Recherche d'information dirigée par les interfaces utilisateur : approche basée sur l'utilisation des ontologies de domaine

Amir Zidi

Université de Valenciennes et du Hainaut-Cambrésis

Thèse de doctorat

Pour obtenir le grade de Docteur de l'Université de

VALENCIENNES ET DU HAINAUT-CAMBRESIS

Sciences et Technologie, Mention : Informatique

Présentée et soutenue par Amir, ZIDI.

Le 26/03/2015, à Valenciennes

Ecole doctorale :
Sciences Pour l'Ingénieur (SPI)

Equipe de recherche :
Decsion, Interaction and Mobility (DIM)

Laboratoire :
Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH)
UMR CNRS 8102

Recherche d'information dirigée par les interfaces utilisateur : Approche basée sur l'utilisation des ontologies de domaine

JURY

Rapporteurs

-Pr. Mohand BOUGHANEM, Univ. Paul Sabatier Toulouse, France.
-Pr. Bernard ESPINASSE, Univ. Aix-Marseille 3, France.

Examinateurs

-Pr. Lotfi BEN ROMDHANE, Univ. Sousse, Tunisie
-Dr. Valérie MONFORT, Univ. Paris 1 Panthéon Sorbonne, France.
-Pr. Abdelhakim ARTIBA, Univ. Valenciennes et du Hainaut Cambrésis, France. (Président du jury)

Directeur de thèse

-Pr. Mourad ABED, LAMIH, Univ. Valenciennes, France.

Résumé

Ce mémoire porte sur l'utilisation des ontologies dans les systèmes de recherche d'information SRI dédiés à des domaines particuliers. Il se base sur une approche à deux niveaux, à savoir la formulation et la recommandation des requêtes. La formulation consiste à assister l'utilisateur dans l'expression de sa requête en se basant sur des concepts et des propriétés de l'ontologie de domaine utilisée. La recommandation consiste à proposer des résultats de recherche en utilisant la méthode du raisonnement à partir de cas. Dans cette méthode, une nouvelle requête est considérée comme un nouveau cas. La résolution de ce nouveau cas consiste à réutiliser les anciens cas similaires qui ne sont que des requêtes traitées auparavant. Afin de valider l'approche proposée, un système OntoCBRIR a été développé et un ensemble d'expérimentations a été élaboré. Enfin, les perspectives de recherche concluent le présent rapport.

Mots clés : Recherche d'Information, Ontologies de domaine, raisonnement à partir du cas, règles sémantiques.

Abstract

This thesis study the using of ontologies in information retrieval system dedicated to a specific domain. For that we propose a two-level approach to deal with i) the query formulation that assists the user in selecting concepts and properties of the used ontology ; ii) the query recommendation that uses the Case-based reasoning method, where a new query is considered as a new case. Solving a new case consists of reusing similar cases from the history of the previous similar cases already processed. For the validation of the proposed approaches, a system was developed and a set of computational experimentations was made. Finally, research perspectives conclude that this present report.

Keywords : Information Retrieval, domain Ontologies, Case-based reaso-

ning, semantic rules.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Introduction générale

Motivations

Avec le développement exponentiel des nouvelles technologies, les données, rendues disponibles par les systèmes informatiques, ne cessent pas de s'accroitre, formant ainsi des masses de plus en plus grandes et hétérogènes. Ces données sont souvent stockées dans des sources d'informations distinctes et de types différents, ce qui rend les tâches de la gestion et la recherche d'information complexes.

Le problème n'est donc plus la disponibilité de l'information mais plutôt la sélection de l'information pertinente, répondant aux besoins précis d'un utilisateur. De ce fait, pour qu'ils restent utilisables, les systèmes de recherche d'information devraient subir des mutations profondes afin de répondre aux nouvelles exigences. Ceci est particulièrement visible à travers plusieurs aspects, notamment la prise en compte de la sémantique du contenu informationnel. L'objectif est de faire face aux limites des techniques de traitement statistique de l'information qui utilisent des listes de mots-clés pour décrire le contenu de l'information dont la fréquence des occurrences de ces mots est déterminante dans la sélection des résultats aux utilisateurs.

Des ressources sémantiques, à l'instar des thésaurus et des ontologies, ont été largement utilisées dans la recherche d' information/donnée où le mot n'est plus seulement une simple chaine de caractères mais aussi une référence à des concepts ou à des entités d'un domaine spécifique ou à des relations entre ces entités [73]. En outre, les ontologies permettent aux utilisateurs d'interroger plusieurs sources d'informations par une simple interaction avec une seule base de connaissances, donnant ainsi un seul accès à ces sources.

Problématique

Bien que l'intégration des ontologies nécessite des langages de requêtes caractérisés par un formalisme complexe (e.g.[1] SPARQL), l'utilisation des langages naturels ou des langages visuels représente une alternative. Quant au langage naturel, il existe des limites relatives aux ambiguïtés sémantiques, ce qui affecte la pertinence des résultats fournis. Les langages visuels, caractérisés par l'incorporation des éléments non textuels (formulaires, icônes, images, graphes, etc.) [47] nous semblent adéquats. L'objectif derrière l'utilisation d'un tel paradigme est d'assister l'utilisateur dans sa formulation de requête même s'il ne dispose pas de connaissances au préalable sur l'ontologie utilisée dans le système de recherche d'information. Une telle vision s'avère concevable surtout lorsqu'il s'agit de rechercher des informations ou des données dans un domaine particulier.

Par ailleurs, la pertinence des informations fournies et leur adaptation aux préférences des utilisateurs sont devenues des facteurs clés de succès ou de rejet des systèmes de recherche d'information. Afin de répondre à ce critère, la personnalisation se présente comme une solution appropriée. Généralement, la personnalisation consiste à instaurer un environnement interactif qui aide de manière coopérative l'utilisateur dans ses différentes tâches tout en lui assurant une recherche efficace [53]. Ce rôle peut se traduire, en plus de la saisie de sa requête, par la validation des résultats fournis par le système.

La problématique étudiée dans ce travail de recherche consiste à apporter une solution qui s'articule autour de la question suivante :
Quelle approche utiliser pour concevoir un système de recherche d'information guidé par les ontologies et dédié à un domaine particulier en tenant compte des aspects cités ci-dessus ?

1. http ://www.w3.org/TR/rdf-sparql-query/

Organisation de la thèse

Le présent mémoire est organisé en deux parties principales. La première partie est composée de premiers chapitres dédiés à l'étude bibliographique. La deuxième partie, organisée autour de deux chapitres, présente l'ensemble de nos contributions.

Dans le premier chapitre, nous explorons une introduction au domaine de la recherche d'information RI où nous présentons ses principaux processus et les principaux modèles qui sont utilisés dans la littérature. Nous soulignons ensuite l'intérêt de la nouvelle variante de la RI qui est la recherche conceptuelle ou sémantique. Cette étude nous permet de présenter une des notions clés de cette thèse à savoir les ontologies. Pour ce faire, nous définissons les ontologies, leurs classifications ainsi que les différents langages et outils permettant leur manipulation. Enfin, nous terminons ce chapitre par une conclusion.

Le deuxième chapitre aborde un état de l'art sur les différents aspects de l'utilisation des ontologies dans la RI. Ces aspects couvrent tous les processus de la RI, à savoir l'indexation, l'appariement et le classement des résultats, la représentation des requêtes, l'interface utilisateur et la personnalisation. Étant donné que nous nous intéressons à l'inclusion de l'utilisateur dans la boucle de la pertinence en se basant sur ses préférences, nous présentons ensuite les modèles de préférences existants afin de positionner notre travail. La dernière partie de ce chapitre, nous l'avons consacré à l'étude des principaux travaux qui portent sur l'utilisation des ontologies et le raisonnement à partir de cas RàPC. L'objectif de cette étude est de montrer le rôle d'une telle méthode dans la personnalisation.

Le troisième chapitre est dédié à la présentation de notre approche principale de formulation et de recommandation des requêtes. Dans la première

partie de ce chapitre, nous expliquons la méthode de traitement des requêtes et leur représentation par des règles sémantiques SWRL. Nous montrons ensuite, le rôle des ces règles, d'une part dans la modélisation des préférences des utilisateurs, d'autre part dans l'enrichissement de l'ontologie utilisée. La deuxième partie est consacrée à la présentation de la méthode du raisonnement à partir de cas (RàPC) utilisée lors de la recommandation des requêtes. Une description détaillée du cycle RàPC est également présentée dans ce chapitre.

Afin de valider l'approche proposée, nous présentons dans le quatrième chapitre un système intitulé OntoCBRIR. Nous commençons par la description de l'architecture globale de ce système. Ses fonctionnalités sont ensuite présentées en se référant à l'ordre des étapes présentées de l'approche proposée dans le troisième chapitre. Le quatrième chapitre se termine par une étude expérimentale visant à montrer la fiabilité de l'approche proposée.

Ce mémoire se termine par une conclusion portant sur le bilan de notre recherche, sur l'ensemble des contributions apportées par cette thèse et finalement ses limites qui représentent aussi les perspectives .

Chapitre 1

Recherche d'Information : fondements et modèles conceptuels

1.1 Introduction

Dans sa version de base, un Système de Recherche d'informations (SRI) est un système informatique qui permet de retourner, à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin en information d'un utilisateur. L'approche classique suivie pour développer un SRI s'articule principalement autour de l'appariement requête-document. Elle se base sur le principe que le système renvoie essentiellement les documents qui contiennent les termes constituant la requête. L'inconvénient majeur de cette approche est le nombre énorme de documents renvoyés, ce qui provoque un certain bruit difficile à maîtriser par les utilisateurs. La sémantique de l'information est capitale pour pallier ce problème. Les travaux s'orientent actuellement vers une variante de la RI qui est la recherche d'information conceptuelle.

Dans ce chapitre, nous optons pour un aperçu du domaine de recherche d'information dans la première section. LA deuxième section est consacrée pour la présentation des concepts de base de la recherche d'information (RI). Une architecture générique du système de recherche d'information SRI ainsi que les principaux modèles de RI sont présentés dans les troisième et quatrième sections. Ensuite, nous soulignons les limites de la RI dite classique

dans la cinquième section. Dans l'objectif d'introduire une de ses variantes qui est la RI conceptuelle, laquelle est présentée dans la sixième section. La septième section est dédiée aux ontologies comme modèle de représentation conceptuelle. Enfin nous clôturons le chapitre avec la conclusion.

1.2 Concepts de base de la Recherche d'Information (RI)

La recherche d'information RI, en tant que problématique de recherche, a été largement étudiée dans la littérature. En effet, nous trouvons plusieurs définitions ayant pour objet la description des processus de la recherche d'information (RI). Ces définitions, quoiqu'informelles, sont quasiment identiques. Pour les auteurs [81] [7], la recherche d'information traite la représentation, l'organisation, le stockage et l'accès aux éléments de l'information qui peuvent être des documents, des pages Web, des catalogues en ligne, des enregistrements structurés/semi-structurés ou des objets multimédias. La représentation et l'organisation doivent être élaborées d'une manière permettant aux utilisateurs un accès facile à l'information. La définition originale est la suivante :

Définition 1. *"Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information items should be such as to provide the users with easy access to information of their interest"* [81] [7].

Dans une autre définition, l'objectif de la RI est de recenser, à partir de larges collections d'objets (généralement des documents textuels sous forme électronique), les informations pertinentes par rapport à un besoin en information d'un utilisateur. Ainsi, [63] définissent la RI de cette manière :

Définition 2. *"Information Retrieval is finding material (usually docu-*

ments) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)".

Au regard de ces définitions, la RI s'articule principalement autour d'un certain nombre de concepts clés [8] [90] :

- Collection de documents : en théorie, tout type de supports d'informations peut faire l'objet d'une recherche d'information. Cependant, nous nous focalisons sur les systèmes de recherche d'information (SRI), où les objets manipulés font référence à des « documents » textuels. De ce fait, la collection de documents constitue les informations exploitables et accessibles. Selon [81], ces informations peuvent faire l'objet de deux types de recherche : recherche d'information (*information retrieval*) et recherche de données (*data retrieval*). La recherche d'information se rapporte aux termes ainsi que leurs synonymes pouvant décrire le sujet de la requête utilisateur, et ce, pour déterminer les documents appropriés. Ce type de recherche est caractérisé par l'ambiguïté et le manque de précision des résultats. La recherche de données consiste à traiter les informations ayant une structure ou une sémantique bien définie. En conséquence, les résultats de recherche doivent être concrets et précis. Nous trouvons plus de détails dans les travaux de [45] qui s'articulent autour de la définition des principales propriétés de chaque type de recherche ainsi que leurs degrés de complexité.

- Document : le document est une granule d'informations élémentaires d'une collection de documents. Notons que la notion des documents textuels dans la RI s'applique à un large spectre de supports textuels, à savoir des parties complètes (articles, livres, pages Web) ou juste des sections (phrases, paragraphes, chaines de caractères).

- Besoin en information : l'utilisateur dans sa recherche d'information exprime ses besoins par un ensemble de mot clés. Cet ensemble constitue une requête qui peut être exprimée sous plusieurs formes.

- Pertinence : la pertinence en RI indique dans quelle mesure les docu-

ments retournés par le système de RI répondent au besoin d'information de l'utilisateur. Cette notion représente un critère majeur de l'évaluation des performances du système de RI [86]. La pertinence, qui est l'objet principal de tout système de RI, constitue une notion fondamentale en RI. Elle peut être définie comme la correspondance entre un document et une requête ou encore une mesure d'informativité du document à la requête.

1.3 Système de recherche d'information SRI

Du point de vue du système, la RI est effectuée suivant un cycle représenté schématiquement par le modèle en U de recherche d'information. La figure 1.1 illustre une abstraction de l'architecture d'un SRI, qui prend en entrée une requête formulée par un utilisateur (formulation de besoins) puis va sélectionner des documents estimés pertinents (résultats de recherche) au sein d'une collection préalablement indexée. Comme illustré dans l'architecture (figure 1.1), d'un côté se trouve la collection de documents qui n'est que le résultat de l'information accessible dans le système, de l'autre côté le besoin en information de l'utilisateur qui est exprimé par une requête. Notons que l'accès à l'information nécessite un traitement optimal de la collection de documents et de la requête afin de les rendre exploitables. Pour ce faire, les processus suivants sont mis en œuvre [63] [90] :

– Processus de représentation (indexation) : il s'agit de fournir un modèle de description expressif pour la collection de documents. Cette modélisation se traduit par la conversion des données textuelles, qui constituent la granule d'information (document), vers un format qui facilite une recherche rapide. Ce processus de conversion est appelé indexation. L'index inversé, en tant que structure de donnée, est le plus fréquemment utilisé lors de l'indexation [25] [70]. Il est composé d'un vocabulaire et une liste d'affectation. Le vocabulaire contient une liste de termes significatifs pour les documents, auxquels sont associés des

poids mesurant leur degré de représentativité. Chaque terme du vocabulaire est associé à une liste d'affection qui indique les documents ou ce terme apparait.

- Processus de formulation de requête : le besoin en information (requête) est traduit, en utilisant des modèles et des langages adéquats, sous des formes exploitables. En général, les termes qui constituent la requête utilisateur, sont traités par le même algorithme utilisé lors de la sélection des termes d'indexation [36]. La plupart des SRI attendent une requête textuelle, soit en langage libre (généralement une liste de mots clés), soit en combinant les termes via les opérateurs ensemblistes (ET, OU, SAUF). Alternativement, certains systèmes offrent la possibilité d'interroger leurs collections par navigation. Dans ce cas, l'utilisateur navigue généralement dans une ressource terminologique (thésaurus, liste de termes, hiérarchie de concepts, ontologie) pour sélectionner les éléments qui lui semblent répondre à son besoin d'information. Certains systèmes peuvent supporter un processus de reformulation automatique de requêtes dans l'objectif d'améliorer la qualité des résultats fournis en réponse aux requêtes des utilisateurs. L'idée principale est de prendre en compte des retours ($f$eedback) sur les documents jugés pertinents auparavant ou des précisions à partir des ressources de connaissances externes.
- Processus d'appariement : il s'agit de rechercher l'information en réponse à des requêtes utilisateurs et la restitution des résultats pertinents pour ces requêtes. Ce processus comprend la fonction d'appariement qui permet de calculer le degré de similarité entre la requête de l'utilisateur et les documents.
- Processus de classement : le résultat du processus d'appariement est une liste de documents, dont certains sont plus ou moins pertinents que les autres par rapport aux requêtes utilisateurs. Il devient donc nécessaire de déterminer le degré de pertinence des résultats de recherche. Pour ce faire, des algorithmes de classement sont utilisés pour attribuer un score à chaque document retourné. Dans l'objectif d'améliorer la fiabilité des SRI, d'autres processus peuvent être mis en place notamment

la personnalisation de la recherche :

– L'objectif de la personnalisation de l'information est d'intégrer l'utilisateur dans le processus global d'accès à l'information en vue d'adapter les différentes étapes (recherche, filtrage, visualisation) à son contexte. L'idée est d'adapter les résultats de recherche selon les intérêts et les préférences des utilisateurs.

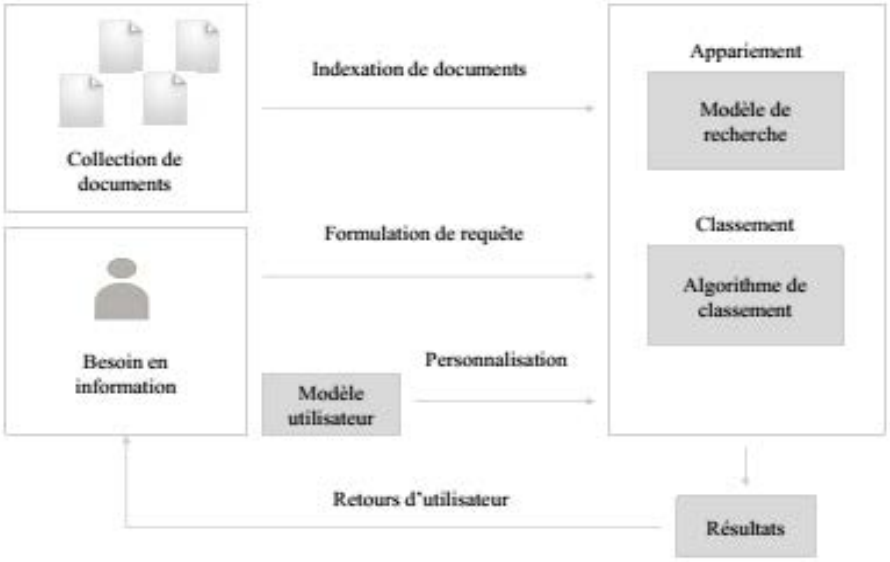


FIGURE 1.1 – Architecture générique d'un système de recherche d'information

En réalité, les résultats d'une recherche effectuée par l'utilisateur peuvent contenir à la fois des informations pertinentes et non-pertinentes. Un SRI peut donc être jugé efficace s'il a une meilleure capacité de comprendre le besoin en information (requête utilisateur) et par conséquent récupérer plus d'informations pertinentes et rejeter les non-pertinentes.

1.4 Principaux modèles de pertinence en RI

En suivant le cadre théorique utilisé pour l'indexation et l'appariement, plusieurs modèles de pertinence en RI sont à distinguer notamment : le modèle booléen où les documents et les requêtes sont simplement représentés par l'ensemble de termes d'indexation, le modèle vectoriel qui utilise les vecteurs pour représenter les documents et les requêtes dans un espace t-dimensionnel et le modèle probabiliste qui est basé sur la théorie de probabilité. Dans sa version basique, un modèle RI est peut être vu comme suivant :

- D est un ensemble de représentations logiques de documents.
- Q est un ensemble de représentations logiques des requêtes utilisateurs.
- F est un modèle mathématique adopté.
- R est une fonction de classement qui défini une relation d'ordre entre les documents par rapport à une requête q.
- Généralement, nous avons des termes d'indexation $t_1...t_n$.
- $w_{i,j}$ est un poids qui reflète l'importance d'un terme t_i dans un document d_j

1.4.1 Modèle booléen

Le modèle booléen basique ou étendu [85] est basé sur la théorie des ensembles et l'algèbre de Bool. En effet, les termes d'indexation sont vus comme des prédicats de la logique du premier ordre, où les poids des termes d'indexation $w_{i,j} \in \{0, 1\}$. De ce fait, les documents et les requêtes sont représentés par une combinaison logique des termes d'indexation. Par exemple, la requête

$q = t_1 \wedge (t_2 \vee (\neg t_3))$, peut être mise en correspondance avec la forme normale disjonctive, où nous avons une série de disjonctions ($q = 100 \wedge \vee 110 \vee 111$). L'appariement document-requête est donc effectué selon un critère de jugement binaire qui indique le degré de pertinence d'un document par rapport à une requête. Ce modèle est caractérisé par sa simplicité et son propre formalisme, cependant il ne prend pas en compte l'ordre des documents retournés en réponse à la requête utilisateur. En outre, les termes dans un document sont considérés comme indépendants les uns des autres.

1.4.2 Modèle vectoriel

Dans l'objectif d'améliorer le modèle booléen en supprimant la restriction de poids binaires pour des termes d'indexation, la théorie des espaces vectoriels [87] définit la similitude entre une requête de l'utilisateur et chacun des documents d'un corpus par l'angle qu'ils forment dans un espace dont les dimensions sont les termes. Cette théorie considère que les requêtes et les documents peuvent être représentés par des vecteurs de poids non binaires, $d_j = (w_{1,j}, w_{2,j}, ...w_{n,j})$ et $q = (w_{1,q}, w_{2,q}, ...w_{n,q})$. Ces poids sont associés aux termes d'indexation en se basant sur leurs fréquences. L'appariement document-requête est effectuée par des fonctions qui mesurent la colinéarité des vecteurs documents et requêtes dont les plus courantes sont décrites comme suit :

Le produit scalaire :

$$\mathrm{sim}(d_j, q) = \sum_{i=1}^{n} (w_{i,j} \times w_{i,q}) \tag{1.1}$$

La mesure de Jaccard :

$$\mathrm{sim}(d_j, q) = \frac{\sum_{i=1}^{n} (w_{i,j} \times w_{i,q})}{\sum_{i=1}^{n} (w_{i,q})^2 + \sum_{i=1}^{n} (w_{i,j})^2 - \sum_{i=1}^{n} (w_{i,j} \times w_{i,q})} \tag{1.2}$$

La mesure de Dice :

$$\text{sim}(d_j, q) = \frac{2 \times \sum_{i=1}^{n}(w_{i,j} \times w_{i,q})}{\sum_{i=1}^{n}(w_{i,q})^2 + \sum_{i=1}^{n}(w_{i,j})^2} \tag{1.3}$$

La mesure de cosinus :

$$\text{sim}(d_j, q) = \frac{d_j \times q}{|d_j|.|q|} = \frac{\sum_{i=1}^{n}(w_{i,j} \times w_{i,q})}{\sqrt{\sum_{i=1}^{n}(w_{i,j})^2} \times \sqrt{\sum_{i=1}^{n}(w_{i,q})^2}} \tag{1.4}$$

La mesure de cosinus consiste à déterminer la similarité entre le document et la requête en fonction de l'angle que forment leurs vecteurs dans l'espace vectoriel d'indexation $cos(a, b) = \frac{a \times b}{|a|.|a|}$ (figure 1.2).

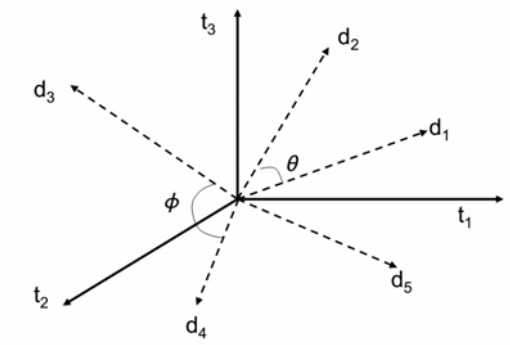


FIGURE 1.2 – Représentation vectorielle des documents et des requêtes

Le succès du modèle vectoriel dépend plus de la stratégie de pondération des termes mise en œuvre que de la structure de l'espace vectoriel adopté [39]. Pour trouver les termes du document qui représentent le mieux son contenu, [82] a défini la fonction de pondération des termes dans un document sous la forme de $tf - idf$. La fréquence d'un terme tf dans un document quantifie

le degré de représentativité du document par un terme. Plus il est présent, mieux il représente le document. En outre, si un terme est fréquent dans toute la collection des documents il sera moins représentatif de leur contenu. Cette mesure est représentée par idf. Ainsi, le poids d'un terme est calculé par la formule suivante :

$$w_{i,j} = f_{i,j} \times \log(\frac{N}{n_i}) \tag{1.5}$$

Avec :

- $f_{i,j}$ une normalisation de la fréquence d'un terme t_i dans un document d_j (tf).
- N est le nombre du document dans la collection.
- n_i est le nombre de documents contenant le terme t_i.

L'avantage du modèle vectoriel par rapport au modèle booléen réside particulièrement dans le classement des documents sélectionnés selon leurs pertinence. Cependant, à l'instar du modèle booléen l'inconvénient majeur de l'approche vectorielle réside dans le fait que l'association entres les termes d'indexation n'est pas considérée.

1.4.3 Modèle probabiliste

L'idée provient de l'hypothèse selon laquelle la recherche d'information traite une information incertaine dans la représentation des documents et des requêtes ainsi que dans la phase d'appariement. La solution consiste donc à quantifier cette incertitude par les probabilités [22]. Le premier modèle probabiliste a été proposé par [82], dans lequel deux probabilités conditionnelles sont utilisées lors du processus d'indexation :

- La probabilité pour que le terme t_i apparaisse dans un document donné sachant que ce document est pertinent pour la requête.
- La probabilité pour que le terme t_i apparaisse dans un document donné sachant que ce document est non pertinent pour la requête.

Pour une étude détaillée du modèle probabiliste, les travaux de [59] peuvent être cités. Bien qu'il existe d'autres extensions de ces modèles (booléen, vectoriel et probabiliste), l'objectif reste le même. Il s'agit de modéliser une certaine conception de la pertinence d'un document pour un utilisateur ayant un besoin d'information [39].

1.5 Limites de la RI classique

Les approches suivies pour développer un SRI s'articulent principalement autour de l'appariement requête-document. Elles se basent sur le principe que le système renvoie essentiellement les documents qui contiennent les termes constituant la requête ou les termes ayant une certaine proximité logique (similarité entre chaines de caractères). La pertinence des résultats issus de cet appariement dépendra donc de la manière comment les documents ont été représentés. Néanmoins, la représentation la plus répandue dans les modèles de RI dite classique est celle du "sac de termes" dans lequel le document est vu comme un ensemble de termes que l'on estime les plus discriminants pour décrire son contenu. L'hypothèse fondamentale de cette représentation,est l'indépendance des termes d'indexation [39]. Cela peut affecter la performance du SRI, d'autant que l'utilisateur ne connait pas forcément les sources de données qu'il interroge ni leur description ni leur contenu. En conséquence, sa requête ne traduit plus un besoin précis mais une intention qui doit être affinée en fonction des sources de données disponibles au moment de l'interrogation [58]. A l'issue de ses interactions avec le système, l'utilisateur examine la liste des documents retournés par le SRI pour trouver ceux qui sont susceptibles de répondre à ses besoins. S'il est chanceux, il trouvera dans cette liste le document qui satisfait pleinement ses besoins. En effet, la qualité des résultats retournés dépend également de la façon dont la requête a été formulée. Ainsi, il est nécessaire d'assister l'utilisateur dans la formulation de sa requête, ce qui n'est pas souvent possible en présence du modèle "sac de termes", pour lequel certaines limites peuvent être exister, notamment :

– Polysémie : un terme revêt plusieurs sens.

- Synonymie : deux mots se rapportant à un même sens.
- Relations entre les termes d'une requête.
- Inférence : déduction des nouvelles connaissances à partir des termes de la requête.

En outre, les informations pertinentes peuvent être dispersées dans différents documents, l'utilisateur devrait donc rassembler et agréger les sections d'informations afin de construire la réponse la plus appropriée à ses besoins.

Ces limites ont fait l'objet d'un nouveau paradigme en RI. Il s'agit de la recherche conceptuelle, qui se rapporte, pour la représentation d'un document ou d'une requête, aux sens des termes ainsi qu'aux relations qui les lient [94]. L'objectif de la recherche conceptuelle est de faciliter l'expression du besoin de l'utilisateur et lui permettre d'obtenir des résultats pertinents en exploitant au mieux la sémantique de l'information. Plus précisément, il s'agit d'enrichir l'ensemble de documents et/ou de requêtes par des connaissances supplémentaires ou des annotations afin de supporter un haut niveau de conceptualisation de l'information. L'illustration de cette conceptualisation repose sur une architecture de concepts caractérisés par des définitions formelles. C'est ce caractère formel qui permet à l'information d'être traitée par les machines. Le recours à des ontologies, pour mettre en place l'architecture de concepts, devient intéressant. En effet, les ontologies qui sont une spécification explicite, formelle d'une conceptualisation partagée [44], permettent d'optimiser la formulation des requêtes et d'améliorer la precision des résultats de recherche. Quant aux SRI, il doit y avoir un accès transparent à l'ensemble de sources de données représentées par les ontologies. L'accès peut être vu comme un processus itératif et exploratoire au cours duquel l'utilisateur s'engage activement avec le système [99].

