



HAL
open science

Interrogation d'un réseau sémantique de documents : l'intertextualité dans l'accès à l'information juridique

Nada Mimouni

► **To cite this version:**

Nada Mimouni. Interrogation d'un réseau sémantique de documents : l'intertextualité dans l'accès à l'information juridique. Informatique et langage [cs.CL]. Université Sorbonne Paris Cité, 2015. Français. NNT : 2015USPCD084 . tel-02099789

HAL Id: tel-02099789

<https://theses.hal.science/tel-02099789v1>

Submitted on 15 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interrogation d'un réseau sémantique de documents : l'intertextualité dans l'accès à l'information juridique

THÈSE

présentée et soutenue publiquement le 27 Janvier 2015

pour l'obtention du

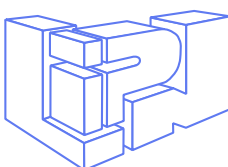
Doctorat de l'Université Paris 13 – Sorbonne Paris Cité
(spécialité informatique)

par

Nada Mimouni

Composition du jury

<i>Rapporteurs :</i>	Sylvie Calabretto	Professeur, INSA de Lyon
	Ollivier Haemmerlé	Professeur, Université Toulouse - Jean Jaurès
<i>Examineurs :</i>	Danièle Bourcier	Directrice de recherche CNRS, CERSA - Paris
	Aldo Gangemi	Professeur, Université Paris 13
	Amedeo Napoli	Directeur de recherche CNRS, LORIA - Nancy
	Chantal Reynaud	Professeur, Université Paris Sud
<i>Encadrants :</i>	Adeline Nazarenko	Professeur, Université Paris 13 (directrice)
	Sylvie Salotti	Maître de conférences, Université Paris 13 (co-encadrante)



Remerciements

En tout premier lieu, j'aime exprimer ma vive gratitude et reconnaissance à ma directrice de thèse Adeline Nazarenko qui m'a aidée et guidée durant toutes les étapes de mon travail. Ses conseils étaient très précieux tout le long de mon aventure et allaient bien au-delà de l'obtention d'un titre universitaire et l'accomplissement d'un projet de recherche. Merci pour le professionnalisme, la motivation et surtout pour la patience. Merci de m'avoir enseignée que les nouvelles idées sortent d'un esprit ouvert et évoluent en s'ouvrant sur les idées des autres.

Je suis particulièrement reconnaissante pour le soutien et les conseils de ma co-encadrante Sylvie Salotti, pour l'enthousiasme et l'encouragement et d'avoir eu confiance en moi. Merci surtout pour la touche personnelle qui a fait que cette aventure soit sereine et rendu le travail plus facile.

Je tiens également à remercier tous les partenaires académiques et industriels du projet Légilocal, en particulier Danièle Bourcier, Meritxell Fernandez et Ève Paul, les experts juristes du projet, pour toutes les discussions et les précieux conseils qui m'ont aidé à découvrir un domaine qui m'était inconnu auparavant.

Mes remerciements vont également à M. Claudio Fabiani, chef d'unité au Parlement Européen et responsable du projet IT4AM, M. Michel Brogard, chef de l'unité production documentaire, et M. Pierre Henrard, responsable de l'atelier SGML/XML du PE à Luxembourg, pour leur disponibilité et leur collaboration.

Je tiens à remercier tous les membres passés et présents de l'équipe RCLN pour avoir été de bons collègues et pour leur attitude amicale ainsi que tous les membres du laboratoire LIPN en particulier Faouzi Boufarès, Pierre Boudes et tous ceux avec qui j'ai travaillé sur des modules d'enseignement. Mes remerciements vont également à Brigitte Guéveneux, Marie Fontanillas et les autres secrétaires du LIPN, très débordées mais toujours souriantes et disponibles.

Merci à mes collègues doctorants Nouha Omrane, Sarra Ben Abbes, Aicha Ben Salem, Leila Abidi, Sondes Bannour, Ines Bannour, Ines Chebil, Hanène Ochi, Manisha Pujari, Abdoulaye Guisse, Zied Yakoubi, Ehab Hassan et tous les autres : vous avez fait des petites pauses un vrai moment de plaisir dans les longues journées de travail. Je vous souhaite tout le meilleur pour un avenir plein de réalisations.

Je ne manque pas de remercier vivement toute personne qui a participé, de près ou de loin, à la bonne réalisation de ce travail et son déroulement dans les meilleures conditions.

Enfin, je remercie chaleureusement mes parents, mes sœurs et mon frère qui ont toujours su être présents quand il le fallait. Qu'ils trouvent tous ici l'expression de mon profond respect.

Cela va sans dire, personne n'a été plus important dans la poursuite de ma thèse que mon mari..

*Je dédie ce travail
à ma mère, à mon père,
à Nizar et Lilou*

Table des matières

Table des figures	xi
Liste des tableaux	xv
1 Introduction	1
1.1 Contexte général	1
1.1.1 Recherche d'information	1
1.1.2 Recherche d'information sémantique et sur le web	2
1.2 Contexte applicatif : le projet Légilocal	2
1.3 Enjeux de la recherche d'information juridique	3
1.4 Objectif et contributions	5
1.5 Structure du document	6
2 Accès à l'information juridique	9
2.1 Introduction	9
2.2 Caractéristiques des données juridiques	10
2.2.1 Structure et hiérarchie des sources de loi	10
2.2.2 Réseau de réglementations et complexité juridique	11
2.3 Efforts de structuration de l'information juridique	14
2.3.1 Création ou édition de la réglementation	14
2.3.2 Représentation des documents	15
2.3.3 Ontologies du droit	17
2.3.4 Synthèse	20
2.4 Méthodes d'accès à l'information juridique	20
2.4.1 Portails généralistes de sources de droit	20
2.4.2 Outils spécialisés	22
2.4.3 Données gouvernementales ouvertes sur le web	24
2.5 Traitement de l'intertextualité	24
2.6 Conclusion	26

3	Recherche d'information et graphe de documents	29
3.1	Introduction	29
3.2	Recherche d'information classique	30
3.2.1	Indexation ou processus de représentation	30
3.2.2	Appariement ou processus de recherche	31
3.2.3	Tri de résultats	32
3.2.4	Reformulation de requêtes	32
3.2.5	Modèles de RI	32
3.2.6	Mesures d'évaluation	33
3.2.7	Interface utilisateur	33
3.3	Recherche d'information sémantique	34
3.3.1	Annotation sémantique	34
3.3.2	Modèles de RI numériques et à base de connaissances	35
3.3.3	Modèles logiques de RI	37
3.4	RI et Analyse de liens	38
3.4.1	Intertextualité dans les systèmes de RI existants	38
3.4.2	Analyse de graphes de citation	39
3.4.3	Analyse des liens hypertextes (algorithmes Page Rank et HITS)	39
3.4.4	Analyse socio-sémantique	40
3.5	Conclusion	40
4	Méthodes pour la modélisation et l'interrogation de données complexes	43
4.1	Introduction	43
4.2	AFC et ARC : fondements théoriques	45
4.2.1	Notions de base de la théorie des treillis	45
4.2.2	L'Analyse Formelle de Concepts	46
4.2.3	L'Analyse Relationnelle de Concepts	52
4.3	Applications de l'AFC et ARC	60
4.4	Web sémantique et web de données	62
4.4.1	Les technologies du web sémantique	63
4.4.2	Le web de données et les données liées sur le web	69
4.4.3	Les ontologies	70
4.5	Application à l'analyse documentaire dans le web sémantique	73
4.5.1	Vocabulaires conceptuels et annotation sémantique	73
4.5.2	Ontologies documentaires	74
4.6	Synthèse	75

5	Interrogation d'un réseau sémantique de documents : application aux sources de droit	79
5.1	Introduction	79
5.2	L'enjeu de l'intertextualité dans Légilocal	80
5.2.1	Objectif de la thèse	80
5.2.2	Intertextualité dans les sources de droit	81
5.3	Modélisation des collections documentaires	83
5.3.1	Caractéristiques des collections documentaires	83
5.3.2	Les collections comme graphes de documents	83
5.3.3	Exemples de collections juridiques	84
5.4	Interrogation des collections documentaires	88
5.4.1	Langage de requêtes	89
5.4.2	Exemples	90
5.4.3	Analyse des besoins des juristes	91
5.4.4	Jeu de requêtes types	97
5.4.5	Discussion	99
5.5	Conclusion	100
6	RI et intertextualité : approche conceptuelle	101
6.1	Introduction	101
6.2	Collection documentaire et choix de modélisation	102
6.3	Modélisation du contenu sémantique par l'AFC	103
6.3.1	Construction des treillis formels	104
6.3.2	Interprétation des structures conceptuelles	105
6.4	Modélisation des liens intertextuels par l'ARC	107
6.4.1	Modèle de données	107
6.4.2	Construction des treillis relationnels	108
6.4.3	Interprétation de la structure relationnelle	109
6.4.4	Modèle de la collection documentaire	111
6.5	Interrogation du modèle documentaire	111
6.5.1	Stratégie de recherche dans le modèle documentaire	112
6.5.2	Requêtes simples	113
6.5.3	Requêtes relationnelles	114
6.5.4	Déroulement sur un exemple	117
6.6	Navigation dans la structure conceptuelle	119
6.6.1	Raffinement et expansion des résultats	120
6.6.2	Recherche par exemple de documents	123

6.6.3	Recherche de réponses approchées	127
6.7	Algorithmes d'interrogation et de navigation	129
6.8	Requêtes exprimables par le modèle	132
6.9	Conclusion	136
7	RI et intertextualité : approche sémantique	139
7.1	Introduction	139
7.2	Bonnes pratiques pour la construction de vocabulaires	140
7.3	Première ontologie documentaire	141
7.3.1	Structure globale de l'ontologie	142
7.3.2	Modélisation de la collection documentaire	144
7.3.3	Modélisation des documents	147
7.3.4	Modélisation sémantique des contenus textuels	152
7.4	Deuxième ontologie documentaire	154
7.4.1	Gestion des versions d'un document	156
7.4.2	Gestion des références	156
7.4.3	Structure globale de l'ontologie	163
7.4.4	Positionnement par rapport au standard juridique Metalex	165
7.5	Mise en œuvre des ontologies documentaires	166
7.5.1	Instanciation et interrogation dans la première ontologie	166
7.5.2	Instanciation et interrogation dans la deuxième ontologie	173
7.6	Conclusion	179
8	Experimentation	181
8.1	Introduction	181
8.2	Corpus OIT	182
8.2.1	Description du corpus	182
8.2.2	Requêtes OIT et réponses pertinentes	183
8.2.3	Approche conceptuelle : AFC/ARC	183
8.2.4	Approche sémantique : première ontologie	187
8.2.5	Discussion	191
8.3	Corpus LÉGILOCAL	191
8.3.1	Description du corpus	191
8.3.2	Requêtes LÉGILOCAL et réponses pertinentes	193
8.3.3	Exécution sur la première ontologie documentaire	193
8.3.4	Exécution sur la deuxième ontologie documentaire	198
8.3.5	Discussion	201

9 Conclusion et perspectives	203
9.1 Conclusion	203
9.2 Perspectives	204
Bibliographie	207

Table des figures

2.1	Activités d'un système d'information législatif [Sartor et al., 2011].	10
2.2	Hiérarchie des sources de loi.	12
2.3	Exemple de documents juridiques et de types de liens qui existent entre eux.	12
2.4	Extrait d'un document décrit en Metalex	16
2.5	Extrait d'un document décrit en Metalex : identification de références	16
2.6	La plate-forme Légilocal [Amardeilh et al., 2013].	23
3.1	Vue générale d'un système de recherche d'information.	31
4.1	Le treillis de concepts \mathcal{L}_P correspondant au contexte formel \mathcal{K}_P donné dans la table 4.1.	50
4.2	Le treillis de concepts \mathcal{L}_F correspondant au contexte formel \mathcal{K}_F donné dans la table 4.2.	51
4.3	Le treillis relationnel $\mathcal{L}_{P_F}^{\forall,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage universel par la relation "Like" par rapport au treillis \mathcal{L}_F	55
4.4	Le treillis relationnel $\mathcal{L}_{P_F}^{\exists,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage existentiel par la relation "Like" par rapport au treillis \mathcal{L}_F	56
4.5	Le treillis relationnel $\mathcal{L}_{P_P}^{\forall,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage existentiel par la relation Ami.	58
4.6	Le treillis relationnel $\mathcal{L}_{P,F}^{\forall,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage universel par les relations Ami et "Like".	59
4.7	Architecture du web sémantique (<i>semantic web stack</i>).	63
4.8	Graphe de données décrivant la relation "Like" entre un utilisateur d'un réseau social et un film.	64
4.9	Graphe RDF avec types sémantiques des sujets et des objets.	65
4.10	Le nuage de données liées (<i>Linked data cloud diagram</i>). Chaque cercle représente un ensemble de données publiées selon les principes des données liées. La taille des cercles représente le nombre de triplets qu'ils contiennent. Le jeu de couleurs identifie les domaines. Les flèches indiquent qu'au moins 50 triplets relient les ensembles de données.	71
4.11	Ontologie correspondant aux données (Personne, Film).	76
5.1	Arrêté du 25 Avril 2003 relatif à la limitation du bruit dans les établissements d'enseignement citant l'article R111-23-2 du Code de la construction et de l'habitation.	83

5.2	Langage de graphes : description des graphes de collections documentaires. Les éléments du vocabulaire terminal sont notés entre guillemets simples (ex. ‘(’), les non-terminaux sont en italiques (ex. <i>prédictat</i>) et les métasymboles utilisés sont la flèche de réécriture (\leftarrow), les crochets pour former les groupes ([]), la barre d’alternative () et l’étoile de Kleene pour marquer la répétition de l’élément ou du groupe précédent pour un nombre quelconque d’occurrences (*).	84
5.3	Exemple de graphe modélisant une collection documentaire comportant 4 unités documentaires. Pour des questions de lisibilité les attributs et relations partagés par plusieurs documents sont représentés en double. Les unités documentaires sont représentées par des cercles. Les relations sont notées comme des flèches. les attributs sont reliés aux documents par des traits pleins (descripteurs sémantiques) ou pointillés (types de documents).	85
5.4	Exemple de collection juridique avec annotations sémantiques et lien de référence.	85
5.5	Collection BRUIT. Pour des questions de lisibilité les descripteurs sémantiques partagés par plusieurs documents sont représentés en double. Les unités documentaires sont représentées par des cercles. Les relations sont notées comme des flèches. les attributs sont reliés aux documents par des traits pleins (descripteurs sémantiques) ou pointillés (types de documents).	87
5.6	Langage de requêtes. Les éléments du vocabulaire terminal sont notés entre guillemets simples (ex. ‘(’), les non-terminaux sont en italiques (ex. <i>prédictat</i>) et les métasymboles utilisés sont la flèche de réécriture (\leftarrow), les crochets pour former les groupes ([]), la barre d’alternative () et l’étoile de Kleene pour marquer la répétition de l’élément ou du groupe précédent pour un nombre quelconque d’occurrences (*).	89
6.1	Schéma d’un exemple de collection de documents juridiques.	102
6.2	Ensemble de contextes correspondant à la collection juridique de la figure 6.1. . .	104
6.3	Le treillis de concepts \mathcal{L}_{arr} correspondant au contexte formel des arrêtés \mathcal{K}_{arr} . .	106
6.4	Le treillis de concepts \mathcal{L}_{dec} correspondant au contexte formel des décrets \mathcal{K}_{dec} . .	106
6.5	Ensemble de contextes correspondant à la collection juridique de la figure 6.1. . .	107
6.6	Treillis relationnel \mathcal{L}_{arr}^+ résultant de l’enrichissement relationnel entre les objets du contexte des arrêtés et du contexte des décrets.	110
6.7	Correspondance entre le schéma des données (documents dans la collection) et le graphe de la requête relationnelle	115
6.8	Requête simple Q_s^{dec} sur le treillis des décrets $\mathcal{L}_{Q,dec}$	118
6.9	Requête relationnelle Q_r sur la FTR $(\mathcal{L}_{Q,arr}^+, \mathcal{L}_{Q,dec})$	119
6.10	Exemple de navigation par généralisation basée sur une requête simple	121
6.11	Exemple de navigation par généralisation à partir d’une requête relationnelle . .	122
6.12	Un exemple de navigation pour retourner des réponses approchées dans le cas d’une requête simple	129
6.13	Un exemple de navigation pour retourner des réponses approchées dans le cas d’une requête relationnelle	130
6.14	Conjonction de requêtes simples sur le treillis des décrets \mathcal{L}_{dec}	133
6.15	Aperçu de l’approche conceptuelle de RI relationnelle.	137
7.1	Ontologie de collection documentaire : modules et dépendances	143
7.2	Les concepts de haut niveau de l’ontologie documentaire.	144
7.3	Une décision de justice	145

