicales dans les documents techniques*. Thèse de doctorat, Université Paul Sabatier.

[Simon *et al.*, 2003] SIMON, L., DESMONTILS, E. et JACQUIN, C. (2003). Utilisation de techniques d'enrichissement d'ontologie pour améliorer le processus d'indexation structurée. *In Actes des journées francophones d'Ingénierie des Connaissances (IC'2003)*, pages 145–160.

[Singhal *et al.*, 1995] SINGHAL, A., BUCKLEY, C., MITRA, M. et SALTON, G. (1995). Pivoted document length normalization. Rapport technique, Ithaca, NY, USA.

[Skuce, 1993] SKUCE, D. R. (1993). A system for managing knowledge and terminology for technical documentation. *In Terminology and Knowledge Engineering*, pages 428–441.

[Slodzian, 2000] SLODZIAN, M. (2000). L'émergence d'une terminologie textuelle et le retour du sens. *In* BÉJOINT, H. et THOIRON, P., éditeurs : *Le sens en terminologie*. Presses Universitaires de Lyon.

[Sowa, 1984] SOWA, J. F. (1984). *Conceptual structures : information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Sowa, 2000] SOWA, J. F. (2000). *Knowledge representation : logical, philosophical and computational foundations*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA.

[Specia et Motta, 2007] SPECIA, L. et MOTTA, E. (2007). Integrating folksonomies with the semantic web. *In Proc. of ESWC '07*.

[Spärck-Jones, 1972] SPÄRCK-JONES, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

[Stojanovic *et al.*, 2002] STOJANOVIC, L., MAEDCHE, A., MOTIK, B. et STOJANOVIC, N. (2002). User-driven ontology evolution management. *In EKAW '02 : Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 285–300, London, UK. Springer-Verlag.

[Stollberg *et al.*, 2004] STOLLBERG, M., ZHDANOVA, A. et FENSEL, D. (2004). h-TechSight – A Next Generation Knowledge Management Platform. *Journal of Information and Knowledge Management*, 3 (1):1–22.

[Studer *et al.*, 1998] STUDER, R., BENJAMINS, V. R. et FENSEL, D. (1998). Knowledge engineering : Principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197.

[Sunagawa *et al.*, 2005] SUNAGAWA, E., KOZAKI, K., KITAMURA, Y. et MIZOGUCHI, R. (2005). A Framework for Organizing Role Concepts in Ontology Development Tool : Hozo. *AAAI Fall Symposium Technical Report FS-05-08*, pages 136–143.

[Szulman et Biébow, 2004] SZULMAN, S. et BIÉBOW, B. (2004). OWL et Terminae. *In Actes des 15es journées francophones d'ingénierie des connaissances*, pages 41–52. Presses Universitaires de Grenoble.

[Szulman *et al.*, 2002] SZULMAN, S., BIÉBOW, B. et AUSSENAC-GILLES, N. (2002). Structuration de Terminologie à l'aide d'outils de TAL avec TERMINAE. *In Traitement Automatique des Langues*, volume 43, pages 103–128.

[Thieu *et al.*, 2004] THIEU, M., STEICHEN, O., ZAPLETAL, E. et JAULENT, M.-C. (2004). Mesures de similarité pour l'aide au consensus en anatomie pathologique. *In 15es journées francophones d'ingénierie des connaissances*, pages 225–236.

[Tran *et al.*, 2007] TRAN, T., CIMIANO, P., RUDOLPH, S. et STUDER, R. (2007). Ontology-based interpretation of keywords for semantic search. *In ISWC/ASWC*, pages 523–536.

[Turhan *et al.*, 2006] TURHAN, A.-Y., BECHHOFER, S., KAPLUNOVA, A., LIEBIG, T., LUTHER, M., MÖLLER, R., NOPPENS, O., PATEL-SCHNEIDER, P., SUNTISRIVARAPORN, B. et WEITHÖNER, T. (2006). Dig 2.0 : Towards a flexible interface for description logic reasoners. *In Proc. of the International Workshop on OWL : Experiences and Directions (OWLED'06)*.

[Turtle et Croft, 1991] TURTLE, H. et CROFT, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9(3):187–222.