1.6 Vers une vue conceptuelle pour la RI

Dans la RI textuelle, il devient de plus en plus indispensable de catégoriser les informations non structurées pour réduire les limitations des algorithmes

d'appariement basés sur les "sacs de termes". Dans certains cas, les catégories ne sont pas encore connues au moment de l'analyse des documents et il devient alors nécessaire d'exploiter la puissance des solutions d'analyse conceptuelle ou d'extracteur d'entités afin d'extraire les différents thèmes et concepts se rapportant au document en cours et d'en déduire le nom de la catégorie à laquelle ce dernier appartient. Une fois ces thèmes et concepts déterminés, il faut les lier au dit document et seront les descripteurs de ce document.

Un concept représente une idée générale et abstraite que se fait l'esprit humain d'un objet de pensée concret ou abstrait, et qui lui permet de rattacher à ce même objet les diverses perceptions qu'il en a et d'en organise les connaissances (Larousse 2012)). Selon [40], un concept fait référence à un groupe d'objets ou *d'êtres* partageant des caractéristiques et des propriétés permettant de les identifier comme appartenant à un même ensemble. Cependant, l'extension d'un concept concerne l'ensemble des objets ou *êtres* dont il représente une abstraction []. Dans la recherche d'information (RI), la représentation conceptuelle peut être effectuée en utilisant différents modèles formels, à savoir les taxonomies, les facettes de classification et les ontologies [90].

1.6.1 Les taxonomies

Les taxonomies sont des schémas de classification hiérarchique. Elles permettent de faciliter la recherche d'un terme en fonction des relations hiérarchiques entre les concepts. Il s'agit principalement d'une relation de subsomption, dite "est-un" (*super-subcategory*). Comme exemple de taxonomie, nous pouvons citer WordNet [69] pour lequel un concept est représenté par un Synset, c'est-à-dire l'ensemble des termes (mots ou groupes de mots) synonymes qui peuvent le désigner [8]. Les concepts reliés sémantiquement par une relation donnée à un Synset sont représentés par une classe qui porte le nom de la relation. La relation de base entre les termes dans WordNet est la Synonymie. Les Synsets sont liés par des relations spécifiques, génériques

(hyponyme-hyperonyme *is-a*) et la relation de composition(meronymie holonymie.) Ces relations sont illustrées sur la figure 1.3. En outre, le filtrage des contenues dans la (RI) a fait l'objet d'autres taxonomies notamment *OpenDirectoryProject ODP*[1] et *Yahoo Directory*[2].

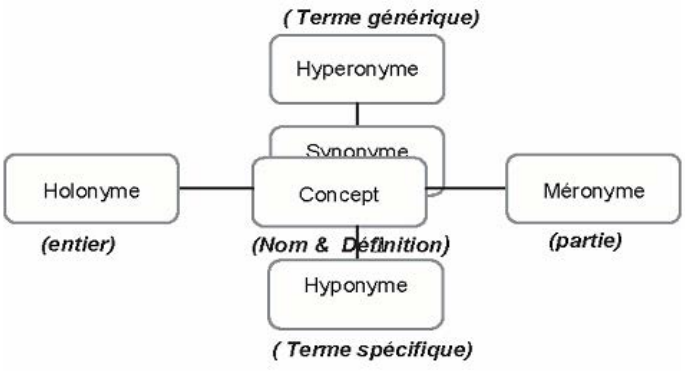


FIGURE 1.3 – Types de relations dans Wordnet

1.6.2 Les facettes de classification

Initialement développée pour une bibliothèque des sciences [103], la classification à facettes définit une façon de décrire une ressource selon plusieurs catégories conceptuelles (les facettes), chaque facette contenant des termes qui peuvent être décrits dans un thésaurus, un terme appartenant à une seule facette. Cette multi-classification est utilisée dans la "recherche à facettes" ou "navigation par facettes". Par exemple, une navigation par facettes pour les informations d'un département Marketing peut se traduire par la possibilité de consulter les documents selon les marchés, selon les produits, selon les spécialités, etc.

1. http ://www.dmoz.org/
2. https ://dir.yahoo.com/

1.6.3 Les ontologies

Les ontologies permettent de représenter formellement les connaissances, de décrire le raisonnement sur ces connaissances et de les partager et les réutiliser par plusieurs applications [44]. Nous donnons par la suite les définitions d'une ontologie ainsi que sa représentation formelle et syntaxique.

Définitions

La définition la plus convenue de l'ontologie a été proposée par [44] dans laquelle il considère l'ontologie comme étant une spécification formelle et explicite d'une conceptualisation partagée. Les auteurs [43] ont expliqué ces critères de la manière suivante :

- "Conceptuel" : se réfère à un modèle abstrait des concepts pertinents caractérisant des phénomènes dans le monde.
- "Explicite" : précise que le type de concepts utilisés et les contraintes sur leur utilisation sont explicitement définis.
- "Formelle" : définit qu'une ontologie doit être exprimée par un langage de représentation de connaissances qui fournit une sémantique formelle. Cela garantit que l'ontologie peut être exploitable par des machines et interprétable d'une manière bien définie.
- "Partagée" : reflète qu'une ontologie doit capturer la connaissance consensuelle acceptée par les différentes communautés.
- "Spécificité du domaine" : il ne permet pas de créer une ontologie de haut niveau qui engloberait toutes les autres, et par conséquent, l'idée est d'opter pour une spécification qui se limite à des connaissances sur un domaine d'intérêt particulier. En effet, la conception d'ontologie se concentre plutôt sur les détails dans ce domaine que de couvrir un large éventail de sujets connexes. De cette manière, la définition explicite de connaissances du domaine peut être modulaire et exprimée en utilisant plusieurs ontologies différentes avec des domaines d'intérêt distincts.

Afin de formaliser l'ontologie, le W3C [3] l'a défini de la manière suivante : *"L'ontologie est définie comme étant les termes définissant un domaine spécifique ainsi que les relations entre eux".*
Ainsi, nous déduisons de cette définition que les éléments de base composant l'ontologie sont :

-Concepts ou classes : représentent les entités ou bien les objets définissant l'existence du domaine ciblé par l'ontologie.

-Instances (ou individus) : constituent la définition extensionnelle de l'ontologie. Ce sont des objets concrets d'une classe.

-Relations : permettent de définir les liens entre les classes.

-Propriétés : servent pour la définition de la sémantique exprimée par chaque concept défini.

-Axiomes : permettent de définir la sémantique des termes (classes, relations), leurs propriétés et toute contrainte liée à leur interprétation.

A l'égard de ces définitions, nous ne pouvons pas affirmer que les différents types d'ontologies se résument en un seul qui serait l'ontologie de domaine. En réalité, les ontologies peuvent présenter différents niveaux d'abstraction dans les détails des conceptualisations capturées, ce qui entraine une variété dans leurs classifications [6]. Parmi ces classifications nous pouvons souligner celle de [38]. Cependant, nous nous focalisons sur les ontologies de domaine qui permettent de décrire un domaine spécifique (la médecine, le droit, le transport, etc.) sous forme d'une base de connaissances. Nous donnons par la suite une description formelle d'une base de connaissances.

Langages de description des ontologies

D'un point de vue opérationnel, les ontologies peuvent être développées par des langages classiques de la programmation logique (par exemple : Prolog, Lisp, etc). Avec l'émergence du Web sémantique et sa modélisation de

3. http ://www.w3.org/

connaissance, il a fallu opter pour des langages spécifiques afin de construire des ontologies. Cependant, aujourd'hui, différents modèles et langages de description notamment, F-logic [56], *Resource Description Framework* RDF [65], DAML+OIL [66] et *Web Ontology Language* OWL [67]. Les langages RDF et OWL sont les plus fréquemment utilisés étant donné qu'ils fournissent une meilleure représentation formelle des connaissances sémantiques. Dans RDF, l'ensemble des ressources constitue un gigantesque graphe conceptuel qui sera sérialisé sous la forme de triplets, c'est-à-dire des relations de type sujet - prédicat - objet. Plus précisément, chaque arc du graphe est étiqueté par un prédicat et relie un nœud source (sujet) à un nœud cible (objet) (4.1).

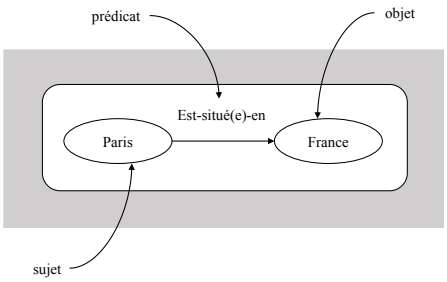


FIGURE 1.4 – Exemple d'un triple RDF (sujet-prédicat-objet)

Dans l'objectif de donner plus de spécification aux concepts, un nouveau schéma *Resource Description* RDFS [18] a été introduit. Ce dernier, fournit un vocabulaire de base pour décrire les propriétés et les classes des ressources RDF. En utilisant RDFS, il est possible de créer des hiérarchies de classes et de propriétés. SPARQL [4] est un langage de requête et un protocole permet-

4. http ://www.w3.org/TR/rdf-sparql-query/

tant, à l'instar de SQL, de rechercher, d'ajouter, de modifier ou de supprimer des données RDF. Il existe cependant d'autres langages de requête. Par exemple, la requête donnée, ci-dessous, permet de récupérer l'ensemble des villes ainsi que le pays dans lequel elles sont implantées :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdfsyntax-ns#>
PREFIX Lieux: <http://example.org/Lieux#>
SELECT?Ville ?Pays
WHERE {
?Pays rdf:type Lieux:Lieu.
?Ville rdf :type Lieux :Ville.
?Ville Lieux :est_situé(e)_en ?Pays
```

OWL a été développé comme une extension de RDF. Basé sur la logique de description [5], le langage OWL permet par exemple de rajouter des contraintes supplémentaires, en particulier, des cardinalités , ou des caractéristiques des propriétés telle la transitivité. Afin de permettre une meilleure utilisation du langage, celui-ci a été structuré en trois parties : OWL-Lite, OWL-DL et OWL-Full.

- OWL-Lite : permet d'établir une hiérarchie simple de concepts et contraintes.
- OWL-DL : comprend toutes les structures de OWL et possède une expressivité plus importante, avec complétude de calcul.
- OWL-Full : permet une expressivité maximale, une liberté syntaxique sans garantie de calcul et un partage des instances entre différentes classes.

Au delà des relations binaires et des propriétés qui peuvent caractériser une classe, nous pouvons exploiter des relation n-aires entre les concepts par le biais des Règles spécifiques comme *Semantic Web Rule Langage* SWRL [5] [52].

De point de vue création, il existe plusieurs outils d'édition des ontologies. Parmi ces outils, nous pouvons citer Protégé [77]. Il est l'un des outils les plus

5. http ://www.w3.org/Submission/SWRL/

fréquemment utilisés grâce à l'éditeur intuitif qu'il fournit. Il permet la création des ontologies et la communication avec plusieurs moteurs d'inférence notamment Pellet [92], Fact++[98] et HermiT [42]. Ces derniers ont pour rôles la vérification de la consistance de l'ontologie et la déduction des nouvelles connaissances en se basant sur les relations et les règles préalablement définies.

Ontologies et base de connaissances

Une ontologie et l'ensemble des instances individuelles des concepts constituent une base de connaissances. Une frontière subtile marque la fin d'une ontologie et le début d'une base de connaissances. Les auteurs de [73] ont présenté les différents éléments d'une base de connaissance de la façon suivante, en donnant les équivalents OWL de chaque notation :

A) **Une ontologie de domaine** O^i est définie par le quadruplet (C_{O^i}, R_{O^i}, A_{O^i}, X_O^i) où :

-C_{O^i} est l'ensemble des concepts (i.e. owl : Class).

-R_{O^i} est l'ensemble de relations : propriétés dont le domaine et le co-domaine sont des concepts (i.e owl : DatatypePropoerty).

-A_{O^i} est l'ensemble des attributs : propriétés dont le domaine est un concept et le co-domaine est un littéral (i.e. owl : DatatypeProperty).

- X_O^i est un ensemble d'axiomes définissant les caractéristiques des concepts et des propriétés. Nous listons quelques exemples :

- domain(P,C) indique que le domaine de la propriété P est C (i.e <P rdfs : domain C>).
- range(P,C) indique que le co-domaine de la propriété P est C (i.e <p rdfs : range C>)
- subClass(C_1,C_2) indique que C_1 est un sous-concept de C_2 (i.e <C_1 rdfs : subClass C_2>)
- subProperty(P_1,P_2) indique que P_1 est une spécialisation de P_2 (i.e <P_1

rdfs : subProperty P_2>)

B) **Un ensemble de faits noté** $T^j_{O^i}$ décrivant les instances de O^i.

Une base de connaissances est en fait constituée de deux couches :

– Une couche conceptuelle qui est la couche de plus haut niveau définissant les types d'objets (les classes) ainsi que les relations existantes entre ces derniers. Cette couche permet de donner un sens à des termes, c'est-à-dire de définir les notions qu'ils désignent et de justifier leur place dans la terminologie. Techniquement, cette couche peut être définie, soit en RDFS pour les ontologies dites légères (OWL-Lite), soit en OWL pour des ontologies plus complètes (OWL-DL et OWL-Full).

– Une couche d'instanciation (peuplement) qui constitue la couche correspondant à la réalité.

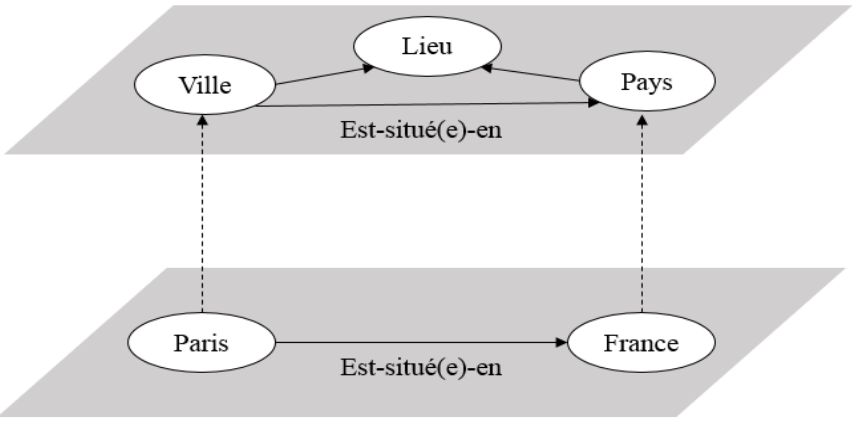


FIGURE 1.5 – Les deux niveaux d'une ontologie peuplée

La figure 1.5 donne un exemple d'une base de connaissances. Dans cet exemple, nous pouvons constater, d'une part trois concepts (e.g. Lieu, Ville, Pays) et une propriété (e.g. est-situé-en) formant la couche conceptuelle et d'autre part un ensemble d'individus (e.g.. Paris, France) formant la couche d'instanciation.

Inférence de connaissances sémantiques

Hérité de l'Intelligence Artificielle, le système expert est sans doute l'un des points forts des technologies sémantiques, puisqu'il repose son raisonnement sur les ontologies des bases de faits et de règles en utilisant son moteur d'inférence.

Prenons l'exemple de l'ontologie de la figure 1.6 qui montre l'appartenance implicite d'un individu ind_1 de la classe D à la classe B. Ainsi si on déclare, un nouveau individu $ind2$ et qu'on souhaite afficher les individus de la classe B, alors l'individu $ind2$ en fera partie des résultats. il s'agit d'une inférence par

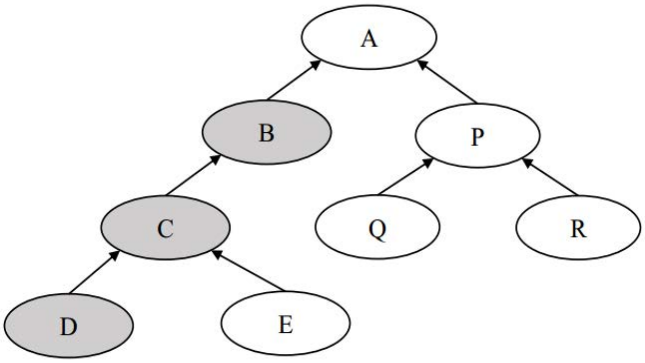


FIGURE 1.6 – fig :Déduction des nouvelles relations

déduction de type généralisation qui est souvent mise en œuvre nativement. Il est également possible de définir des règles supplémentaires à l'aide de différents langages comme SWRL. La règle ci-dessous indique que si nous avons x, y et z des instances de la classe $Personne$, que y et z sont les parents de x par la relation $a - parent$, il sera alors possible de déduire que x est une instance de la classe $EnfantDeParentsMaries$.