7.4	Un acte local	145
7.5	Un document éditorial	146
7.6	Hierarchie des types de documents.	147
7.7	Types de liens entre les documents et leur hiérarchie.	147
7.8	Les classes modélisant la structure d'un document.	149
7.9	Gestion du cycle de vie d'une unité documentaire (document ou article).	151
7.10	Dates associées à une unité documentaire ou un article.	151
7.11	Gestion de versions d'un article.	152
7.12	Ressources et annotation sémantique.	153
7.13	Concepts terminologiques représentant les ressources sémantiques. Hiérarchie entre concepts de la ressource Environnement.	153
7.14	Gestion des versions des documents d'une collection et la relation de réalisation entre un document (œuvre) et ses versions (expression). La classe <code>DocumentText</code> représente les fragments de documents qui peuvent être annotés et la classe <code>CitableDocumentObject</code> représente les unités documentaires qui peuvent en outre être citées.	157
7.15	Transposition de la directive 2004/114/CE, cible de la relation de transposition (la source de la relation est le texte de loi Loi n° 2006 – 911 du 24 Juillet 2006) et objet de l'opération de transposition (le résultat est l'article Art. L221 – 33 (M) du Code monétaire et financier).	158
7.16	Gestion des liens intertextuels.	159
7.17	Classe Citation.	159
7.18	Opération documentaire de modification : participants et liens de référence et citation résultants.	161
7.19	Classe <code>DocumentaryOperation</code>	161
7.20	Codification de l'Article 46 quater-00 A bis du 4 juillet 1992.	163
7.21	Ontologie de collection documentaire avec gestion des versions et des références (relations ternaires).	164
7.22	Graphes réponses à une requête relationnelle.	166
7.23	Exemple 1	167
7.24	Exemple 2	168
7.25	Exemple 3	169
7.26	Modélisation de la collection arrêtés-décrets.	171
7.27	Annotations sémantiques des arrêtés et des décrets.	172
7.28	Codification de l'article L362 – 1 du code de l'environnement par l'Ordonnance n°2000 – 914.	174
7.29	Modification de l'article L362 – 1 du code de l'environnement par la Loi n°2006 – 436.	175
7.30	Modification de l'article L362 – 1 du code de l'environnement par l'Ordonnance n°2012 – 34.	176
7.31	Abrogation de la Loi n°2006 – 436 par l'Ordonnance n°2000 – 914.	177
8.1	Treillis des conventions avant enrichissement relationnel.	185
8.2	Graphes réponses exactes et approchées de la requête OIT1-2.	187
8.3	Graphe réponse de la requête OIT1-1.	187
8.4	Graphe RDF sur la première ontologie : instances de la classe <code>CodifiedText</code>	195
8.5	Opération documentaire de modification de l'article L2213-1 : l'œuvre, les deux versions qui réalisent l'œuvre et le texte source de modification.	200

Liste des tableaux

2.1	Exemples de types de relations entre les sources de droit	13
2.2	Les éléments de structure des textes juridiques dans les systèmes francophone et anglophone.	14
2.3	Thésaurus et catalogues juridiques.	18
2.4	Ontologies juridiques.	19
4.1	Contexte formel \mathcal{K}_P décrivant des utilisateurs d'un réseau social.	47
4.2	Contexte formel \mathcal{K}_F décrivant les films associés à leurs catégories.	48
4.3	Contexte relationnel Amis décrivant la relation d'amitié entre les utilisateurs du réseau social.	53
4.4	Contexte relationnel "Like" liant les utilisateurs du réseau social et les films. . .	53
4.5	SPARQL vs. Algèbre relationnelle (AR).	67
4.6	Mapping FCA/RCA vers OWL DL.	76
4.7	Tableau comparatif RDF/SPARQL vs AFC/ARC.	78
5.1	Composition de la collection BRUIT	86
5.2	Vocabulaire utilisé pour l'annotation sémantique de la collection BRUIT	86
5.3	Vocabulaire utilisé pour la modélisation de la collection OIT et les requêtes associées. Les types et les identifiants de documents ont une majuscule à l'initiale ; les identifiants comportent en outre un indice ; les noms de relations et les descripteurs sémantiques ont une initiale minuscule mais les noms de relations sont des verbes.	93
5.4	Vocabulaire utilisé pour la formation de la collection Légilocal et des requêtes associées	95
5.5	Vocabulaire utilisé dans le jeu de requêtes-types	98
6.1	Le contexte formel des arrêtés \mathcal{K}_{arr}	104
6.2	Le contexte formel des décrets \mathcal{K}_{dec}	105
6.3	Relation : fait_référence	108
6.4	Le contexte formel des arrêtés \mathcal{K}_{arr}^1 à l'itération 1 du processus d'enrichissement relationnel (dans les attributs $rf : ci$, les ci correspondent aux concepts du treillis des décrets).	109
6.5	Tableau récapitulatif de la typologie des requêtes exprimables par l'AFC et l'ARC et leur correspondance avec les requêtes-types issues de l'analyse des besoins. . .	136
7.1	Classes et propriétés réutilisés par le vocabulaire LIDO.	155
7.2	Classes et propriétés reliées à la classe <code>DocumentaryOperation</code>	162

8.1	Requêtes OIT avec réponses pertinentes.	183
8.2	Propriétés de la collection OIT : Nb. objets, Nb. attributs, Nb. concepts dans le treillis, Nb. arcs, Nb. niveaux (hauteur) du treillis.	184
8.3	Description de la collection LÉGILOCAL : les documents, leurs types et leurs relations.	192
8.4	Requêtes LÉGILOCAL avec réponses pertinentes.	193
8.5	Vocabulaire utilisé pour la formation de la collection LÉGILOCAL et des requêtes associées	194

Chapitre 1

Introduction

Sommaire

1.1	Contexte général	1
1.1.1	Recherche d'information	1
1.1.2	Recherche d'information sémantique et sur le web	2
1.2	Contexte applicatif : le projet Légilocal	2
1.3	Enjeux de la recherche d'information juridique	3
1.4	Objectif et contributions	5
1.5	Structure du document	6

1.1 Contexte général

Au cours des dernières années, les « données liées » sont apparues comme une nouvelle tendance qui a régi l'évolution du web et l'a transformé d'un espace d'information global de documents liés (avec des liens hypertextes) à un espace d'information où documents et données sont liés avec des liens qui sont typés. En effet, dans le modèle hypertexte classique, la nature de la relation entre deux documents liés est implicite [Heath and Bizer, 2011], ceci est dû au fait que le format de données (HTML) est expressivement insuffisant pour permettre à des entités individuelles décrites dans un document particulier d'être reliées par des liens typés à des entités connexes. Le terme « données liées » (ou *Linked Data*) décrit une méthode de publication des données structurées (provenant de différentes sources) de sorte qu'ils peuvent être interconnectés. Pour ce faire, les relations entre les données doivent être explicitées afin de créer cet espace global de données interdépendantes (par opposition à une simple collection d'ensembles de données) qui peuvent être interrogées. Déterminer comment représenter (quel modèle) et interroger (quelle technique de recherche d'information) une collection de documents inter-reliés est l'enjeu global auquel nous proposons de répondre dans cette thèse.

1.1.1 Recherche d'information

Avec la croissance continue de l'information disponible et librement accessible en ligne, il est devenu essentiel d'automatiser le processus de représentation des données et d'avoir un processus de recherche et de gestion de contenus capable de traiter toute cette information. Dans la plupart des cas, l'information est représentée par des documents et les utilisateurs exploitent les collections de documents afin de satisfaire leurs besoins en information. Les systèmes de recherche d'information permettent d'automatiser le processus de recherche en construisant une

représentation adaptée des documents et des requêtes (opération d'indexation) puis en comparant la représentation des requêtes et des documents pour déterminer si le document est pertinent pour la requête (opération d'appariement). Les techniques classiques de description du contenu et de traitement des requêtes en recherche d'information (RI) sont basées sur des mots-clés. Les systèmes de RI basés sur le modèle classique représentent les documents comme des sacs de mots auxquels sont assignés des poids mesurant leur importance dans le texte (poids binaire, fréquence, etc.). La recherche est ensuite faite sur cet ensemble de mots pondérés. Les moteurs de recherche actuels utilisant une technique de recherche par mots-clés (par ex. Google) introduisent constamment de nouvelles fonctionnalités pour améliorer l'expérience de recherche des utilisateurs (nouveaux mécanismes pour gérer le contenu multimédia, personnalisation des résultats en utilisant l'information contextuelle, etc.).

1.1.2 Recherche d'information sémantique et sur le web

Visant à résoudre les limitations des modèles par mots-clés, la recherche sémantique (recherche par le sens plutôt que par les chaînes de caractères) a fait l'objet d'une grande vague de recherche dans les communautés de la RI et du web sémantique.

Dans le domaine de la RI, plusieurs approches sémantiques ont été définies. Certaines sont basées sur des méthodes statistiques qui étudient la co-occurrence des termes dans le texte, d'autres appliquent des algorithmes basés sur des techniques de traitement du langage naturel tout en s'appuyant sur des thésaurus et des taxonomies (par ex. Wordnet).

Le web sémantique a été lancé pour automatiser des tâches qui nécessitent un certain niveau de compréhension conceptuelle des objets impliqués et permettre à des logiciels de combiner les informations et les ressources d'une manière cohérente [Fernández et al., 2011]. L'utilisation des ontologies [Gruber, 1993], élément clé dans les nouvelles technologies du web pour la représentation des connaissances, a permis de surmonter les limites de la recherche par mots-clés dans le domaine de la RI (par ex. en utilisant les annotations sémantiques des documents [Kiryakov et al., 2004a]). La RI sur le web sémantique est différente de la RI sémantique par le fait qu'elle traite principalement des objets, par la complexité des interfaces d'interrogation initialement destinées à manipuler des bases de connaissances et par l'absence des algorithmes de classement de résultats à une grande échelle qu'est le web.

1.2 Contexte applicatif : le projet Légilocal

Notre travail s'inscrit dans le projet Légilocal¹ qui vise à rendre l'acte administratif et juridique facilement accessible au citoyen et aux collectivités locales de façon adaptée à leurs besoins [Amardeilh et al., 2013].

Le besoin des citoyens d'être informés et d'interagir dans un espace public est reconnu comme un droit dans la société de l'information. Les citoyens, mais aussi le monde des affaires souhaitent savoir qui est en charge de la conduite des affaires de la communauté, de comprendre les décisions qui sont prises par les collectivités locales et leurs représentants, et d'anticiper les décisions qui peuvent influencer sur leur vie quotidienne (par exemple en matière d'urbanisme). De plus, ces acteurs s'interrogent sur les fondements ou la validité juridique des actes qui leur sont apposés. Toutefois, si l'information produite par l'état et par l'UE est généralement disponible à partir de Legifrance² pour les citoyens et les acteurs français, les informations produites par les

1. Projet FUI, <http://www.mondeca.com/fr/R-D/Projets/LegiLocal-Projet-FUI-9-Cap-digital-2010-2013>

2. www.legifrance.gouv.fr/

communautés locales ne sont pas systématiquement disponibles en ligne.

Un autre défi pour les administrations locales est la qualité des documents juridiques qu'elles produisent. Les petites municipalités et groupements de municipalités ont peu de personnel. Le secrétaire de mairie qui produit et publie des documents juridiques n'est généralement pas un avocat en soi et de nombreux actes municipaux sont attaqués (peut-être de l'ordre de 15-20%) pour des motifs de procédure, comme l'inadéquation de visas. Les secrétaires de mairie sont souvent isolés, alors qu'ils ont besoin d'interagir les uns avec les autres, à partager leur expérience et à harmoniser la législation et les décisions locales entre les différents niveaux de collectivités locales ou avec les collectivités voisines.

Le projet Légilocal, « La loi locale tout simplement partagée », vise à résoudre ces problèmes en développant les outils et l'infrastructure qui aident les administrateurs locaux à préparer et à publier les actes locaux, décisions et règlements de telle façon qu'ils soient faciles à rechercher pour eux-mêmes et pour les citoyens, assurant ainsi l'accessibilité, la transparence et la qualité de la législation locale. L'originalité de l'approche suivie par le projet consiste à combiner des outils de gestion de contenu et des services de gestion d'interaction dans une plate-forme unique et facile d'accès pour les agents administratifs et les citoyens par le biais de widgets intégrés dans des outils de bureautique (pour l'édition) ou les sites web des municipalités (pour l'accès aux documents) [Amardeilh et al., 2013].

Le projet est mené par Victoires Editions, un éditeur juridique spécialisé dans le droit des communautés locales. Il réunit des partenaires industriels et académiques en charge du développement des technologies et des ressources sur lesquelles la plate-forme Légilocal est construite. Un groupe pilote de petites municipalités est également associé au projet à des fins de test.

Pour résumer, le projet possède un triple objectif :

- Faciliter l'accès aux données administratives et juridiques locales (interrogation, consultation) pour les citoyens.
- Faciliter l'accès aux ressources et la collaboration au sein des collectivités locales (la prise de décision locale doit s'appuyer sur les décisions similaires antérieures).
- Permettre aux citoyens de s'informer et de commenter les décisions des collectivités locales.

Ces nouvelles fonctionnalités sont mises en œuvre en intégrant les techniques du web sémantique pour une meilleure exploitation du contenu des documents juridiques :

- des ontologies et des standards juridiques sont utilisés pour permettre l'interopérabilité documentaire entre les collectivités locales et l'ouverture aux citoyens ;
- des services web simples sont créés pour la recherche d'information par interrogation (formulation de requêtes) ou par consultation (navigation) ;
- un réseau social unique à l'ensemble des collectivités locales est développé.

1.3 Enjeux de la recherche d'information juridique

Les documents juridiques sont des documents structurés fortement interconnectés. L'accès à l'information dans ce domaine est aussi problématique pour les citoyens qui essayent de comprendre la norme qui s'applique à leur cas particulier que pour les juristes professionnels qui doivent déterminer comment la loi s'applique sur des cas particuliers. Le domaine juridique pose de ce fait des questions spécifiques en terme de recherche d'information.

Structure d'un document La structure du document est importante à prendre en compte. Un texte juridique, notamment le texte d'une loi, est composé d'articles qui ont un cycle de vie autonome. Ils peuvent être modifiés ou même abrogés indépendamment de la loi considérée dans

son ensemble. Il est essentiel pour un juriste de pouvoir consolider un texte de loi, c'est-à-dire retrouver toutes les modifications qui s'appliquent à ce texte, et retrouver la version en vigueur à une date donnée, parce qu'il faut pouvoir déterminer le droit qui s'applique à un moment particulier du passé. Il faut également pouvoir ajuster la granularité documentaire (texte complet ou article de ce texte) aux besoins de l'utilisateur et prendre en compte la complexité du cycle de vie du document juridique qui peut être signé, publié, entré en vigueur, promulgué, modifié et abrogé à des dates différentes. Les systèmes actuels d'accès à l'information juridique, comme Normattiva³ ou UK Legislation⁴, prennent partiellement en compte ce type de propriétés quand ils proposent un accès temporel aux sources juridiques (*point in time access*).

Document indépendant vs. collection documentaire Le plus souvent cependant, dans ces systèmes, les notions de modification ou d'abrogation – qui sont en réalité des relations intertextuelles – sont modélisées comme des attributs de documents. On peut savoir quel est le statut d'un document juridique mais on n'a pas directement accès au texte qui lui confère ce statut. La dimension intertextuelle des collections de documents juridiques est mal prise en compte. Elle est pourtant centrale dans la compréhension du raisonnement juridique : un texte ne s'interprète pas isolément, indépendamment de la jurisprudence et des interprétations auxquelles il a donné lieu, des textes qui sont venus le modifier ou des décrets qui en précisent l'application. La dimension intertextuelle des collections juridiques est reconnue comme un facteur de complexité majeur [Bourcier, 2011] pour la compréhension du droit. Ouvrir cette complexité est aujourd'hui un défi majeur pour l'accès à l'information juridique⁵.