[Uren *et al.*, 2006] UREN, V., CIMIANO, P., IRIA, J., HANDSCHUH, S., VARGAS-VERA, M., MOTTA, E. et CIRAVEGNA, F. (2006). Semantic annotation for knowledge management : Requirements and a survey of the state of the art. *Journal of Web Semantics : Science, Services and Agents on the World Wide Web*, 4:14–28.

[Uschold *et al.*, 1998] USCHOLD, M., HEALY, M., WILLIAMSON, K., CLARK, P. et WOODS, S. (1998). Ontology reuse and application. *In Proc. of the International Conference on Formal Ontology in Information Systems - FOIS'98*, pages 179–192. IOS Press.

[Vallet *et al.*, 2005] VALLET, D., FERNÁNDEZ, M. et CASTELLS, P. (2005). An ontology-based information retrieval model. *In* GÓMEZ-PÉREZ, A. et EUZENAT, J., éditeurs : *ESWC*, volume 3532 de *Lecture Notes in Computer Science*, pages 455–470. Springer.

[van Heijst *et al.*, 1997] van HEIJST, G., SCHREIBER, A. T. et WIELINGA, B. J. (1997). Roles are not classes : a reply to Nicola Guarino. *Int. J. Hum.-Comput. Stud.*, 46(2):311–318.

[van Rijsbergen, 1979] van RIJSBERGEN, C. J. (1979). *Information retrieval*. Butterworths, London, 2 édition.

[Velardi *et al.*, 2001] VELARDI, P., FABRIANI, P. et MISSIKOFF, M. (2001). Using text processing techniques to automatically enrich a domain ontology. *In Proc. of the international conference on Formal Ontology in Information Systems*, pages 270–284, New York, NY, USA. ACM Press.

[Vogel, 1988] VOGEL, C. (1988). *Génie cognitif*. Masson, Paris.

[Volz *et al.*, 2003] VOLZ, R., STUDER, R., MAEDCHE, A. et LAUSER, B. (2003). Pruning-based identification of domain ontologies. *J. UCS*, 9(6):520–529.

[Vrandecic *et al.*, 2006] VRANDECIC, D., VÖLKER, J., HAASE, P., TRAN, D. T. et CIMIANO, P. (2006). A metamodel for annotations of ontology elements in owl dl. *In* SURE, Y., BROCKMANS, S. et JUNG, J., éditeurs : *Proceedings of the 2nd Workshop on Ontologies and Meta-Modeling*, Karlsruhe, Germany.

[W3C, 2004a] W3C (2004a). RDF Schema . http ://www.w3.org/TR/rdf-schema/.

[W3C, 2004b] W3C (2004b). Web Ontology Language OWL. http ://www.w3.org/2004/OWL/.

[Wahlster, 2004] WAHLSTER, W. (2004). Smartweb : Mobile applications of the semantic web. *In* BIUNDO, S., FRÜHWIRTH, T. W. et PALM, G., éditeurs : *KI*, volume 3238 de *Lecture Notes in Computer Science*, pages 50–51. Springer.

[Wüster, 1976] WÜSTER, E. (1976). La théorie générale de la terminologie - un domaine interdisciplinaire impliquant la linguistique, la logique, l'ontologie, l'informatique et les sciences des objets. *In Actes du colloque international de terminologie*.

[Wu et Palmer, 1994] WU, Z. et PALMER, M. (1994). Verb semantics and lexical selection. *In 32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico.

[Zhong *et al.*, 2002] ZHONG, J., ZHU, H., LI, J. et YU, Y. (2002). Conceptual graph matching for semantic search. *In ICCS '02 : Proceedings of the 10th International Conference on Conceptual Structures*, pages 92–196, London, UK. Springer-Verlag.

[Zipf, 1949] ZIPF, G. K. (1949). *Human Behaviour and the Principle of Least Effort*. Addissley Press.