```
Personne(?x)a-parent(?x, ?y),
a-parent(?x, ?z),
a-conjoint(?y, ?z)
-> EnfantDeParentsMaries(?x)
```

1.7 Les ontologies comme modèle de représentation de connaissances dans la RI

Les ontologies comme modèle de représentation de connaissances liées à l'émergence du Web sémantique ont suscité un fort développement des approches conceptuelles en RI. L'objectif est, d'une part, de pouvoir faire un certain nombre de raisonnements fondés sur la structure de l'ontologie et les relations entre ses éléments afin d'assister au mieux l'utilisateur, d'autre part de répondre au mieux à ses besoins. Ceci a fait l'objet de plusieurs travaux notamment, [96] [20] [36]. Ces auteurs ont étudié l'utilisation des ontologies en tant que bases de connaissances. L'idée est d'annoter les documents par les instances de l'ontologie et d'y appliquer un mécanisme de raisonnement. Ils utilisent ensuite les modèles classiques de la RI (booléen, algébrique, etc.) lors de l'appariement document-requête.

Les auteurs de [83] ont proposé un système de recherche d'information basé sur une ontologie ; il permet aux utilisateurs d'effectuer des recherches sur des instances de l'ontologie au lieu de rechercher des pages Web arbitrairement. Le système commence par créer la base de connaissances après avoir peuplé l'ontologie par des instances (URI). Une requête, initialement exprimée par un ensemble de mots-clés, est utilisée pour interroger la base de connaissances. Ensuite, le système renvoie ensuite un ensemble des instances comme résultat initial.

Les auteurs [107] ont proposé un modèle permettant la recherche dans des portails sémantiques en interrogeant des bases de connaissances. La requête utilisateur est définie formellement par un ensemble de concepts de l'ontologie. Les résultats de recherche peuvent être des instances de ces concepts ou bien des informations collectées à partir de la représentation textuelle des relations sémantiques.

Les auteurs [105] et [106] ont proposé respectivement deux systèmes de recherche d'information dans le domaine du e-commerce et le domaine de la chaine logistique. Dans les deux systèmes proposés, la représentation sémantique des documents reposent sur trois éléments : concepts, propriétés et

valeurs.

Les travaux que nous venons de citer reposent sur la logique de description [5] qui ne met pas des restrictions sur les relations sémantiques définies dans l'ontologie. Néanmoins, il existe d'autres travaux [8] [94] [33] qui se rapportent principalement aux relations de subsomption d'une ontologie. Cette restriction peut être représentée par un graphe dont les nœuds sont les concepts et les arcs sont les liens (*is-a*). Nous expliquons cet aspect par la suite.

Enfin, les auteurs de[34] et [90] ont présenté des revues importantes des techniques de la RI basée sur les ontologies.

1.8 Conclusion

Dans ce chapitre, nous avons présenté comment mettre en œuvre les concepts de basse la recherche d'information RI en introduisant brièvement ses principaux modèles de pertinence. Ensuite, nous avons souligné les limites de la RI dite classique ce qui a conduit à l'émergence d'une vue conceptuelle pour la représentation des documents et des requêtes. Enfin, nous avons consacré une partie importante pour la présentation des ontologies comme modèle de représentation de connaissances. Nous pouvons conclure que cet état de l'art nous a permis de mettre en évidence la nécessité de considérer l'ontologie comme une composante importante dans le cadre de la RI conceptuelle. Cependant, nous ne retenons dans notre travail les ontologies de domaine comme un modèle de représentation de connaissance.

Dans ce contexte, nous poursuivons cet état de l'art dans le second chapitre en mettant en exergue l'exploitation des ontologies comme modèles de connaissances pour le contenu informationnel, en particulier les requêtes des utilisateurs et nous nous focalisons sur l'utilisation des ontologies dans la boucle pertinence et nous nous intéressons plus précisément de l'intégration de l'utilisateur dans cette boucle à travers l'expression de ses préférences.

Chapitre 2

Exploitation des ontologies pour la Recherche d'Information

2.1 Introduction

Les ontologies ont pour vocation l'amélioration de la capacité d'un SRI à diminuer le "silence" (absence de documents pertinents dans les résultats du SRI) et le "bruit" (proportion de documents non pertinents parmi ceux fournis). Cette hypothèse a conduit l'exploitation des ontologies en tant que modèle de connaissances lié à l'émergence du Web sémantique. Par conséquent, l'expression de la requête et la stratégie pour trouver des résultats pertinents devaient être repensées. C'est pour cela qu'il existe différents niveaux d'utilisation des ontologies dans la RI à savoir la formulation de la requête, la modélisation de l'utilisateur, l'indexation des documents, la visualisation des résultats ou bien une utilisation multiple.
Dans ce chapitre, nous étudions l'usage et la mise en œuvre des ontologies dans la RI à travers différentes approches. Comme nous l'avons mentionné dans le chapitre précédent, le scénario de base de la RI ne s'arrête pas à la représentation du contenu informationnel (document/requête) et l'interrogation. L'utilisateur y joue un rôle central puisqu'il est le seul juge des résultats. Dans cette optique, nous étudions la manière dont l'utilisateur est inclus dans la boucle de pertinence à travers des approches visant à proposer un modèle de préférences pertinent. Nous étudions également les approches

qui sont étroitement liées avec l'aspect de la personnalisation que nous traitons dans nos travaux de thèse. Ces approches repose sur la combinaison des ontologies et de la méthode de raisonnement à partir de cas (RàPC).

2.2 Usage des ontologies dans la RI

L'utilisation des ontologies dépend du degré de leur implication dans le processus de la RI : une utilisation restreinte ou partielle et une utilisation avancée ou étendue [8].

L'utilisation avancée est traduite par la représentation sémantique des documents. Les travaux de [49] [49] consistent à identifier les concepts importants dans les documents en fonction de deux critères, la co-occurrence et les relations sémantiques. Une étape de désambiguïsation est effectuée ensuite par l'intermédiaire d'une ontologie externe WordNet [64]. Les résultats du *mapping* entre l'ontologie et les documents sont des réseaux sémantiques dont les concepts indexés représentent les nœuds et les arcs des liens pondérés. Parmi les limites de cette approche, la complexité de l'étape de désambiguïsation et la couverture limitée de WordNet (seuls les termes les plus communs sont référencés) peuvent être soulignées. Dans la même optique, [9] utilisent WordNet afin de récupérer les termes reliés à la requête par des relations de synonymie, généralisation et spécialisation. Les auteurs de [55] ont proposé un système de recherche d'information par mots-clés. Ce dernier système crée une ontologie de domaine (football) à partir des informations extraites des sources d'informations. Le système génère ensuite des fichiers OWL à partir de l'ontologie. Les données sémantiques de ces fichiers seront indexées d'une manière structurée et seront utilisées pour répondre à des requêtes par mots-clés. L'ontologie est utilisée dans l'extraction des informations, l'inférence et la phase de recherche. Les résultats obtenus ont montré l'efficacité d'une telle approche par comparaison avec des approches d'indexation classique et des méthodes de reformulation de requêtes. Dans [75], une ontologie de domaine a été combinée avec une technique d'analyse du langage naturel pour

extraire des concepts importants à partir des documents Web et construire ensuite le contenu sémantique de ces documents. Le système commence par extraire les termes représentatifs des documents Web en suivant l'approche classique [87] ; par la suite, un mapping entre ces termes et l'ontologie de domaine est effectué pour constituer une portion du document sémantique (résultat de l'indexation). À la fin, le document sémantique sera complété par les concepts confirmés par l'utilisateur à travers ses interactions avec le système. L'inconvénient de cette approche est le fait d'intégrer l'utilisateur dans le processus de la formulation des requêtes, ce qui augmente son interaction avec le système. Dans ce contexte, l'interrogation sémantique se fait à travers des requêtes SPARQL formulées suivant le vocabulaire d'une ontologie et couvrant généralement un domaine d'application particulier. Les bases de connaissances RDF(S)/OWL interrogées représentent des informations sous la forme (sujet, relation, objet) et permettent d'obtenir des réponses de différents types :

- Des concepts et des relations définis dans l'ontologie.
- Des instances de concepts.
- Des instances de relations.
- Des valeurs littérales correspondant à des valeurs d'attributs d'instances.

Les ontologies sont utilisées également en amont des SRI pour améliorer l'expressivité du besoin de l'utilisateur. L'idée principale est de prendre en compte des connaissances détaillées sur l'utilisateur en définissant un modèle basé sur une ressource sémantique externe pour décrire ses préférences et ses intérêts [28] [54]. Dans [28], [1]l'ODP est vue comme une ontologie où ses concepts sont utilisés pour représenter sémantiquement les centres d'intérêts dans les profils utilisateurs. Les auteurs [3] ont proposé une approche pour l'accès contextuel à l'information en utilisant une ontologie de domaine dans un système de recommandation. Une autre approche proposée par [68] consiste à créer les profils des utilisateurs à partir de leurs comportements et de leurs retours de pertinence (*relevance feedback*) ; les termes modélisant

1. http ://www.dmoz.org/

les profils sont déterminés à partir d'une ontologie destinée à classifier des papiers scientifiques. Pour répondre à une requête utilisateur, un système de recommandation collaboratif détecte et recommande les papiers similaires entre les profils ayant les mêmes intérêts.Les auteurs [91] ont défini un espace pour le profil utilisateur sous forme d'une ontologie UPS (*User Profile Space*). Il s'agit d'un nouveau modèle ayant une structure dynamique basée sur le retour de pertinence et les interactions avec les utilisateurs. En outre, ce travail présente un modèle d'expansion de requête multi-ontologie, où chaque utilisateur est représenté par une ontologie. Dans ce modèle, des arbres couvrants sont générés à partir des graphes de l'ontologie UPS en se basant sur les requêtes initiales. L'objectif est d'améliorer la représentation des requêtes et de permettre également une recherche personnalisée sur une base sémantique.

2.3 Mise en œuvre des ontologies dans la RI

Dans cette section, nous détaillons la mise en œuvre des ontologies comme modèle de représentation de connaissances dans les différentes tâches qui constituent la RI. Nous tirons de cette étude, la méthode d'intégration d'ontologie utilisée dans cette thèse.

2.3.1 Indexation conceptuelle

L'indexation consiste à produire une représentation formelle du contenu en déterminant et extrayant les termes représentatifs d'un document ou d'une requête qui couvrent au mieux leurs contenus sémantiques. Elle est généralement élaborée en suivant deux étapes : l'extraction des termes et leur pondération. Comme résultat, nous pouvons avoir une liste ou groupe de termes significatifs pour l'unité textuelle correspondante. Chaque terme est généralement assorti d'un poids représentant le degré de représentativité du contenu sémantique de l'unité qu'ils décrivent. D'un point de vue concep-

tuelle, l'indexation selon [33] permet de décrire un document en utilisant des éléments (concepts) uniques qui abstraient les notions compréhensibles par l'humain. En effet, une nouvelle représentation est obtenue et elle ne se rapporte pas seulement aux sens des termes mais aussi à leurs unifications quand il s'agit de sources de données hétérogènes. Ceci implique de pouvoir exploiter différentes relations sémantiques et mettre en place un mécanisme du raisonnement qui aboutit à la production des nouveaux faits. En conséquence, il devient nécessaire d'adapter la structure d'indexation pour qu'elle couvre la notion de concepts.

Structures d'indexation conceptuelle

La conceptualisation du contenu informationnel peut se faire en suivant deux approches génériques.

- La première approche découle d'une tradition documentaire et vise la classification des objets par des relations taxonomiques [30] [31][94]. En effet, la structure d'index est basée sur le modèle dit "sac de concepts" ; elle est souvent adoptée lorsqu'il s'agit d'utiliser des ontologies ayant principalement des relations de subsomption (e.g. domaine médical), chaque concept étant pris indépendamment des autres.
- La deuxième approche consiste à annoter les documents, elle se distingue par les types de connaissances ontologiques utilisés pour l'annotation et par la nature et la granularité des parties de document annotées [73] [20] [36] [55].

Le processus clé dans les deux approches consiste à détecter les concepts appropriés d'une taxonomie ou une ontologie pour décrire au mieux le contenu du document ou une requête. Ce processus peut se faire, soit manuellement en utilisant des outils comme Protégé [2] [77] et CREAM [46], soit semi-automatiquement en se basant sur les interactions des utilisateurs [57], soit automatiquement grâce à des outils de traitement de langage automatique [79] [26] [84]. Les concepts détectés ainsi que les relations sémantiques, dé-

2. http ://protege.stanford.edu/

finies explicitement dans l'ontologie, seront ensuite affectés aux documents. Nous parlons précisément de l'assortiment (*mapping*) entre les concepts et le contenu informationnel. Pour ce faire, [36] ont utilisé les techniques d'extraction d'information (IE) pour identifier dans les documents les termes ou groupes de termes qui peuvent potentiellement refléter les entités sémantiques de l'ontologie (classes, propriétés, instances ou littérales) (figure 2.1). Une telle démarche nécessite un ensemble de conditions :

- L'annotateur sémantique identifie les entités de l'ontologie (classes, propriétés, instances ou littéraux) dans les documents textuels et génère les annotations correspondantes en utilisant les techniques du traitement automatique du langage TAL (filtrage des termes). Ceci est équivalent à un processus d'indexation de la RI classique où les unités d'indexation sont des entités ontologiques (sens des termes) au lieu des simples mots-clés.
- Les processus d'annotation effectués ne visent pas à peupler les ontologies mais à identifier les connaissances sémantiques déjà disponibles dans les documents. De cette manière, les informations sémantiques et les documents restent découplés.
- Tout document peut être associé ou lié à une ontologie sans aucune restriction prédéfinie.

Pour faire face aux limitations d'évolutivité les auteurs de [36] ont proposé l'utilisation de plusieurs indexes : (a) les ontologies et les bases de connaissances correspondantes sont analysées et stockées dans des indexes inversés utilisant Lucene[3] ; (b)les documents sont pré-traités et également stockés dans des indexes inversés utilisant Lucene ; (c) les annotations sémantiques sont stockées dans une base de données relationnelles. Pour chaque annotation, une entrée est générée dans la base de données. Cette entrée contient les identifiants de l'entité sémantique correspondante (sens du terme) et du document, ainsi qu'un poids indiquant le degré de pertinence de l'entité sémantique dans le document.

3. http ://lucene.apache.org/core/

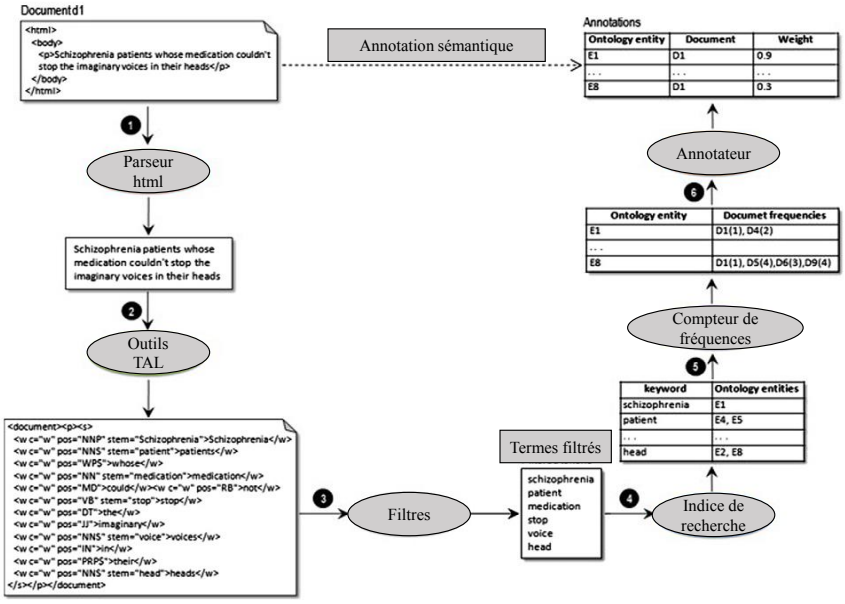


FIGURE 2.1 – Annotation sémantique de documents [36]

Pondération des concepts

Une fois un ensemble de concepts extrait d'un document et une structure d'indexation choisie, il est important de considérer les importances relatives des concepts extraits. Différentes méthodes de pondération de concepts ont donc été proposées. Dans[33] [30], la fréquence d'occurrence dans un document d'un concept c_1 d'une ontologie O dépend de la présence de ses instances (présence lexicale) et de celles de ses hyponymes $c_2 \in \text{Desc}(c_1)$ [101].

Dans [9], la méthode de pondération proposée consiste à exploiter la statistique d'occurrence d'un concept dans une collection. Pour ce faire, un schéma de pondération dit CF-IDF, qui étend la pondération TF-IDF, a été adopté. La fréquence d'un concept c_i dans un document d_j est calculée par la formule suivante :

$$\text{Score}(c_i, d_j) = cf_{d_j}(c_i) \times \ln\left(\frac{N}{df}\right) \tag{2.1}$$

Avec N est le nombre total de documents, *df* (fréquence de documents) est le nombre de documents dans lesquels apparait un concept c_i. $cf_{d_j}(c_j)$ est la fréquence locale cf d'un concept c_i composé de n termes dans un document d_j. Cette fréquence dépend du nombre d'occurrences du concept lui-même et de ses sous-concepts. Si un concept apparaît dans tous les documents, sa fréquence sera 0. Ainsi, chaque document est représenté par un ensemble de concepts pondérés. L'étape suivante consiste à calculer les similarités sémantiques entre les différents sens pouvant être attribués à un seul concept. Il existe une autre adaptation du modèle de pondération TF-IDF dans [33] où la quantité d'information donnée par la présence d'un concept dans un document dépend de la profondeur de ce concept dans le graphe de l'ontologie, du nombre de fois qu'il apparaît dans le document et du nombre de fois qu'il se produit dans toute la collection. Les auteurs de[10] ont également adopté un modèle de pondération CF-ICF qui permet de déterminer approximativement la pertinence d'un concept dans un document. La formule utilisée est la suivante :

$$W_{i,j} = \frac{W_{ij}}{\sum (W_{ij})^2} \tag{2.2}$$

Avec N est le nombre total des concepts d'une ontologie de domaine, $W_{ij} = (1 + \ln(CF_{ij})) \times ICF$, CF (fréquence de concept) est la fréquence d'occurrences de tous les termes qui sont représentés par un concept donné, ICF (l'inverse de la fréquence du concept) est calculé par la formule donnée dans [9].

Nous trouvons une autre adaptation dans [20]et [36] où la formule utilisée est la suivante :

$$d_x = \left(\frac{frec_{x,d}}{max_y \times freq_{y,d}} \right) \times \ln \left(\frac{n}{n_x} \right) \tag{2.3}$$

Avec $frec_{x,d}$ est le nombre d'occurrences des termes attachés à un concept x, $max_y \times freq_{y,d}$ est la fréquence des concepts ayant plus d'occurrences dans

un document d. Le nombre de documents annotés avec x est donné par n_x et D représente la collection des documents. Ainsi, le nombre d'occurrences d'un concept dans un document est essentiellement défini comme le nombre de fois que le label de ce concept apparaît dans le texte du document, 1 si le document est annoté par ce concept et 0 sinon. Dans [84], les annotations de chaque documents sont stockées dans une base de données relationnelle avec leurs poids qui sont calculés par une extension du modèle TF-IDF en utilisant l'équation suivante :

$$\text{tf-idf}_{i,d} = \left(\frac{n_{i,d}}{\sum k(n_{k,d})}\right) \times \log\left(\frac{|D|}{N_i}\right) \tag{2.4}$$

Avec $n_{i,d}$ est le nombre d'occurrences d'une entité ontologique i dans le document d, $\sum k(n_{k,d})$ est la somme d'occurrences de toutes les entités ontologiques dans un document d, $|D|$ est le nombre total de documents et N_i le nombre de documents annotés avec i.

2.3.2 Formulation des requêtes

La formulation des requête représente une étape cruciale pour la qualité des résultats fournis par un SRI basé sur les ontologies. Selon [50] et [51], il existe deux approches pour la formulation des requêtes dans les SRI basés sur les ontologies.

Dans la première approche, l'utilisateur doit saisir des requête formelles (SPARQL, RDQL)[76] [61]. Une telle formulation nécessite une bonne maitrise de la syntaxe utilisée et des connaissances préalables sur la structure de l'ontologie et le schéma de la base de connaissances. Cela entraine une limitation pour cette approche [11].

La deuxième approche consiste à limiter la complexité de la recherche sémantique de l'approche précédente, en la rendant efficace et facile à utiliser. L'objectif est de déduire avec précision les intentions de l'utilisateur à partir des mots-clés pour être en mesure de fournir l'information désirée. La

meilleure caractéristique de la recherche par mots-clés est sa simplicité [62]. Une telle formulation est peut être mise en place en affectant les termes de la requête aux différents éléments de l'ontologie.

L'interprétation de ce type de requête passe par une ou plusieurs étapes [90] : déduction sémantique à partir des termes saisis ; assortiment entre ces termes et l'ontologie ; transformation de la requête en utilisant un langage structuré. Pour élaborer les deux premières étapes, les mêmes approches utilisées lors de la phase d'indexation peuvent être appliquées. Ces étapes dépendent du modèle de recherche (appariement document-requête) utilisé, s'il s'agit d'une représentation vectorielle des concepts [31][88], les deux premières étapes sont suffisantes. La troisième étape consiste à transformer, ré-écrire ou formuler la requête en langage interprétable par la base de connaissances.

Le système proposé par [60] permet à l'utilisateur de saisir sa requête en langage naturel (deux ou plusieurs mots-clés). Les mots-clés saisis sont transformés automatiquement en une requête formelle basée sur le langage SeRQL [4]. Les termes de la requêtes sont ensuite comparés avec des triplets (sujet, prédicat, objet) dans la base de connaissances sémantique. Le système proposé par [97], les termes de la requête sont transformés en un ensemble de requêtes conjonctives en se basant sur la logique de description. Cette transformation est effectuée en trois étapes : les termes sont initialement affectés aux éléments de l'ontologie ; les relations ainsi que les concepts sélectionnés sont représentés par des sous-graphes ; des requêtes formelles sont ensuite générées à partir de ces sous-graphes.

Les deux systèmes intègrent une base de connaissances unique qui oblige l'utilisateur à conceptualiser ses besoins d'information en des termes compatibles avec l'ontologie utilisée. Pour surmonter ce problème, le système proposé par [37] suggère à l'utilisateur de passer à la recherche classique par mots-clés quand aucun élément de l'ontologie ne satisfait les termes de la requête. Une autre approche d'adaptation de mots-clés a été proposée dans [109]. Dans ce travail, un système nommé "SPARQ" est présenté qui trans-

4. http ://www.w3.org/2001/sw/wiki/SeRQL

forme automatiquement les requêtes par mots-clés en des requêtes formelles logiques SPARQL. Comme dans [97], cette transformation se fait en trois étapes : assortiment sémantique (Wordnet) et morphologique (techniques de comparaison de chaines de caractères) entre les termes de la requêtes et les éléments de l'ontologie ; construction des graphes pour chaque requête en utilisant l'algorithme d'arbre couvrant de poids minimal ; utilisation d'un modèle probabiliste pour sélectionner la requête SPARQL la plus appropriée.

Dans l'objectif de masquer la complexité liée à la formulation de requêtes SPARQL pour interroger une base de connaissances (entrepôt de triplets RDF), [23] ont proposé un mécanisme permettant de générer de telles requêtes à partir de mots-clés fournis par l'utilisateur. Le système associe aux mots-clés les concepts et instances correspondantes puis sélectionne un ensemble de patrons de requêtes jugées pertinentes par rapport à ces concepts. Les patrons sont des modèles de requêtes pré-établis traduisant des besoins récurrents en information pour un domaine d'application donné. Les patrons sont modifiés afin de tenir compte des mots-clés proposés par l'utilisateur qui peut alors choisir, grâce à des descriptions en langage naturel, la requête correspondant effectivement le mieux à ses besoins. Une requête SPARQL est finalement générée.

Dans [13], la requête utilisateur est constituée de deux parties indépendantes : une partie sémantique correspondant à une requête SPARQL et une partie mots-clés. Par exemple, il est possible de poser une requête qui permet de rechercher tous les documents qui réfèrent à une instance de *Composant Électronique* et dont le texte contient la chaine de caractères "Résistance". Les résultats de la requête utilisateur considérée dans sa globalité sont les documents obtenus à la fois par les parties mots-clés et sémantiques de la requête. En cas d'absence de réponses à l'une ou l'autre des parties de la requête, les réponses correspondent à la partie mots-clés seule ou à la partie sémantique seule.

2.3.3 Appariement et classement des résultats

Dans la plupart des SRI basés sur les ontologies, il n'y a pas de distinction claire entre le processus d'appariement et le processus de classement [90]. Certains systèmes évaluent simplement l'appariement entre la requête formelle et la base de connaissances et ne donnent aucune description du processus de classement, c'est surtout le cas dans les portails sémantiques où la recherche d'information est basée sur le modèle booléen et donc aucun classement n'est fourni. Ainsi, les résultats de recherche sont souvent précis puisque la notion d'approximativité n'y existe pas. Ce modèle est fiable quand le contenu informationnel peut être totalement représenté par une base de connaissances basée sur les ontologies, ce qui n'est pas souvent trivial. Cependant, différentes approches de classement ont été proposées [90].

Les auteurs de [20]et [36] ont adapté le modèle vectoriel classique pour la recherche par mots-clés basée sur l'ontologie. La requête utilisateur est transformée en une requête formelle SPARQL. Après l'interrogation de la base de connaissances par la nouvelle requête, les résultats récupérés sont utilisés pour générer une représentation vectorielle. Dans le modèle vectoriel adapté, il s'agit d'attribuer des poids aux variables de la clause SELECT de la nouvelle requête. Les poids peuvent être calculés, soit explicitement par l'utilisateur, soit automatiquement par le système en se basant sur la fréquence des entités sémantiques dans les documents ou sur les préférences de l'utilisateur. Une fois que les vecteurs sont créés, la mesure de similarité entre un document d et la nouvelle requête q est calculée par la formule du modèle vectoriel classique :

$$\mathrm{sim}(q, d) = \left(\frac{d \times q}{|d|.|q|} \right) \tag{2.5}$$

Il existe d'autres approches qui reposent sur la structure hiérarchique des concepts d'une ontologie. Les auteurs de[24] définissent la distance d'un document à une requête par la somme des distances de ses concepts à ceux de la requête. Dans les approches qui se rapportent aux graphes pour représenter

les réseaux sémantiques reliant les concepts de la requête et des documents, l'algorithme de diffusion de l'activation est utilisé. Dans cet algorithme, l'activation d'un nœud (concept) représente son importance : plus un nœud est activé plus son classement est élevé. Chaque nœud a une activation initiale qui se propage à ses voisins. L'activation finale est le résultat de la somme de l'activation des nœuds associés multipliée par la force de l'association. Cet algorithme a été utilisé dans [89]. Une structure de réseau à deux niveaux est intégrée, elle consiste en une couche de concepts conjonctifs et la couche de ressources correspondante. Les poids des liens entre les concepts (arcs) sont calculés par une mesure de similarité conceptuelle tandis que les poids des liens entre les documents sont calculés par une mesure de similarité textuelle. L'algorithme de diffusion d'activation est appliqué en deux couches, ce qui peut conduire à la recherche de documents sans annotation sémantique précédente.

2.3.4 Interface utilisateur

Les auteurs de [48] ont fait une étude approfondie sur les interfaces utilisateur dans les SRI. Cependant nous nous concentrerons uniquement sur les caractéristiques des interfaces où l'utilisateur doit saisir des requêtes formelles que ce soit en langage spécifique (SPARQL, RDQL, etc.) ou en langage naturel (mot-clés). Afin de remédier à la complexité de la formulation des requêtes et à la nécessité d'avoir des connaissances préalables sur la base de connaissances, plusieurs approches ont été développées pour assister l'utilisateur dans le processus de formulation de sa requête.

Au delà d'un seul champ de saisie de texte, plusieurs systèmes offrent des fonctionnalités supplémentaires. À titre d'exemple, les nuages de mot-clés qui représentent des concepts/instances d'une ontologie sont utilisés pour visualiser le contenu de la collection de documents. L'idée est d'initier un processus de recherche d'information [102]. En outre, les interfaces à base de formulaire permettent à l'utilisateur de sélectionner explicitement les classes de l'ontologie et entrer les valeurs des relations ou des propriétés [57]. Ce type

d'interface découle de la possibilité de convertir les éléments de l'ontologie sélectionnés en des listes et des formulaires. Les auteurs de [11] proposent une interface à base de formulaire pour saisir les requêtes en langage naturel.

Une autre approche consiste à rechercher l'information graphiquement. Il s'agit d'un nouveau paradigme d'interaction qui permet à l'utilisateur de créer des requêtes graphiques à travers la navigation et la sélection au sein d'une ontologie. Les auteurs de [104] ont proposé un système qui assiste l'utilisateur dans sa formulation des requêtes sémantiques liées à un domaine donné. L'utilisateur commence par saisir une requête par mots-clés. Il sera ensuite guidé automatiquement en suivant un processus incrémental de raffinement afin de déterminer tous les sens possibles de la requête. Pour ce faire, le système calcule toutes les requêtes sémantiques possibles en se basant sur des triplets RDFS. Ces requêtes sont ensuite utilisées pour permettre une navigation à base de graphes pendant le processus de raffinement. Enfin, l'utilisateur choisit la requête sémantique la plus appropriée parmi celles qui ont été générées automatiquement par le système. Si aucune de ces requêtes n'est considérée comme appropriée, l'utilisateur peut orienter le système vers la génération d'une requête appropriée en fournissant des précisions supplémentaires (par exemple, en indiquant si un mot-clé donné doit être conçu comme une propriété ou une entité, etc.).

2.4 Modélisation des préférences utilisateur

Au delà de la pertinence des résultats en réponse à une requête, un SRI doit satisfaire des besoins d'information spécifiques d'un utilisateur. Nous parlons précisément de la personnalisation de la recherche. L'idée est d'intégrer l'utilisateur dans le processus global d'accès à l'information en vue d'adapter les différentes étapes (recherche, filtrage, visualisation) à son contexte. La modélisation de l'utilisateur devient donc une étape indispensable pour établir une stratégique de personnalisation. En général, les informations collectées sur l'utilisateur sont stockées dans un profil qui peut