Des efforts sont faits pour développer des modèles technologiques qui facilitent l'accès et assurent l'interopérabilité des données dans le domaine juridique. Ces technologies ne sont implémentées que de façon limitée [Sartor et al., 2011] par les systèmes d'accès à l'information juridique existants. Ainsi, même si une grande quantité de données juridiques est disponible sur le web, son exploitation reste limitée du fait qu'elle est stockée dans différents formats (word, pdf, html, xml, etc.), elle est interrogeable par des moteurs de recherche (avec un bon rappel, mais avec beaucoup de bruit et peu de pertinence).

Ouvrir la complexité dans l'accès à l'information juridique suppose d'être en mesure de lancer des requêtes relationnelles sur un moteur de recherche et de retrouver non pas une liste de documents autonomes mais une liste de graphes de documents qui respectent les contraintes relationnelles formulées en entrée par l'utilisateur.

Les requêtes des utilisateurs peuvent porter sur les cas d'application d'une règle de droit (*Quels sont les textes de jurisprudence qui ont appliqué un texte de loi donné?*), une date de validité (*Quels sont les textes locaux qui parlent de bruit et qui sont valides à une date donnée?*), des liens de modification (*Quels sont les lois qui modifient un code donné?*), ou porter sur plusieurs contraintes à la fois (*Quels sont les textes de jurisprudence relatifs au texte de loi donné avant la date d'abrogation de ce dernier?*).

Contenu d'un document Au-delà de ces besoins particuliers au domaine juridique, il faut également fournir des outils sémantiques d'accès au contenu pour permettre aux utilisateurs de retrouver des documents à partir de leurs métadonnées d'identification (date de publication, titre, type de document, numéro d'un article, etc.) mais aussi de certaines notions clés.

3. <http://www.normattiva.it/ricerca/avanzata/vigente>

4. <http://www.legislation.gov.uk/search/point-in-time>

5. Les efforts de simplification juridique actuels portent essentiellement sur la normalisation et le contrôle du lexique, à ce jour.

Le domaine juridique et la RI logique Il est essentiel de comprendre que le tri des résultats retournés par un moteur de recherche n'est pas central dans le domaine juridique, où la recherche d'information se doit d'abord d'être exhaustive. La sécurité juridique impose en effet de prendre connaissance de tous les documents qui se rapportent à un cas particulier. Il est préférable de laisser le contrôle au juriste qui peut progressivement affiner sa requête en fonction de ses besoins plutôt que de lui présenter un sous-ensemble de documents sélectionnés en fonction d'un critère de pertinence défini *a priori*. En cela, la recherche d'information juridique se distingue clairement des moteurs de recherche généralistes sur le web.

Les logiques formelles ont été utilisées efficacement dans la RI du fait qu'elles sont bien adaptées pour la représentation des connaissances [Baader et al., 2003] et pour la construction de modèles de RI intégrant formellement des connaissances dans le processus de recherche. Leur utilisation dans le web sémantique en est le témoin. En effet, les logiques de descriptions forment la théorie mathématique qui est à la base de certaines technologies du web sémantique comme par exemple OWL-DL sur lequel s'appuient les ontologies. Dans les modèles de RI basés sur la logique, l'appariement entre les documents et les requêtes est principalement binaire (une correspondance existe ou non), ce qui est en adéquation avec les besoins dans le domaine juridique en terme d'exhaustivité des résultats.

En relation étroite avec les logiques formelles, la théorie des treillis a été utilisée comme base pour des modèles de RI où l'implication logique devient une relation d'ordre partiel. Une des premières études qui ont exploité la structure algébrique des treillis dans la RI est présentée dans [Mooers, 1958] et a été reprise par [Priss, 2000] avec l'AFC (Analyse Formelle de Concepts). Le processus de recherche est la recherche booléenne classique. Le travail présenté dans cette thèse s'inscrit dans le cadre de modèles de RI basés sur la logique.

1.4 Objectif et contributions

Dans ce travail, nous nous intéressons à la recherche d'information dans une collection documentaire, où les documents sont inter-reliés par différents types de relations et où l'interprétation d'un document se fait en référence à son contexte (nous définissons le contexte d'un document par l'ensemble des documents auxquels il est relié).

Le but de ce travail est d'exploiter la richesse de collections de documents (dans le cas général) en essayant d'intégrer les liens et le contenu dans le processus de recherche qui comporte donc un double aspect relationnel et sémantique. Dans le domaine juridique, il s'agit de rendre compte à la fois de la complexité sémantique et de la complexité relationnelle des sources du droit, en s'appuyant sur une typologie des documents qui les composent. Sur le plan sémantique les sources de droit utilisent un langage juridique complexe qui n'est pas le même pour les différents types de documents. Sur le plan relationnel, les liens sont régis par une typologie partiellement induite par les documents qu'ils relient.

Prendre en compte l'intertextualité (les relations entre documents) dans un processus de RI pour en améliorer les résultats forme l'enjeu global de ce travail de thèse. Pour répondre à cet enjeu, nous nous fixons les objectifs suivants :

- mettre en évidence l'intérêt de la prise en compte des relations entre les documents dans un processus de recherche dans une collection documentaire ;
- proposer un modèle unifié pour la prise en compte de la sémantique des documents dans une collection documentaire et des liens qu'ils entretiennent ;
- définir des méthodes d'accès et d'exploitation de ce modèle dans un processus de recherche d'information ;

- appliquer ce modèle et ces techniques d'accès au domaine juridique, le contexte applicatif de notre travail.

Le travail effectué dans cette thèse peut être réparti en trois grandes parties commençant par une analyse de besoins dans le domaine juridique, puis la proposition de deux approches, conceptuelle et sémantique, pour résoudre la problématique de la thèse. Nos principales contributions peuvent être résumées par les points suivants :

1. Analyse et identification des besoins en RI dans le domaine juridique. Un ensemble de requêtes formulées par les spécialistes du domaine juridique est collecté parmi lesquelles les requêtes relationnelles sont fréquentes.
2. Définition d'un formalisme logique pour décrire la collection de documents et le langage de requêtes. Une liste des types de requêtes importantes à traiter dans un système de recherche d'information juridique est tirée de l'analyse des requêtes recueillies auprès des juristes interviewés.
3. Utilisation des treillis de concepts formels et relationnels pour la classification de documents inter-reliés d'une collection documentaire. Il s'agit de s'appuyer sur la théorie de l'Analyse Formelle de concepts (AFC) et l'Analyse Relationnelle de Concepts (ARC) pour organiser l'ensemble des documents en fonction de leurs annotations sémantiques et des relations intertextuelles qu'il entretiennent.
4. Définition de méthodes de recherche de documents pertinents dans les treillis formels et relationnels. La recherche peut être effectuée soit de manière directe en interrogeant les treillis de concepts par l'intermédiaire de requêtes simples et relationnelles, soit de manière progressive en naviguant dans le treillis, soit en combinant les deux modes (interrogation et navigation).
5. Utilisation des technologies du web sémantique pour la modélisation de collections documentaires en prenant en compte toutes les caractéristiques de la collection : la typologie des documents, les liens intertextuels et leurs différents types, la structure des documents, leur contenu sémantique et leur cycle de vie :
 - Première proposition d'ontologie documentaire pour les textes juridiques contenant trois modules : module document (structure), module collection (types des documents et liens intertextuels) et module sémantique (ressources sémantiques pour les concepts de domaine).
 - Propositions d'améliorations de l'ontologie pour permettre la gestion avancées des opérations documentaires (versions des documents et relations n-aires).

1.5 Structure du document

Le chapitre 2 aborde la problématique de la connaissance juridique face aux techniques du web sémantique. Il décrit la complexité du domaine juridique et les efforts pour la structuration du contenu des documents. Il liste les principales méthodes d'accès à l'information juridique et de traitement de l'intertextualité dans ce domaine.

Le chapitre 3 présente et définit brièvement les concepts de base de la RI classique et de la RI sémantique ainsi que les différents modèles de cette dernière (numérique et logique). Il décrit également les principaux modèles de traitement de l'intertextualité dans les systèmes existants de RI.

Le chapitre 4 présente l'état de l'art des approches candidates pour notre travail (l'approche conceptuelle et l'approche sémantique).

Le chapitre 5 introduit un formalisme logique pour décrire les collections de documents et le langage de requêtes. Il liste les types de requêtes à traiter dans un système de recherche d'information juridique.

Le chapitre 6 détaille l'approche conceptuelle, basée sur l'analyse formelle et l'analyse relationnelle de concepts, pour la création d'un modèle unifié de collections documentaires et la définition de méthodes d'accès par interrogation et par navigation.

Le chapitre 7 détaille l'approche sémantique, basée sur les technologies du web sémantique, pour la modélisation et l'interrogation de collections documentaires.

Le chapitre 8 décrit les expérimentations réalisées pour évaluer les approches proposées de recherche d'information dans une collection de documents.

Le chapitre 9 conclut le travail et en donne les principales perspectives.

Chapitre 2

Accès à l'information juridique

Sommaire

2.1	Introduction	9
2.2	Caractéristiques des données juridiques	10
2.2.1	Structure et hiérarchie des sources de loi	10
2.2.2	Réseau de réglementations et complexité juridique	11
2.3	Efforts de structuration de l'information juridique	14
2.3.1	Création ou édition de la réglementation	14
2.3.2	Représentation des documents	15
2.3.3	Ontologies du droit	17
2.3.4	Synthèse	20
2.4	Méthodes d'accès à l'information juridique	20
2.4.1	Portails généralistes de sources de droit	20
2.4.2	Outils spécialisés	22
2.4.3	Données gouvernementales ouvertes sur le web	24
2.5	Traitement de l'intertextualité	24
2.6	Conclusion	26

2.1 Introduction

Le modèle documentaire classique a fait ses preuves dans la recherche d'information généraliste qui se caractérise par le volume de documents appréhendés, la diversité des requêtes des utilisateurs et la redondance de l'information. Dans des domaines spécialisés, comme la médecine ou le domaine réglementaire, ce modèle trouve ses limites. C'est en particulier le cas dans le domaine juridique, où les documents sont de plusieurs types (législation, jurisprudence, etc.) et liés par différents types de relations. Ces relations sont en général le résultat de l'activité d'un agent sur un document donné (document législatif, décret d'application, etc.) dans un système d'information juridique. Par exemple, dans le cas de la législation, un système d'information juridique comporte plusieurs activités comme le montre la figure 2.1.

Les documents dans le domaine juridique sont liés les uns aux autres par des relations d'amendements, de dérivation, de transposition, de complémentation, de jurisprudence, etc. et cette intertextualité est reconnue comme un source importante de complexité juridique [Bourcier, 2011].

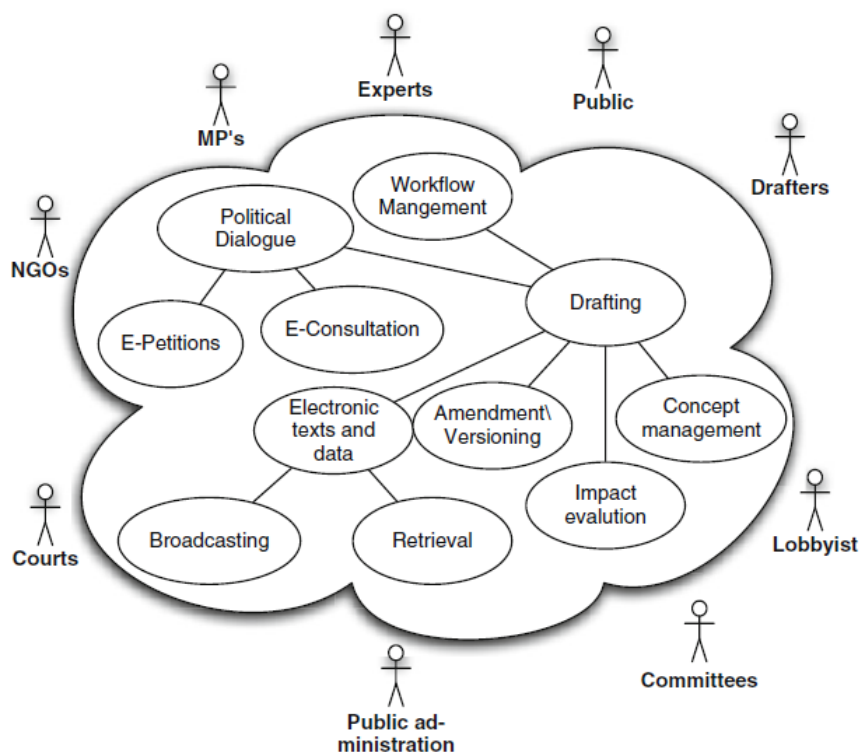


FIGURE 2.1 – Activités d'un système d'information législatif [Sartor et al., 2011].

Dans les sections suivantes nous décrivons les caractéristiques des sources juridiques (section 2.2), leur structuration (section 2.3) ainsi que les méthodes d'accès à cette information (section 2.4).

2.2 Caractéristiques des données juridiques

Les documents dans le domaine juridique présentent des caractéristiques qui les distinguent dans le traitement des corpus textuels habituellement manipulés par les outils de recherche d'information. D'un côté, les documents possèdent des structures internes bien spécifiques à chaque type de documents, d'un autre côté ils sont fortement inter-connectés avec plusieurs types de liens entre eux. L'analyse et l'accès à ces documents présente un problème différent de celui des corpus textuels. Nous étudions ces deux caractéristiques importantes des documents juridiques dans ce qui suit.

2.2.1 Structure et hiérarchie des sources de loi

La structure des documents réglementaires possède trois caractéristiques spécifiques [Lau, 2004] qui font de l'étude et l'analyse de ces corpus un problème intéressant.

- Les réglementations possèdent une hiérarchie arborescente profonde. Ce sont des documents semi-structurés organisés dans une structure d'arbre. Selon les types des documents, les sous-parties structurées peuvent être des sections ou des articles. Par exemple l'article 11.4.5(a) peut être interprété comme sous-partie ou noeud fils de l'article 11.4.5 et un noeud frère de l'article 11.4.5(b). Cette structure est cruciale pour la compréhension en

contexte des différentes parties des documents.

- Pour certains types de réglementations, les articles sont massivement inter-reliés dans un même texte réglementaire (liens internes). Par exemple, l'article 11.4.5(a) peut faire référence à l'article 8.2 pour des besoins de conformité. Dans l'analyse et l'exploitation des provisions, cette information de lien est très importante, puisque les règles prescrites dans un article ne sont complètes que par l'inclusion des références.
- Les termes importants utilisés dans une réglementation particulière sont généralement définis dans une partie relativement avancée de cette réglementation. La définition des termes ajoute clairement une information sémantique aux phrases spécifiques au domaine et aide la compréhension des réglementations. Un traitement automatique de ces définitions s'avère utile pour l'analyse des phrases différentes qui partagent les mêmes définitions.

Nous ajoutons une quatrième caractéristique à celles citées ci-dessus, que nous estimons très importante pour l'analyse d'un corpus de documents réglementaires (étudiée en détail dans la section 2.2.2).

- Les textes réglementaires sont fortement interconnectés entre eux (liens externes). Par exemple, l'arrêté du 23 avril 2012 JORF 2 mai 2012 (Avenant n° 80 du 16 novembre 2011) qui parle du droit des salariés aux congés payés sous certaines conditions fait référence aux dispositions de l'article L.122 – 26 – 10 du code du travail.

Nous constatons, dans l'exemple cité, que le lien va d'un arrêté vers un article de code. Ceci est typique aux relations de hiérarchie qui réglementent et ordonnent les types de documents juridiques. La figure 2.2 montre une hiérarchie de types de documents relativement à leurs portées. La figure 2.3 donne un exemple de documents juridiques de différents types et les relations qu'ils entretiennent⁶. En règle générale, un document juridique ne peut pas contredire un autre document juridique hiérarchiquement supérieur. Par exemple, un arrêté ne peut pas contredire une loi.

2.2.2 Réseau de réglementations et complexité juridique

La complexité du droit et des textes juridiques a souvent été soulignée, notamment la dispersion des règles de droit dans différents textes qui crée une forte interconnexion entre ces textes (exprimée par des relations). Le tableau 2.1 cite différents types de relations qui peuvent exister entre les sources de droit.

La multiplicité et la diversité de ces relations fait de la structure de la collection documentaire un aspect important à prendre en compte si nous souhaitons satisfaire au mieux les besoins d'un utilisateur en termes de recherche d'information.