Liste des figures

Liste des tableaux

Glossaire

Ambiguïté d'un terme : caractère d'un terme ayant plusieurs sens possibles dont l'interprétation est incertaine ; synonyme de *polysémie*.
Anaphore : procédé grammatical assurant une reprise sémantique d'un précédent segment textuel (l'antécédent) par un mot ou un syntagme.
Annotation sémantique : méta-donnée correspondant à la sémantique d'un fragment textuel et formalisée par un élément ontologique ; produit de l'*indexation sémantique*.
Base de faits : partie extensionnelle de l'ontologie, constituée des instances de concepts, de leurs valeurs d'attributs et des relations sémantiques entre instances.
Base de recherche : ensemble des documents sur lequel s'applique le *SRI*.
Concept : entité de base d'une ontologie symbolisant une idée du domaine modélisé ; on peut définir cette notion en intension (via ses propriétés caractéristiques) et/ou en extension (via l'ensemble de ses instances).
Corpus : "*collection de textes constituée [...] pour évaluer une hypothèse linguistique ou répondre à un besoin applicatif*" [Condamines, 2003].
Co-occurrence : présence simultanée et récurrente de deux (ou plusieurs) mots dans un ensemble de documents.
Décidabilité : un langage ontologique est décidable si dans ce formalisme, tout raisonnement logique se conclut dans un temps fini.
Domaine (resp. codomaine) de relation : ensemble de départ (resp. d'arrivée) de la fonction définie par la relation sémantique en question ; dans l'ontologie, le domaine (ou le codomaine) d'une relation sémantique correspond à un concept nommé ou à l'union des instances de plusieurs concepts.
Hyponymie : un concept est hyponyme d'un autre s'il est plus spécifique (i.e. que son extension est incluse dans celle du second) ; antonyme d'une *relation hypéronymique*.
Indexation : recensement, dans une structure donnée (l'index), des différents ensembles de ressources textuelles qui mentionnent un terme donné parmi tous ceux du langage d'indexation, en vue de retrouver ces documents ultérieurement.
Indexation sémantique : association, pour chaque document de la *base de recherche*, d'un ensemble de méta-données sous forme d'éléments ontologiques modélisant tout ou partie de son sens.
Langage d'indexation : ensemble des termes reconnus par le *SRI* ; il peut être libre (aucune contrainte sur la nature des termes) ou contrôlé (lexique figé).
Maintenance de RTO : opération consistant à faire évoluer la RTO par l'ajout, la suppression ou la modification d'éléments terminologiques et/ou ontologiques.

Monosémie : caractéristique d'un terme au sens univoque ; antonyme de *polysémie*.
Ontographe : ingénieur modélisateur en charge de construire ou de maintenir la RTO.
Ontologie : "*spécification normalisée représentant les classes des objets reconnus comme existant dans un domaine*" [Charlet, 2002].
Polysémie : caractéristique d'un terme au sens plurivoque ; antonyme de *monosémie*.
Relation hypéronymique (ou taxonomique) : relation sémantique selon laquelle l'extension du premier concept englobe celle du second, plus spécifique ; antonyme d'*hyponymie*.
Relation méronymique : relation sémantique partitive ; A est méronyme de B s'il en constitue une partie.
Relation transverse : relation sémantique autre que la *relation taxonomique*.
Requête : expression par l'utilisateur du *SRI* de ses besoins en information.
Réseau d'instances : ensemble d'instances de concepts reliées entre elles par une ou plusieurs relations sémantiques.
Ressource Termino-Ontologique (RTO) : ressource comportant une composante conceptuelle (e.g. une *ontologie*) et une composante lexicale (e.g. une *terminologie*).
Similarité sémantique : fonction permettant de comparer, sur la base de propriétés communes, deux éléments ontologiques (concepts ou instances) à travers le calcul d'une valeur numérique comprise entre 0 (éléments totalement différents) et 1 (éléments identiques).
Système de Recherche d'Information (SRI) : ensemble de logiciels assurant les fonctionnalités nécessaires à la Recherche d'Information dans une *base de recherche*, i.e. entre autres l'*indexation* des documents et leur comparaison avec les *requêtes* de l'utilisateur.
Terme : entité lexicale décrite par son label, son contexte d'usage, sa (ses) signification(s) et/ou ses relations avec d'autres entités de même nature.
Terminologie : ensemble des termes d'un domaine.
Traitement Automatique du Langage Naturel (TALN) : discipline à la frontière de la linguistique et de l'intelligence artificielle, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain (écrit ou oral).
Web Sémantique (WS) : ensemble de technologies visant à rendre le contenu sémantique des ressources Internet accessible et exploitable par des agents logiciels, grâce à un système de méta-données formelles fondées sur des langages développés par le W3C.