être représenté par différentes structures, à savoir des vecteurs de termes ou de concepts, des réseaux sémantiques de termes ou de concepts ou bien une combinaison des deux. Même si l'ensemble des approches de personnalisation s'accorde sur l'importance de disposer d'un profil utilisateur, il manque un consensus sur la typologie des connaissances le constituant ainsi que sur la manière de représenter ces connaissances. Pour la majorité des approches, le profil contient la description des centres d'intérêts et préférences de l'utilisateur. Nous trouvons plus de détails sur les sources de données utilisées pour la construction et la mise à jour des profils utilisateurs dans[41] et [32].

La personnalisation dans la RI conceptuelle est un axe de recherche à part entière. Cependant le but de notre étude n'est pas de recenser les différentes approches sémantiques ou ontologiques de la personnalisation ; nous nous focalisons plutôt sur l'étude de quelques aspects sémantiques traités dans la littérature pour l'intégration de l'utilisateur dans la boucle de pertinence à travers ses préférences. Selon [41], nous pouvons distinguer deux catégories d'approches de personnalisation de la recherche en se basant sur les préférences des utilisateurs : approches pour l'adaptation des requêtes ; approches pour l'adaptation des résultats. Notons que cette classification s'avère applicable dans les approches de la RI conceptuelle.

Afin d'adapter la requête saisie par l'utilisateur, plusieurs techniques ont été proposées, notamment :

- Modification de la requête : il s'agit d'étendre les requêtes saisies par un utilisateur, en élargissant le champ de recherche au moyen des nouveaux termes, concepts d'ontologies [1] ou même par des préférences sémantiques [95]. Les techniques qui permettent l'obtention des nouveaux termes peuvent être implicites en se basant sur un modèle d'utilisateur [108] ou bien en se basant sur une source de connaissances externe (thésaurus, ontologies) [9]. Le retour de pertinence, quant à lui, représente une technique explicite qui consiste à prendre en compte les éléments sélectionnés par l'utilisateur à travers ses interactions avec le système. Pour avoir plus de détails sur cet aspect, nous pouvons citer les travaux de [14][19].

- Recommandation des requêtes : il s'agit de proposer des requêtes en fonction des informations collectées explicitement ou implicitement par le système [93]. La recommandation peut être collaborative en prenant en compte des requêtes saisies par les autres utilisateurs.

Quant à l'adaptation des résultats nous pouvons citer les techniques suivantes :

- Re-classement des résultats : il s'agit de re-ordonner les résultats fournis initialement en passant par une étape de calcul des nouveaux scores. Généralement, ces nouveaux scores incluent les degrés de similarité entre les documents et la requête utilisateur d'une part, entre les documents et le modèle utilisateur d'autre part.
- Filtrage de résultats : cette technique peut être considérée comme un cas particulier ou une variante de la technique précédente [41]. Il s'agit d'éliminer des résultats ayant un score inférieur à un certain seuil, au lieu d'afficher tous les résultats re-calculés en fonction du modèle utilisateur.

.

L'utilisation des ontologies pour la modélisation des préférences a fait l'objet de plusieurs travaux. Parmi ceux-ci, nous soulignons les travaux de [100] et [74] dans lesquels l'ontologie représente à la fois la collection de documents et les préférences des utilisateurs (figure 2.2). Ils ont proposé un algorithme d'appariement qui fournit une mesure de pertinence personnelle (PRM) d'un document pour un utilisateur, en fonction de ses préférences sémantiques. La mesure est calculée en fonction des préférences de l'utilisateur et les annotations sémantiques des documents. Ces deux dernières sont représentées par deux vecteurs dans un espace vectoriel de dimension N, où N est le nombre d'éléments dans l'ontologie O. Les préférences d'un utilisateur u sont représentées par un vecteur de poids (entre -1 et 1) correspondant à l'intensité de l'intérêt de l'utilisateur pour chaque concept de l'ontologie, les valeurs négatives étant indicatives d'un désintérêt pour ce concept. Enfin, la PRM est représentée par la similarité algébrique entre les deux vecteurs.

Dans [15], les auteurs ont proposé une stratégie de personnalisation en

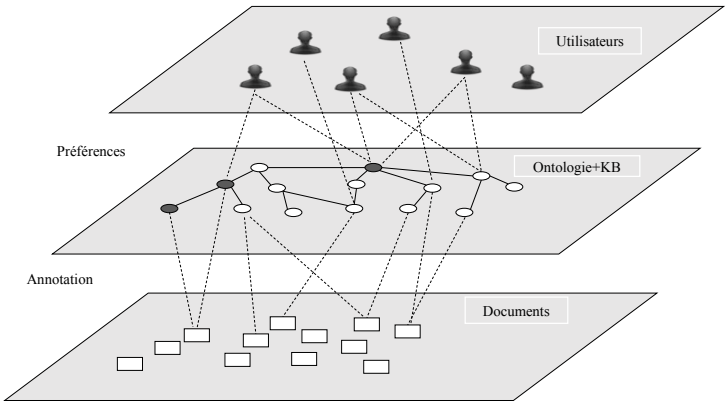


FIGURE 2.2 – Représentation des préférences utilisateur par les ontologies

appliquant des techniques d'inférence issues du Web sémantique afin de déduire des relations sémantiques entre les termes représentant les préférences utilisateur. Ces relations sémantiques apportent des connaissances supplémentaires sur les préférences de l'utilisateur et permettent au système de les comparer avec les documents (programmes TV) de manière plus efficace. La méthodologie d'inférence proposée s'effectue suivant ces étapes :

1. Localiser les préférences de l'utilisateur dans la base de connaissances OWL.
2. Explorer successivement les séquences de propriétés (relation sémantiques) à partir de l'ensemble de programmes TV.
3. Filtrer les instances pertinentes contenues dans les séquences analysées.

En suivant le modèle de RI proposé par [35], les auteurs de [94] ont considéré la pondération des critères de recherche par l'utilisateur, représentés par des concepts de l'ontologie, comme moyen pour déterminer ses préférences.

Ils ont défini une requête conceptuelle pondérée Q comme suit :

$$Q = \{(c_r, w_r), r = 1..n, w_r \in \mathbf{R}, c_r \in C\} \tag{2.6}$$

où les poids w_r associés aux concepts d'une requête sont fournis manuellement par un utilisateur. c_r est un concept sélectionné à partir de l'ensemble de concepts C de l'ontologie. La dimension n correspond au nombre de concepts des requêtes conceptuelles. La sémantique de ce poids est relative à la préférence qu'un utilisateur peut avoir pour un concept plutôt que pour un autre. Quant à la similarité entre un document d_j et une requête q, elle est calculée en suivant ces trois étapes :

1. Calculer les similarités sémantiques entre un concept c_r de la requête Q et chaque concept c_s du document d_j.

$$Q = (\{\pi(c_1, c_s)\}, ..., \{\pi(c_r, c_s)\}, ..., \{\pi(c_n, c_s)\} \tag{2.7}$$

2. Calculer les degrés des performances élémentaires du document d_j par rapport à c_r avec $agreg$ est un opérateur d'agrégation.

$$X_r(d_j, c_r) = \text{agreg}(\pi(c_r, c_s))\text{avec}(c_s, w_s) \in C(d_j) \tag{2.8}$$

3. Déterminer si un document d_j est préféré à un document d_i, $U(d_j) > U(d_i)$ avec A un opérateur d'agrégation.

$$U(c_1, ..., c_n = A(X_1(c_1, d_j), ..., X_r(c_r, d_j), ..., X_n(c_n, d_j)) \tag{2.9}$$

L'intégration de l'utilisateur dans la boucle de pertinence se traduit par la prise en compte des ses préférences ou de ses intérêts lors du processus d'appariement document-requête. La re-classement, en tant qu'approche de personnalisation, se rapporte à l'utilisation d'un modèle utilisateur. Ce modèle sera utilisé lors d'un processus d'appariement qui sera effectué sur deux étapes : appariement (document-requête) appariement modèle utilisateur-requête. Le calcul de similarité peut donc se faire par la formule suivante

[90] :

$$\text{Score}(u, d, q) = \alpha \times \text{score}(d, q) + (1 - \alpha) \times \text{score}(d, q) \tag{2.10}$$

où d et q représentent respectivement un document et une requête, tandis que le modèle d'utilisateur dédié à cette approche peut être représenté par un vecteur de préférences pondérées u. Les similarités sont calculées par la mesure cosinus.

De point de vue modélisation de l'utilisateur, les concepts des ontologies sont utilisés pour représenter sémantiquement les centres d'intérêts et les préférences dans les profils utilisateur. Les auteurs de[80] sont parmi les premiers à utiliser les ontologies de domaine pour modéliser l'utilisateur. Quant à [54], ils ont proposé un modèle utilisateur basé sur une ontologie de domaine pour le re-classement des résultats. Leur modèle de préférences n'est pas seulement représenté par des concepts de l'ontologie mais aussi par des relations sémantiques. Le modèle utilisateur est maintenu au moyen des statistiques ainsi qu'un mécanisme d'inférence appliqué sur les éléments ontologiques. Le re-classement des résultats est effectué après avoir déduit les concepts pertinents, à partir du modèle utilisateur, en réponse à la requête posée. Dans une autre approche proposée par [27], le modèle utilisateur est représenté par un graphe de concepts de l'ontologie ODP [5]. Les concepts, en tant que nœuds du graphe, sont représentés par des arbres hiérarchiques composés par les relations hyponyme-hyperonyme (is-a). Les concepts sont sélectionnés en faisant correspondre les concepts de l'ontologie aux documents qui intéressent l'utilisateur. Le re-classement des résultats est effectuée par une mesure de distance à base de graphe en suivant l'hypothèse selon laquelle "un document est mieux classé s'il contient un maximum de concepts issus du modèle utilisateur."

Par ailleurs, les ontologies ont été combinées avec d'autres méthodes afin de personnaliser les résultats en réponse aux préférences utilisateur. Le raisonnement à partir de cas RàPC, issu de l'intelligence artificielle, est l'une de

5. http ://www.dmoz.org/

ces méthodes qui ont été utilisées dans ce contexte pour plusieurs fins, notamment la re-formulation des requêtes, les systèmes de questions-réponses ou encore pour améliorer la précision des résultats de recherche. Le point fort de la méthode RàPC est sa capacité de résoudre les problèmes en retrouvant des cas analogues (anciennes expériences) dans sa base de cas et en les adaptant au cas considéré. Elle se déroule généralement en cinq étapes : représentation des cas ; recherche de cas similaires ; réutilisation de cas et adaptation ; révision et apprentissage. L'idée de cette combinaison est encore récente, c'est pour cela que nous nous focalisons sur les travaux qui sont étroitement liés avec les travaux de cette thèse.

Dans [10], chaque utilisateur est associé à une base de cas locale qui indexe les documents consultés. Les cas créés seront partagés par les autres utilisateurs. La base de cas est utilisée pour plusieurs finalités à savoir : la re-formulation de nouvelles requêtes sur la base des anciennes requêtes, la proposition des recommandations sous forme de requêtes similaires et leurs résultats (qui partagent le même topique de recherche) à partir de la base de cas, la classification et le filtrage des documents et finalement, la création et l' enrichissement des modules ontologiques. Un cas est le triplet composé d'un problème, d'une solution et d'un score d'évaluation. Le problème est décrit par : le type de but de recherche, le domaine ou le thème de recherche, le concept pivot du module ontologique concerné et la classe des requêtes similaires. La solution est constitués par les résultats pertinents de recherche. En outre, cette approche permet d'enrichir le module ontologique avec de nouveaux concepts ou de nouvelles relations avec d'autres modules en se basant sur l'apprentissage.

Dans [12], les auteurs proposent un système hybride de recherche d'information intégrant la méthode RàPC et la composition d'ontologies. Cette hybridation est traduite par une amélioration à plusieurs niveaux, notamment la re-formulation de requêtes, la prédiction des scores des résultats (classement) et la satisfaction de l'utilisateur. Pour la re-formulation de la requête, l'ontologie modulaire permet d'enrichir la requête utilisateur par les concepts les plus proches de ses termes. Pour cela, la similarité sémantique a

été calculée entre les concepts de l'ontologie et les termes de la requête. Ensuite, les concepts ayant des poids les plus élevés sont utilisés pour enrichir la requête. L'avantage de l'enrichissement de la requête avec l'ontologie modulaire permet d'améliorer le processus de recherche même si les connaissances de l'utilisateur dans le domaine sont limitées.

Ces approches peuvent, d'une part enrichir les ontologies utilisées à travers l'apprentissage, d'autre part améliorer le processus de recherche à travers la formulation, la re-formulation ou la recommandation des requêtes. Toutefois ces approches ne peuvent pas détecter facilement les préférences ou les intérêts des utilisateurs quand il s'agit d'une recherche basée sur plusieurs critères, ce qui pourrait avoir un impact négatif sur la pertinence des résultats. Cela peut s'expliquer par l'inexistence d'un modèle de préférence basé sur des opérateurs d'agrégation. C'est dans ce cadre que nous orientons une partie de nos travaux.

2.5 Conclusion

Dans ce chapitre, nous avons présenté une étude de l'exploitation des ontologies dans les différentes phases de la RI. Cependant un SRI est vu comme un outil qui répond à des requêtes formulées par les concepts et les relations d'une ontologie de domaine en les alignant avec des concepts modélisant les documents cibles. Cette vue ne se concrétise que si les contenus de tous les documents peuvent être représentés par des instances de concepts ou de relations définis dans une ontologie donnée. Des techniques d'apprentissage peuvent être y appliquées pour, d'une part mieux formuler la requête, d'autre part inclure l'utilisateur dans la boucle de pertinence, notamment le raisonnement à partir de cas RàPC. C'est dans ce cadre que nous avons choisi d'orienter nos travaux de thèse. Étant données les critères d'un utilisateur exprimés dans sa requête et une source de données, nous proposons un modèle de préférences associé à l'ontologie. Nous adoptons cette vision dans cette thèse dans l'objectif d'assister l'utilisateur à deux niveaux. Le premier

niveau est la formulation de la requête où il s'agit de structurer l'expression de ses besoins afin que sa requête soit en adéquation avec les sources de données. L'idée est de modéliser la requête en se basant sur les concepts définis dans l'ontologie intégrée. Le deuxième niveau est la recommandation des résultats à partir des anciennes requêtes, au cours de laquelle le système tient compte des informations collectées à partir des interactions précédentes avec l'utilisateur ou profite de l'expérience des autres utilisateurs. Il s'agit d'un processus d'apprentissage réalisé à partir des préférences rendues par les utilisateurs.

Nous présentons l'approche proposée dans le chapitre suivant. Au cours de cette présentation nous détaillons les deux niveaux de l'approche et le modèle de préférences adopté.

Chapitre 3

Une approche de formulation et de recommandation des requêtes pour la RI basée sur les ontologies de domaine

3.1 Introduction

La formulation des requêtes utilisateurs et la prise en compte de leurs préférences ont été reconnues, dans le chapitre précédent, comme étant des facteurs clés de la qualité des résultats fournis par un SRI basé sur les ontologies. Pour traiter ces deux aspects, nous proposons, dans ce chapitre, une approche de formulation et de recommandation des requêtes qui permet une recherche personnalisée. Cette approche repose sur l'utilisation des concepts d'une ontologie de domaine comme étant un langage pivot pour l'expression des requêtes. Ainsi, une requête initialement saisie par l'utilisateur, à travers son interface visuelle à base de formulaire, est formulée par un ensemble de triplets d'entités sémantiques.

Dans l'objectif de s'adapter au mieux aux préférences utilisateur, nous proposons d'une part, de représenter sa requête formulée par des règles sémantiques SWRL et, d'autre part, de lui fournir des résultats de recherche à partir des requêtes recommandées.

– Les règles sémantiques permettent de créer un modèle simple de préférence où il s'agit d'orienter et cibler la recherche après avoir déduit des

nouveaux concepts (concepts topiques) caractérisant l'utilisateur. Ces nouveaux concepts sont utilisés, par le système automatiquement, pour enrichir l'ontologie de domaine utilisée. L'ontologie peut également être enrichie explicitement par des concepts saisis par l'utilisateur à travers son interface.

– La recommandation des requêtes repose sur l'utilisation de la méthode du *raisonnement à partir de cas* (RàPC). Au cours de cette méthode, un nouveau cas émerge lorsqu'une nouvelle requête est considérée comme étant un nouveau problème. La résolution de ce cas consiste à réutiliser les anciens cas qui lui sont similaires et qui ne sont autres que des requêtes qui ont été traitées auparavant. Cette similarité repose sur l'intégrale de Choquet qui consiste à agréger les critères de recherche des requêtes traitées. La recommandation se termine par la validation des résultats de recherche par l'utilisateur.

Les différentes étapes de l'approche proposée sont illustrée par la figure 3.1.

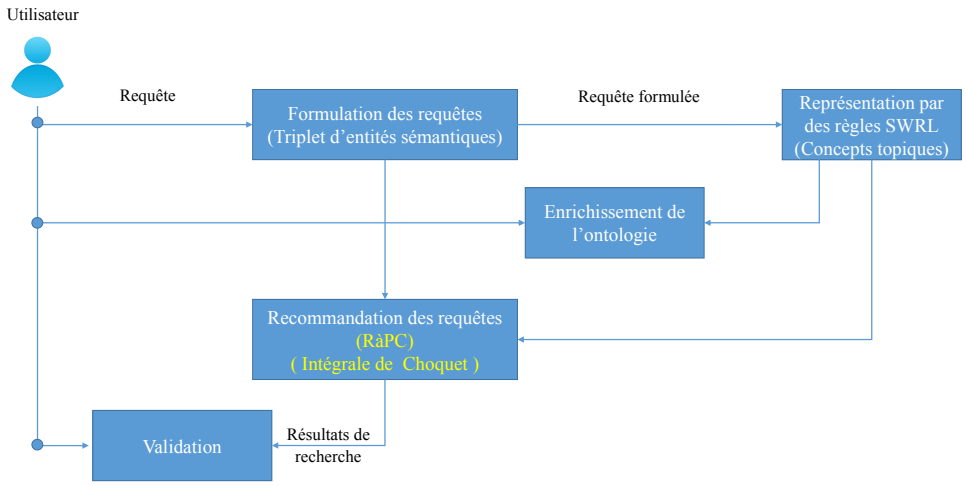


FIGURE 3.1 – Les étapes du processus de recherche dans l'approche proposée

Dans ce chapitre, la première section est consacrée pour la présentation

globale de l'approche. Nous décrivons ensuite les techniques utilisées au cours de la formulation des requêtes dans la deuxième section. La troisième section est consacrée pour l'explication des processus du cycle RàPC à savoir la représentation, l'indexation, la recherche et l'adaptation de cas.

3.2 Présentation globale de l'approche

En suivant les processus de la RI présentés dans le premier Chapitre, l'approche détaillée dans ce troisième Chapitre fait référence à la phase d'interrogation de la collection de documents. Dans la suite, nous supposons prédisposer d'une collection de documents représentés et indexés par les concepts d'une ontologie de domaine. Cette hypothèse s'articule sur la possibilité de retourner à l'utilisateur des résultats pertinents sans avoir forcément passé par tous les processus de la RI. En d'autres termes, avant d'interroger directement la collection de documents (en réponse à la requête utilisateur), il serait possible de retourner les résultats en tenant compte des expériences d'autres utilisateurs.

Pour une meilleure visibilité de l'approche, nous illustrons toutes ses étapes dans la figure 3.2.

En effet, les étapes définies dans les deux niveaux (formulation et recommandation) sont les suivantes :

- (1) Créer la requête à partir des concepts et des propriétés de l'ontologie de domaines en s'appuyant sur les interfaces à base de formulaire.
- (2) Lister les triplets des entités sémantiques constituant la requête.
- (3) Détecter, si possible, le concept topique de la requête saisie en utilisant des règles SWRL.
- (4) Représenter la nouvelle requête sous la forme d'un nouveau cas et rechercher ensuite les cas similaires permettant de recommander des résultats appropriées.

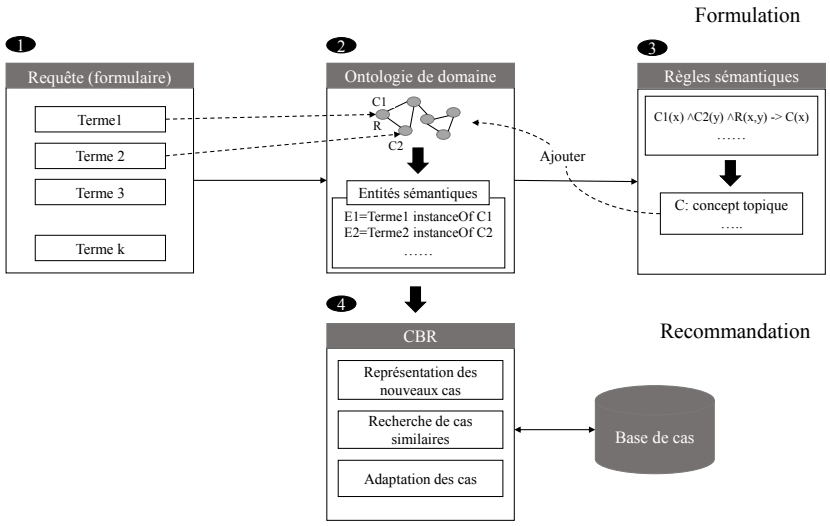


FIGURE 3.2 – Étapes de l'approche proposée

3.3 Formulation des requêtes guidée par les ontologies

3.3.1 Représentation sémantique

Le processus de recherche commence par la formulation des requêtes Nous commençons par formaliser le niveau de la formulation des requêtes.Cette formulation repose sur la modélisation établie par [72]. Dans cette modélisation, une ontologie de domaine est définie, en donnant ses équivalents OWL, comme suit :

A) **Une ontologie de domaine** O^i est définie par le quadruplet (C_{O^i} , R_{O^i}, A_{O^i}, X_O^i) où :

- C_{O^i} est l'ensemble des concepts (i.e. owl : Class).
- R_{O^i} est l'ensemble de relations : propriétés dont le domaine et le co-domaine sont des concepts (i.e owl : ObjectPropoerty).
- A_{O^i} est l'ensemble des attributs : propriétés dont le domaine est un concept et le co-domaine est un littéral (i.e. owl : DatatypeProperty).

- X_O^i est un ensemble d'axiomes définissant les caractéristiques des concepts et des propriétés. Nous listons quelques exemples :
- subProperty(P_1,P_2) indique que P_1 est une spécialisation de P_2 (i.e <P_1 rdfs : subProperty P_2>)

- domain(P,C) indique que le domaine de la propriété P est C (i.e <P rdfs : domain C>).
- range(P,C) indique que le co-domaine de la propriété P est C (i.e <p rdfs : range C>)
- subClass(C_1,C_2) indique que C_1 est un sous-concept de C_2 (i.e <C_1 rdfs : subClass C_2>)
- subProperty(P_1,P_2) indique que P_1 est une spécialisation de P_2 (i.e <P_1 rdfs : subProperty P_2>)

B) **Un ensemble de faits, noté** $T_{O^i}^j$ affirmés par les propriétés (relations et attributs). Ces faits décrivent les membres des concepts et leurs instances de O^i.

- Les relations (owl : ObjectProperty), relient les instances de deux ou plusieurs concepts.
- Les attributs (owl : DatatypeProperty), sont des relations entre une valeur ou donnée et une instance d'un concept. Elles sont appelées littéraux en RDF.

Les requêtes formulées par les éléments de l'ontologie de domaine O^i recherchent des instances de concepts du domaine, vérifiant potentiellement des relations sémantiques (*ObjectProperty*) avec d'autres instances, ou des valeurs d'attributs littéraux (*DatatypeProperty*). Ainsi, nous définissons une requête comme un ensemble des triplets des entités sémantiques. Chaque triplet E est défini par $< domaine, \text{propriété}, co - domaine >$ (tableau 3.1). Notons qu'un triplet de la forme $< s, rdf : type, c >$ indique que s est une instance du concept c.

Une fois toutes les triplets des entités sémantiques ont été listées à partir de la requête saisie par l'utilisateur (à travers son interface à base de formulaire), l'étape suivante consiste à déduire de nouvelles connaissances

—	Relation	Attribut
Domaine	instance	instance
Co-domaine	instance	valeur (littéral)

TABLEAU 3.1 – Structure d'un triplet des entités sémantiques

en se basant sur la conjonction de ces entités, il s'agit plus précisément de l'application des règles sémantiques SWRL.

3.3.2 Enrichissement de l'ontologie

Cette étape a pour but d'inférer les préférences utilisateurs. Elle considère certains concepts utilisés dans la requête comme critères de recherche qui doivent être satisfaits. L'existence de ces concepts définit ainsi les critères de recherche dans l'ontologie de domaine utilisée. La conjonction des entités contenant ces critères de recherche permet d'identifier la catégorie à laquelle appartient un utilisateur donné. Cette catégorie sera classée dans une taxonomie des nouveaux concepts que nous appelons : *des concepts topiques* C_t.

Une règle sémantique donnée peut être formulée comme suit :

$$C_t(x) \longleftarrow \wedge_1^n \quad \exists y_j, \exists R \quad \theta_i(x, y_j) \text{avec } j \in \{1, \ldots, n\} \tag{3.1}$$

avec :

- C_t : concept topique.
- $\wedge$: fonction de conjonction des entités contenant les critères de recherche
- y_j : les instances des concepts représentant les critères de recherche.
- x : l'instance du concept avec lequel sont liés les concepts représentant les critères de recherche par la relation R.
- θ_i : l'entité ayant comme domaine x et co-domaine y_i.
- $R : C \times C_j$: deux classes C et C_j, ayant comme instances x et y_j et

une relation R.

Au vu de sa complexité, l'enrichissement d'une ontologie existante est un axe de recherche à part entière. Selon [78], il n'existe que des approches semi-automatiques qui nécessitent l'intervention des experts du domaine pour valider l'ajout des nouveaux éléments et pour vérifier la consistance de l'ontologie enrichie. En générale, la tâche d'enrichissement se traduit par (1) un premier processus de fouille de données à partir d'un corpus qui consiste à sélectionner des termes pouvant refléter des nouveaux concepts ou des nouvelles relations dans l'ontologie en question. Un deuxième processus consiste à placer les nouveaux éléments dans l'ontologie, et ce, en utilisant différentes techniques, à savoir des heuristiques, des algorithmes d'apprentissage, des algorithmes d'analyse syntaxiques et bien d'autres. Dans notre travail, nous ne traitons pas les processus d'extraction et de sélection des nouveaux concepts. Par contre, notre approche est capable de détecter les nouveaux concepts topiques par le biais des règles sémantiques SWRL comme nous l'avons expliqué précédemment. De même, l'utilisateur peut introduire de nouveaux concepts à travers l'interface dédiée aux requêtes. Nous illustrerons ce point, par un exemple, dans le chapitre suivant. Par la suite, il reste à placer les nouveaux concepts dans l'ontologie existante. Pour ce faire, nous adoptons deux démarches :

- Concepts topiques : ils sont placés automatiquement par le système sous la forme d'une nouvelle taxonomie. Les concepts de cette taxonomie sont liés avec l'ancienne ontologie par la relation *has_case*.
- Autres concepts : ils sont également placés automatiquement par le système. Il s'agit principalement de rajouter une relation de subsomption *is_a* avec les concepts sélectionnés (concepts pères) dans l'interface. En effet, les autres relations sont héritées des concepts pères. Ajouter une relation similaire aux fils (nouveaux concept) et aux pères (anciens concepts) n'a donc aucun sens car cela entraine une redondance d'information.

Ces deux démarches constituent d'une technique qui permet de placer de nouveaux concepts (concepts topiques) dans l'ontologie et d'ajouter de nouvelles relations entre ces concepts. Ces relations sont principalement des re-

lations de subsomption (*est-un*) formant ainsi une nouvelle taxonomie. Nous illustrons par la suite la génération des concepts topiques par un exemple (figure 3.3). Dans cette figure nous considérons :

- Une liste de concepts $(C_1, C_2, C_3, C_4, C_5, C_6, C_7)$ avec (C_4, C_5, C_6) sont des concepts représentant les critères de recherche.
- Une liste des instances $(X_1, X_2, Y_1, X_2, X_3)$.
- Une liste des valeurs d'attributs(T_1, T_2).
- Une relation R.

Supposons que nous avons de deux requêtes qui correspondent à deux utilisateurs différents dont la première contient les concepts C_2, C_6 tandis que la deuxième contient les concepts (C_3, C_5, C_7). Comme le montre la figure (figure 3.3) , la première étape consiste à lister toutes les entités sémantiques, y compris celles qui contiennent les critères de recherche. La deuxième étape se traduit par l'application des deux règles sémantiques possibles. Ces règles consistent à générer deux concepts topiques C_{t1} et C_{t2} qui, à leur tour, permettent de catégoriser les deux utilisateurs suivant leurs critères de recherches. Afin de les réutiliser dans des prochaines recherches, les nouveaux concepts topiques seront ajoutés à l'ontologie sous la forme d'une taxonomie (relations de subsomption).

Une fois la requête est formulé et représentée par une règles sémantique, la processus de recherche se déclenche et les résultats de recherche sont fournis à partir des requêtes recommandées. Nous présentons dans le section suivante le deuxième niveau de l'approche proposée, la recommandation des requêtes.

3.4 Recommandation des requêtes fondée sur la méthode RàPC

Cette étape consiste à formuler des recommandations fondées sur les connaissances et les expériences des requêtes qui ont été traitées auparavant. Pour ce faire, notre choix s'est porté sur l'utilisation de la méthode

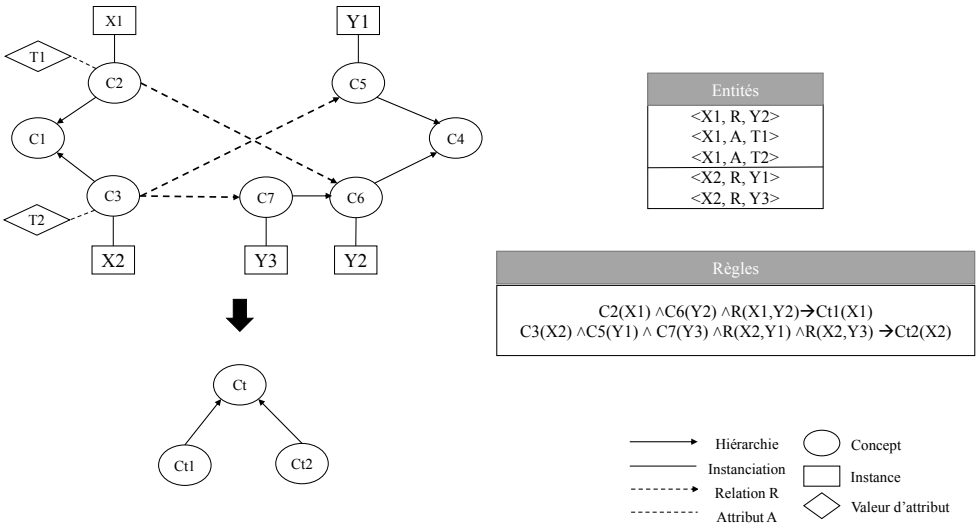


FIGURE 3.3 – Génération des concepts topiques

du raisonnement à partir de cas (RàPC). Ainsi, nous considérons chaque nouvelle requête introduite comme un nouveau problème formant ainsi un nouveau cas. La solution à ce nouveau cas est construite en raisonnant sur les anciens cas similaires. En effet, notre approche ne prend pas seulement en compte la similarité en termes de contenu mais aussi la similarité entre les préférences d'un utilisateur et celles d'un autre. Nous détaillons par la suite les différentes phases de la méthode utilisée, à savoir la représentation, l'indexation, la recherche et l'adaptation de cas.

3.4.1 Représentation de cas

Un cas étant traité par un raisonnement, il constitue une unité de connaissances, et non pas une simple donnée. Cependant, à l'instar d'une requête formulée par l'ontologie de domaine O^i , notre représentation de cas repose sur le même modèle de connaissances. Par conséquent, les règles sémantiques SWRL définies auparavant sont utilisées afin de réduire la taille de la base de cas en des sous-ensembles qui ne comprennent que des cas ayant des structures appropriées. Notons qu'une base de cas CB est un ensemble fini de cas,

généralement muni d'une structure. Dans notre approche, un cas est défini par le n-uplet $< P, S >$ où P désigne un problème et S désigne la solution. En effet, résoudre un problème P, c'est trouver (ou construire) sa solution S.

- Le problème P : est défini par un ensemble d'attributs qui correspondent aux concepts et propriétés de l'ontologie O^i constituant la requête. Un problème P est également caractérisé par le concept topique C_t déduit à partir de la règle sémantique utilisée lors de la phase de la formulation de la requête.
- La solution S : désigne un ensemble de réponses à des anciennes requêtes qui réfèrent aux cas similaires. En d'autres termes, si l'objectif de recherche de l'utilisateur est de rechercher dans une collection de documents, la solution est un ensemble de requêtes similaires avec leurs documents sélectionnées. Si l'utilisateur recherche une réponse plus précise qui peut être un ensemble des entités sémantiques (instance ou un concept de l'ontologie O^i), la solution est alors un ensemble de requêtes similaires avec leurs réponses sélectionnées. La représentation de la solution repose sur deux vecteurs *textuel* et *numérique* dont le premier désigne les concepts ayant des instances textuelles ou alphanumériques et le deuxième désigne des concepts ayant des instances numériques. Ce dernier désigne, entre autres, les concepts modélisant les critères de recherche. Nous donnons plus de détails sur ces deux vecteurs dans la section suivante. Dans la suite de la thèse nous désignons un nouveau cas par C^{new} et un cas en mémoire (ancien cas) par C^{mem}.

Nous illustrons par l'exemple suivant la représentation des cas par l'ontologie de domaine suivante (figure 3.4) :

Exemple. Soit la base de cas CB suivante :

- Cas 1 : $C_1(?x_1) \wedge C_3(?y_1) \wedge C_4(?z_1) \wedge R_1(?x_1, ?z_1) \wedge R_2(?x_1, ?y_1) \rightarrow C_{t1}(?x_1)$
- Cas 2 : $C_2(?x) \wedge C_5(?k_1) \wedge C_4(?z_2) \wedge R(?x, ?k_1) \wedge R_1(?x, ?z_2) \rightarrow C_{t3}(?x)$
- Cas 3 : $C_1(?x_2) \wedge C_3(?y_2) \wedge R_2(?x_2, ?y_2) \rightarrow C_{t3}(?x_2)$

Soit le nouveau cas suivant :

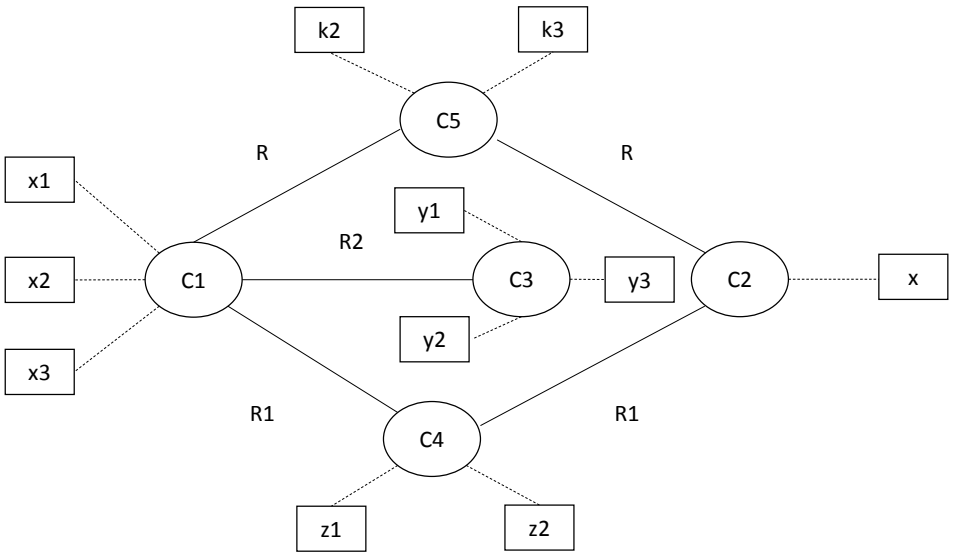


FIGURE 3.4 – Exemple d'ontologie de domaine utilisée

– $C_1(?x_3) \wedge C_3(?y_3) \wedge C_5(?k_3) \wedge R(?x_3, ?k_3) \wedge R_2(?x_3, ?y_3) \rightarrow C_{t1}(?x_3)$

Dans cet exemple nous avons les concepts $(C_1, C_2, C_3, C_4, C_5)$ dont (C_3, C_5) sont des concepts qui représentent les critères de recherche. Les ensembles des instances de chaque concept sont respectivement (x_1, x_2, x_3), (x), (y_1, y_2, y_3), (z_1, z_2) et (k_2, k_3). Les instances textuelles $(x_1, x_2, x_3, x, z_1, z_2)$ sont composées par les termes suivants :

– $x_1 = (terme_1, terme_2)$.
– $x_2 = (terme_1)$.
– $x_3 = (terme_1, terme_5)$.
– $x = (terme_2, terme_3)$.
– $z_1 = (terme_6, terme_{10})$.
– $z_2 = (terme_{10}, terme_7)$.

Les instances numériques $(y_1, y_2, y_3, k_2, k_3)$ sont quantifiées par les valeurs suivantes : $y_1 = (ch_1, val_1)$, $y_2 = (ch_1, val_2)$, $y_3 = (ch_1, val_3)$, $k_1 = (ch_2, val_4)$ et $k_2 = (ch_2, val_5)$. ch_i représente le contenu textuel de l'instance. L'idée