Geist [Geist, 2009] observe que le réseau des citations réglementaires doit bien être le réseau de citations le plus ancien, le plus grand et le mieux documenté jamais créé. Les juristes apprennent à l'utiliser sans forcément en connaître la structure globale. Des travaux récents

6. Les noms des relations dans la figure et les types des documents qu'elles relient résultent des premières discussions que nous avons eues avec un expert du domaine juridique. Une description évoluée et plus précise des types des documents et des relations entre eux est donnée dans les chapitres suivants.

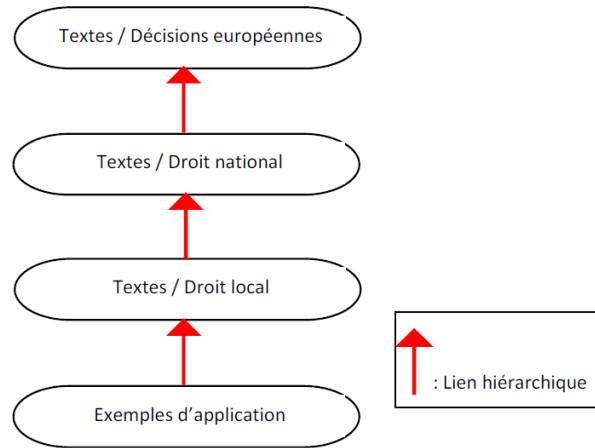


FIGURE 2.2 – Hiérarchie des sources de loi.

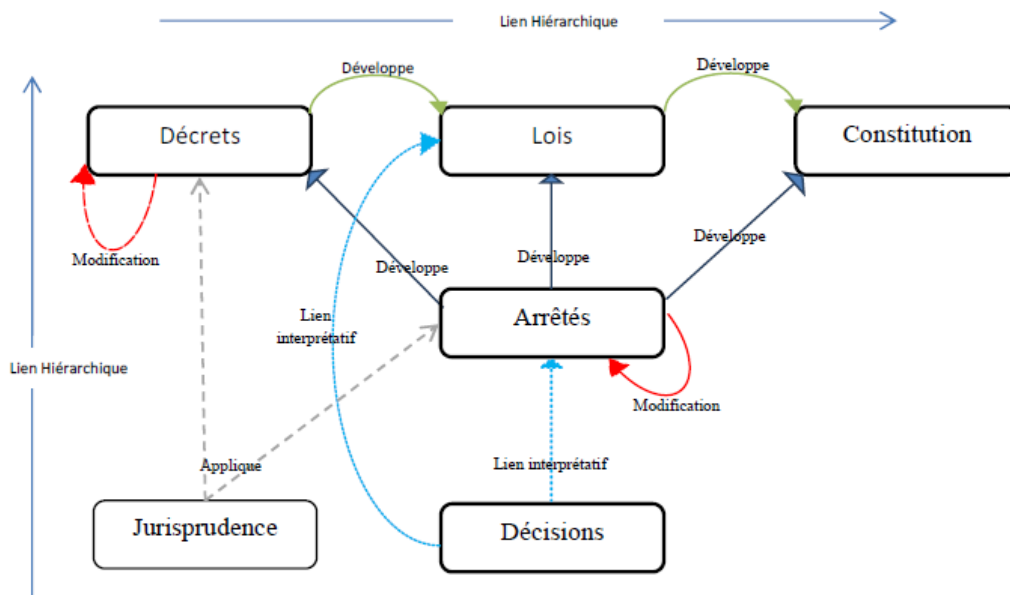


FIGURE 2.3 – Exemple de documents juridiques et de types de liens qui existent entre eux.

TABLE 2.1 – Exemples de types de relations entre les sources de droit

Liens entre sources de droit	Type de relation
Une loi en modifie une autre	Modification/Amendement
Un texte local interprète un texte national	Interprétation/Dérivation
Un décret applique une loi	Application
Une directive européenne est transposée en droit national	Transposition
Un texte codifie une loi ou un article	Codification
La jurisprudence (les jugements) s'appuie sur des lois (cas concret d'application)	Jurisprudence

se sont particulièrement intéressés à l'étude des réseaux de réglementation pour comprendre leurs topologies et étudier l'impact du facteur d'interconnexion (références entre les sources de loi) sur la complexité de l'accès à l'information juridique [Bourcier, 2011, Boulet et al., 2011, Winkels and de Ruyter, 2011, Winkels et al., 2013].

[Bourcier, 2011] cite parmi les principales sources de complexité des systèmes juridiques « l'auto-organisation d'un système textuel fortement interconnecté » et « la lecture enchevêtrée des textes pour un usager du droit (citoyen, décideur, juge) ». L'intertextualité, définie comme étant une interaction entre un ensemble de textes, identifiée par l'ensemble des liens qui les relient, forme ainsi une source incontournable de la complexité des systèmes juridiques. Ce qui explique que cette intertextualité soit un facteur majeur de complexité, c'est la multiplicité et la diversité des types de liens entre les sources de droit. L'évolution des textes de droit donne lieu à la création, l'abrogation ou à la codification d'autres textes.

L'hypothèse faite par Bourcier [Bourcier, 2011] "que le droit est normalement complexe et que cette complexité doit être maintenue, gérée, exploitée par des modèles adéquats" ainsi que l'observation de Geist [Geist, 2009] justifient la perspective de notre travail qui vise à proposer un modèle permettant d'exploiter le matériau juridique tout en tenant compte de la complexité due à son facteur d'intertextualité.

Les opérateurs juridiques sont de plus en plus conscients de la complexité du droit. Celle-ci peut être abordée de plusieurs points de vue, selon le degré de détail des lois [Tulloch, 1995] ou en fonction d'autres paramètres comme les renvois. Les sciences qui étudient les systèmes complexes ont été utilisées pour favoriser l'émergence de nouvelles approches pour l'analyse du droit et des systèmes juridiques sous la forme de réseaux [Ruhl, 1997]. On met généralement l'accent sur la complexité du droit en lien avec les nombreuses citations croisées entre les textes juridiques. Certains travaux ont construit des cartographies des branches du droit sur la base des renvois entre les articles de différents codes [Bourcier and Mazzega, 2007a, Bourcier and Mazzega, 2007b, Boulet et al., 2011]. Cette analyse du réseau juridique vise à construire une méta-représentation des relations (graphe de relations) entre les codes. Cette représentation a fait apparaître certaines proximités entre codes, en conformité avec les groupements obtenus à partir du partitionnement de graphe, qui étaient jusque-là inconnus des juristes. Cette nouvelle cartographie du système juridique ouvre aux chercheurs de nouvelles possibilités d'analyse, une nouvelle échelle dans la perception du droit, et de nouvelles pistes en matière d'ingénierie de la loi dans les "usines du droit" [Boulet et al., 2011].

En plus du facteur d'intertextualité, l'évolution du système juridique, c'est-à-dire les modifications et abrogations des sources juridiques, forme aussi une source importante de complexité du droit mais les deux aspects sont liés puisqu'une modification crée une nouvelle version pour un document existant.

Nous faisons l'hypothèse que la modélisation du système juridique sous la forme d'un réseau documentaire va permettre d'améliorer l'accès à l'information juridique. Le travail de cette thèse propose de s'appuyer sur les liens du réseau de réglementation pour offrir une nouvelle forme d'interrogation des corpus juridiques et de leurs réseau de relations intertextuelles.

2.3 Efforts de structuration de l'information juridique

Les textes juridiques possèdent des structures complexes et variables selon le type des documents. Des efforts sont faits pour structurer les documents juridiques et faciliter l'échange et l'exploitation de ces données. Des outils d'aide à l'édition réglementaire sont proposés. En parallèle, plusieurs standards XML juridiques sont définis pour normaliser la structure des textes de loi et assister la production de ces textes. Des efforts sont également faits pour rendre ces données compatibles avec les standards et normes définis dans le web sémantique (XML, RDF, SPARQL) et définir des modèles sémantiques (ontologies) pour différents domaines. Ces efforts ont pour but d'assurer l'interopérabilité des données, faciliter leur gestion et leur accès par les utilisateurs.

2.3.1 Création ou édition de la réglementation

[Engeljehringer and Schefbeck, 2006] indique que le terme écriture de la loi (*legislative drafting* ou *writing law*) fait référence au cadre formel de l'exécution de cette tâche. Il décrit le processus de rédaction comme un processus itératif composé de trois étapes :

1. comprendre, analyser les règles et les instructions (qui ne sont pas toujours écrites) ;
2. modéliser, composer, structurer, éditer les documents législatifs ;
3. ajouter des détails en toute liberté (par ex. pour la formulation des définitions).

Selon le type du document, la création du texte doit respecter un certain nombre de contraintes sur la structure. Le tableau 2.2 donne les principaux éléments de structure des documents dans les deux systèmes francophone et anglophone (comme décrit par G.Schefbeck).

TABLE 2.2 – Les éléments de structure des textes juridiques dans les systèmes francophone et anglophone.

	Système francophone	Système anglophone
Unités de haut niveau	livre, titre, chapitre, sous-chapitre	chapitre, partie, division, sous-division
Unité de base	article	section
Unités de bas niveau	numéro, alinéa	sous-section, paragraphe, sous-paragraphe, item, sous-item

Il existe des ressources qui sont mises à la disposition des agents pour les aider dans la tâche d'écriture de la loi. Par exemple :

- La collection de manuels/aide au drafting (essentiellement pour les pays anglophones)⁷ ;
- Commission manual⁸ (en Europe) ;
- Practical Legal Drafting Guidelines⁹ (pour les pays de l'Afrique).

Des outils d'aide à l'écriture de la loi sont également disponibles (par exemple : *Meta-vex* [Ven et al., 2007], *Bungeni Editor*¹⁰). Leur but est d'assister les secrétaires de mairies dans la tâche lourde et fastidieuse de l'édition du contenu des réglementations et de leur faire gagner du temps. Des outils d'aide à l'édition et à la gestion de la modification des textes existants sont aussi disponibles : par exemple, *AT4AM* (*Authoring Tool for Amendments*) est un outil web d'édition d'amendements pour le Parlement Européen, il est accessible en open source¹¹.

2.3.2 Représentation des documents

La mise à disposition des documents juridiques sur le web enrichis avec de l'information lisible et traitable par la machine contribuent à l'émergence du web sémantique juridique. Dans ce cadre, des nouvelles technologies se développent et se multiplient [Sartor et al., 2011]. Des standards sont définis pour

- identifier les ressources juridiques : chaque document juridique, produit par n'importe quelle autorité, peut être identifié de façon unique (et par conséquent peut être retrouvé) ;
- structurer les documents juridiques, de n'importe quel type, respectant des définitions XML bien spécifiques.

Des ontologies juridiques sont créées (et liées aux ontologies générales) pour :

- organiser et annoter les documents juridiques ;
- permettre de faire du raisonnement sur ces documents.

Les standards XML fournissent une description uniforme des documents de différentes sources assurant une meilleure interopérabilité. Ces standards facilitent la production de documents (en utilisant les mêmes outils dans des systèmes juridiques différents), la présentation des documents (affichage, impression) et l'accessibilité (interconnexion des réseaux de documents) [Biasiotti et al., 2008].

Différentes initiatives dans plusieurs pays (en Europe, en Afrique ou aux États Unis) ont introduit des standards pour la description et l'identification des documents juridiques : *Metalex*¹², *LexDania*¹³, *AkomaNtoso*¹⁴, *NormeInRete*¹⁵, *Formex*¹⁶, etc. Nous avons étudié en détails trois de ces standards :

Metalex [Boer et al., 2002, Boer et al., 2007, Boer et al., 2008, Boer, 2009] (Pays-Bas) définit les éléments partagés entre les documents juridiques de différentes juridictions avec possibilité d'ajouter des détails pour décrire les éléments spécifiques à chaque juridiction. La structure d'un document est décrite par des articles, considérés comme des éléments indépendants du reste du document, qui contiennent une ou plusieurs phrases et peuvent être regroupés en parties (voir figure 2.4). Le schéma affecte des URIs aux documents, permet

7. <http://www.ili.org/ld/manuals.htm>

8. http://ec.europa.eu/governance/better_regulation/documents/legis_draft_comm_en.pdf

9. <http://www.apkn.org/lrp/guidelines/guidelines>

10. <http://code.google.com/p/bungeni-editor/>

11. OPEN Authoring Tool for Amendments (OPEN at4am) : <https://code.google.com/p/open-at4am/>.

12. <http://www.metalex.eu/>

13. <http://lawin.org/lexdania/>

14. <http://www.akomantoso.org/>

15. http://www.interno.gov.it/mininterno/export/sites/default/it/sezioni/sala_stampa/notizie/internet/app_notizia_15466.html

16. <http://formex.publications.europa.eu/>

la gestion du temps en définissant un certain nombre de dates pour chaque document et permet l'identification des références entre documents et vers des concepts (voir figure 2.5). L'initiative CEN-Metalex (Comité Européen de Normalisation), basée sur Metalex, vise à proposer un standard européen au delà des standards nationaux.

CEN MetaLex [Winkels et al., 2003] est un standard XML pour la représentation, la publication et l'échange de la structure des sources juridiques. CEN MetaLex est un format d'échange, considéré comme le plus petit dénominateur commun pour d'autres normes. Il n'est pas destiné à remplacer les normes spécifiques à la juridiction et les formats propriétaires dans le processus de publication, mais d'imposer une vue normalisée sur les documents juridiques à fin d'échange d'informations et d'interopérabilité dans le cadre du développement d'outils spécialisés. Pour répondre à ces exigences, CEN MetaLex définit un mécanisme pour l'extension de schéma, les métadonnées ajoutant des références croisées, la construction des documents composites et une convention de nommage. Pour chaque élément de structure d'un document juridique, un identifiant basé sur un IRI (*Internationalized Resource Identifier*¹⁷) est fixé comme indiqué par la convention standard de noms (*standard naming convention*) [Law, 2009].

MetaLex définit une ontologie qui contient l'information de ce qui est considéré comme une métadonnée de MetaLex, la façon dont elle est stockée dans un document MetaLex, les classes d'entités et les propriétés (prédicats).

```

<Sentence-id="d0e240">
  <TextVersion-id="d0e242" xml:lang="fr">Celui-ci peut faire l'objet d'une division ou d'une affectation, mais dans la seule mesure prévue par la loi.</TextVersion>
  <TextVersion-id="d0e245" xml:lang="en">The patrimony may be divided or appropriated to a purpose, but only to the extent provided by law.</TextVersion>
</Sentence>
<Annotation-id="d0e249">
  <TextVersion-id="d0e251">
    <Cite-id="_1991c64a2_">1991, c. 64, a. 2.</Cite>
  </TextVersion>
</Annotation>

```

FIGURE 2.4 – Extrait d'un document décrit en Metalex

```

<Annotation-id="d0e199">
  <TextVersion-id="d0e201" xml:lang="fr">
    <Reference-id="d0e202" xlink:href="_1991c64a1_">1991, c. 64, a. 1.</Reference>
  </TextVersion>
</Annotation>
<Annotation-id="d0e206">
</Annotation>
</Article>

```

FIGURE 2.5 – Extrait d'un document décrit en Metalex : identification de références

17. <http://www.w3.org/2001/Talks/0912-IUC-IRI/paper.html>

AkomaNtoso [Palmirani et al., 2003] produit des DTD pour les documents parlementaires, législatifs et judiciaires de plusieurs pays africains. Les Schémas XML AkomaNtoso rendent "visibles" la structure et la sémantique des composants pertinents de documents numériques afin de soutenir la création de services d'information à forte valeur ajoutée et accroître l'efficacité et la responsabilité dans le contexte parlementaire, législatif et judiciaire. Akoma Ntoso [Barabucci et al., 2011] propose une gestion avancée des références et des modifications [Palmirani et al., 2009, Palmirani and Cervone, 2009, Palmirani and Brighi, 2010] utilisant une base de données XML native et des éléments XML spécifiques (*passiveRef*, *activeRef*) pour permettre l'accès aux citations qui ont modifié le document original ou qui modifient le document actuel. Il permet aussi une gestion automatique des mises à jour [Brighi and Palmirani, 2009]. Une autre caractéristique intéressante de AkomaNtoso est la façon dont il gère la distinction entre les annotations et les interprétations des agents de l'autorité ou de tiers. Il propose une structure en couches qui permet la séparation du contenu original créé par les chambres du Parlement (données) et le contenu ajouté par les différents acteurs (métadonnées).