derrière la quantification de ces instances découle de la nature des critères de recherche dans la requête qui sont souvent définis par un ensemble de choix présenté par le SRI. En outre, les relations utilisées sont (R, R_1, R_2) avec (R, R_2) ont comme co-domaines des concepts représentant les critères de recherche. Les concepts topiques générés sont (C_{t1}, C_{t3}). Nous pouvons remarquer que le premier et le troisième cas ont le même concept topique que le nouveau cas.

3.4.2 Indexation de cas

Notre schéma d'indexation est basé sur l'analogie entre la recherche dans une collection de documents textuels et la recherche dans une base de cas. La structure d'indexation adoptée est basée sur l'indice inversé. Ce dernier est composé d'une liste d'entités sémantiques, que nous appelons le vocabulaire. Pour chaque entité e_i dans le vocabulaire, l'indice contient une liste inversée, qui enregistre un identifiant pour tous les cas dans lesquels l'entité e_i existe. Nous montrons par la suite la procédure qui consiste à calculer le poids de chaque entité.

3.4.3 Recherche de cas

Il existe plusieurs méthodes de recherche de cas notamment, la méthode des k-plus proche voisins, la méthode inductive, la méthode guidée par les connaissances ou bien une combinaison de ces méthodes. Cependant, la méthode des k-plus proche voisins s'avère la plus adéquate avec notre démarche de recommandation. Par analogie, la recherche de cas consiste à sélectionner les cas en mémoire ayant des problèmes similaires ou proches de celui du nouveau cas. Pour mener à bien ce processus, il doit y avoir une technique d'appariement permettant de déterminer le degré de similarité d'un cas en mémoire C^{mem} par rapport à un nouveau cas C^{new} . Souvent, la similarité est modélisée par une ou des mesures qui permettent d'exprimer :

– Que deux problèmes P_1 et P_2 sont similaires suivant un seuil de simi-

larité prédéfini.

– Qu'un problème P_1 est plus similaire qu'un problème P_2 par rapport à un nouveau cas.

Dans notre approche, nous traitons la similarité en tenant compte de deux types de concepts, à savoir les concepts constituant les éléments de base de la requête et les concepts représentant les critères de recherche. L'origine de cette distinction découle de la nécessité d'affecter des poids pour chaque type de concept. En effet, l'objectif de cette pondération est double. D'une part, pour calculer la similarité entre les instances (textuelles et numériques) dans les deux cas C^{new} et C^{mem}, d'autre part pour prendre en compte l'agrégation des critères de recherche de ces deux cas. De cette manière, la similarité entre les cas est traitée en deux niveaux, *contenu* et *préférences* utilisateurs.

La notion d'agrégation découle de la nature du problème de recherche de cas similaires, auquel nous sommes confrontés. Plus précisément, il s'agit d'un problème de prise de décision multicritère où il faut trouver un consensus sur le classement d'un ensemble de cas selon un ensemble de critères en réponse à un nouveau cas (requête utilisateur). L'agrégation est la combinaison des scores partiels des cas, obtenus sur chaque concept représentant un critère de recherche. Ceci est effectué par le biais d'un opérateur d'agrégation approprié.

Avant de passer à la section suivante, nous illustrons par cet exemple [71] la motivation qui a été à l'origine de notre choix de l'opérateur d'agrégation :
Exemple. Nous considérons un problème de prise de décision multicritère avec deux concepts critères de recherche (C_1, C_2) et Ca_1, Ca_2 et Ca_3 ayant les scores partiels suivants (scores saisis initialement par l'utilisateur dans sa requête) :

– $c_1(Ca_1) = 0.3, C_1(Ca_2) = 0, C_1(Ca_3) = 1$

– $C_2(Ca_1) = 0.3, C_1(Ca_2) = 1, C_1(Ca_3) = 0$

Supposons que Ca_1 est préférée à Ca_2 et Ca_3 ($Ca_1 \succ Ca_2 \sim Ca_3$). Pour modéliser cette préférence, il serait nécessaire d'utiliser un opérateur d'agrégation qui permet de trouver les poids de préférence w_1 w_2 de C_1 et C_2

respectivement :

- $Ca_2 \sim Ca_3 \Leftrightarrow w_1(C_1(Ca_2)+w_2(C_2(Ca_2)) = w_1(C_1(Ca_3)+w_2(C_2(Ca_3)) \Rightarrow w_1 = w_2$
- $Ca_1 \succ Ca_2 \Leftrightarrow w_1(C_1(Ca_1)+w_2(C_2(Ca_1)) > w_1(C_1(Ca_2)+w_2(C_2(Ca_2)) \Rightarrow 0.3 \times (w_1 + w_2)$

Par conséquent, nous obtenons : $0.6 \times w_2 > w_2$, ce qui est impossible. Ainsi, nous pouvons remarquer la limite de cet opérateur qui utilise la combinaison linéaire des scores pondérés par les poids des critères. A titre d'exemple, nous pouvons citer, la méthode d'agrégation analytique *AHP* [4]. Bien qu'il existe des approches qui permettent de pallier ce problème en utilisant un opérateur de priorité sur des critères individuels ($c_i \succ c_j \succ c_k$) à l'instar de [**?**], cet opérateur atteint sa limite dès qu'il s'agit de considérer des sous-ensembles de critères ($\{c_i, c_j\} \succ \{c_k, c_l\}$). C'est dans cette optique que, nous proposons de traiter le problème d'agrégation à l'aide de l'intégrale de Choquet [21]. Cet opérateur permet de définir des poids d'importance non seulement sur des critères uniques mais aussi sur des combinaisons de critères en se basant sur une mesure floue μ tel que ($\mu\{c_i, c_j\} \succ \mu\{c_k, c_l\}$) où $\mu\{.\}$ représente le degré d'importance d'un critère ou un sous ensemble de critères.

Nous décrivons maintenant l'algorithme de recherche de cas proposé (figure 3.5). Une fois le processus de recherche est déclenché, l'algorithme commence par rechercher dans la base de cas ceux qui ont le même concept topique C_t ou, en d'autres termes, ceux qui ont la même structure des triplets d'entités sémantiques. S'il existe un cas C^{mem} qui satisfait ce premier niveau de similarité, que nous appelons similarité entre concepts, l'algorithme calcule ensuite la similarité entre les instances de ces concepts. Deux mesures de similarité sont proposées à ce niveau :

- Mesure de similarité textuelle basée sur une extension de la mesure TF-IDF , elle est utilisée pour déterminer le degré d'appariement entre deux cas en termes de contenu.
- Mesure de similarité numérique basée sur l'intégrale de Choquet [21], elle est utilisée pour calculer la similarité entre deux cas en termes de préférences des utilisateurs.

La mesure de similarité textuelle est utilisée pour déterminer le degré d'appariement entre deux cas en termes de contenu, tandis que la mesure de similarité numérique repose sur deux étapes : la première consiste à calculer la similarité numérique locale entre deux instances des deux cas C^{new} et C^{mem}, la deuxième étape consiste à calculer la similarité numérique globale de toutes les instances numériques de ces deux cas. Enfin, l'algorithme calcule la similarité globale entre C^{new} et C^{mem} qui est la somme des mesures utilisées dans les deux niveaux multipliée par un coefficient de normalisation.

Dans le cas où l'algorithme ne trouve pas des cas en mémoire ayant le même concept topique C_t que le nouveau cas C^{new}, il calcule directement la similarité entre les instances textuelles des deux cas et sélectionne ensuite les k-cas proches suivant un seuil prédéfini.

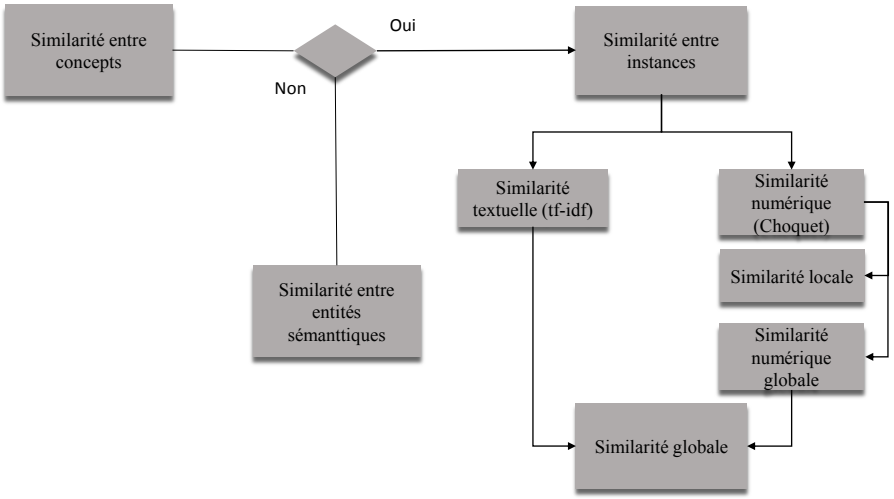


FIGURE 3.5 – Les différentes mesures de similarités

Avant de passer à l'explication de chaque mesure de similarité, nous présentons quelques notations et définitions qui seront utilisées dans les mesures de similarités.

Notation 1. Les instances textuelles (alphanumérique) : sont représentées par un vecteur T. Ceci est le n-uplet $((w_1 I_1^t)), (w_2 I_2^t), \ldots, (w_n I_n^t))$ avec $I_i^t \in O^i$ et $i \in \{1, \ldots, n\}$. w_i représente un poids calculé par la méthode TF-IDF.

Notation 2. Les instances numériques : sont représentées par un vecteur N de scores. $(f_1^{I^N}, f_2^{I^N}, \ldots, f_m^{I^N})$ avec $j \in \{1, \ldots, m\}$ et $f_m^{I^N}$ représente un score initialement défini par l'utilisateur.

Soient les définitions suivantes :

Définition 1. Soit $\mathcal{C} = \{I^N{}_1, I_2^N, \ldots, I_n^N\}$ est un ensemble des instances de concepts représentant les critères de recherche et $N_\mathcal{C}$ est un ensemble de tous les sous-ensembles de ces critères. Une capacité (mesure floue) et une fonction μ de de mesure floue sur $N_\mathcal{C}$ dans $[0, 1]$ vérifiant $\mu(\emptyset) = 0$, $\mu(I_\mathcal{C}) = 0$ et $\mu(A) \leqslant \mu(B)$ si $A \subseteq B$ (monotonie). Cette condition de monotonie provient du fait que l'importance d'un sous-ensemble de critères ne peut décroître si nous ajoutons un critère à ce sous-ensemble.

Définition 2. Soient CB la base de cas et $Ca_i \in CB$. Le score global de Ca_i, donné par l'intégrale de Choquet selon une capacité μ et un ensemble $\mathcal{C}$ de critères de pertinence, est défini par :

$$C_\mu(f) = \sum_{i=1}^{n} [f(\sigma t(i)) - f(\sigma(i-1))].\mu(A_i)$$

$$= \sum_{i=1}^{n} [f(\mu(A_i)) - f(\mu(A_{i+1}))].\mu(f(\sigma(i))) \tag{3.2}$$

avec $A_i = \{\sigma(i) \ldots \sigma(n)\}$ peut être interprété comme le degré d'importance de la combinaison A_i d'un ensemble de concepts critères de recherche, $f : \mathcal{C} \to \Re$ une fonction représentant les scores d'un cas Ca_i sur les n concepts

critères. Dans notre cas, ces sores sont initialement attribués par l'utilisateur lors de la saisie de sa requête. $\sigma(i)$ le score obtenu selon un concept critère donné σ est une permutation sur $\mathcal{C}$ tel que $f(\sigma(1)) \leq f(\sigma(2)) \leq \ldots f(\sigma(n))$.

3.4.4 Mesures de similarités de cas

Dans cette section, nous présentons les mesures de similarités utilisées lors du processus de recherche de cas. La valeur de similarité entre C^{new} et C^{mem} est comprise entre 0 et 1.

Similarité textuelle. Dans chaque cas C, le poids d'une instance textuelle est calculé par une extension de la mesure TF-IDF proposée par [36]. Cette mesure est basée sur la fréquence d'une instance I_i^t dans chaque cas. Le poids w_i de I_i^t est calculé comme suit :

$$w_i = \frac{frec_{I_i^t,C}}{max_J \times freq_{J,C}} \times \log \frac{N_c}{n_{I_i^t}} \tag{3.3}$$

Avec $frec_{I_i^t,C}$ est le nombre d'occurrences des termes attachés à l'instance I_i^t ,$max_J \times freq_{J,d}$ est la fréquence de l'instance ayant plus d'occurrences dans un cas C. $n_{I_i^t}$ est le nombre de cas annotés avec x est donné par I_i^t et N_c représente le nombre total de cas. La similarité textuelle entre C^{new} et C^{mem} est calculée par la mesure de cosinus entre les vecteurs :

$$\text{sim}_T(T^{new}, T^{mem}) = \frac{T^{new} \times T^{mem}}{|T^{new}| \,.\, |T^{mem}|} \tag{3.4}$$

Avec T^{new} et T^{mem} représentent respectivement les instances textuelles des C^{new} et C^{mem}.

Similarité numérique. Comme nous l'avons mentionné précédemment, l'objectif de cette similarité est de chercher les cas les plus proches en termes de préférences. La similarité numérique entre C^{new} et C^{mem} est calculée en deux étapes, à savoir similarité numérique locale et similarité numérique glo-

bale. La première étape calculer la similarité locale entre C^{new} et C^{mem} par rapport à un concept critère j (ayant comme instance I_j^N) comme suit [16] :

$$\text{sim}(N_i, N_{new}) = 1 - \frac{|N_i - N_n ew|}{\text{range}} \tag{3.5}$$

Avec $j \in \{1, \ldots, m\}$, N_i, N_i est la valeur du score partiel du C_i^{mem} selon le concept critère j avec $i \in \{1, \ldots, n\}$, N_{new} est la valeur du score partiel du C^{new} selon le concept critère j, *range* est la valeur absolue de la différence entre la borne supérieure et la borne inférieure de l'ensemble des valeurs des scores de tous les cas selon le critère j. Ensuite, une similarité globale sera calculée en agrégeant les similarités locales grâce à la formule de l'intégrale de Choquet 2-additive suivante :

$$\begin{aligned}\text{Sim}_{glob}(C^{new}, C^{mem}) = \sum_{i=1}^{n} [&\text{sim}_{\sigma(j)}(C^{mem}, C^{new}) \\ &- sim_{\sigma}(j-1)(C^{mem}, C^{new})].\mu(A_j)\end{aligned} \tag{3.6}$$

Dans l'objectif de calculer la capacité (mesure floue) μ des différents sous-ensembles de concepts critères, nous avons utilisé le programme quadratique suivant [16] :

$$Minimize \sum_{a \in CB} \left[C_{\mu}^{mem}(a) - C_{\mu}^{new}(a)\right]^2 \tag{3.7}$$

Avec $A_i = \{\sigma(j) \ldots \sigma(m)\}$, $\text{sim}_{\sigma(0)}(C^{mem}, C^{new}) = 0$ et σ est une permutation sur N tel que $\text{sim}_{\sigma(1)} \leq \text{sim}_{\sigma(2)} \leq \ldots \text{sim}_{\sigma(m)}$.

En outre, les contraintes qui doivent être prises en compte sont les suivantes :

- $\mu C_i \leq \mu C_j \forall C_a \subset CB$
- $\mu(i,j) \leq \text{or} \geq \mu(i) + \mu(j)$
- $C_{\mu}^{mem}(a) - C_{\mu}^{new}(a') \geq \delta \forall a, a' \in CB$

Avec CB est la base de cas, $C_{\mu}^{new}(a)$ est le score globale du nouveau cas C^{new}, C_i et C_j sont des sous-ensembles de concepts critères dans k, i et j, relatifs

aux concepts critères dans N, δ est un degré de différence fixé et a, a' sont des cas dans CB.

Similarité globale. Une fois les deux mesures textuelles et numériques sont calculées, il reste à calculer la similarité globale suivante :

$$\text{Sim}_{glob}(C^{new}, C^{mem}) = (\text{sim}_T + \text{sim}_N) \times \alpha \text{ avec } \alpha \in [0, 1] \tag{3.8}$$

Avec α un coefficient de normalisation.

En se basant sur ces mesures de similarité, l'algorithme de recherche de cas est le suivant(algorithme 3.1)

Nous avons présenté les mesures de similarité utilisées quand il s'agit d'une intersection entre deux cas (en termes de concepts topiques). Il reste une dernière mesure de similarité à utiliser lorsque l'algorithme ne trouve pas de cas en mémoire disposant du même concept topique C_t que le nouveau cas C^{new}. Nous parlons de la similarité entre entités qui consiste à chercher le cas le plus proche en terme d'entités. La formule utilisée pour cette similarité n'est qu'une généralisation de la similarité textuelle 3.3 mais en remplaçant l'instance par une entité qui peut être un concept, une instance ou bien une propriété.

$$w_e = \frac{frec_{e_i,C}}{max_J \times freq_{J,C}} \times \log \frac{N_c}{n_{e_i}} \tag{3.9}$$

Avec $frec_{e_i,C}$ est le nombre d'occurrences des termes attachés à l'entité e_i ,$max_J \times freq_{J,d}$ est la fréquence de l'entité ayant plus d'occurrences dans un cas C. n_{e_i} est le nombre de cas annotés avec e_i est donné par e_i et N_c représente le nombre total de cas. La similarité entre C^{new} et C^{mem} est calculée par la mesure de cosinus entre les vecteurs :

$$\text{sim}(V^{new}, V^{mem}) = \frac{V^{new} \times V^{mem}}{|V^{new}| \,.\, |V^{mem}|} \tag{3.10}$$

Avec V^{new} et V^{mem} représentent respectivement les entités du C^{new} et C^{mem}.

Algorithme 3.1 : Algorithme de recherche de cas

Fonction Meilleur_voisin_améliorant(x)
Entrées : Un nouveau cas C^{new}
Sorties : Degré de similarité d'un nouveau cas avec la base de cas
Un cas en mémoire C^{mem} est décrit comme suit :
n Concepts : $C_1^{mem}, C_2^{mem}, \ldots, C_q^{mem}$
tel que : $C_q^{mem}(q = 1, \ldots, n) \in O^c$
Type d'instances : {Contenu (instance textuelle) , préférence (instance numérique)}
Le poids W est le poids des instances textuelles (TF-IDF)
3.3 Le score f est le score partiel C^{mem} par rapport aux instances numériques
begin
  Comparer les concepts de C^{new} et de C^{mem}
  pour *chaque* C_p^{new} *de* C^{mem} **faire**
    Comparer les valeurs des instances de C_p^{new} et de C_q^{mem} :
    si *Type d'instance == Contenu* **alors**
      (
    finsi
    Calculer 3.4) **si** *Type d'instance == préférence* **alors**
      (
    finsi
    Calculer 3.6) Calculer le degré de similarité du cas recherché par rapport à (3.8)
  fin
end

3.4.5 Adaptation de cas

Ce processus permet la construction d'une solution au problème actuel du C^{new} en modifiant localement une ou plusieurs sources de solutions C^{mem} de celles qui ont été conservées au cours de l'étape d'extraction et dont la similarité est supérieure à un seuil prescrit donné. Nous utilisons l'adaptation comme une tâche de recherche dans l'espace des solutions. L'état initial est la solution d'un cas en mémoire C^{mem} récupéré et l'état final est une solution pour le nouveau cas C^{new}. Cette recherche est effectuée par l'application des opérateurs d'adaptation, qui sont des transformations réalisées dans l'espace des solutions. Plusieurs types d'opérateurs d'adaptation sont utilisés dans la littérature pour modifier la source de solution :

- Opérateur de copie qui n'effectue aucune transformation, il copie seulement la solution de C^{mem} dans la solution de C^{new}. Nous utilisons cet opérateur quand il s'agit d'une similarité entre deux cas supérieure à un certain seuil τ(ayant le même concept topique) C_t. Cela s'explique par le fait que les deux cas ont des structures semblables, ce qui ne nécessite pas vraiment une transformation ou une modification (algorithme 3.2).
- Opérateur de substitution qui établit des modifications dans la solution du C^{mem} en ajoutant, supprimant ou en remplaçant un certain nombre d'éléments. Nous utilisons cet opérateur quand la similarité entre C^{new} et C^{mem} est supérieure à un certain seuil ς (n'ayant pas le même concept topique C_t)(algorithme 3.3).

Algorithme 3.2 : Algorithme d'adaptation de cas par copie

Entrées : Un nouveau cas C^{new}
Une solution sol^{new}
Un cas en mémoire C^{mem}
Une solution sol^{mem}
Sorties : Cas adapté
begin
  si $sim\ ((C^{new}, C^{mem}) > \tau)$ **alors**
    $sol^{new} \leftarrow sol^{mem}$
  finsi
end

Algorithme 3.3 : Algorithme d'adaptation de cas par substitution

Entrées : Un nouveau cas C^{new}
Un cas en mémoire C^{mem}
Sorties : Cas adapté
begin
  si *sim* $(C^{new}, C^{mem}) > \varsigma$ *et* $(C^{new} \subset C^{mem})$ **alors**
    pour *chaque* $e_i \in C^{mem}$ **faire**
      si $(e_i \notin C^{new})$ **alors**
        C^{new}. ajouter(e_i)
      finsi
    fin
  finsi
end

3.4.6 Insertion de cas

Après avoir saisi une requête, si des cas similaires existent et leurs solutions sont validées par l'utilisateur, une étape d'insertion de ce nouveau cas est mise en œuvre.En effet il s'agit d'insérer les attributs du nouveau problème (nouveau cas) qui correspondent aux concepts et propriétés de l'ontologie O^i constituant la requête. Les attributs insérés sont structurés suivant les règles sémantiques qui caractérisent les problèmes des cas en mémoire similaires. Ensuite les solutions sélectionnées sont également insérées. Si aucun cas similaire n'existe dans la base de cas, un nouveau cas est inséré sur la base d'une nouvelle règle.

3.5 Conclusion

Dans ce chapitre, nous avons décrit les niveaux de l'approche proposée dans les travaux de cette thèse, à savoir la formulation et la recommandation des requêtes. L'utilisation de l'ontologie de domaine comme un modèle de connaissances se rapporte, principalement, à la représentation sémantique du contenu informationnel. Cela s'explique d'une, part par la formulation

des requêtes utilisateurs par des entités sémantiques, d'autre part par la structuration de la base de cas dans le processus de la recommandation des requêtes. Nous avons démontré qu'à l'aide des règles sémantiques, l'ontologie de domaine utilisée peut être enrichie par des nouveaux concepts. Cet enrichissement s'avère d'une utilité importante, elle permet : une classification intuitive des utilisateurs suivant leurs préférences (conjonction de critères de recherche de la requête) ; une recherche facile des cas similaires dans la base de cas. Ceci qui permet une recommandation avec une précision importante. De point de vue de la RI, les solutions des requêtes recommandées peuvent être des documents annotés par ces entités sémantiques. Il peuvent être aussi un ensemble des entités sémantiques, à savoir des concepts, des relations et leurs instances, c'est le cas de notre cas d'étude qui sera présenté au chapitre suivant. Ainsi, le chapitre suivant met en œuvre les différentes phases de l'approche proposée sur un système de recherche d'itinéraires dans le domaine du transport logistique OntoCBRIR.

Chapitre 4

OntoCBRIR : Système de recherche d'itinéraires

4.1 Introduction

Dans ce chapitre, nous présentons OntoCBRIR (*Ontology Based Information Retrieval using Case-Based Reasoning System*) un système de recherche d'information personnalisé dont le but est de rechercher les itinéraires qui correspondent le mieux aux préférences des utilisateurs. Le domaine d'application est le transport de marchandises en milieu urbain. Ce domaine est caractérisé par la diversité des sources d'information et l'hétérogénéité des informations relatives aux acteurs de la chaine logistique, ce qui rend la tâche de recherche des informations relatives à un itinéraire plus complexe. En effet, l'utilisateur ne souhaite avoir à disposition que peu d'informations, juste celles qui l'intéressent directement, c'est-à-dire celles qui sont adaptées à ses besoins et à ses préférences. Ces informations sont alors personnalisées, en d'autres termes, destinées à un utilisateur spécifique et non à un utilisateur générique.

L'objectif de cette présentation est de montrer les différents modules mises en œuvre depuis le chargement des données jusqu'à la réponse à une requête utilisateur. Bien que le système propose des résultats en partant d'une source de données structurées ou semi structurées (base de données, fichiers CSV), il peut être adapté à une recherche dans une collection de documents classiques. Nous menons par la suite une évaluation de ce système sous la forme d'un cas d'étude de recherche d'information relative aux itinéraires en transport

de marchandises.

La première section est consacrée à la présentation de l'architecture et les fonctionnalité de OntoCBRIR. La deuxième section décrit l'étude expérimentale menée. Enfin nous clôturons le chapitre avec la conclusion.

4.2 Présentation de OntoCBRIR

Cette section décrit la mise en œuvre du système OntoCBRIR. Cependant, nous jugeons utile de commencer par présenter le modèle de connaissances ou l'ontologie de domaine utilisée. Nous passons ensuite à la présentation des fonctionnalités de ce système, qui se rapportent aux approches proposées dans le chapitre précédent.

4.2.1 Modèle de connaissances

Le système proposé est destiné à un domaine particulier qui porte sur la gestion des opérations de transport de marchandises en milieu urbain. L'objectif est d'assurer une productivité optimale et un service fiable tout en réduisant les impacts environnementaux, les émissions de pollution de l'air, la consommation d'énergie et la congestion du trafic. Cependant, l'association d'un flux d'information ou de données au flux physique de la logistique urbaine nécessite l'intégration d'un modèle de connaissances riche, d'une part pour faciliter le partage des connaissances en utilisant différentes terminologies, d'autre part pour surmonter l'hétérogénéité des informations disponibles aux acteurs lors de la prise des décisions (Tableau 4.2). Pour ce faire nous utilisons initialement l'ontologie de domaine *GenCLOn* proposée par [2] comme modèle de connaissances. Cette ontologie sera enrichie en utilisant OntoCBRIR . La figure 4.1 montre un aperçu, qui comprend :

- Taxonomie des concepts représentant les objectifs
- Taxonomie des concepts représentant les acteurs
- Relations sémantiques entre les différents concepts

Nous montrons par la suite comment cette ontologie peut être enrichie automatiquement au cours de l'utilisation du système proposé.

Acteurs	Objectifs
Les résidents	– Coût minimum – Impact environnemental minimum
Les détaillants	– Rentabilité – Compétitivité
Les autorités	– Accessibilité – Respecter la législation et la restriction, – Coût minimum – Impact environnemental minimum – Assurer la sécurité des biens
Fournisseur	– Croissance du marché – Rentabilité
Transporteur	– Éviter la congestion – Respecter la législation et les restrictions – Coût de transport minimum – Consommation de carburant minimale – Assurer la sécurité des biens – Réduire le temps de chargement-déchargement

TABLEAU 4.1 – Objectifs des acteurs de la logistique urbaine

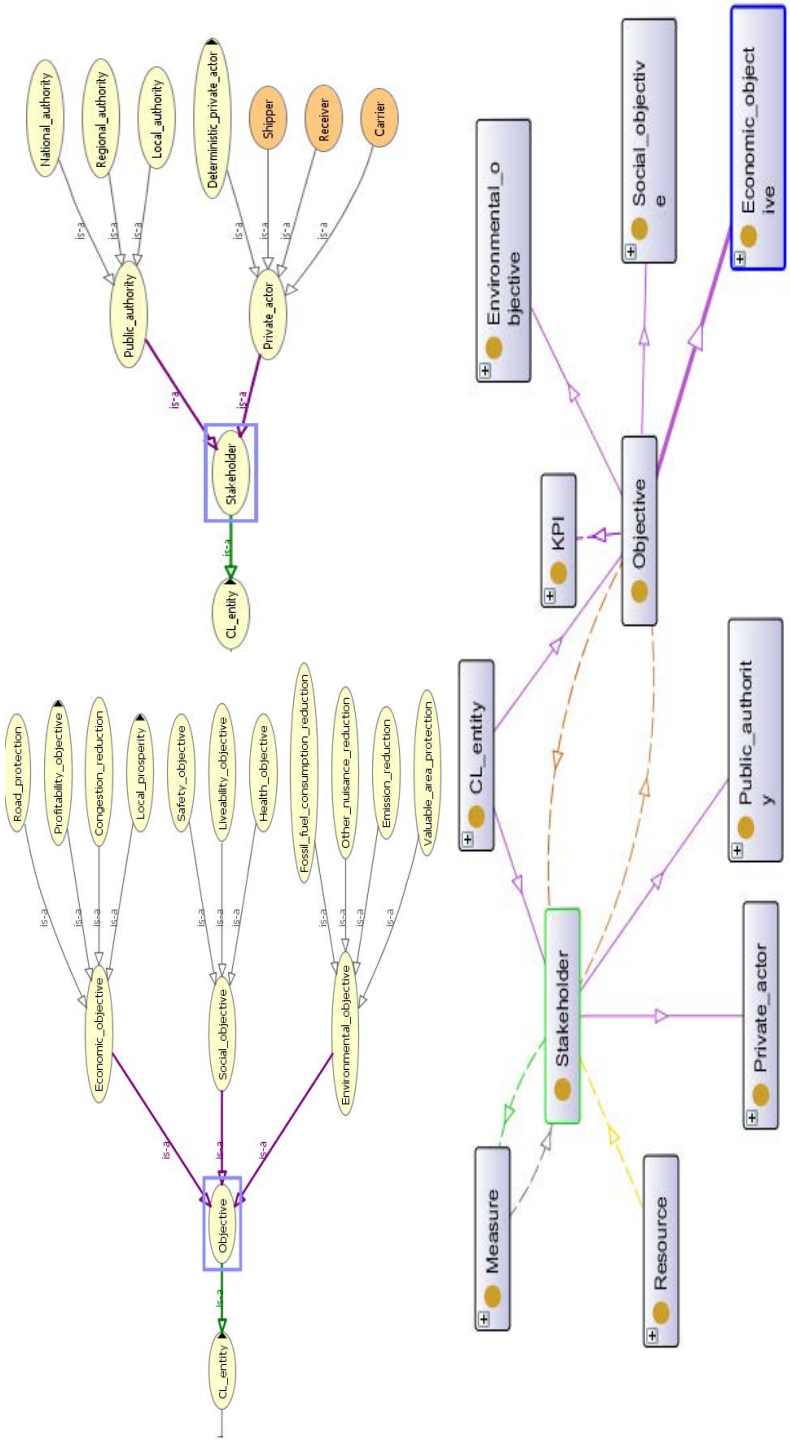


FIGURE 4.1 – Un aperçu sur l'ontologie de la logistique urbaine *GenCLOn*

4.2.2 Architecture et fonctionnalités

Du point de vue de la RI, le système proposé consiste à rechercher, à l'instant voulu, des informations relatives à des itinéraires en transport de marchandises. Il est destiné aux différents acteurs du transport urbain privé ou public (*carrier*, *receiver*, *shipper*, etc.) qui sont, entre autres, les utilisateurs potentiels de ce système. Il existe d'autres utilisateurs comme les managers du transport. En suivant l'architecture générique présentée dans le premier chapitre, notre système se compose principalement de trois parties (figure 4.2) :

- Une couche métier qui implémente les approches proposées dans le chapitre précédent.
- Une couche données qui gère l'ontologie de domaine et la collection de données.
- Une partie client qui correspond à l'interface destinée aux requêtes utilisateurs et la visualisation des résultats.

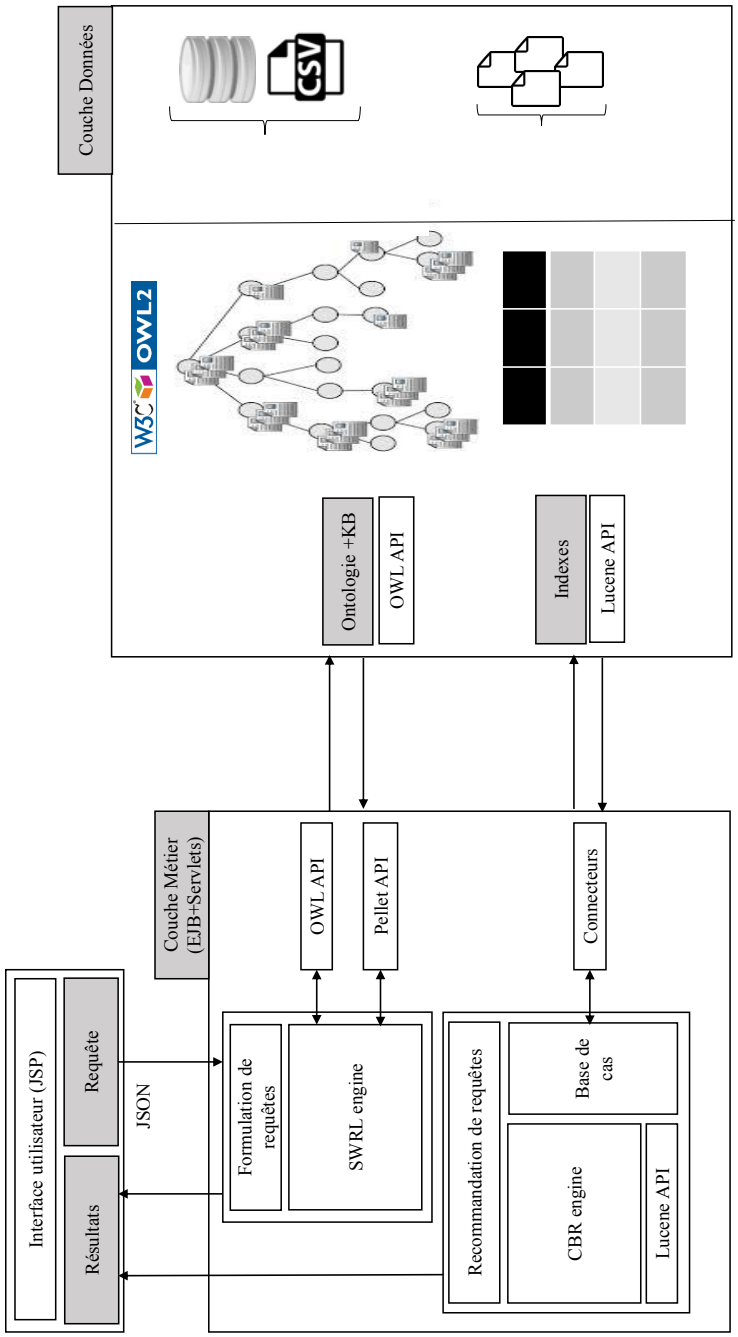


FIGURE 4.2 – Architecture globale du Système *OntoCBRIR*

La couche métier est implémentée sous forme de librairie Java englobée dans un composant EJB et déployée dans un conteneur J2EE(*Glassfish*) sous forme d'un portail Web accessible. L'avantage de ce déploiement est que les données (l'ontologie et les indexes) sont centralisées et chargées une seule fois [39]. Le composant ne fait dès lors que répondre à des requêtes qui lui sont adressées via le Web. La réponse aux requêtes utilisateurs est renvoyée en se basant sur les approches proposées dans le chapitre précédent, à savoir la formulation et la recommandation des requêtes. Il est possible de renvoyer la réponse sans forcément passer par l'étape de la recommandation des requêtes.
Les données traitées par le système proposé proviennent principalement des bases de données ou des collections de fichiers CSV. Néanmoins, nous pouvons étendre la couche données en prenant en compte la manipulation des collections des documents textuels. C'est dans cette couche données qu'est effectué l'assortiment entre les données et l'ontologie de domaine utilisée. Nous parlons précisément de l'instantation de l'ontologie.
Par ailleurs, nous proposons une interface utilisateur interactive, dans laquelle l'utilisateur sélectionne les concepts acteurs et objectifs. Un module d'ontologie sera ensuite instancié et visualisé sous forme d'un formulaire.

Notre système fait appel à plusieurs interfaces de programmation, à savoir :

- OWLAPI : Interface de programmation qui permet d'accéder à l'ontologie et la base de connaissances et manipuler les entités sémantiques lors de la phase de la formulation des requêtes.
- PelletAPI : Interface de programmation qui permet de raisonner sur les règles SWRL.
- LuceneAPI : Interface de programmation qui permet d'indexer, d'une part la collection de données dans la couche données, d'autre part la base de cas dans la couche métier.

Nous décrivons par la suite les fonctionnalités de OntoCBRIR en se basant sur l'approche proposée dans le chapitre précédent. Pour ce faire, nous commençons par illustrer la phase de la formulation des requêtes. Elle repose sur

des entités sémantiques qui peuvent être des classes, des propriétés (relations ou attributs) ou bien des instances. Le tableau 4.2 présente quelques entités sémantiques qui peuvent être utilisées lors de la formulation des requêtes.

Termes de requêtes	Concepts	Propriétés(relations ou attributs)
Origine/Destination	– OWLClass-00000046661641579586 Annotations : rdfs :label ”Shipper” – OWLClass-00000046661637366480 Annotations : rdfs :label ”Carrier” – OWLClass-0000004666164059677 Annotations : rdfs :label ”Receiver”	– OWLObjectProperty-00000003714219380954 Annotations :rdfs :label ”connect” – OWLObjectProperty-00000043948583785090 Annotations : rdfs :label ”origin” – OWLObjectProperty-00000044016042214862 Annotations : rdfs :label ”destination” – OWLObjectProperty-00000045309184500823 Annotations : rdfs :label ”ship-address
Vehicle Type	– OWLClass-00000048681345885021 Annotations : rdfs :label” Road-freight-vehicle”	– OWLObjectProperty-000000485818 Annotations : rdfs :label ”has-resource”
Fuel consumption	– OWLClass-00000017260945781935 Annotations : rdfs :label ”Fossil-fuel-consumption-reduction”	– OWLClass-00000022452118786605 Annotations : rdfs :label ”Logistics-cost-reduction” – OWLObjectProperty-00000049659148982342 Annotations : rdfs :label ”has-objective”
Transportation cost	– OWLClass-00000017260940625694 Annotations : rdfs :label ”Transport-cost-reduction”	– OWLClass-00000022452118786605 Annotations : rdfs :label ”Logistics-cost-reduction” – OWLObjectProperty-00000049659148982342 Annotations : rdfs :label ”has-objective”
Emission Co2	– OWLClass-00000017260944876793 Annotations : rdfs :label ”Emission-reduction”	– OWLClass-00000022452118786605 Annotations : rdfs :label ”Logistics-cost-reduction” – OWLObjectProperty-00000049659148982342 Annotations : rdfs :label ”has-objective”

TABLEAU 4.2 – Exemples des entités sémantiques utilisées dans la formulation des requêtes

Avant de détailler l'interface utilisateur de OntoCBRIR, nous illustrons par l'algorithme 4.1 le cycle des interactions d'un utilisateur avec le système dans une session de recherche. L'utilisateur commence par saisir sa requête qui sera analysée par le système sous la forme des entités sémantiques. Une règle SWRL est ensuite générée pour initialiser le processus de la recommandation. A l'issue de ce processus, le système renvoie les résultats de recherche à l'utilisateur. Deux scénarios sont alors possibles :

- Si l'utilisateur accepte les résultats, le processus de recherche prendra fin.
- Si l'utilisateur n'accepte pas les résultats, le système reprendra le processus de recherche d'une manière classique en interrogeant directement les sources de données indexées (rechercher(E)).

Comme l'indique l'ontologie GenCLon, il existe une liste exhaustive des acteurs (privés et publics) ayant différents objectifs (voir Tableau 4.2). Selon les objectifs sélectionnés par les utilisateurs dans leurs requêtes, le système détermine à quelles catégories de cas ces utilisateurs seront affectés. Les règles sémantiques sont appliquées en utilisant la conjonction des triplets des entités sémantiques. A titre d'exemple (a), si dans une requête l'utilisateur cherche des itinéraires permettant de satisfaire ses objectifs économiques (*fuel consumption*, *transportation cost*) et qui sont relatifs à un acteur de type *Receiver*, les triplets des entités sémantiques qui seront utilisés sont les suivants :

- $< Receiver(x), has\_objective, Fuel\_consumption(y) >$
- $< Receiver(x), has\_objective, Transportation\_cost(z) >$
- $< x, rdf : type, Receiver >$
- $< y, rdf : type, Fuel\_consumption(y) >$
- $< z, rdf : type, Transportation\_cost >$

Dans ce cas, la règle utilisée consiste à classer cette requête comme un nouveau cas (case) destiné aux acteurs de type *Receiver*. Cela implique la création d'une nouvelle taxonomie dans l'ontologie *GenCLOn* comme le montre la figure 4.3. Cette taxonomie est constituée par de nouveaux concepts topiques qui sont insérés automatiquement par le système et qui sont liés avec l'ancienne ontologie par la relation (has_case).

Algorithme 4.1 : Algorithme de recherche dans OntoCBRIR