NormeInRete définit des DTDs et des schémas XML pour la législation italienne. Ces schémas représentent les métadonnées nécessaires pour automatiser la gestion du cycle de vie des documents législatifs. Ils représentent des informations structurelles et des informations administratives et sémantiques. En rendant les documents disponibles en XML, ils permettent de fournir des fonctionnalités avancées de recherche.

Dans le cadre du projet Légilocal, une réflexion a eu lieu pour choisir un standard pour la description des documents. Le format de la DILA (Direction de l'Information Légale et Administrative)¹⁸ est adopté pour les documents du projet étant donné qu'une grande partie des documents de la collection Légilocal est extraite de Legifrance. Dans le cadre de notre travail, et dans une perspective d'ouverture de données sur le web, nous nous positionnons par rapport au standard Metalex, et plus spécifiquement par rapport à l'ontologie qu'il définit (voir chapitre 7).

2.3.3 Ontologies du droit

Avec la numérisation des documents juridiques et la définition des standards XML, des ressources ontologiques et terminologiques sont parallèlement créées pour représenter et spécifier le contenu sémantique de ces documents [Shaheed, 2005, Gangemi et al., 2005, Després and Szulman, 2007, Hoekstra et al., 2009, Mommers, 2010]. Ces ressources existent sous plusieurs formes : des catalogues et index numériques non structurés (vocabulaires contrôlés destinés à l'indexation de contenus), des thésaurus (ensemble de descripteurs structurés à travers des relations d'équivalence, de généralité ou de spécificité, par ex. Eurovoc), des ontologies lexicales (ressources terminologiques structurées sur la base de relations linguistiques : hyperonymie, hyponymie, synonymie) et des ontologies (ressources sémantiques contenant des classes, des attributs, des relations et des instances) [Bourcier and Fernández-Barrera, 2012]. Selon le degré d'abstraction du domaine couvert, les ontologies peuvent être classées en trois catégories :

- ontologies de haut-niveau ou *top ontologies* (par ex. DOLCE),
- ontologies noyaux ou *core ontologies* (par ex. LKIF core, CLO),
- ontologies de domaine.

Les tableaux 2.3 et 2.4 donnent une description de quelques ressources sémantiques définies pour le domaine juridique.

18. Format d'échange de publication de données, utilisé dans Legifrance.

TABLE 2.3 – Thésaurus et catalogues juridiques.

Ressource	Description
EUROVOC ¹⁹	thésaurus multilingue (16 langues officielles) qui couvre tous les domaines d'activité de la communauté européenne : la politique, les relations internationales, le droit, l'économie, le commerce, etc. Quelques domaines sont plus développés que d'autres parce qu'ils sont plus proches des centres d'intérêt de la communauté. Ainsi, par exemple, les noms des régions de chaque état membre de la communauté est dans Eurovoc mais pas ceux des pays qui n'appartiennent pas à la communauté. Il a pour objectif de représenter d'une façon non équivoque (univocal way) les documents et les concepts de recherche. Il contient des descripteurs (mots ou expressions qui décrivent sans ambiguïté les concepts), des non-descripteurs (mots ayant des sens équivalents ou expressions pour les descripteurs) et des relations sémantiques entre descripteurs d'une part et entre descripteurs et non-descripteurs d'autre part. Les concepts d'Eurovoc sont utilisés pour décrire les documents, pour faire une recherche par mots-clés et aussi pour étendre la recherche à d'autres documents décrits par le même concept. Il contient 6501 descripteurs qui sont répartis dans 21 domaines et 127 microthésaurus. Consultable en ligne, ou en fichiers pdf (gratuitement).
ECLAS ²¹	Thésaurus ECLAS (European Commission Library Automated System) : édition de janvier 2005. Bilingue français/anglais. Mis à jour 2 à 3 fois par an. Édité par la Bibliothèque Centrale de la Commission Européenne. Domaines d'activités de l'Union Européenne. Environ 6 300 descripteurs complétés par 12 000 non-descripteurs dans d'autres langues, répartis dans 19 domaines. Consultable en ligne.
Interdoc ²²	Édité par Interdoc, l'association des documentalistes de Conseils généraux (France). Représente un langage documentaire, hiérarchisé et normalisé, et a pour ambition d'harmoniser et de rendre cohérent le traitement documentaire des services documentation des collectivités territoriales. Contient 8563 descripteurs et 1012 non-descripteurs répartis dans 21 domaines.
Urbamet ²³	Thésaurus bilingue français/anglais. Édité par l'Association Urbamet. Il couvre les champs thématiques de l'urbanisme, l'aménagement, l'habitat, la construction, l'architecture et les équipements. Contient 4151 descripteurs, 497 non descripteurs et 348 termes reliés par la relation associative, répartis dans 24 champs sémantiques. Consultable en ligne.
Jurivoc ²⁴	Thésaurus juridique trilingue français/allemand/italien, mis à jour mensuellement et édité par le Tribunal fédéral suisse. Environ 9500 descripteurs et 20 000 non-descripteurs par langue, répartis dans 37 champs sémantiques. Téléchargeable et consultable en ligne.

TABLE 2.4 – Ontologies juridiques.

Ressource	Description
FOLaw [Breuker and Hoekstra, 2004]	<i>Functional Ontology of Law</i> . Ontologie noyau du droit (<i>core ontology</i>) développée dans le Leibniz Center of Law pour définir une base réutilisable comme le dénominateur commun des différents domaines juridiques. FOLaw forme le point de départ d'un certain nombre d'ontologies et systèmes de raisonnement juridiques dans divers projets européens.
LRI-Core [Breuker and Hoekstra, 2004]	Ontologie noyau du droit basée sur les notions de sens-commun (<i>common sense</i>). Elle se compose de cinq grandes parties ("worlds") : classes physiques, classes mentales, classes abstraites, rôles et événements.
LKIF-Core [Hoekstra et al., 2009]	Développée dans le cadre du projet européen Estrella (European project for Standardised Transparent Representations in order to Extend Legal Accessibility). Trois types différents d'utilisateurs sont visés : les citoyens, les professionnels et les juristes. Les trois groupes d'utilisateurs ont fourni des termes de domaine qui ont été classifiés selon leur degré d'abstraction. Cette classification initiale a donné lieu à des clusters de concepts qui ont été reliés à des catégories de LRI-Core.
DOLCE [Gangemi et al., 2002]	<i>Descriptive Ontology for Linguistic and Cognitive Engineering</i> . Ontologie fondationnelle (OF) développée dans le cadre du projet EU WonderWeb. Les OFs sont indépendantes d'un domaine, contiennent une axiomatisation riche de leurs vocabulaires. DOLCE est une OF top-level. DOLCE+ est une extension de DOLCE qui contient quelques modules dédiés aux ontologies noyaux de contextes, temps, espace, etc.
CLO [Gangemi et al., 2005]	<i>Core Legal Ontology</i> . CLO formalise les catégories du domaine juridique qui existent dans n'importe quel système juridique, comme loi, norme juridique, régulation, agent juridique et rôle juridique, etc. Ces catégories sont connectés à l'ontologie fondationnelle DOLCE+.

2.3.4 Synthèse

Le web sémantique, par ses différentes techniques, offre beaucoup d'opportunités pour le traitement de l'information juridique, en particulier la législation. Ces techniques ont permis de faciliter le processus de production interne aux instances juridiques (écriture de la loi, maintenance des sources de loi, gestion des workflows et procédures législatives), améliorer l'interaction avec les acteurs externes (publication des procédures et des informations, communication avec les citoyens, dialogue avec les institutions nationales et internationales). La définition des standards et la création de nouvelles techniques appropriées pour les documents législatifs peut effectivement créer et nouer le lien entre la production de la législation et son utilisation dans la communauté juridique [Sartor et al., 2010].

2.4 Méthodes d'accès à l'information juridique

La présentation de l'information juridique a évolué. Ce changement est essentiellement dû à l'évolution des technologies de l'information et de la communication et les développements dans l'informatique juridique. La progression rapide de la numérisation de l'information juridique fait qu'une large quantité de textes de loi est disponible au format électronique : législation, réglementations, décisions administratives, jurisprudence, contrats, données fiscales, etc. Cette numérisation a permis entre autres le transfert et l'échange sur internet des textes de loi. Dans certains domaines, le web est d'ores et déjà la principale source d'information juridique pour les juristes et les citoyens.

Un des résultats de la numérisation des textes de loi, c'est la diversité, toujours en croissance, des fournisseurs de l'information juridique. Entre organismes publics et entreprises privées, le duel dure depuis le début des années 1970 [Sartor et al., 2010]. En même temps, internet a favorisé l'émergence de nouveaux acteurs dans la fourniture de l'information juridique. Instituts d'information juridique, établissements d'enseignement, associations professionnelles, cabinets juridiques et centres de recherche offrent une très grande quantité d'information juridique en libre accès. Il existe aussi des portails qui ont pour objectif spécifique d'offrir l'accès à des ressources juridiques en ligne.

2.4.1 Portails généralistes de sources de droit

Legifrance ²⁵ est le portail officiel des données juridiques du gouvernement français. Il offre un service public pour la diffusion du droit national, européen et international. Les documents sont accessibles en ligne et en libre accès. Le site présente ou fait référence à tous les textes en vigueur depuis 1539 et la jurisprudence des tribunaux supérieurs depuis 1986. Plusieurs liens vers d'autres sites juridiques sont également répertoriés sur Legifrance. Le portail offre un large éventail de types de documents juridiques qui sont classés par catégories :

- droit français : lois et règlements (constitution, codes en vigueur, autres textes législatifs et réglementaires), jurisprudence (constitutionnelle, administrative, judiciaire), conventions collectives ;
- droit européen : traités européens, journal officiel de l'Union Européenne, transposition des directives, jurisprudence européenne ;
- droit international : traités internationaux, jurisprudence internationale.

25. <http://www.legifrance.gouv.fr>

L'accès aux documents de la base de données de Legifrance se fait soit par interrogation soit par navigation. L'interrogation se fait en introduisant un ou plusieurs mots-clés et le résultat est une liste de documents ou parties de documents (articles) qui contiennent au moins l'un des mots-clés de la requête. La navigation se fait selon les types et les thèmes des documents qui sont organisés ainsi dès la page d'accueil et en suivant les liens hypertextes entre les documents.

EUR-Lex ²⁶ est un portail d'accès au droit de l'union européenne : traités, législation, jurisprudence, travaux préparatoires, questions parlementaires. EUR-Lex donne un accès libre aux documents juridiques officiels publiés par les institutions de l'union européenne ainsi qu'aux autres documents considérés comme publics. Il est géré par l'office de publications de l'union européenne. Le site contient aux alentours de 3 650 000 documents dans 23 langues avec des textes postérieurs à 1951. La base de données est mise à jour quotidiennement. Chaque année, à peu près 15 000 nouveaux documents sont ajoutés à la base.

Le site offre des fonctionnalités simples (par mots clés, numéro de document, date, référence du Journal Officiel, numéro CELEX, etc.) ou avancées (selon le type des documents : traités, accords internationaux, législation en vigueur, législation consolidée, jurisprudence, questions parlementaires ou travaux préparatoires) pour l'accès aux documents.

UK Legislation Le site *UK legislation* ²⁷ est le lieu officiel de publication de la législation récemment adoptée au Royaume-Uni. Les versions originales (adoptées) et révisées de la législation sont publiées par et sous l'autorité du Contrôleur de HMSO (Her Majesty's Stationery Office). Le site comporte la plupart des types de la législation avec leurs documents explicatifs. Toute la législation postérieure à 1988 et une grande partie de la législation antérieure sont disponibles sur le site. La plupart des types de législation primaire (par exemple, lois, mesures, ordonnances du conseil) sont sous la forme 'révisé' : les modifications apportées par la législation ultérieure sont incorporées dans le texte. Le site permet de rechercher les modifications apportées par la loi depuis 2002.

Normattiva ²⁸ est le site web officiel de l'état italien publié le 19 Mars 2010 et créé par le décret-loi du 22 Décembre 2008, n.200, sur les mesures urgentes pour la simplification de la législation, qui vise à la création d'un service gratuit de consultation des lois italiennes. Il contient, à l'heure actuelle, les normes italiennes depuis 1940. Le site, en plus de mettre à jour immédiatement la base de données avec les nouvelles normes publiées au Journal officiel de la République Italienne, offre un service de consultation en mode multi-validité ²⁹, qui permet la consultation d'une norme à une date donnée, puis l'affichage du document dans sa précédente version et éventuellement toutes les modifications qui ont eu lieu après la date indiquée.

Portails privés Le site de Legifrance propose une liste de portails privés ³⁰ (éditeurs juridiques, universités, centres de recherche ou associations, etc.). Cette liste est proposée dans le but de « développer la synergie entre la mission de service public de diffusion des données essentielles du droit français assurée par Legifrance, et la valeur ajoutée apportée par les sites juridiques privés, payants ou non, grâce aux sélections, commentaires et enrichissements de toutes sortes qu'ils effectuent ».

26. <http://eur-lex.europa.eu/fr/index.htm>

27. www.legislation.gov.uk/

28. <http://www.normattiva.it/>

29. Le terme "multi-validité" signifie, en particulier, le mode d'édition utilisé pour mettre à jour l'instrument juridique qui permet à l'utilisateur de visualiser le chemin historique de la loi et les changements qu'elle a subis au fil du temps, avec les dates correspondantes de validité.

30. <http://www.legifrance.gouv.fr/Sites/Portails-juridiques>

2.4.2 Outils spécialisés

Plusieurs techniques et outils ont été proposés pour l'exploitation du contenu de la réglementation [Lau, 2004, Geist, 2009, Chieze et al., 2010, Palmirani et al., 2003, Amardeilh et al., 2013].

Dans [Chieze et al., 2010] les auteurs présentent leur système DecisionExpress qui offre un bulletin des décisions récentes des *Canadian federal courts* et *provincial tribunals*. Le système traite automatiquement les décisions juridiques et fait en sorte que les informations quotidiennes utilisées par les juristes soient plus accessibles en les présentant sous forme de résumés. Le système permet aussi d'extraire l'information essentielle de l'ensemble de ces décisions de même type et de les présenter de façon accessible sous forme de feuillets d'information (*factsheets*). Les auteurs proposent un outil de recherche permettant de faire des recherches dans la base de données du *Canadian federal courts and tribunals*. L'outil offre de nouvelles fonctionnalités de recherche, en plus de celles proposées par la plupart des fournisseurs canadiens de l'information juridique (QuickLaw³¹, Westlaw-Carswell³²) qui se basent sur la recherche dans les *factsheets*. L'utilisateur peut formuler sa requête en se basant sur le nom du juge, sa conclusion, le domaine de la loi, le sujet de la décision, les mots-clés, etc.

Au niveau national, une plate-forme d'accès à l'information juridique a été développée dans le cadre du projet Légilocal [Amardeilh et al., 2013]. La plate-forme a deux caractéristiques principales :

- Elle permet aux juridictions locales de rendre leurs actes publics et accessibles en ligne sur leurs sites web (en consultation et en recherche). Ce résultat est obtenu grâce à l'utilisation de vocabulaires sémantiques (les annotations des documents) qui sont enrichis de méta-données pour la recherche, et un moteur de recherche sémantique qui permet aux acteurs (personnel administratif, représentants ou citoyens) de trouver les documents pertinents.
- Elle permet aux secrétaires de mairies d'éditer des actes plus rapidement et de produire des actes plus sûrs. Ce résultat est obtenu par le partage de documents entre les différentes municipalités et avec des experts, grâce à des outils qui soutiennent la gestion de contenu et l'interaction humaine dans ce contexte et à travers un système de détection basé sur la vérification de la validité de documents.

La plate-forme Légilocal (figure 2.6) se base sur quatre fonctionnalités principales : la gestion des documents, l'enrichissement sémantique des documents, la recherche sémantique et la mise en réseau des acteurs et des documents. Elle est composée d'un ensemble complexe d'outils et de référentiels et présente deux fonctionnalités : le réseau social REZODAC pour l'usage interne des secrétaires de mairies et un moteur de recherche sémantique intégré dans les sites web des municipalités pour les citoyens.

Sur la figure 2.6, trois catégories d'utilisateurs peuvent interagir avec la plate-forme Légilocal :

- l'éditeur, administrateur de la plate-forme, met en place et maintient REZODAC, gère les ressources sémantiques et publie les sources des documents éditoriaux ;
- les secrétaires de mairies sont responsables de l'édition et de la publication des documents administratifs (à la fois dans REZODAC et sur le web), mais ils peuvent également rechercher des documents et de l'expertise au sein du réseau REZODAC. Toutes ces tâches sont effectuées dans le réseau sur lequel les employés doivent être connectés ;
- les citoyens ont accès au moteur de recherche sémantique Légilocal, qui est intégré comme un simple widget dans les sites des mairies.

31. www.lexisnexis.ca/en-ca/products/quicklaw-full-service.page

32. <http://www.carswell.com/products/westlawnext-canada/>

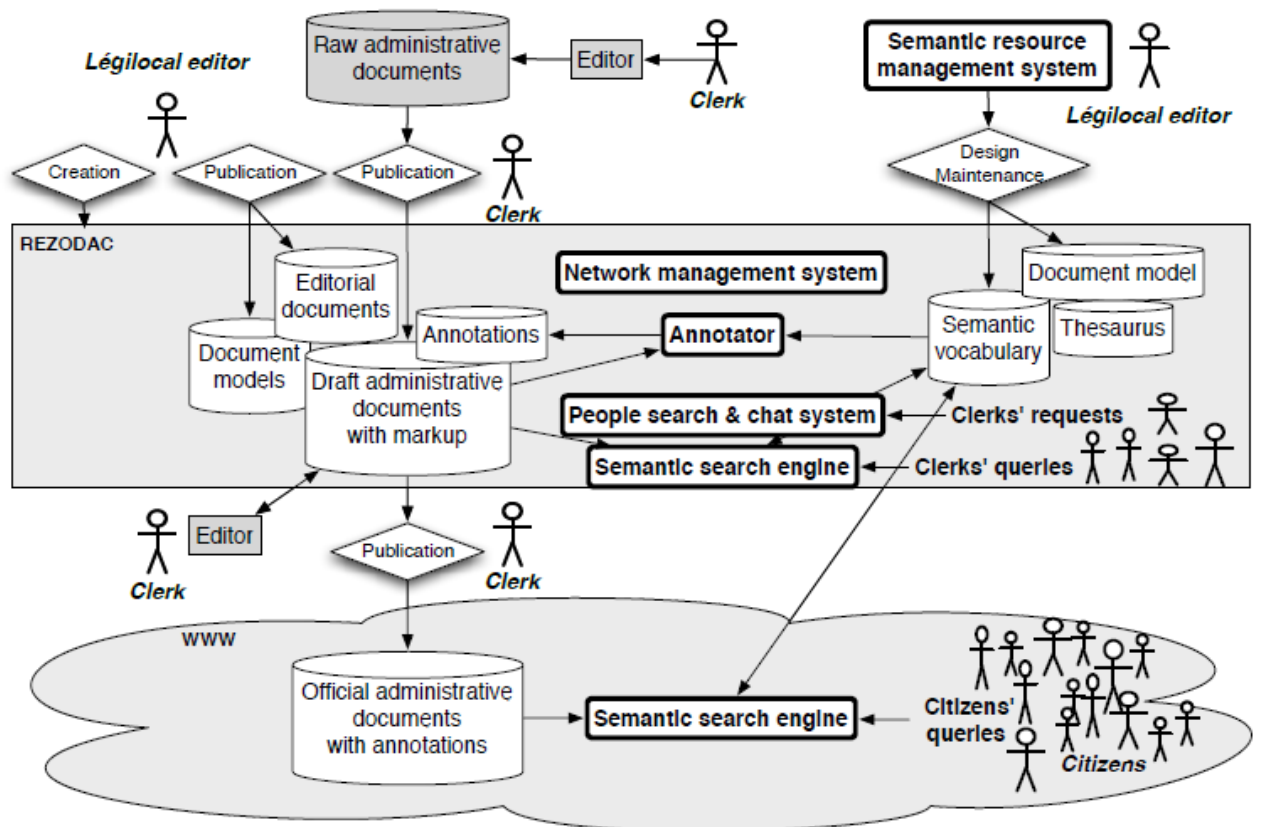


FIGURE 2.6 – La plate-forme Légilocal [Amardeilh et al., 2013].

La gestion unifiée des contenus et interactions sur le contenu repose sur des outils développées ou adaptés dans le cadre du projet Légilocal [Amardeilh et al., 2013] :

1. le système de gestion de réseau permet l'installation du réseau REZODAC qui contrôle la circulation de l'information entre les acteurs et le traitement des documents ;
2. l'annotateur qui enrichit les documents avec des balises, des métadonnées et des annotations sémantiques ;
3. un moteur de recherche sémantique qui permet la navigation à facettes basée sur des documents et le raffinement de requêtes ;
4. un système de mise en réseau qui permet la recherche experte et le *chat* dans REZODAC.

Un système de gestion de ressources, qui ne fait pas partie de la plate-forme Légilocal, est utilisé pour la conception et l'entretien des ressources sémantiques. La plate-forme s'appuie également sur divers types de ressources. Les documents sont regroupés dans une grande base de documents. Certains sont publiés par l'éditeur pour assister les secrétaires de mairies dans la production des actes, mais la plus grande partie de la base est composée d'actes administratifs produits par les secrétaires de mairies en REZODAC et progressivement enrichis avec des métadonnées et des annotations sémantiques. Les ressources sémantiques sont utilisées pour annoter et rechercher dans la base de documents.

2.4.3 Données gouvernementales ouvertes sur le web

Des organismes gouvernementaux et des organisations du secteur public produisent une grande quantité de données : données statistiques, données économiques, registres d'entreprises, résultats de vote des élus, etc. Dans de nombreux pays, une partie significative de ces données est mise en ligne par souci de transparence. Ces données sont devenues facilement accessibles et ont permis à des utilisateurs de les analyser et d'en tirer de nouvelles connaissances qui peuvent produire de nouveaux services (par ex. la proposition sous forme structurée des emplois de la fonction publique³³). L'utilisation des techniques du web sémantique pour la publication des données sous forme de données liées (Linked Open Data [Heath and Bizer, 2011]) a montré qu'elles facilitent l'accès aux données gouvernementales, comme dans le cas des initiatives data.gov.uk³⁴ et data.gov³⁵.

2.5 Traitement de l'intertextualité

Dans le domaine juridique, la cohérence des composants de la loi est exigée. La vérification de la cohérence ne peut se faire qu'à travers l'étude des liens intertextuels entre les sources de droit (vérification des interactions inter-réglementaires). L'avancée des techniques de traitement et d'accès à l'information juridique a rendu plusieurs tâches, difficiles et fastidieuses il y a quelques années, plus faciles pour les utilisateurs (juristes, secrétaires de mairies, citoyens). Les techniques et outils proposés ont traité la structure d'un document dans tous ses détails (structure logique du texte, les concepts, les dates, etc.) ce qui permet une interrogation plus précise sur le contenu d'un document. L'étude de la structure de la collection documentaire (les documents considérés dans leur ensemble aussi bien qu'individuellement) dans un but de recherche d'information a reçu moins d'attention.

33. <http://www.civilservice.gov.uk/>

34. <http://data.gov.uk/linked-data>

35. <http://www.data.gov/semantic>

L'un des défis de tout système de RI juridique est de gérer la complexité du réseau de sources juridiques qui contient les informations nécessaires à l'utilisateur. Habituellement, cette information est répartie sur les différents documents de la collection. En d'autres termes, la connaissance juridique est structurée en morceaux contenus dans divers documents et le but de l'utilisateur est de les identifier et de les interpréter conjointement. Un système de RI juridique doit donc permettre de suivre les « traces des connexions » entre des éléments de connaissances juridiques et de les présenter de manière cohérente à l'utilisateur. Ces traces sont définies comme des références explicites et implicites. Identifier les références implicites demande des connaissances très spécialisées (ontologies, règles, etc.) contrairement aux références explicites qui sont plus directement accessibles par leur représentation textuelle [Brighi and Palmirani, 2009]. L'identification des références explicites a permis de mesurer la complexité juridique en termes d'intertextualité, fournissant ainsi une idée approximative de la quantité de références croisées que les professionnels du droit doivent connaître sur le domaine réglementaire étudié.

La complexité du droit est bien illustrée par plusieurs travaux qui se situent à l'intersection du domaine juridique et de la théorie des graphes. Ces travaux sont classés en deux grandes catégories : analyse du réseau à haut-niveau (*macro-level network analysis*), qui explorent la structure globale du réseau de citations, et à bas-niveau (*micro-level network analysis*) qui se concentrent sur une granularité plus fine des documents [Gultemen and van Engers, 2013]. Par exemple, une analyse du réseau de citations de la cour suprême des États-Unis est présentée dans [Chandler, 2005, Fowler et al., 2007, Fowler and Jeon, 2008]; le code des États-Unis est également analysé comme un réseau dans [Bommarito and Katz, 2009]; dans [Boulet et al., 2009] les auteurs font une analyse similaire du réseau constitué par les citations dans le code de l'environnement français et dans [Winkels and de Ruyter, 2011] un réseau de citations de la cour suprême néerlandaise (15053 décisions entre 1965 et 2008 avec 106559 citations) a été étudié. Ces travaux sur la structure des réseaux de réglementations ont fourni un certain nombre d'indicateurs aux acteurs du domaine qui aident à la compréhension des caractéristiques et au suivi de l'évolution de ces réseaux mais pas à des fins de recherche d'information.

À ce jour, les liens juridiques entre les documents ont été exploités de façon limitée par les systèmes de recherche d'information à des fins d'interrogation. Dans ce qui suit nous présentons quelques exemples de la gestion des liens entre des documents juridiques dans des systèmes opérationnels de recherche documentaire juridique.

Légifrance Les liens explicites sont traités le plus souvent manuellement. Certains d'entre eux sont formalisés sous la forme de liens navigables (liens hypertextes) mais les liens ne sont parfois conservés que sous la forme d'états juridiques (attributs) associés aux documents : Vigueur (V), Vigueur différée (VD), Vigueur avec terme (VT), Abrogé (Ab), Annulé (A), Disjoint (D), Modifié (M), Périmé (P), Substitué (S), Transféré (T).

UK Legislation L'utilisateur peut interroger la base de données en spécifiant la législation modifiée ou la source juridique qui introduit le changement. La liste des résultats indique également les types de modifications (mots abrogés, insertion, abrogation partielle, cessation d'effets, renumérotation, etc.) effectuées sur les documents. Le système traite le lien général « modifie / modifié par » comme une relation entre documents mais des types de modifications plus spécifiques ne sont représentées que comme des attributs de documents. *UK Legislation* permet également d'accéder à la version d'un document juridique en vigueur à une certaine date. « En vigueur » est un attribut qui peut être associé aux documents juridiques dans le système de recherche d'information de *UK Legislation*.

Normattiva Le site permet aussi l'accès *point-in-time* à la législation, de telle sorte que l'uti-

lisateur puisse récupérer les différentes versions d'un document en vigueur à des dates différentes.

À partir de l'analyse de ces systèmes, nous pouvons distinguer quatre façons d'exploiter les liens explicites entre les documents juridiques. La liste ci-dessous décrit les différentes techniques de représentation des liens classées de la moins opérationnelle à la plus opérationnelle [Mimouni et al., 2013] :

- les liens sont représentés comme des chaînes de caractères dans le texte du document : en général, ils apparaissent dans la partie finale du document et sont ajoutés manuellement (par une équipe éditoriale). Les liens intégrés dans le texte ne peuvent être interrogés que par des requêtes en plein texte ;
- les liens sont encodés sous la forme d'hyperliens qui sont navigables lors de la consultation du document. Les liens sont des références qui pointent vers des objets de la collection (d'autres documents juridiques ou des fragments de ces documents). Cette représentation rend les liens plus opérationnels car ils permettent à l'utilisateur d'accéder aux documents cités directement sans interroger à nouveau la base de données ;
- les liens sont représentés comme des statuts juridiques qui sont interrogeables comme des attributs de documents. L'utilisateur peut chercher des documents avec des attributs comme « modifié » ou « abrogé » ;
- Les liens juridiques sont intégrés dans la base documentaire comme des liens entre documents qui sont interrogeables *via* des requêtes relationnelles. Seule la base *UK Legislation* présente cette fonctionnalité mais pour une unique relation générique de « modification » sans précision sur le type de la modification.

En résumé, les trois systèmes présentés prennent en compte les liens juridiques, mais de façon limitée et rarement sous la forme de relations entre documents à proprement parler. Les systèmes traitent le statut juridique résultant des relations entre les documents, plutôt que des relations elles-mêmes : au lieu de représenter le fait que « le document x modifie le document y », le système encode le fait que le document y a un statut juridique « modifié ».

L'objectif de notre travail est d'aller plus loin dans le traitement de l'intertextualité en représentant plusieurs types de liens juridiques comme des relations entre les documents de la collection et en exploitant ces relations dans un système de recherche d'information juridique acceptant des requêtes relationnelles. Nous estimons que cette représentation reflète de manière plus précise la façon dont les professionnels du droit conçoivent le réseau des normes et permettra une interaction plus naturelle entre l'utilisateur et le système.

2.6 Conclusion

L'accès aux connaissances juridiques présente des défis particuliers pour les systèmes de recherche d'information :

- les connaissances juridiques sont souvent exprimées dans des formes linguistiques complexes et possèdent des structures complexes ;
- la complexité due au facteur d'intertextualité et aux différents types de liens qui existent entre les documents ;
- le besoin d'exhaustivité des résultats.

Les systèmes d'accès à l'information juridique existants ne proposent pas de solutions directes pour prendre en compte une recherche d'information qui porte aussi bien sur le contenu sémantique que sur les liens intertextuels. Ils contournent cette difficulté avec des techniques simples, par exemple, en modélisant les liens comme des attributs qui sont intégrés dans la base (par exemple « modifié par », « abrogé par ») et qui peuvent être interrogés. Les résultats retournés

ne se présentent pas comme des graphes et l'utilisateur est amené à parcourir les liens hypertextes pour construire le contexte de la réponse.

On ne peut se contenter de visualiser les citations et de proposer des systèmes pour naviguer de proche en proche dans les bases documentaires (un utilisateur peut facilement s'y perdre sans trouver ce qu'il cherche). Il faut proposer des outils de recherche d'information axés sur l'intertextualité pour retrouver les documents en fonction des liens qu'ils entretiennent. Ceci représente l'objectif principal de ce travail de thèse.

Le chapitre suivant décrit les méthodes de recherche d'information classique et sémantique existantes et donne les définitions de base des techniques utilisées. Nous positionnons notre travail par rapport à ces approches, notamment celles qui intègrent la dimension intertextuelle.

Chapitre 3

Recherche d'information et graphe de documents

Sommaire

3.1	Introduction	29
3.2	Recherche d'information classique	30
3.2.1	Indexation ou processus de représentation	30
3.2.2	Appariement ou processus de recherche	31
3.2.3	Tri de résultats	32
3.2.4	Reformulation de requêtes	32
3.2.5	Modèles de RI	32
3.2.6	Mesures d'évaluation	33
3.2.7	Interface utilisateur	33
3.3	Recherche d'information sémantique	34
3.3.1	Annotation sémantique	34
3.3.2	Modèles de RI numériques et à base de connaissances	35
3.3.3	Modèles logiques de RI	37
3.4	RI et Analyse de liens	38
3.4.1	Intertextualité dans les systèmes de RI existants	38
3.4.2	Analyse de graphes de citation	39
3.4.3	Analyse des liens hypertextes (algorithmes Page Rank et HITS)	39
3.4.4	Analyse socio-sémantique	40
3.5	Conclusion	40

3.1 Introduction

Le but de ce chapitre est d'étudier comment l'intertextualité est prise en compte dans les systèmes de RI existants. Les systèmes de recherche d'information servent d'interface entre la collection de documents et les utilisateurs. Ils proposent des fonctionnalités de stockage, d'organisation, de recherche d'information en réponse à des requêtes et de retour de l'information pertinente pour ces requêtes. Différents modèles de représentation de l'information (stockage, organisation), de mécanismes d'appariement (recherche d'information en réponse à des requêtes) et

d'interfaces (retour de l'information pertinente pour ces requêtes) ont été proposés pour améliorer les performances des systèmes de RI³⁶.

Les systèmes de RI ont également vu une amélioration grâce à l'essor des techniques sémantiques. En effet, l'énorme augmentation de la quantité et la complexité de l'information accessible sur le web a provoqué une demande pour des outils et des techniques qui peuvent traiter les données sémantiquement. L'approche classique de recherche d'information repose principalement sur la recherche par mots-clés dans les textes des documents, eux-mêmes modélisés avec des sacs de mots sans tenir compte de l'information sémantique. Des modèles de représentation de connaissances, principalement les ontologies, sont proposés pour faire face à ce problème en ajoutant une couche de sémantique aux textes bruts (métadonnées, concepts). Ils forment actuellement la base de tout système de RI sémantique.