```
Entrées : Requête rq,
          Requête [] rqs,
          Entités [] E,
          Bool slect
Sorties  :  Résultat [] res
begin
    Analyser rquête()
    si requêteSWRL(rq) alors
     |  E = lister les entités (rq)
    finsi
    alors
     |  E = lister les entités ()
    finsi
    Recommandation()
    rqs=moduleCbr(rq)
    res=SelectRésultat(rqs)
    Afficher (res)
    si select = vrai alors
     |  Fin de recherche
    finsi
    alors
     |  res = rechercher (E)
    finsi
end
```

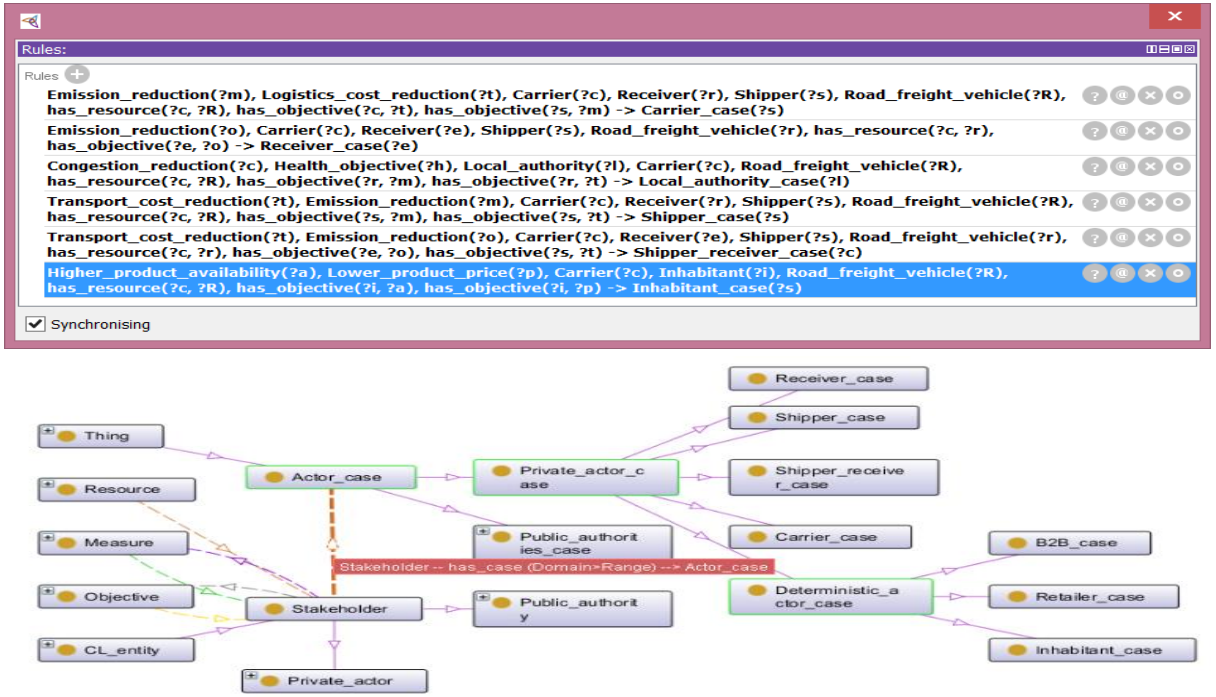


FIGURE 4.3 – Visualisation sur Protégé des nouveaux concepts topiques générés à partir des règles sémantiques SWRL

Comme c'est indiqué dans la figure 4.4, en utilisant le formulaire (*user information*), la première étape consiste à sélectionner les acteurs présents dans le menu déroulant. L'étape suivante consiste à sélectionner le type des objectifs dans l'itinéraire recherché. Si l'objectif sélectionné est économique, un affichage automatique des critères correspondants sera effectué dans le formulaire de recherche (*Search*). Le champ du texte (*add new actor*) permet de rajouter un nouveau acteur en cas où l'utilisateur n'est pas représenté dans la liste des acteurs présents dans *GenLog*. Le second formulaire (*Search*) permet, d'une part de saisir l'origine et la destination de l'itinéraire, d'autre part de saisir les critères de recherche qui ont été générés automatiquement lors de la sélection du type d'objectif dans le premier formulaire. Pour déterminer le poids des critères de recherche, l'utilisateur doit saisir un nombre entre $\{1, \ldots, 5\}$ après chaque critère saisi. Le système repose sur ces valeurs pour calculer les poids de préférences en utilisant l'opérateur d'agrégation (intégrale de Choquet).

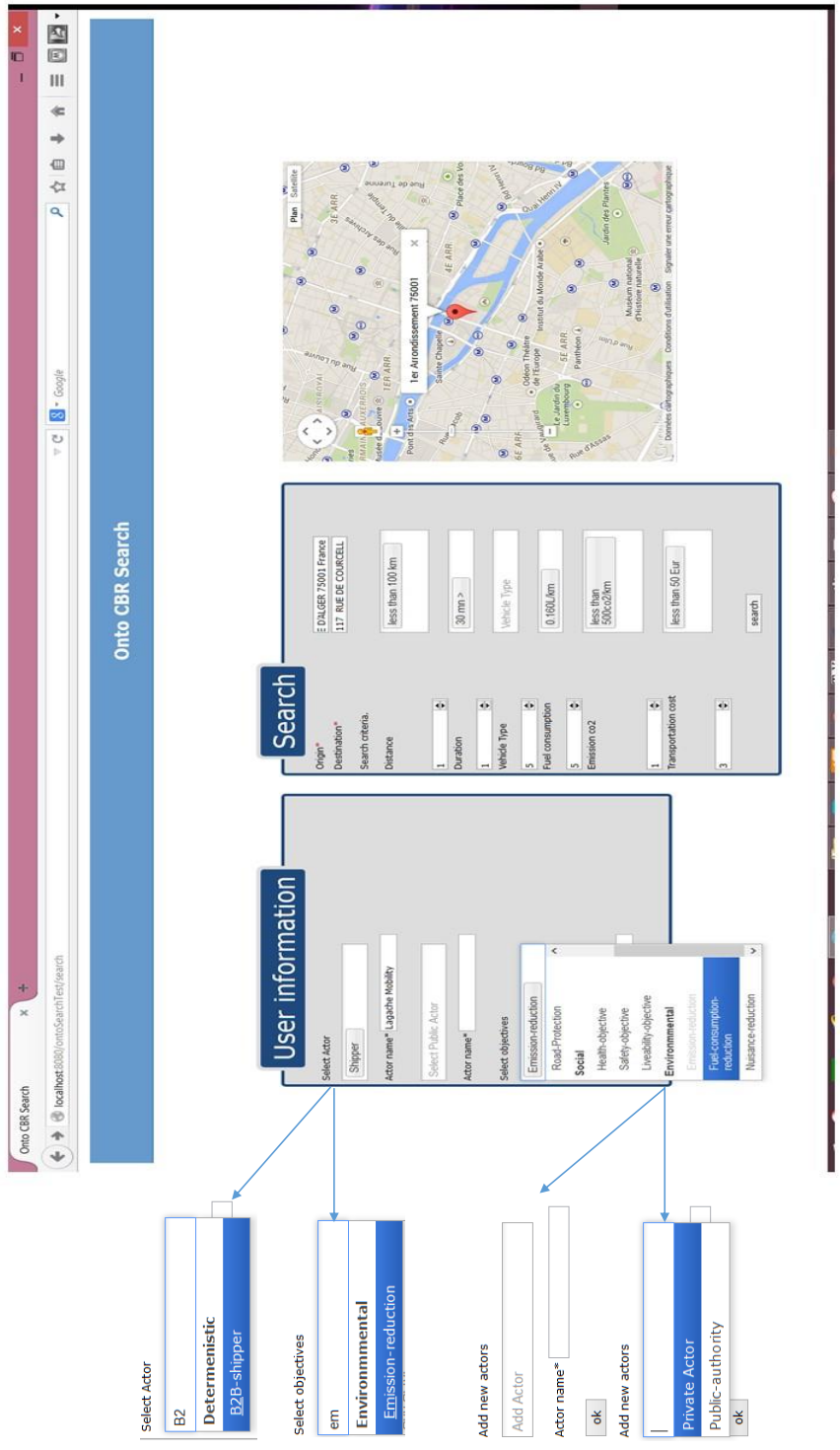


FIGURE 4.4 – Interface de saisie de requêtes conceptuelles avec des fonctionnalités d'auto complétion

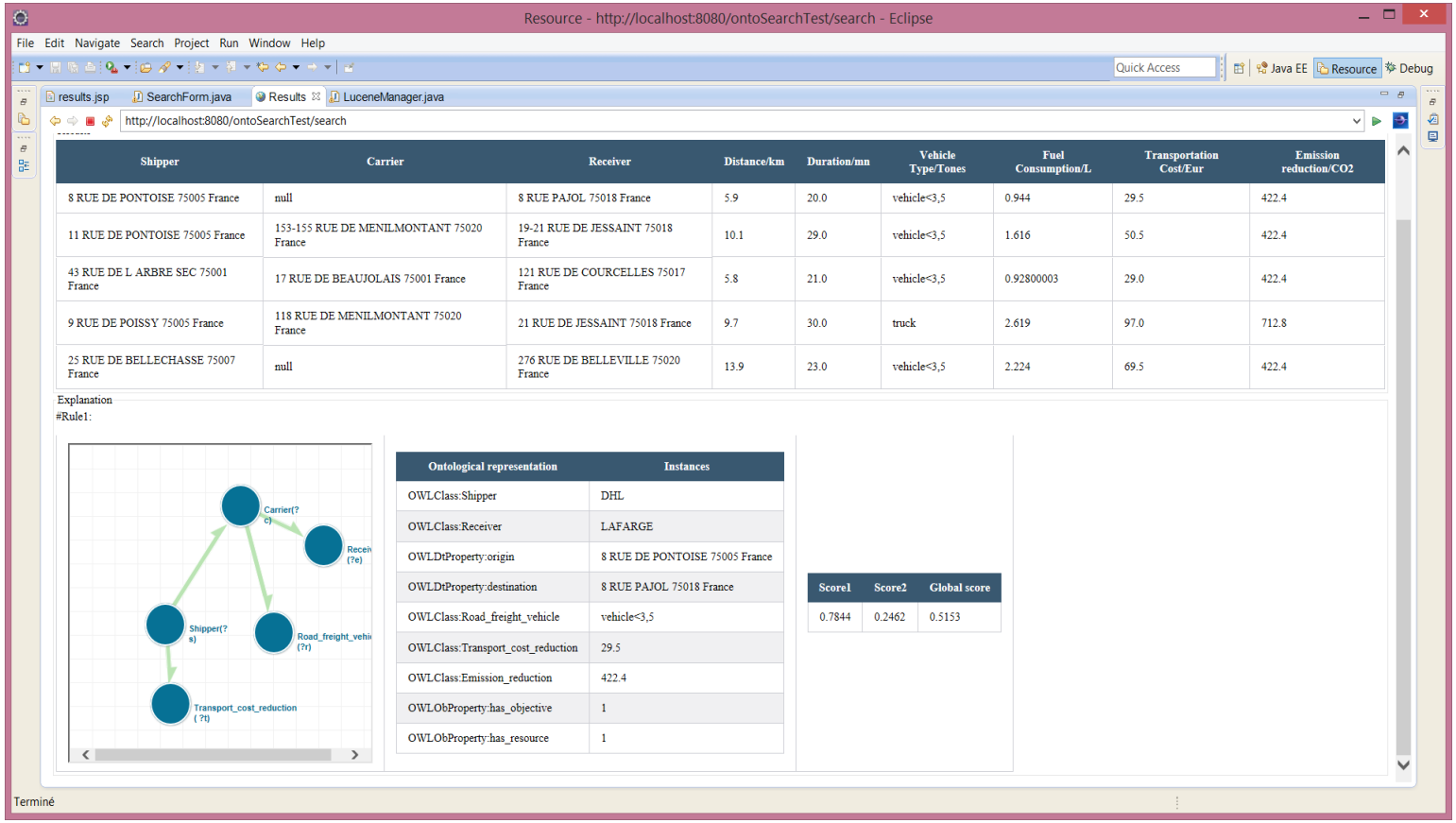
Après avoir formulé la requête utilisateur, le système procède à la phase de la recommandation. Nous rappelons que chaque utilisateur est associé à la base de cas locale, ainsi les créés seront partagés par les autres utilisateurs. La base de cas est utilisée principalement pour la proposition des recommandations sous forme de requêtes similaires et leurs résultats (qui partagent le même focus de recherche). Pour ce faire, OntoCBRIR utilise les règles sémantiques SWRL qui permettent de représenter la nouvelle requête sous forme d'un nouveau cas (problème et solutions). Le principal avantage de cette méthode réside dans le fait qu'elle permet de cibler la recherche en se basant sur les solutions relatives aux anciens cas ayant la même structure que le nouveau cas. De même, nous pouvons associer la règle suivante à l'exemple de recherche d'itinéraires cité auparavant (a) :

$$\begin{gathered} Transport\_cost\_reduction(?t) \\ \wedge Other\_nuisance\_reduction(?o) \\ \wedge Carrier(?c) \wedge Receiver(?s) \wedge Shipper(?e) \\ \wedge Driver(?d) \wedge Road\_freight_v ehicle(?r) \wedge \\ has\_ressource(?c, ?d) \wedge has\_ressource(?c, ?r) \\ \wedge has\_objective(?s, ?t) \wedge has\_objective(?s, ?o) \\ \longrightarrow Case\_receiver(?s) \end{gathered}$$

La règle ci-dessus est composée de deux type concepts :

- Concepts textuels : les instances de ces concepts représentent les informations relatives aux itinéraires recherchés, à savoir les labels des propriétés (relations), les labels/noms des acteurs, l'adresse et la destination ainsi que les labels des objectifs.
- Concepts numériques : ces concepts représentent les critères de recherche dans la requête, à savoir le coût du transport, le type de véhicule, la consommation de carburant, la distance parcourue et le temps.

Cette requête sera l'entrée de l'étape suivante permettant de rechercher les cas similaires. Si des cas similaires existent dans la base de cas, les solutions ramenées à partir de cette base de cas sont affichées à l'utilisateur. La similarité est calculée ainsi en fonction des deux types de concepts textuels et numériques.



Shipper	Carrier	Receiver	Distance/km	Duration/mn	Vehicle Type/Tones	Fuel Consumption/L	Transportation Cost/Eur	Emission reduction/CO2
8 RUE DE PONTOISE 75005 France	null	8 RUE PAJOL 75018 France	5.9	20.0	vehicle<3,5	0.944	29.5	422.4
11 RUE DE PONTOISE 75005 France	153-155 RUE DE MENILMONTANT 75020 France	19-21 RUE DE JESSAINT 75018 France	10.1	29.0	vehicle<3,5	1.616	50.5	422.4
43 RUE DE L ARBRE SEC 75001 France	17 RUE DE BEAUJOLAIS 75001 France	121 RUE DE COURCELLES 75017 France	5.8	21.0	vehicle<3,5	0.92800003	29.0	422.4
9 RUE DE POISSY 75005 France	118 RUE DE MENILMONTANT 75020 France	21 RUE DE JESSAINT 75018 France	9.7	30.0	truck	2.619	97.0	712.8
25 RUE DE BELLECHASSE 75007 France	null	276 RUE DE BELLEVILLE 75020 France	13.9	23.0	vehicle<3,5	2.224	69.5	422.4

Ontological representation	Instances
OWLClass:Shipper	DHL
OWLClass:Receiver	LAFARGE
OWLDtProperty:origin	8 RUE DE PONTOISE 75005 France
OWLDtProperty:destination	8 RUE PAJOL 75018 France
OWLClass:Road_freight_vehicle	vehicle<3,5
OWLClass:Transport_cost_reduction	29.5
OWLClass:Emission_reduction	422.4
OWLObProperty:has_objective	1
OWLObProperty:has_resource	1

Score1	Score2	Global score
0.7844	0.2462	0.5153

FIGURE 4.5 – Interface utilisateur : résultats de recherche

La figure 4.5 illustre une interface avancée pour les résultats de la recherche :

- La première partie de l'interface représente un ensemble de résultats personnalisés (résultats des k anciens cas).
- La seconde partie de l'interface illustre les explications qui sont générées automatiquement lorsque l'utilisateur sélectionne un itinéraire donné.
- Le diagramme généré représente la règle SWRL utilisée.
- Le tableau de la représentation ontologique (la tableau de taille en moyenne) illustre les concepts et les instances d'un itinéraire sélectionné.
- Le tableau montre les scores résultant de la similarité numérique (Score1), la similarité textuelle (Score2) et le score global de l'itinéraire choisi.

4.3 Expérimentations et évaluation

4.3.1 Première partie : Test de la méthode RàPC

Expérimentions

Dans cette section, nous présentons les principaux résultats des expérimentations menées pour évaluer OntoCBRIR. En raison de l'inexistence de méthodes de recherche d'itinéraires basées sur l'intégration des ontologies et le raisonnement à partir de cas RàPC, nous avons mené des études comparatives entre la méthode proposée par [16] et l'approche proposée dans le cadre de cette thèse. L'objectif est de montrer l'apport de l'ontologie et les nouvelles mesures de similarités adaptées dans l'amélioration du processus de recommandation de requêtes. Nous rappelons que ce processus est basé sur la méthode RàPC et que l'hypothèse retenue est que plus la similarité entre les requêtes est importante, plus la recommandation et précise. Par conséquent, les résultats de recherche devraient être plus pertinents.

Les expérimentations réalisées s'appuient sur des données récupérées sous

forme des fichiers CSV, à partir du site *Paris Open Data*[1]. La base de connaissances a été ensuite créée et indexée en effectuant l'instanciation de l'ontologie par les données récupérées. Un échantillon de 100 requêtes a été retenu en se basant sur le fichier log du système, formant ainsi un premier ensemble de requêtes (*query-set*). La figure 4.6 illustre la structure d'une requête utilisée lors des expérimentations.

```
<query>
<num>3</num>
<content>
Actor name: Lagache Mobility
Origin: 4 RUE D'ALGER 75001 France
Destination: 117  RUE DE COURCELLES 75017 France
Distance: Less Than 100        weight: 1
Duration: 22. Less Than 60     weight: 2
Fuel Cosumption:0.180/km   weight: 3
Transportation Cost: Less than 50 weight: 1
Emission reduction: Less than 0.180/km weight: 4
</content>
</query>
```

FIGURE 4.6 – Structure des requêtes

Deux expérimentations ont été menées (voir 4.7). L'objectif de la première est de créer des ensemble de requêtes pouvant être exécutables sur le système basé sur l'approche [16]. Les principales étapes de cette expérimentation sont les suivantes :

- Exécuter le premier ensemble de requêtes.
- Formuler les requêtes par les entités sémantiques.
- Générer, si possible, les règles appropriées pour chaque requête.
- Pour chaque nouveau cas, chercher des solutions à partir des cas similaires dans la base de cas.
- Retourner les résultats.
- Classifier les requêtes selon les règles sémantiques SWRL utilisées.

L'objectif de la deuxième expérimentation est de comparer OntoCBRIR avec le système basé sur l'approche [16] en exécutant tous les sous-ensembles de

1. http ://opendata.paris.fr/page/home/

requêtes générés à l'issue de la première expérimentation. Les principales étapes de la deuxième expérimentation sont les suivantes :

- Exécuter les sous-ensembles de requête dans les deux systèmes.
- Retourner les résultats.
- Évaluer les résultats à l'aides des courbes (precision-rappel).

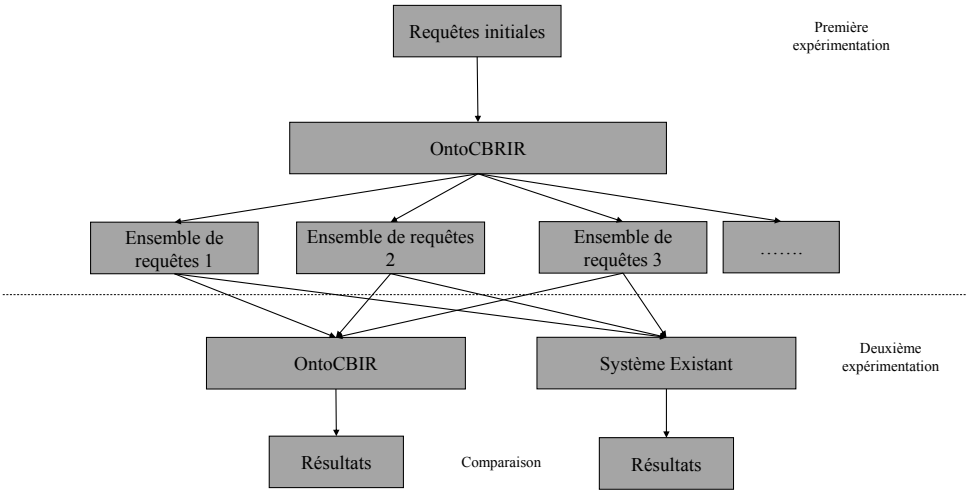


FIGURE 4.7 – Organigramme des expérimentations

Après avoir établi la première expérimentation, OntoCBRIR a été en mesure de créer les nouveaux cas sur la base de trois règles générées automatiquement (Tableau 4.3). La première règle *Rule1* est destinée aux acteurs de type (*receiver*) ayant des objectifs environnementaux. La deuxième règle, quant à elle, *Rule2* est destinée aux acteurs de type (*Shipper*) ayant des objectifs économiques tandis que la troisième règle *Rule3* est destinée à deux acteurs ayant tous les deux des objectifs dans le même itinéraire.

La base de cas contient 95 cas, dont 40 cas représentés par *Rule1*, 35 par *Rule2*, 20 par *Rule3* tandis que le reste n'a été représenté que par des entités sémantiques. Le tableau 4.4 montre la répartition des cas selon leurs règles générées ainsi que le nombre des instances relatives aux concepts textuels et concepts numériques (critère de recherche) dans chaque groupe de cas.

Concept topique	Règle SWRL
Receiver_case	$Emission\_reduction(?o) \wedge Carrier(?c)$ $\wedge Receiver(?e) \wedge Shipper(?s)$ $\wedge Road\_freight\_vehicle(?r) \wedge has\_ressource(?c, ?r)$ $\wedge has\_objective(?e, ?o) \rightarrow Receiver\_case(?e)$
Shipper_case	$Transport\_cost_reduction(?t) \wedge Carrier(?)$ $Receiver(?e) \wedge Shipper(?s)$ $\wedge Road\_freight_vehicle(?r) \wedge has\_ressource(?c, ?r)$ $\wedge has\_objective(?s, ?t) \rightarrow Shipper\_case(?e)$
ShipRec_case	$Transport\_cost\_reduction(?t) \wedge Emission_reduction(?o)$ $\wedge Carrier(?) \wedge Receiver(?e) \wedge Shipper(?s)$ $\wedge Road_freight_vehicle(?r) \wedge Itinerary_pattern(?p)$ $\wedge has_ressource(?c, ?r) \wedge has_objective(?s, ?t)$ $\wedge has_objective(?e, ?o) \rightarrow Shipper_receiver_case(?p)$

TABLEAU 4.3 – Règles SWRL générées par OntoCBRIR

Règles SWRL	Concepts textuels	Concepts numériques	Propriétés	Nombre de cas
Rule1	7	3	2	40
Rule2	7	3	2	35
Rule3	7	5	4	20

TABLEAU 4.4 – Répartition des cas en fonction des règles sémantiques SWRL

Évaluation

À ce stade l'objectif de l'évaluation est d'étudier l'utilité l'approche de recommandation basée sur l'utilisation de l'ontologie et le raisonnement à partir de cas comparativement à l'approche de recommandation classique basée sur seulement le raisonnement à partir de cas RàPC [16]. Pour cela nous avons effectué l'évaluation sur trois sous-ensembles de requêtes relatifs aux règles précédemment générées par le système lors de la première expérimentation. Nous avons ensuite procédé à l'évaluation en utilisant les mesures standards utilisées pour évaluer les approches de classification et de recherche d'information (4.5). Plus le système retourne de résultats, plus la possibilité de retourner tous les résultats pertinents est élevée : il s'agit de la notion de précision. Moins le système en retourne, plus le taux de résultats pertinents retournés est élevé : il s'agit de la notion de rappel. Ainsi, la fiabilité de notre système dépend de la capacité de présenter, d'une part un rappel élevé , d'autre part une précision élevée.

Soit N la collection de cas, n est le nombre de cas pertinents, r le nombre de cas pertinents récupérés à partir de N et k le nombre de tous les cas récupérés. Notons que le jugement de la pertinence des résultats de chaque requête (nouveau cas) est défini sur la base des seuils de similarité maximale définis dans le chapitre précédent. Plus la similarité entre un nouveau cas C^{new} et un cas en mémoire C^{mem} est proche du seuil, plus le cas résultat (C^{mem}) est pertinent par rapport au nouveau cas ou la nouvelle requête (C^{new}). Comme expliqué dans le chapitre précédent, ces similarités s'effectuent en utilisant différentes mesures, à savoir la mesure TF-IDF pour calculer la similarité en termes de contenu et la mesure numérique basée sur l'opérateur d'agrégation (intégrale de Choquet).

$$\text{Rappel} = \frac{a}{a+c} \tag{4.1}$$

Documents	Pertinents	Non Pertinents	Total
Récupérés	a	b	a+b=k
Non Récupérés	c	d	c+d=n-k
Total	a+c=r	b+d=n-r	a+b+c+d=N

TABLEAU 4.5 – Mesures d'évaluation : Précision-Rappel

$$\text{Précision} = \frac{a}{a+b} \tag{4.2}$$

$$P_{interp}(r) = \max(P(r')pourr' \geq r \tag{4.3}$$

Ces mesures d'évaluation sont ensuite illustrées par des courbes précision-rappel. Ces courbes sont calculées par la fonction $f(R) = P$ en utilisant la règle d'interpolation [63] :

$$P_{interp}(r) = \max(P(r')pourr' \geq r \tag{4.4}$$

Avec P_{interp} est la précision par interpolation à un niveau de rappel r, elle est définie par la plus grande précision trouvée pour n'importe quel niveau de rappel $r' \geq r$. Les trois courbes correspondent aux trois sous-ensembles de requêtes qui on été obtenus lors de la première expérimentation. Comme le montrent les figures 4.8 4.9 4.10, nous pouvons constater l'amélioration du taux de précision pour les trois sous-ensembles de requêtes. Ainsi, nous obtenons une performance meilleure lors de la recommandation des requêtes. Cela s'explique par le fait que l'approche proposée a l'avantage de récupérer les cas similaires sur la base des nouvelles mesures de similarité sémantique, ce qui n'est pas réalisable avec l'approche RàPC [16]. En outre l'utilisation des entités sémantiques et les règles sémantiques SWRL a amélioré l'expressivité dans la requête, ce qui augmente encore le niveau de la personnalisation dans le processus de recherche. Nous présentons les pourcentages des améliorations dans le tableau 4.6.

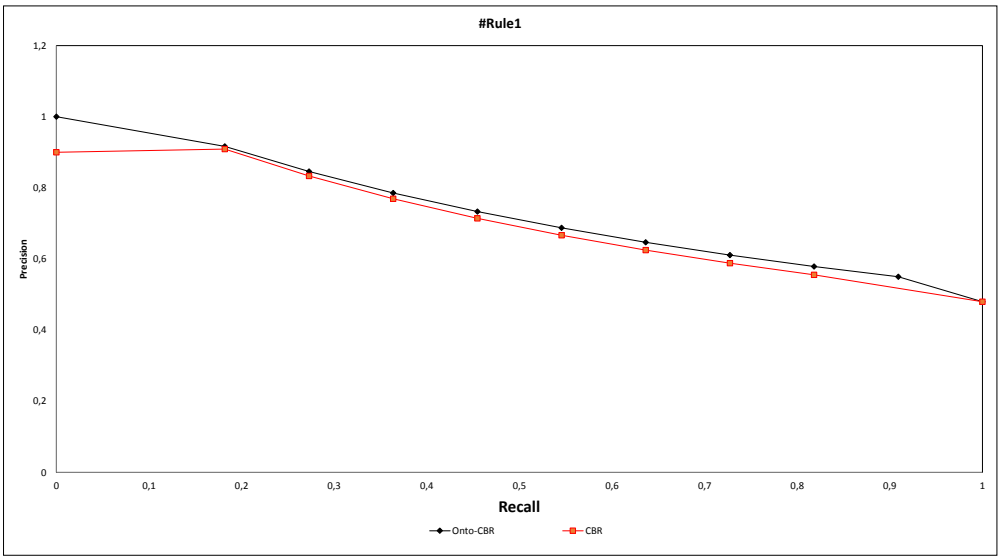


FIGURE 4.8 – Courbe précision-rappel pour le sous-ensemble de requêtes : Règle 1

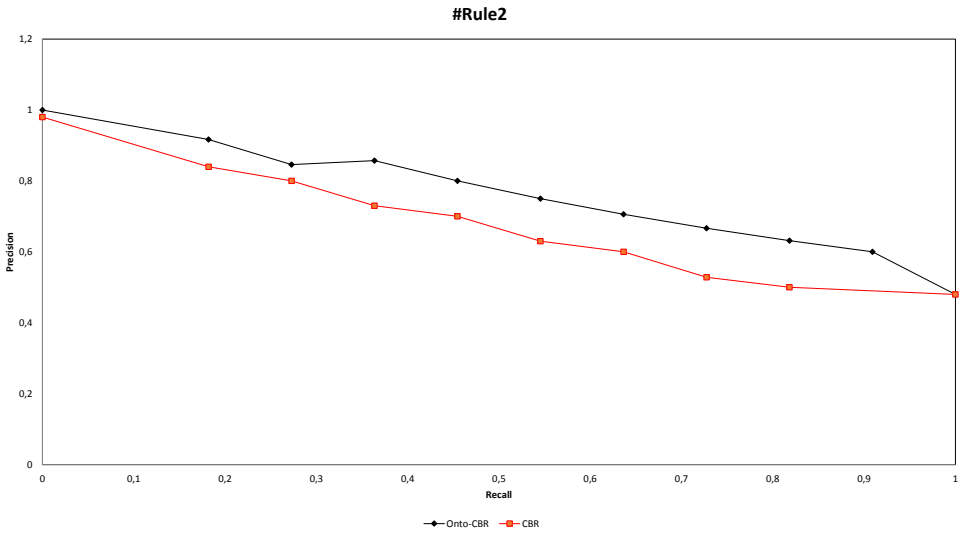


FIGURE 4.9 – Courbe précision-rappel pour le sous-ensemble de requêtes : Règle 2

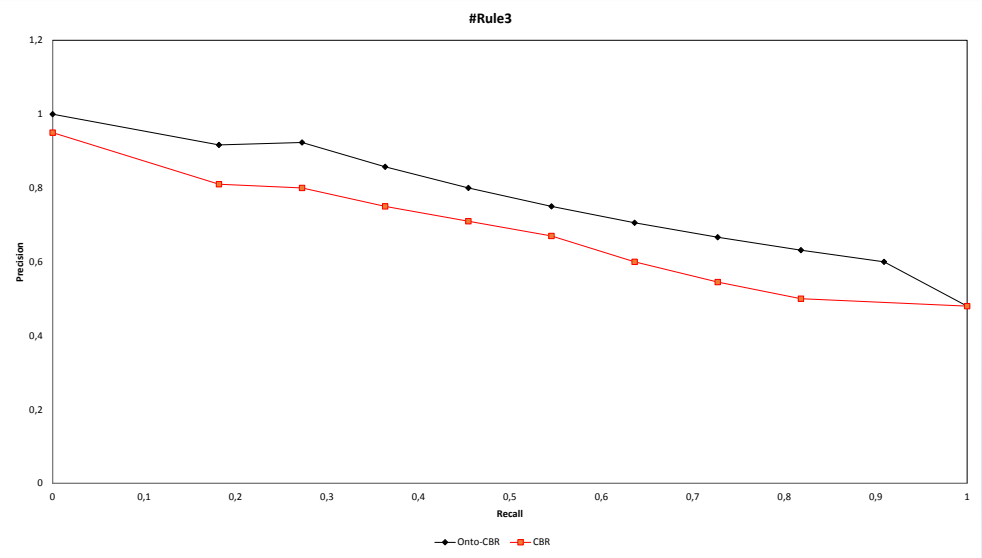


FIGURE 4.10 – Courbe précision-rappel pour le sous-ensemble de requêtes : Règle 3

Règles SWRL	Amélioration (%)
Sous-ensemble 1	3,4
Sous-ensemble 2	13,8
Sous-ensemble 3	14,3

TABLEAU 4.6 – Pourcentages d'amélioration sur les trois sous-ensembles de requêtes

4.3.2 Deuxième partie : Test de processus de recherche

Dans l'objectif de montrer l'impact de l'approche proposée dans le processus de recherche, nous avons effectué une étude comparative concernant les résultats obtenus d'une requête (figure 4.6). Ces résultats sont relatifs à deux scénarios.

Scénario 1

Ce scénario consiste à tester la requête sur un SRI [111] implémenté en suivant l'approche proposée par [29]. Dans ce système, la base de connais-

sances est créée en effectuant l'assortiment entre l'ontologie et les sources de données. Ainsi, nous obtenons un graphe (ensemble de triplets RDF/OWL ayant la forme $Entit - Attribut - Valeur$) étiqueté dont les nœuds sont des instances des concepts et les arcs représentent les propriétés [29]. Les résultats des requêtes sont obtenus en utilisant une combinaison booléenne de paires attribut-valeur basée sur l'opérateur logique $\wedge$, $\vee$ et $\neg$. Notons que la requête est traitée sous la forme des mots-clés saisis par l'utilisateur. Les résultats proviennent des entités qui correspondent à ces mot-clés.

Structure de la requête. Modèle EAV, l'entité est désigné par e, l'attribut est désignée par att et la valeur est notée v. Compte tenu de l'état de sélection de mots-clés c et une relation R, l'opérateur de sélection des mots clés $\sigma_c(R)$ est défini comme un ensemble d'instances de relations $\{r|r \in R\}$ pour laquelle la condition c est vrai. La condition c consiste à tester si un mot donné désigné par k est produit dans l'un des champs (e, att, v) d'une relation R, qui est désignée par $f : k$ où la fonction du test est notée W. Par exemple, si nous testons si le mot clé k est produit dans relation d'instance r (désignée par$r.v$) , alors : $\sigma_v :_k (R) : \{r|r \in R, k \in W(r.v)\}$.

Avec $\pi_f(R)$ désigne l'opérateur de projection, qui permet l'extraction d'une colonne spécifique d'un champ F à partir d'une relation R. L'opérateur de projection peut être utilisé pour extraire plus qu'une colonne. Par exemple, π_e, $d(R)$ renvoie une relation avec seulement deux colonnes, l'ensemble de données et l'entité.

Exemple. Dans l'exemple 4.6, une partie de la requête peut être formulée comme suit :
$Q = \pi_{e,att,v}(\sigma_v : "RueD'Alger" \wedge v : "RuedeCourcelles"(R))$.

Scénario 2

Ce scénario consiste à tester la requête sur OntoCBRIR. Comme le montre la figure 4.11, la base de connaissances est indexée en se basant sur l'approche

proposée par [36].

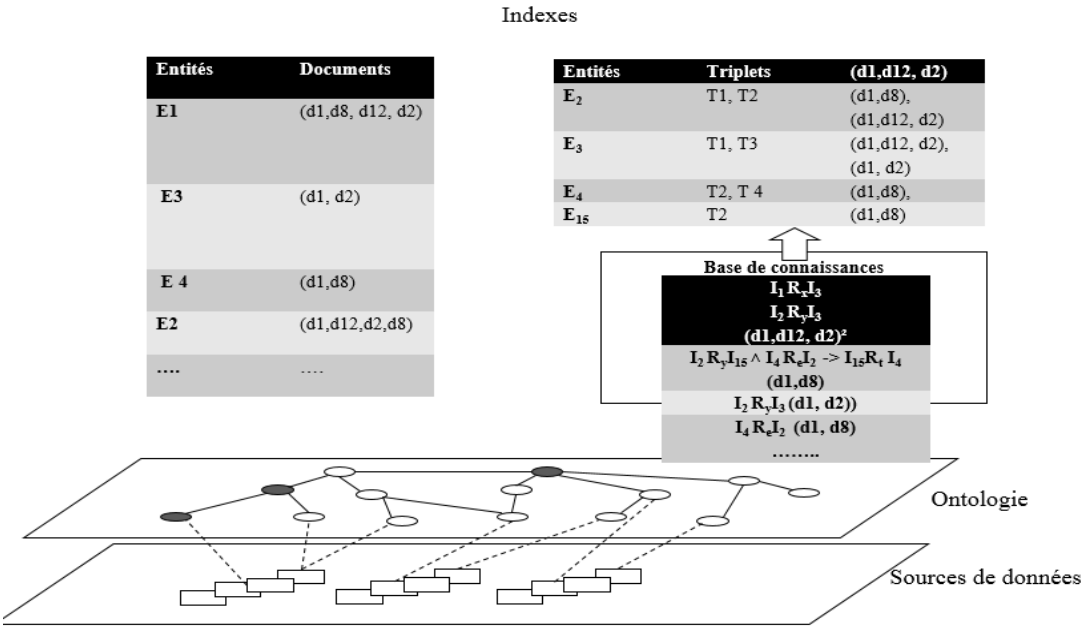


FIGURE 4.11 – Indexation suivant l'approche [36]

Le tableau 4.6 présente les valeurs de la précision moyenne obtenues pour les deux scénarios. Nous pouvons remarqué l'amélioration de la précision moyenne au niveau du deuxième scénario. C'est là qu'apparaît le rôle de la personnalisation dans notre système. En effet, le représentation de la requête par des règles sémantiques permet de capter et déduire les préférence utilisateur. Cela se traduit par le fait de catégoriser l'utilisateur en se basant sur le nouveau concept topique généré à partir de la règle utilisée. Ensuite, la recherche des cas sémantiquement similaires permet de cibler la recherche et d'augmenter la corrélation entre les résultats retournés et les préférences des utilisateurs. Bien que les résultats montrent une tendance à l' amélioration, cette étude est préliminaire et nous souhaitons la continuer dans les prochains travaux.

Résultats (itinéraires)	Précision (Scénario 1)	Précision (Scénario 2)
15	0,694	0,777(+8,3 %)
25	0,419	0,910(+49,1%)
30	0,344	0,862(+51,8%)
35	0,294	0,735(44,1%)
40	0,282	0,666(+38,4%)
45	0,252	0,590(+33,8%)
50	0,244	0,552(+30,8%)
Précision moyenne	0,158	0,318(+16%)

TABLEAU 4.7 – Précision Moyenne des résultats de recherche

4.4 Conclusion

Le système de recherche d'itinéraires décrit dans ce chapitre concrétise bien l'approche proposée dans le chapitre précédent. A travers les expérimentations menées nous pouvons constater que l'utilisation des entités sémantiques permet une formulation de requête assistée et optimale et qui ne nécessite pas des connaissances au préalable sur la structure de l'ontologie. En outre, l'utilisation des règles sémantiques SWRL avec le raisonnement à partir de cas permet de capturer les informations relatives aux préférences des utilisateurs et leur proposer des résultats personnalisés sans construire un modèle d'utilisateur explicite.

Par ailleurs, notre système intègre l'utilisateur dans la boucle de pertinence à travers l'expression de ses préférences mais nous pouvons aller encore plus loin dans cette prise en compte. Il s'agit notamment de lui permettre d'ajouter, dans un premier temps, des concepts relatifs aux acteurs, ce qui enrichira l'ontologie et la base de cas. Enfin, nous pouvons étendre les fonctionnalités du système proposé à d'autre types de recherche, notamment la recherche documentaire.

Conclusion générale

Les travaux de cette thèse sont articulés autour de la recherche d'information (RI) guidée par les ontologies de domaine. Ce paradigme a été exploré en réponse à différents verrous qui sont principalement relatifs au manque de représentativité du contenu informationnel. En effet, l'utilisation des concepts d'une ontologie de domaine permet d'identifier les liens possibles entre les concepts exprimés dans la requête, et par conséquent, d'améliorer davantage la qualité des résultats fournis en réponse à cette requête. Cette démarche a d'autant plus d'impact qu'il s'agit d'un système de recherche d'information dédié à un domaine particulier dans lequel la représentation sémantique du contenu informationnel peut être exhaustive. Néanmoins, l'utilisateur doit être familiarisé avec un tel environnement de ce système, ce n'est pas souvent le cas. L'expression de ses besoins reste ainsi une tâche complexe. C'est dans ce contexte que nous avons exposé notre problématique de recherche et évoqué la nécessite de développer des techniques permettant :

- D'assister l'utilisateur dans la formulation de sa requête.
- De proposer à l'utilisateur des résultats en réponse à ses besoins et surtout ses préférences.
- De faire évoluer l'ontologie de domaine utilisée, d'une manière supervisée, au cours de l'utilisation du système de recherche d'information (SRI).

Nous résumons dans la suite nos principales contributions ainsi que les perspectives ouvertes par nos travaux.

Contributions

Les travaux présentés dans cette thèse entrent dans le sillage des approches conceptuelles de la RI, en particulier la recherche de données. Nous entendons par recherche de données, la recherche dans des bases de connaissances qui représentent des sources de données de différents types.

Nous avons proposé une approche de recherche à deux niveaux permettant de tenir en compte les préférences des utilisateurs.

Formulation des requêtes. Dans l'objectif d'explorer au mieux l'ontologie de domaine utilisée dans le processus de la formulation de requête, nous avons proposé à l'utilisateur, en nous inspirant du principe des requêtes visuelles, une interface permettant de spécifier les concepts ainsi que les propriétés (relations et attributs) qui constituent sa requête. La requête est ensuite analysée sous la forme des triplets des entités sémantiques. Notons que l'utilisateur peut intégrer des nouveaux concepts suivant des structures définies dans l'ontologie utilisée. C'est l'un des aspects de l'enrichissement de l'ontologie utilisée.

Par ailleurs une base de règles sémantiques a été incorporée pour détecter les préférences des utilisateurs. Nous pouvons dire que l'objectif derrière l'utilisation de ces règles est double :

- Enrichir l'ontologie, c'est le deuxième aspect enrichissement.
- Structurer la base de cas qui sera utilisée dans le processus de recommandation et donc cibler la recherche.

Recommandation des requêtes. Ce processus est le cœur de notre approche. Nous avons opté pour l'utilisation de la méthode du raisonnement à partir de cas (RàPC). En effet, l'avantage majeur de la réutilisation d'expérience dans les systèmes de recherche est d'intégrer des connaissances fiables, puisque vécues et validées par les anciens utilisateurs. Dans l'approche proposée, une expérience ou un cas est caractérisée par une requête et ses résultats. Ces résultats devaient être validés par les utilisateurs lors de leurs anciennes recherches. La difficulté majeure de l'approche de réutilisation est de mettre en évidence des contextes ou situations proches (cas en mémoire similaires) de recherche afin de les réutiliser. Cette contrainte nous a amené à combiner plusieurs mesures de similarités afin de sélectionner les cas les plus proches en termes de contenu et de préférences. En effet, plus la similarité entre deux cas est importante, plus les préférences des deux utilisateurs concernés sont similaires. Nous obtenons par la suite un niveau de personnalisation impor-

tant.
Pour finir, nous avons présenté notre système OntoCBRIR pour lequel nous avons mis en œuvre l'approche proposée. La première étude expérimentale, comparant OntoCBRIR avec un système combinant la méthode RàPC et l'intégrale de Choquet, a montré que l'ajout des règles SWRL permet une meilleure adaptation aux préférences utilisateurs, en conséquence une personnalisation supérieure et finalement des recommandations très appropriées. Ces résultats sont renforcés par la deuxième étude expérimentale comparant OntoCBRIR avec un SRI basé sur les mots-clés. Ces travaux on fait l'objet de trois publications [17][112][110].
Enfin, l'approche proposée est indépendante du domaine et sa généricité peut être validée dans des domaine divers. En outre, les solutions proposées dans notre approche peuvent être mises à la disposition des partenaires industriels de différentes manières.

Positionnement

En nous basant sur les caractéristiques des approches basées sur les ontologies [90], nous présentons un tableau comparatif permettant de positionner notre proposition (Tableau 4.8) :

- Enrichissement de l'ontologie : cet aspect a été traité dans les approches [10] et [12] où il s'agit d'enrichir des ontologies modulaires qui correspondent aux différents utilisateurs. Notre approche couvre cet aspect de deux manières. En effet, l'ontologie peut s'enrichir par des concepts déduits automatiquement par le SRI. Ce dernier permet également à l'utilisateur d'intervenir dans ce processus de l'enrichissement en rajoutant des concepts.
- Techniques TAL (traitement automatique du langage) : ces techniques sont utilisées par [36] [55] [10] [12] à plusieurs fins, notamment l'extraction des concepts et leur assortiment avec les sources de données ou pour la désambiguïsation des requêtes. Cet aspect n'a pas été traité

–	Enrichissent de l'ontologie	Techniques TAL	formulation sémantique des requêtes	Personnalisation
[36]	-	☐	☐	-
[55]	-	☐	-	-
[10]	☐	☐	-	☐
[12]	☐	☐	-	☐
[94]	-	-	☐	☐
Notre Approche	☐	-	☐	☐

TABLEAU 4.8 – Comparaison de notre approche avec celles présentées dans le Chapitre 2

dans notre approche.

– Formulation sémantique des requêtes : il s'agit de traiter la requête sous la forme des concepts ou des entités sémantiques. [36] [94] traitent des requêtes (mots-clés) en utilisant des algorithmes de désambiguïsation. Pour faire face à la complexité de l'intégration de ces algorithmes, nous avons adopté la requête visuelle (formulaire) comme alternative.
– Personnalisation : contrairement à la représentation sémantique du contenu informationnel (requêtes, source de données), cet aspect n'a pas été suffisamment traité dans des travaux connexes. Les approches de [10] [12] reposent sur le raisonnement à partir de cas RàPC et les ontologies modulaires pour s'adapter au mieux aux préférences des utilisateurs. Néanmoins, ces approches ne peuvent pas détecter facilement les préférences ou les intérêts des utilisateurs lorsqu'il s'agit d'une recherche basée sur plusieurs critères. Cela peut s'expliquer par l'inexistence d'un modèle de préférence basé sur des opérateurs d'agrégation. Les auteurs de [94] ont proposé un modèle d'agrégation pour inclure l'utilisateur dans la boucle de pertinence. Notons que leur approche atteint sa limite lorsqu'il s'agit d'agréger des sous-ensembles de critères de recherche. Nous avons remédier à cette limite par la combinaison des règles SWRL comme modèle de préférences avec l'intégrale de Choquet comme opérateur d'agrégation.

Perspectives

Si notre approche présente un certain nombre d'avancées, elle présente également un certain nombre de points améliorations qui font partie de nos principales perspectives de recherche :

- Formulation des requêtes : nous souhaitons mettre en place une interface qui permet à l'utilisateur de saisir sa requête en langage naturel avec l'interface visuelle déjà proposée (formulaire). Pour ce faire, nous opterons pour des techniques du TAL (traitement automatique du langage) en utilisant les entités nommées pour analyser la requête.

- Enrichissement de l'ontologie : nous avons présenté une méthode simple d'enrichissement de l'ontologie qui est basée sur l'ajout des relations prédéfinies et limitées. Cependant, nous souhaitons utiliser les techniques d'apprentissage automatique (*Machine Learning*), d'une part pour maintenir la croissance de l'ontologie, d'autre part pour pouvoir l'enrichir par différents types de relations. Cela a déjà fait l'objet d'une nouvelle thèse dans notre équipe, elle traitent également un autre aspect qui est la correspondance entre les ontologies (*Ontology Matching*).

- Raisonnement à partir de cas : dans le cycle RàPC actuel et une fois le cas le plus similaire est récupéré, sa solution est présentée à l'utilisateur. En effet, dans la phase d'apprentissage le système procède à l'ajout du nouveau cas avec sa solution dans la base de cas réelle, ainsi qu'à une mise à jour périodique de la base de cas virtuelle. La périodicité exacte de la mise à jour n'est pas encore fixée dans notre travail. Cependant nous souhaitons développer des algorithmes pour le maintien de la base de cas d'une manière automatique.

- Personnalisation : notre approche peut déterminer les informations relatives à l'utilisateur (préférences) en utilisant, d'une part les nouveaux

concepts (concepts topiques) générés par les règles SWRL, d'autre part l'intégrale de Choquet comme opérateur d'agrégation pour s'approcher des cas similaires. En effet, nous n'avons pas un modèle utilisateur structuré et évolutif. Nous souhaitons donc construire une base de profils utilisateur pour améliorer le niveau de la personnalisation.

– Expérimentations : les expérimentations menées sont limitées du fait que notre objectif a été, en premier temps, de valider la mise en œuvre de notre approche en termes de réalisation et de faisabilité. Cependant, nous souhaitons procéder à une validation en intégrant des bases de connaissances en ligne, à l'instar de DBpedia [2] et YAGO [3]. Concernant la recherche documentaire, nous souhaitons tester notre approche en utilisant les jeux de données dédiés à la recherche d'information comme TREC [4].

2. http ://fr.dbpedia.org/
3. http ://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
4. http ://trec.nist.gov/

Annexes