Bien que les interfaces d'interrogation les plus classiques soient à base de mots clés (comme dans Google), les systèmes spécialisés de RI s'orientent vers les techniques sémantiques basées sur des modèles logiques. C'est le cas dans le domaine juridique, avec plusieurs initiatives d'ouverture de données gouvernementales lesquelles sont annotées avec des métadonnées ou vocabulaires ouverts et partagés. De plus, le besoin d'exhaustivité des résultats dans ce domaine exige une recherche booléenne dans les collections de documents, caractéristique des techniques d'interrogation sémantique.

Dans cette thèse nous nous intéressons à l'accès à l'information dans le domaine juridique. Notre travail s'intègre dans le contexte de la recherche d'information sémantique et en particulier celle qui repose sur un modèle logique.

Ce chapitre commence par présenter et définir brièvement les concepts de base de la RI classique (section 3.2), avant de décrire les notions de base de la RI sémantique (section 3.3) et ses différents modèles (numérique et logique). Une description des principaux modèles de traitement de l'intertextualité dans les systèmes existants de RI est donnée dans la section 3.4.

3.2 Recherche d'information classique

La figure 3.1 présente une vue d'ensemble d'un système de RI : un utilisateur exprime des besoins en information via l'interface qui lui est proposée (en langage naturel, par formulaire, etc.), le système de RI construit une représentation des documents de la collection interrogée et de la requête sous forme d'index, ensuite il compare les deux représentations afin d'établir la correspondance entre eux et identifier, selon des métriques prédéfinies, les documents pertinents pour la requête. Une fonction de classement peut être exécutée par la suite pour trier les documents selon leur degré de pertinence. Selon les résultats, l'utilisateur peut choisir de procéder à un raffinement de requête. Dans le cas d'un système de RI sémantique, des ressources sémantiques (ontologie, thésaurus, etc.) peuvent être utilisées pour l'indexation ou pour la reformulation de requêtes afin d'améliorer les résultats de la recherche. Les étapes de ce processus sont décrits dans les sections suivantes.

3.2.1 Indexation ou processus de représentation

L'indexation est un processus de représentation qui a pour rôle d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu. Le résultat de l'indexation constitue une description du document ou de la requête, qui est une liste de termes

36. Les définitions des notions utilisés dans ce chapitre se basent en partie sur <http://www-nlp.stanford.edu/IR-book/> et [Baziz, 2005].

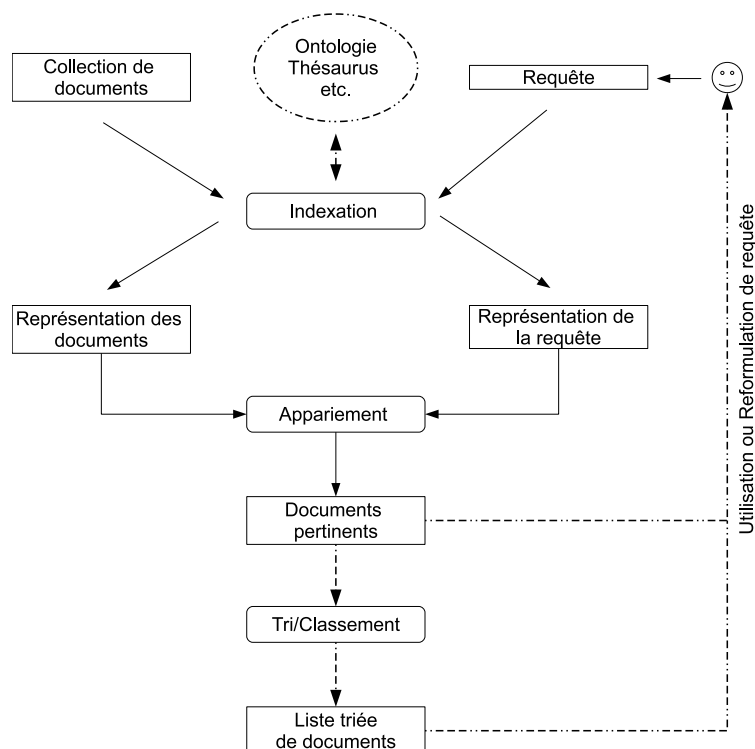


FIGURE 3.1 – Vue générale d'un système de recherche d'information.

significatifs pour l'unité textuelle correspondante, auxquels sont généralement associés des poids pour différencier leur degré de représentativité.

L'indexation est une étape très importante dans un processus de recherche d'information. Un index optimise les performances d'interrogation et améliore considérablement le temps de réponse en stockant les termes dans une structure de fichier inversé. La restitution des documents en réponse à une requête dépend fortement de la qualité de l'indexation. La méthode classique, qu'est l'indexation par sacs de mots, comprend deux étapes : la recherche des termes décrivant le contenu qu'on peut appeler aussi descripteurs (extraction automatique de descripteurs, élimination des mots outils (*stopwords*), lemmatisation, repérage de groupes de mots) et l'évaluation de la représentativité de ces termes (pondération) [Ralalason, 2010].

L'indexation par métadonnées est une méthode qui a été proposée pour améliorer les résultats d'une recherche d'information. Cette méthode s'appuie sur un ensemble d'annotations qu'on appelle métadonnées. Ces informations sont attachées aux documents et décrivent leurs caractéristiques techniques comme la date de publication, l'auteur, etc.

3.2.2 Appariement ou processus de recherche

C'est la base d'un système de RI. Dans cette phase, les termes de la requête sont recherchés dans l'index. Tous les documents contenant des occurrences des termes de la requête sont récupérés. Selon les systèmes, la récupération peut se faire même pour des documents partiellement compatibles.

Les systèmes de recherche d'information se caractérisent par le modèle d'appariement document-requête : la fonction de décision qui permet d'associer à une requête, l'ensemble des documents

pertinents à restituer en mesurant la pertinence d'un document vis-à-vis de la requête. Cette fonction est liée au modèle de représentation des documents et des requêtes (à la phase d'indexation). La valeur de pertinence est calculée à partir d'une fonction de similarité qui tient compte des poids des termes déterminés généralement en fonction d'analyses statistiques et probabilistes (voir section 3.2.5).

3.2.3 Tri de résultats

Selon le degré d'accord entre les documents et les termes de la requête, des scores sont affectés aux documents récupérés. Ils servent à trier les résultats : les documents les plus pertinents sont présentés à l'utilisateur en haut de la liste de résultats. Le processus d'ordonnement dépend fortement du modèle de RI (section 3.2.5), certains modèles ne proposent pas l'ordonnement comme les modèles logiques en RI sémantique (tous les documents extraits sont considérés comme ayant la même pertinence).

3.2.4 Reformulation de requêtes

Certains systèmes proposent la fonctionnalité de reformulation automatique de requêtes afin d'améliorer la précision des résultats retournés. Elle consiste à modifier la requête de l'utilisateur en ajoutant des termes estimés significatifs (par ex. les termes des documents les plus pertinents retournés par le système) ou en modifiant leurs poids.

3.2.5 Modèles de RI

Les systèmes de RI peuvent être classés en trois groupes selon le modèle qu'ils utilisent pour la représentation et l'appariement des documents et des requêtes, modèle qui influe de façon directe les performances des systèmes.

Modèle booléen C'est un modèle simple basé sur la théorie des ensembles et l'algèbre booléenne. Les documents sont représentés par des ensembles de termes et la requête est représentée sous forme d'une expression logique. Les termes qui indexent la requête sont reliés par les connecteurs logiques ET(\wedge), OU(\vee) et NON(\neg). Un document est retourné s'il contient tous les termes exprimés par l'expression logique. Ce modèle est intuitif, facile à mettre en oeuvre et permet pour un utilisateur expérimenté d'avoir une recherche très restrictive. L'approche booléenne a cependant l'inconvénient de ne rien retourner quand aucun document "vraiment" pertinent n'est trouvé.

Modèle vectoriel (VSM) Ce modèle représente à la fois les documents et les requêtes par des vecteurs de termes pondérés. Les documents sont retrouvés en fonction du degré de similarité de leurs vecteurs avec le vecteur de la requête. Les principales mesures de similarité sont le produit scalaire, la mesure de Jaccard et le cosinus. Contrairement au modèle booléen, le modèle vectoriel peut retourner des documents pertinents dont la représentation ne correspond qu'approximativement à la requête [Salton et al., 1975].

Modèle probabiliste Dans ce modèle, un ensemble de documents pertinents par rapport à la requête est précalculé. La recherche se fait en fonction des probabilités d'appartenance à cet ensemble. Le processus de recherche se traduit par un calcul de proche en proche, du degré ou probabilité de pertinence d'un document relativement à une requête. Le principe est le suivant : s'il existe des documents pertinents et non pertinents connus, alors il est possible d'estimer la probabilité d'un terme t apparaissant dans un document pertinent D_i (le terme t est pertinent pour la requête) à apparaître dans un document D_j .

3.2.6 Mesures d'évaluation

La performance des systèmes de RI peut être mesurée à l'aide de plusieurs paramètres d'évaluation. Pour utiliser l'un de ces paramètres, il est nécessaire de préparer une référence (*gold standard*) pour chaque requête afin de décider pour chaque document résultant s'il est considéré comme pertinent ou non par rapport à la requête. Les mesures les plus courantes utilisées dans l'évaluation des systèmes IR sont la précision, le rappel, la F-Mesure et la précision moyenne (Mean Average Precision, MAP). D'autres mesures sont aussi acceptées comme par exemple le temps de réponse d'un système, la présentation des résultats, la clarté et la facilité d'utilisation des interfaces.

3.2.7 Interface utilisateur

Interface d'interrogation

L'interface utilisateur est l'un des aspects les plus importants dans un système de RI. Un compromis doit être fait entre la facilité d'utilisation de l'interface et les performances du système : des interfaces simples sont plus faciles à utiliser mais peuvent entraîner des requêtes ambiguës, alors que des interfaces plus complexes fournissent plus de détails et aident à une formulation précise de la requête, mais elles sont encombrantes et fastidieuses pour l'utilisateur final. Les interfaces à base de mots clés, en langage naturel, par formulaires ou à base de graphes sont quelques-unes des méthodes couramment utilisées dans la littérature [Baeza Yates and R., 1999]. Dans cette thèse, le choix a été fait pour les interfaces à base de formulaire en proposant à l'utilisateur des listes de choix basée sur les termes indexant les documents et les types des liens qu'ils entretiennent.

Interface de résultats

La présentation des résultats est une étape importante dans un système de RI. Plusieurs méthodes de visualisation (ou de restitution) de résultats ont été proposées : textuelles ou graphiques. Les documents retournés peuvent être présentés par :

- une liste à plat : titre avec extraits et adresse. C'est la technique la plus classique et la plus utilisée (utilisée par Google). L'interface propose un affichage linéaire des résultats de recherche sous forme d'une liste triée selon un critère de pertinence ;
- des liens pour divers sous-ensembles des résultats. Le principe du premier point est repris mais en ajoutant une catégorisation des résultats dans des sous-ensembles significatifs, *via* une technique de clustering statique ou à la volée. Les sous-ensembles sont construits par :
 - regroupement par descripteurs (entités nommées extraites des documents par calcul d'un score) ;
 - regroupement par catégories de plan de classement statique (pré-existant), adapté par exemple dans la cas de commerce en ligne ou de fils d'actualité ;
 - regroupement par catégories calculées dynamiquement : *clustering*. Le *clustering* vise à répartir un ensemble de réponses (documents) en sous-ensembles, appelés clusters, de façon à maximiser la cohérence interne à chaque cluster et la différence entre clusters. Cette technique est fondée sur une notion de similarité entre documents et clusters. Nous distinguons plusieurs types de clustering : plat ou hiérarchique, dur (*hard*) ou flou (*soft*) selon que les documents appartiennent à un cluster exclusivement ou potentiellement à plusieurs clusters.

Certaines interfaces de présentation de résultats proposent des méthodes de navigation dans les résultats retournés. La navigation peut être :

- contextuelle, de proche en proche, en suivant des liens entre les documents. En cours de navigation, les documents qui ne contiennent pas les descripteurs recherchés peuvent être filtrés. La navigation peut aussi repartir sur une nouvelle recherche ;
- par similarité avec un document retourné. Ceci correspond à faire d'un document une nouvelle requête (fonctionnalité *more like this*) ;
- par exploration d'un graphe organisant les documents retournés en exploitant les relations (liens hypertextes, similarité, appartenance à un même cluster) qui existent entre les résultats (des techniques de visualisation de graphe sont utilisées).

Une revue des méthodes de visualisation ainsi qu'une proposition d'une méthode d'évaluation de ces interfaces est donnée dans [Nicolas Bonnel, 2006].

Dans cette thèse, nous n'abordons pas la question de l'interaction utilisateur-système. Nous nous intéressons plutôt à la partie modèle de RI. Nous cherchons à construire une représentation des documents qui permette de prendre en compte les différents types de liens qui peuvent exister entre eux (dimension intertextuelle). Nous cherchons aussi à définir des méthodes d'accès par navigation ou par interrogation en définissant des modèles de requêtes adaptés au modèle de représentation des documents.

3.3 Recherche d'information sémantique

Dans la section précédente, nous avons présenté les principales notions d'un système de RI qui se base sur des mots-clés pour représenter l'information contenue dans les textes. Cette représentation ne prend pas en compte les liens sémantiques qui peuvent exister entre les mots ni le contenu sémantique des documents. Afin d'améliorer la qualité des résultats de la RI classique pour répondre au mieux au besoin en information de l'utilisateur, plusieurs travaux ont proposé d'introduire l'information sémantique dans le processus de RI. La RI sémantique a pour objectif de mieux répondre aux besoins en information en prenant en compte le sens des mots aussi bien du côté de la requête utilisateur que celui des documents des corpus. Elle vise à exploiter des ressources sémantiques externes (thésaurus, ontologies, etc.) pour définir le sens de ces descripteurs en annotant le contenu avec des concepts de la ressource sémantique.

Avec l'essor du web sémantique [Berners-Lee et al., 2001], les ressources sémantiques, notamment les ontologies, sont devenues de plus en plus disponibles. Elles sont utilisées dans la couche sémantique pour le raisonnement et pour l'interrogation. Basés sur la logique, de nouvelles techniques et modèles de représentation des données et de RI ont ainsi vu le jour avec le web sémantique.

Dans la suite, la section 3.3.1 donne la définition de l'annotation à base de concepts, la section 3.3.2 décrit les systèmes de RI à base de connaissances (modèle numérique de RI) et la section 3.3.3 décrit le modèle logique de RI dans lequel s'inscrit le travail de cette thèse.

3.3.1 Annotation sémantique

L'annotation sémantique vise à décrire des documents en ajoutant une couche de connaissances liée à ces documents. L'objectif est de rendre l'information textuelle plus compréhensible en utilisant les concepts du domaine dont parle le texte [Haav and Lubi, 2001]. L'annotation sémantique peut être créée manuellement ou de manière automatique. Dans ces deux cas, l'annotateur ou l'outil d'annotation doivent spécifier pour chaque annotation le concept auquel elle se réfère dans une ressource sémantique identifiée (une ontologie ou un réseau sémantique de concepts).

Selon [Desmontiles and Jacquin, 2002], les annotations sémantiques sont des annotations opérationnelles, car elles sont destinées à être traitées par des machines. Les outils sont généralement des éditeurs d'ontologies permettant de choisir une ontologie, les concepts représentant le document et les instances des concepts présents dans le document. Ces annotations (concepts et instances) servent de support d'indexation pour être exploitées par un moteur de recherche.

Les approches d'annotation sémantique se caractérisent par le type de ressource (structure, semi-structuré, etc.), la technique utilisée (apprentissage automatique, patrons, etc.) et le mode d'annotation (manuelle, automatique, etc.).

L'annotation sémantique présente plusieurs avantages par rapport à une recherche d'information classique :

- enrichir les représentations des requêtes et des documents avec des concepts de la ressource sémantique ;
- enrichir la représentation des requêtes par reformulation et raffinement utilisant les concepts de la ressource ;
- avoir un moyen de représenter les documents et les requêtes dans un modèle de référence.