```
public void getAllTriples(String owlfilename) throws FileNotFoundException{
    // list the statements in the Model
     OntModel model = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM);
     InputStream in = new FileInputStream(owlfilename);
     model.read(in,null);
     model.prepare();
     StmtIterator iter = model.listStatements();

     // print out the predicate, subject and object of each statement
     PrintStream out=new PrintStream(new File("triples.txt"));
     while (iter.hasNext()) {
         Statement stmt      = iter.nextStatement();  // get next statement
         Resource  subject   = stmt.getSubject();     // get the subject
         Property  predicate = stmt.getPredicate();   // get the predicate
         RDFNode   object    = stmt.getObject();      // get the object


        out.print(subject.toString());
         out.print(" " + predicate.toString() + " ");
         if (object instanceof Resource) {
            out.print(object.toString());
         } else {
             // object is a literal
             out.print(" \"" + object.toString() + "\"");
         }

         out.println(" .");
     }
```

FIGURE 4.12 – Cette capture d'écran concerne le chapitre 6, il s'agit de lister les triplets des entités sémantiques à partir de la requête saisie par l'utilisateur via le formulaire.

```
    enumerateAllInstances(ontology, reasoner);
    OWLObjectRenderer renderer = new DLSyntaxObjectRenderer();
    // pouplateOntology(ontology, factory, manager);
    useRule(ontology, factory, manager, reasoner);
    String ontologyIRI = "http://www.semanticweb.org/ontologies/2011/3/city_logistic_ontology.owl";
    for (SWRLRule rule : ontology.getAxioms(AxiomType.SWRL_RULE)) {
        System.out.println("Rule: " + renderer.render(rule));
        OWLClass clsA = factory.getOWLClass(IRI.create(ontologyIRI
                + "Shipper_receiver_pattern"));
        String ii=null;
        for (OWLNamedIndividual i : reasoner.getInstances(clsA, true)
                .getFlattened()) {
            ii=i.toString();
        }System.out.println("Inferred Instances from:"+ clsA.toString()+" is: "+ii.toString());
    }
} catch (OWLOntologyCreationException e) {

    e.printStackTrace();
}
```

FIGURE 4.13 – Cette capture d'écran concerne le chapitre 6, elle illustre un exemple d'affichage des instances à partir d'une règle SWRL.

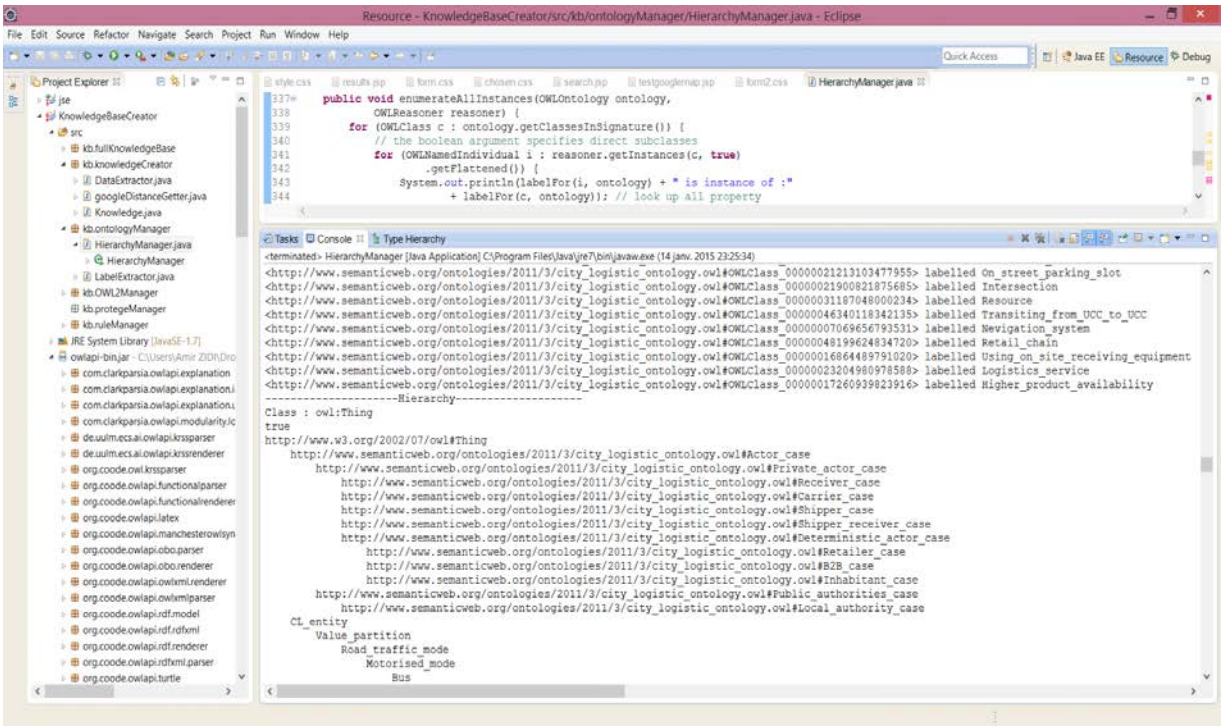


FIGURE 4.14 – Cette capture d'écran concerne le chapitre 6, elle illustre l'affichage de la base de connaissances.

```
public void setW(IndexWriter w) {
    LuceneManager.w = w;
}
private static IndexWriter w=null;
private double sessionTime=0;

public  void indexKnowledgeBase(String indexdirectory, String fileName) throws IOException, ParseException {
   // 0. Specify the analyzer for tokenizing text.
    //    The same analyzer should be used for indexing and searching
    StandardAnalyzer analyzer = new StandardAnalyzer(Version.LUCENE_40);
    // 1. create the index
    File f=new File(indexdirectory);
    FSDirectory index =  FSDirectory.open(f);

    IndexWriterConfig config = new IndexWriterConfig(Version.LUCENE_40, analyzer);
    if (w == null) {
     w = new IndexWriter(index, config);
    }
    knowledgeBase kb= new knowledgeBase( fileName);


   for (int i=0; i<kb.getCaseBase().size();i++){
        KnowledgePattern cas=kb.getCaseBase().elementAt(i);

       indexDocument(cas);
   }
   if (w!= null) {
    w.close();
   }
}
public ArrayList<UserBean> search(String query, float distance, float Time
```

FIGURE 4.15 – Cette capture d'écran concerne le chapitre 6, il s'agit d'une fonction qui permet l'indexation de la base de connaissances en utilisant *Lucene*.

```
            System.out.println("APVsm "+averagePrecisionVsmLoc);
            System.out.println("APGs: "+averagePrecisionGsLoc);
            averagePrecisionVsmTot=averagePrecisionVsmTot+averagePrecisionVsmLoc;
            averagePrecisionGsTot=averagePrecisionGsTot+averagePrecisionGsLoc;
            System.out.println("=============================================================================================");
            System.out.println("nbRelGs "+p3.getNbRleventRetGs()+"nbRelVsm "+p3.getNbReleventRetVsm());
            System.out.println("APVsm "+averagePrecisionVsmLoc);
            System.out.println("APGs: "+averagePrecisionGsLoc);
            System.out.println(p1.getNbReleventRetVsm()+ " "+p1.getNbRleventRetGs()+" "+p1.getNbReleventDoc());
            System.out.println(p1.getNbReleventRetVsm()+ " "+p2.getNbRleventRetGs()+" "+p2.getNbReleventDoc());
            System.out.println(p1.getNbReleventRetVsm()+ " "+p1.getNbRleventRetGs()+" "+p3.getNbReleventDoc());
            System.out.println(p2.getPrecisionGs()+" "+p2.getNbReleventRetVsm());
            System.out.println(p3.getPrecisionGs()+" "+p3.getNbReleventRetVsm());

            System.out.println(p1.getNbReleventRetVsm()/20+ " "+p2.getNbReleventRetVsm()/50+" "+p3.getNbReleventRetVsm()/100);
            System.out.println(p1.getNbRleventRetGs()/20+ " "+p2.getNbRleventRetGs()/50+" "+p3.getNbRleventRetGs()/100);


            System.out.println("APVsm "+averagePrecisionVsmTot);
            System.out.println("APGs: "+averagePrecisionGsTot);




        System.out.println("MAP@ for Vsm= "+averagePrecisionVsmTot/10d);
        System.out.println("MAP@ for Gs= "+averagePrecisionGsTot/10d);

    *
    * get score maximum for normalization
    */
```

FIGURE 4.16 – Cette capture d'écran concerne le chapitre 6, il s'agit d'un code qui permet d'afficher les valeurs de la moyenne de précision/rappel dans les premiers 20, 50 et 100 résultats retournés.

```
public Query getFullerQuery() {
    final SirenTupleQuery tq = new SirenTupleQuery();
     tq.add(termInCell("object property", COLUMN_TUPLE_TYPE), SirenTupleClause.Occur.SHOULD);
     tq.add(termInCell("POINT ARRET berthollet", COLUMN_PERSON_NAME), SirenTupleClause.Occur.SHOULD);
      tq.add(termInCell("is_localised", COLUMN_BIRTH_DATE), SirenTupleClause.Occur.SHOULD);
      tq.add(termInCell("NOEUD RESEAU ROUTIER 11", COLUMN_BIRTH_PLACE), SirenTupleClause.Occur.SHOULD);
  // Create a tuple query that combines the two cell queries
  final SirenTupleQuery tq1 = new SirenTupleQuery();

  final SirenTupleQuery tq2 = new SirenTupleQuery();
  tq2.add(termInCell("object property", COLUMN_TUPLE_TYPE), SirenTupleClause.Occur.SHOULD);
  tq2.add(termInCell("POINT ARRET berthollet", COLUMN_PERSON_NAME), SirenTupleClause.Occur.SHOULD);
  tq2.add(termInCell("is_encercled_by", COLUMN_BIRTH_DATE), SirenTupleClause.Occur.SHOULD);
  tq2.add(termInCell("LOISIR 1", COLUMN_BIRTH_PLACE), SirenTupleClause.Occur.SHOULD);

  // Combine two tuple queries with a Lucene boolean query
  final BooleanQuery q = new BooleanQuery();
  q.add(tq, Occur.MUST);
  List<BooleanClause> list=q.clauses();
  for(BooleanClause str:list)
  System.out.println(str.toString());
  q.add(tq2, Occur.MUST);

  return q;
}
```

FIGURE 4.17 – Cette capture d'écran concerne la travail effectué dans le chapitre 4, il s'agit d'analyser la requête saisie par l'utilisateur (mot clés) sous le forme des clauses booléennes en utilisant *SIREn Lucene*.

Bibliographie

[1] X. Aimé, F. Fürst, P. Kuntz, and F. Trichet. Enrichissement sémantique de requêtes au moyen d'ontologies de domaine personnalisées. *Actes de l'atelier «Personnalisation du Web», 10ième Journées francophones d'Extraction et de Gestion de Connaissances (EGC'2010)*, 2010.

[2] N. Anand, M. Yang, J. Van Duin, and L. Tavasszy. Genclon : An ontology for city logistics. *Expert Systems with Applications*, 39(15) :11944–11960, 2012.

[3] S. S. Anand, P. Kearney, and M. Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Transactions on Internet Technology (TOIT)*, 7(4) :22, 2007.

[4] M. Arora, U. Kanjilal, and D. Varshney. Successful efficient and intelligent retrieval using analytic hierarchy process. In *Agents and Data Mining Interaction*, pages 331–343. Springer, 2012.

[5] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The description logic handbook : theory, implementation, and applications*. Cambridge University Press, New York, NY, USA, 2003.

[6] F. Bacha. *Une approche MDA pour l'intégration de la personnalisation du contenu dans la conception et la génération des applications interactives*. PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2013.

[7] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[8] M. Baziz. *Indexation conceptuelle guidée par ontologie pour la recherche d'information*. PhD thesis, Toulouse 3, 2005.

[9] M. Baziz, M. Boughanem, N. Aussenac-Gilles, and C. Chrisment. Semantic cores for representing documents in ir. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1011–1017. ACM, 2005.

[10] N. Ben Mustapha, H. B. Zghal, M.-A. Aufaure, and H. ben Ghezala. Semantic search using modular ontology learning and case-based reasoning. In *Proceedings of the 2010 EDBT/ICDT Workshops*, page 3. ACM, 2010.

[11] A. Bernstein and E. Kaufmann. Gino–a guided input natural language ontology editor. In *The Semantic Web-ISWC 2006*, pages 144–157. Springer, 2006.

[12] G. Besbes, H. B. Zghal, and H. H. B. Ghézela. Un système hybride de recherche d'information intégrant le raisonnement à partir de cas et la composition d'ontologies. In *EGC*, pages 269–274, 2013.

[13] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli. *Hybrid search : Effectively combining keywords and semantic searches*. Springer, 2008.

[14] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information processing & management*, 43(4) :866–886, 2007.

[15] Y. Blanco-Fernández, J. J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López-Nores, J. García-Duque, A. Fernández-Vilas, R. P. Díaz-Redondo, and J. Bermejo-Muñoz. A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems*, 21(4) :305–320, 2008.

[16] A. Bouhana, A. Fekih, M. Abed, and H. Chabchoub. An integrated case-based reasoning approach for personalized itinerary search in multimodal transportation systems. *Transportation Research Part C : Emerging Technologies*, 31 :30–50, 2013.

[17] A. Bouhana, A. Zidi, A. Fekih, H. Chabchoub, and M. Abed. An ontology-based cbr approach for personalized itinerary search systems for sustainable urban freight transport. *Expert Systems with Applications*, 2014.

[18] D. Brickley and R. V. Guha. {RDF vocabulary description language 1.0 : RDF schema}. 2004.

[19] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1) :1, 2012.

[20] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2) :261–272, 2007.

[21] G. Choquet. Theory of capacities. In *Annales de l ?institut Fourier*, volume 5, page 54, 1953.

[22] S. Clinchant and E. Gaussier. Modélisation probabiliste de collections textuelles et distributions de mots. *Revue des Nouvelles Technologies de l'Information - Apprentissage Automatique et Fouille de Données. Eds Y. Bennani et E. Viennet*, 2009.

[23] C. Comparot, O. Haemmerlé, and N. Hernandez. Production de requêtes sparql à partir de mots-clés et de patrons de requêtes. *TSI. Technique et science informatiques*, 32(7-8) :841–861, 2013.

[24] O. Corby, R. Dieng-Kuntz, and C. Faron-Zucker. Querying the semantic web with corese search engine. In *ECAI*, volume 16, page 705, 2004.

[25] J. S. Culpepper and A. Moffat. Efficient set intersection for inverted indexing. *ACM Transactions on Information Systems (TOIS)*, 29(1) :1, 2010.

[26] H. Cunningham, A. Hanbury, and S. Rüger. Scaling up high-value retrieval to medium-volume data. In *Advances in Multidisciplinary Retrieval*, pages 1–5. Springer, 2010.

[27] M. Daoud, L. Tamine, and M. Boughanem. A personalized graph-based document ranking model using a semantic user profile. In *User Modeling, Adaptation, and Personalization*, pages 171–182. Springer, 2010.

[28] M. Daoud, L. Tamine-Lechani, M. Boughanem, and B. Chebaro. A session based personalized search using an ontological user profile. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1732–1736. ACM, 2009.

[29] R. Delbru, S. Campinas, and G. Tummarello. Searching web data : An entity retrieval and high-performance indexing model. *Web Semantics : Science, Services and Agents on the World Wide Web*, 10 :33–58, 2012.

[30] D. Dinh and L. Tamine. Combining global and local semantic contexts for improving biomedical information retrieval. In *Advances in Information Retrieval*, pages 375–386. Springer, 2011.

[31] D. Dinh and L. Tamine. Towards a context sensitive approach to searching information based on domain specific knowledge sources. *Web Semantics : Science, Services and Agents on the World Wide Web*, 12 :41–52, 2012.

[32] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590. ACM, 2007.

[33] M. Dragoni, C. da Costa Pereira, and A. G. Tettamanzi. A conceptual representation of documents and queries for information retrieval systems by using light ontologies. *Expert Systems with applications*, 39(12) :10376–10388, 2012.

[34] O. Dridi. Ontology-based information retrieval : Overview and new proposition. In *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on*, pages 421–426. IEEE, 2008.

[35] M. Farah and D. Vanderpooten. L'agrégation en recherche d'information : une revue critique des principaux modèles théoriques de recherche d'information. 2007.

[36] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta. Semantically enhanced information retrieval : an ontology-based approach. *Web Semantics : Science, Services and Agents on the World Wide Web*, 9(4) :434–452, 2011.

[37] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic search meets the web. In *Semantic Computing, 2008 IEEE International Conference on*, pages 253–260. IEEE, 2008.

[38] G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, and G. Antoniou. Ontology change : Classification and survey. *The Knowledge Engineering Review*, 23(02) :117–152, 2008.

[39] M. françois Sy et al. *Utilisation d'ontologies comme support à la recherche et à la navigation dans une collection de documents*. PhD thesis, Montpellier 2, 2012.

[40] F. Gandon, O. Corby, and C. Faron-Zucker. *Le Web sémantique : comment lier les données et les schémas sur le web ?* Dunod, 2012.

[41] M. R. Ghorab, D. Zhou, A. O ?Connor, and V. Wade. Personalised information retrieval : survey and classification. *User Modeling and User-Adapted Interaction*, 23(4) :381–443, 2013.

[42] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang. Hermit : an owl 2 reasoner. *Journal of Automated Reasoning*, pages 1–25, 2013.

[43] S. Grimm and S. Grimm. Knowledge representation and ontologies. In *Semantic Web Services*. Citeseer, 2007.

[44] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220, 1993.

[45] N. Gustafson and Y.-K. Ng. Augmenting data retrieval with information retrieval techniques by using word similarity. In *Natural Language and Information Systems*, pages 163–174. Springer, 2008.

[46] S. Handschuh and S. Staab. Authoring and annotation of web pages in cream. In *Proceedings of the 11th international conference on World Wide Web*, pages 462–473. ACM, 2002.

[47] R. Harrathi. Recherche d ?information conceptuelle dans les documents semi-structurés. *month*, 2010.

[48] M. Hearst. *Search user interfaces*. Cambridge University Press, 2009.

[49] N. Hernandez. *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. PhD thesis, Université Paul Sabatier-Toulouse III, 2005.

[50] M. Hildebrand, J. van Ossenbruggen, and L. Hardman. An analysis of search-based user interaction on the semantic web. *Information Systems [INS]*, (E0706), 2007.

[51] P. Hitzler and K. Janowicz. Semantic web–interoperability, usability, applicability. *Semantic Web*, 1(1) :1–2, 2010.

[52] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. Swrl : A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21 :79, 2004.

[53] L. Jéribi, B. Rumpler, and J.-M. Pinon. Système d'aide à la recherche et à l'interrogation de bases documentaires, fondé sur la réutilisation d'expériences. In *INFORSID*, pages 443–463, 2001.

[54] X. Jiang and A.-H. Tan. Learning and inferencing in user ontology for personalized semantic web search. *Information sciences*, 179(16) :2794–2808, 2009.

[55] S. Kara, Ö. Alan, O. Sabuncu, S. Akpınar, N. K. Cicekli, and F. N. Alpaslan. An ontology-based retrieval system using semantic indexing. *Information Systems*, 37(4) :294–305, 2012.

[56] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM (JACM)*, 42(4) :741–843, 1995.

[57] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics : Science, Services and Agents on the World Wide Web*, 2(1) :49–79, 2004.

[58] D. Kostadinov. *Personnalisation de l'information : une approche de gestion de profils et de reformulation de requêtes*. PhD thesis, Université de Versailles-Saint Quentin en Yvelines, 2007.

[59] V. P. Lavrenko. Introduction to probabilistic models in ir. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 905–905. ACM, 2010.

[60] Y. Lei, V. Uren, and E. Motta. Semsearch : A search engine for the semantic web. In *Managing Knowledge in a World of Networks*, pages 238–245. Springer, 2006.

[61] V. Lopez, M. Pasin, and E. Motta. Aqualog : An ontology-portable question answering system for the semantic web. In *The Semantic Web : Research and Applications*, pages 546–562. Springer, 2005.

[62] V. Lopez, V. Uren, M. Sabou, and E. Motta. Is question answering fit for the semantic web ? : a survey. *Semantic Web*, 2(2) :125–155, 2011.

[63] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[64] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[65] F. Manola, E. Miller, B. McBride, et al. Rdf primer. *W3C recommendation*, 10(1-107) :6, 2004.

[66] D. L. McGuinness, R. Fikes, J. Hendler, and L. A. Stein. Daml+ oil : an ontology language for the semantic web. *Intelligent Systems, IEEE*, 17(5) :72–80, 2002.

[67] D. L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10) :2004, 2004.

[68] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1) :54–88, 2004.

[69] G. A. Miller. Wordnet : a lexical database for english. *Communications of the ACM*, 38(11) :39–41, 1995.

[70] A. Moffat and J. Zobel. Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems (TOIS)*, 14(4) :349–379, 1996.

[71] B. Moulahi, L. Tamine, and S. B. Yahia. L'intégrale de choquet discrète pour l'agrégation de pertinence multidimensionnelle. In *CORIA*, pages 399–414, 2013.

[72] Y. Mrabet. *Approches hybrides pour la recherche sémantique de l'information : intégration des bases de connaissances et des ressources semi-structurées*. PhD thesis, Université Paris Sud-Paris XI, 2012.

[73] Y. Mrabet, N. Bennacer, N. Pernelle, M. Thiam, et al. Une approche pour la recherche sémantique de l'information dans les documents semi-structurés hétérogènes. In *COnférence en Recherche d'Infomations et Applications-CORIA 2010, 7th French Information Retrieval Conference, Sousse, Tunisia, March 18-20, 2010. Proceedings.*, pages 195–210, 2010.

[74] P. Mylonas, D. Vallet, P. Castells, M. Fernández, and Y. S. Avrithis. Personalized information retrieval based on context and ontological knowledge. *Knowledge Eng. Review*, 23(1) :73–100, 2008.

[75] S. A. Noah, L. Zakaria, and A. C. Alhadi. Extracting and modeling the semantic information content of web documents to support semantic document retrieval. In *Proceedings of the Sixth Asia-Pacific Conference on Conceptual Modeling-Volume 96*, pages 79–86. Australian Computer Society, Inc., 2009.

[76] V. Nováček, T. Groza, S. Handschuh, and S. Decker. Coraalâ€"dive into publications, bathe in the knowledge. *Web Semantics : Science, Services and Agents on the World Wide Web*, 8(2) :176–181, 2010.

[77] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. *IEEE intelligent systems*, 16(2) :60–71, 2001.

[78] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, and E. Zavitsanos. Ontology population and enrichment : State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution*, pages 134–166. Springer-Verlag, 2011.

[79] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Towards semantic web information extraction. In *Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003)*, volume 20, 2003.

[80] A. Pretschner and S. Gauch. Ontology based personalized search. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 391–398. IEEE, 1999.

[81] B. Ricardo and R. Berthier. Modern information retrieval : the concepts and technology behind search second edition. *Addision Wesley*, 2011.

[82] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3) :129–146, 1976.

[83] C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383. ACM, 2004.

[84] M. Á. Rodríguez-García, R. Valencia-García, and F. García-Sánchez. An ontology evolution-based framework for semantic information retrieval. In *On the Move to Meaningful Internet Systems : OTM 2012 Workshops*, pages 163–172. Springer, 2012.

[85] G. Salton and M. J. McGill. Introduction to modern information retrieval. 1983.

[86] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Information Processing & Management*, 33(2) :193–207, 1997.

[87] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11) :613–620, 1975.

[88] L. Sbattella and R. Tedesco. A novel semantic information retrieval system based on a three-level domain model. *Journal of Systems and Software*, 86(5) :1426–1452, 2013.

[89] P. Scheir, S. N. Lindstaedt, and C. Ghidini. A network model approach to retrieval in the semantic web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 4(4) :56–84, 2008.

[90] K. Schoefegger, T. Tammet, and M. Granitzer. A survey on socio-semantic information retrieval. *Computer Science Review*, 8 :25–46, 2013.

[91] L. Shi and R. Setchi. Ontology-based personalised retrieval in support of reminiscence. *Knowledge-Based Systems*, 45 :47–61, 2013.

[92] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet : A practical owl-dl reasoner. *Web Semantics : science, services and agents on the World Wide Web*, 5(2) :51–53, 2007.

[93] W. Song, J. Z. Liang, X. L. Cao, and S. C. Park. An effective query recommendation approach using semantic strategies for intelligent information retrieval. *Expert Syst. Appl.*, 41(2) :366–372, 2014.

[94] M.-F. Sy, S. Ranwez, J. Montmain, A. Regnault, M. Crampes, and V. Ranwez. User centered and ontology based information retrieval system for life sciences. *BMC bioinformatics*, 13(Suppl 1) :S4, 2012.

[95] T. Tegegne and T. P. van der Weide. Enriching queries with user preferences in healthcare. *Information Processing & Management*, 50(4) :599–620, 2014.

[96] M. Thangaraj and G. Sujatha. An architectural design for effective information retrieval in semantic web. *Expert Systems with Applications*, 41(18) :8225–8233, 2014.

[97] T. Tran, P. Cimiano, S. Rudolph, and R. Studer. *Ontology-based interpretation of keywords for semantic search.* Springer, 2007.

[98] D. Tsarkov and I. Horrocks. Fact++ description logic reasoner : System description. In *Automated reasoning*, pages 292–297. Springer, 2006.

[99] V. Uren, Y. Lei, V. Lopez, H. Liu, E. Motta, and M. Giordanino. The usability of semantic search tools : a review. *The Knowledge Engineering Review*, 22(04) :361–377, 2007.

[100] D. Vallet. Personalized information retrieval in context using ontological knowledge. *Advanced Studies Diploma–EPS-UAM*, 2007.

[101] J. Z. Wang and W. Taylor. Concept forest : A new ontology-assisted text document similarity measurement method. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pages 395–401. IEEE, 2007.

[102] W. Wong, W. Liu, and M. Bennamoun. An ontology-based interface for improving information exploration. In *Proceedings of the first international workshop on Intelligent visual interfaces for text analysis*, pages 29–32. ACM, 2010.

[103] B. S. Wynar, A. G. Taylor, and J. Osborn. *Introduction to cataloging and classification*. Libraries Unlimited, 1985.

[104] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl. From keywords to semantic queriesâ€"incremental query construction on the semantic web. *Web Semantics : Science, Services and Agents on the World Wide Web*, 7(3) :166–176, 2009.

[105] J. Zhai, Y. Liang, Y. Yu, and J. Jiang. Semantic information retrieval based on fuzzy ontology for electronic commerce. *Journal of Software*, 3(9) :20–27, 2008.

[106] J. Zhai, Q. Wang, and M. Lv. Application of fuzzy ontology framework to information retrieval for scm. In *Information Processing (ISIP), 2008 International Symposiums on*, pages 173–177. IEEE, 2008.

[107] L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang. An enhanced model for searching in semantic portals. In *Proceedings of the 14th international conference on World Wide Web*, pages 453–462. ACM, 2005.

[108] D. Zhou, S. Lawless, and V. Wade. Improving search via personalized query expansion using social media. *Information retrieval*, 15(3-4) :218–242, 2012.

[109] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu. *SPARK : adapting keyword query to semantic search*. Springer, 2007.

[110] A. Zidi. Personalization of itineraries search using ontology and rules to avoid congestion in urban areas. In E. Boje, editor, *Proceedings of the 19th IFAC World Congress*. IFAC, Elsevier, aug 2014.

[111] A. Zidi and M. Abed. Towards a framework for ontology-based information retrieval services. *International Journal of Services and Operations Management*, 19(2) :138–150, 2014.

[112] A. Zidi, A. Bouhana, M. Abed, and A. Fekih. An ontology-based personalized retrieval model using case base reasoning. In *18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2014, Gdynia, Poland, 15-17 September 2014*, pages 213–222, 2014.