3.3.2 Modèles de RI numériques et à base de connaissances

Les modèles de RI à base de connaissances sont les modèles qui exploitent explicitement les ressources externes afin de construire une représentation plus précise des documents et des requêtes (*knowledge-based indexing*), ou de construire, pour un système, un jugement de pertinence qui se rapproche le plus de celui d'un être humain (*knowledge-based matching*). Les connaissances sont organisées sous forme de concepts dans des ressources externes, par exemple UMLS³⁷, WordNet³⁸, DBpedia³⁹, etc.

Les documents une fois annotés avec des concepts d'ontologie peuvent être interrogés par mots clés (comme pour la RI classique) qui peuvent être enrichis (par raffinement ou expansion de requête) par les concepts de l'ontologie. Les documents sont ensuite recherchés sémantiquement moyennant des fonctions de similarité sémantique qui évaluent la similitude entre les concepts des documents et ceux de la requête utilisateur. Ces fonctions de similarité enrichissent celles de la RI classique et améliorent ses résultats avec l'ajout de la dimension sémantique. Différentes mesures de similarité conceptuelle ainsi que des techniques de pondération ont été proposées pour estimer la ressemblance entre deux concepts de l'ontologie (pour la reformulation ou l'expansion de requête par exemple), ainsi que des fonctions de similarité sémantique entre requête et document.

Dans ce qui suit nous montrons l'intérêt de l'utilisation de concepts, décrivons les modèles de RI basés sur les concepts avec les définitions de quelques mesures de similarité conceptuelle ainsi que les modèles logiques de RI dans lesquelles s'inscrit le travail de cette thèse.

Les concepts et leurs utilisations en RI

Un concept est défini, d'un point de vue philosophique, comme l'unité de base de la pensée humaine. L'utilisation des concepts dans la RI à la place ou en plus des mots-clés est motivée par plusieurs raisons et présente plusieurs avantages. D'abord, des ressources de connaissances riches et de grandes tailles, qui sont considérées comme les principaux conteneurs de concepts, sont maintenant disponibles (par exemple UMLS, WordNet, etc.). Dans un contexte multilingue, l'utilisation de concepts facilite certaines tâches [Chevallet et al., 2007], comme le fait de se passer

37. Unified Medical Language System (<http://www.nlm.nih.gov/research/umls/>).

38. <http://wordnet.princeton.edu/>.

39. <http://dbpedia.org>.

de la traduction étant donné qu'un concept est censé être indépendant de la langue (« voiture » et « car » correspondent tous les deux à un même concept dans une ressource sémantique). Les concepts contribuent également à résoudre certains problèmes de RI bien connus comme par exemple la polysémie ou le "*term-mismatch*" [Crestani, 2000] : ce dernier se produit lorsque deux termes sont différents mais expriment la même chose ; il est résolu par l'utilisation de concepts au lieu de termes puisqu'un concept est censé englober tous les termes ayant le même sens dans un contexte donné. Des applications avancées de RI sémantique [Ren and Bracewell, 2009] comme dans le web sémantique peuvent faire appel à des structures de connaissances et des raisonnements plus sophistiqués.

Modèles de RI à base de concepts et mesures conceptuelles

Les modèles de RI à base de concepts se divisent en deux grandes familles selon la façon d'utiliser (intégrer) les ressources sémantiques externes (les concepts et leurs relations) dans le processus de RI :

- Utilisation partielle qui consiste à indexer les documents et les requêtes avec les ressources externes et utiliser par la suite un modèle classique pour la recherche [Vallet et al., 2005] et pour l'expansion de requêtes et/ou des documents avec de nouveaux termes (pour résoudre par exemple le problème de *term-mismatch*) [Voorhees, 1994].
- Utilisation globale : qui consiste à intégrer les ressources externes à la fois dans l'étape d'indexation et de recherche. Il s'agit de définir des structures de documents et de requêtes qui s'adaptent au modèle à base de concepts ainsi qu'une fonction d'appariement compatible avec ses structures [Baziz et al., 2005].

Ces systèmes utilisent des mesures de similarité conceptuelle (dites aussi mesures de proximité sémantique) pour mesurer un degré d'adéquation entre une requête et un document. Par exemple la mesure de Rada [Rada et al., 1989] utilise la distance entre deux concepts c_1 et c_2 (nombre d'arcs minimum à parcourir pour aller de c_1 à c_2) pour calculer la similarité sémantique entre eux :

$$Sim_{Rada}(c_1, c_2) = \frac{1}{1 + dist_{edge}(c_1, c_2)}$$

avec :

$dist_{edge}(c_1, c_2)$ est la longueur du plus court chemin entre deux concepts c_1 et c_2 .

Les similarités conceptuelles peuvent être pondérées suivant l'importance des concepts et des instances en calculant leurs poids dans la représentation d'un texte donné. Par exemple, dans le système proposé par [Vallet et al., 2005], aux instances (qui annotent les documents) sont associés des poids qui reflètent l'importance de l'instance dans la construction du sens du document. Le poids est calculé par une adaptation de l'algorithme *TF - IDF*. La mesure proposée calcule le poids $w_{i,j}$ d'une instance I_i dans un document D_j comme suit :

$$w_{i,j} = \frac{freq_{i,j}}{max_k freq_{k,j}} \times \log \frac{N}{n_i}$$

avec :

$freq_{i,j}$	le nombre d'occurrences de I_i dans D_j (nombre de fois où un label de l'instance apparaît dans le texte),
$maxfreq_{k,j}$	la fréquence de l'instance la plus répétée dans D_j ,
n_i	le nombre de documents annotés avec I_i ,
N	le nombre total de documents dans l'espace de recherche.

3.3.3 Modèles logiques de RI

Les logiques formelles ont été utilisées efficacement dans la RI du fait qu'elles sont bien adaptées pour la représentation des connaissances [Baader et al., 2003] et pour la construction de modèles de RI capables d'intégrer formellement les ressources de connaissances dans le processus de recherche. Un modèle logique de RI est un formalisme qui met toutes les notions de RI (documents, requêtes et décision de recherche) dans un cadre logique. Plusieurs modèles logiques de RI ont été proposés dans la littérature. Ils utilisent différents types de logique.

Dans [Losada and Barreiro, 2001], le modèle de RI repose sur la logique propositionnelle (PL). Chaque terme d'indexation est une proposition atomique qui peut être vraie ou fausse pour un document ou une requête donnés. Un document d ou une requête q est une séquence logique formée en utilisant les termes d'indexation. La décision de recherche est une conséquence logique ou implication : d est pertinent pour q si et seulement si $d \models q$.

La logique de description (DL) est une logique plus expressive que la logique propositionnelle mais qui possède un mécanisme de raisonnement plus efficace que le logique du premier ordre (FL). Elle utilise trois éléments de base pour représenter les connaissances :

- individus : pour représenter des objets concrets du monde (Ex. Alice) ;
- concepts : pour définir des classes d'objets (Ex. Personne) ;
- rôles : pour décrire les rôles des objets ou des classes dans les relations.

En plus de la RI, la DL est utilisée avec succès dans une discipline très proche, le web sémantique⁴⁰. La DL constitue la base de langages d'ontologies sur le web [Baader, 2009], comme OWL (Web Ontology Language OWL) et RDFS (Resource Description Framework Schema). Le contenu des documents et des requêtes est transformé en graphes RDF (Resource Description Framework) qui relie les ressources sémantiques aux contenus des documents. Un langage artificiel (par exemple SPARQL) est utilisé pour établir la correspondance entre les graphes RDF des requêtes et des documents. Ce langage est plus expressif qu'un ensemble de mots-clés et permet de poser des requêtes rendant compte des entités et de leurs relations.

Dans ce modèle de RI, l'appariement entre les documents et les requêtes est principalement binaire (une correspondance existe ou non), ce qui est en adéquation avec les besoins dans le domaine juridique (formulés dans le chapitre 2). En effet, les portails existant dans le domaine juridique proposent des fonctionnalités logiques de RI adaptées aux besoins d'exhaustivité des résultats exprimés par les experts du domaine.

En relation étroite avec les logiques formelles, la théorie des treillis a été utilisée comme base pour des modèles de RI et ont prouvé leur intérêt dans plusieurs domaines d'application. Dans ces modèles, l'implication logique devient une relation d'ordre partiel. Une des premières études exploitant la structure algébrique des treillis dans la RI est présentée dans [Mooers, 1958]. Ce travail a été repris par [Priss, 2000] avec l'AFC (Analyse Formelle de Concepts). Le processus de recherche dans ces modèles se base principalement sur la recherche booléenne classique.

Une variante logique de l'AFC, l'Analyse Logique de Concepts (LCA), a été proposée par [Ferré, 2007]. Ce formalisme logique a été utilisé dans un système spécifique pour la RI dans des bases d'images ou pour la navigation dans des graphes d'objets [Ferré, 2010].

40. Détails dans le chapitre 4.

3.4 RI et Analyse de liens

Dans les modèles de RI que nous avons présentés dans les sections précédentes, les documents sont traités indépendamment les uns des autres au moment de la recherche bien qu'ils forment souvent un réseau fortement interconnecté, notamment dans le domaine juridique. L'étude des relations entre objets a fait l'objet de plusieurs travaux qui visent principalement à analyser des structures de graphes d'objets indépendamment de leur contenu. Dans le domaine de la RI, l'analyse de liens a été principalement utilisée pour le tri des résultats comme dans le cas de Page Rank [Page et al., 1999].

Les différentes méthodes prenant en compte des liens entre documents dans les systèmes de RI sont exposées dans ce qui suit. Le Page Rank et les Graphes de citations sont étudiés comme les principales approches qui ont traité l'intertextualité dans une collection de documents. L'analyse socio-sémantique est par la suite présentée comme une nouvelle technique qui vise à combiner les liens avec le contenu pour l'étude des structures des graphes, mais pas à des fins de RI (interrogation avec une requête utilisateur sur les liens entre les documents) sauf pour le cas de l'algorithme Graph Search qui se limite à une RI dans les pages Facebook (présenté plus tard dans ce chapitre).

3.4.1 Intertextualité dans les systèmes de RI existants

Supposons que nous ayons une requête interrogeant sur les liens intertextuels entre documents de la forme « quels sont les documents (d') ayant un type de lien (l) avec un document (d) qui parle d'un sujet donné (s) ? » et regardons comment les systèmes de RI existants permettent de traiter une telle requête.

- Les systèmes de RI généralistes comme Google proposent une exploitation triviale de l'intertextualité. La requête est traitée en deux étapes : une requête classique sur le contenu (s) renvoie le document (d) et l'utilisateur peut alors naviguer dans les hyperliens en fonction du type de lien (l) pour trouver l'ensemble des réponses (d'). Cette catégorie de systèmes ne permettent pas le traitement de requêtes intertextuelles.
- Dans la seconde catégorie, nous classons tous les systèmes qui traitent des requêtes relationnelles *via* des attributs dans la requête, tels que des bases de données natives XML (interrogés avec XPath, XQuery) et les graphes RDF (interrogés avec SPARQL). La requête est traitée dans une première étape comme une requête booléenne sur le contenu (s) pour trouver l'ensemble des documents d, puis une étape de filtrage est effectuée en fonction des éléments XML spécifiés dans la requête (pour les bases de données natives XML), ou l'ensemble des contraintes (dans le cas de requêtes SPARQL).
- Une troisième catégorie pourrait être constituée par les systèmes relationnels tels que les bases de données relationnelles et l'Analyse Relationnelle de Concepts (ARC)⁴¹ appliqués à des objets documentaires. Les deux types de systèmes permettent de coder les relations entre les documents au niveau du modèle mais aussi au niveau de la formulation de requêtes. Étant donné que l'AFC a été appliquée pour des objets documentaires, nous considérons intéressant d'investiguer l'application de l'ARC pour modéliser des collections documentaires. L'originalité de cette approche est que la collection de documents est structurée avant d'être interrogée. Un ensemble de structures conceptuelles (appelé une Famille de Treillis Relationnels) est construit sur la base du contenu sémantique des documents et des liens qu'ils entretiennent entre eux. La requête est exécutée sur ces structures relationnelles

41. Analyse Relationnelle de Concepts (ARC) : extension relationnelle de l'AFC.

pour trouver des réponses pertinentes. L'avantage de cette approche est de permettre la navigation dans les graphes créés pour spécialiser ou généraliser la requête si aucune réponse exacte n'est trouvée⁴².

3.4.2 Analyse de graphes de citation

L'analyse des liens pour la recherche sur le web trouve ses antécédents dans le domaine de l'analyse des citations, qui est lié au domaine de la bibliométrie. Ces disciplines visent à quantifier l'influence des articles scientifiques en analysant le modèle de citations parmi eux. Elles ont inspiré l'analyse de la notoriété des pages sur le web.

Les graphes de citation possèdent plusieurs caractéristiques et leur étude (dynamique, topologie, patrons d'interaction, etc.) est également d'intérêt pour d'autres domaines tels que la physique statistique, la biologie, les mathématiques appliquées. En informatique, plusieurs types de réseaux tirent profit des études faites dans ce domaine [Yan et al., 2011, Pivovarov and Trunov, 2011, Andrews and Fox, 2007, Newman, 2004], comme les réseaux sociaux, les réseaux de neurones ou les graphes de terrains (graphes de grande taille) par exemple pour la création d'une nouvelle thématique de recherche, la prédiction de nouveaux liens dans un graphe, création de nouvelles communautés dans un réseau social, etc.

L'analyse des graphes de citations considère les noeuds comme des objets sans se soucier du contenu sémantique des documents (les articles scientifiques). De plus, l'analyse est principalement faite pour étudier la topologie des graphes et pas interroger la collection.

3.4.3 Analyse des liens hypertextes (algorithmes Page Rank et HITS)

Le web composé de pages HTML statiques avec des hyperliens entre eux est vu comme un graphe orienté dans lequel chaque page web est un noeud et chaque lien hypertexte une arête. L'analyse des liens hypertextes et la structure du graphe du web a joué un rôle dans le développement de la RI sur le web. Les liens hypertextes sont principalement utilisées comme indicateur de notoriété et pour le classement des résultats de recherche. La notoriété de liens est un facteur important pris en compte par les moteurs de recherche pour le calcul de scores de pages web pour une requête donnée [Manning et al., 2008].

Une première technique pour l'analyse des liens attribue à chaque noeud du graphe un score numérique entre 0 et 1, appelé son PageRank [Page et al., 1999]. Le PageRank d'un noeud dépend de la structure des liens dans le graphe. Étant donné une requête, un moteur de recherche calcule un score pour chaque page web qui combine des fonctionnalités telles que la similarité cosinus et la proximité de termes en plus du score PageRank. Ce score composite est utilisé pour fournir un classement de la liste des résultats de la requête.

La deuxième technique attribue à chaque page web, pour une requête donnée, deux scores : score de pivot et score d'autorité. Pour toute requête, l'algorithme HITS [Kleinberg, 1999] calcule deux listes de classement des résultats plutôt qu'une. Le classement d'une liste est induit par le score de pivot et l'autre liste selon le score d'autorité. Il commence par trouver l'ensemble des pages pertinentes par rapport aux termes de la requête puis analyse la structure des liens du sous graphe du web pour calculer les score d'autorité et de pivot. La différence entre HITS et PageRank est que le score de notoriété calculé par HITS dépend de la requête.

Dans ces techniques, la prise en compte des liens influe sur l'ordre de présentation des résultats. Les liens ne peuvent pas être rentrés au moment de l'interrogation dans la requête et ne

⁴². La modélisation de collections documentaires et l'interrogation des structures des treillis relationnels sont décrits dans le chapitre 6.

sont pas pris en compte dans le processus de recherche. Ils peuvent être pris en compte une fois la liste des résultats affichée, en navigant dans les liens hypertextes entre les documents retournés. Cette navigation étant la façon la plus triviale pour la prise en compte des liens intertextuels, elle est disponible dans plusieurs moteurs de recherche généralistes sur le web (comme Google) ou portails spécialisés (comme Legifrance pour le domaine juridique).

3.4.4 Analyse socio-sémantique

L'analyse socio-sémantique [Cointet and Roth, 2009] est une nouvelle approche qui combine l'analyse des liens avec l'analyse du contenu sémantique. Des travaux ont essayé de combiner les propriétés des graphes avec les propriétés des documents [Dang and Viennet, 2012] mais toujours dans le but d'étudier la topologie des graphes (dans les réseaux sociaux, etc).

Récemment, Facebook a lancé une nouvelle fonctionnalité de recherche (qui est intégrée dans ses pages anglophones, à ce jour). Cette fonctionnalité permet de poser des requêtes qui portent directement sur les liens qui peuvent exister entre les pages. Elle se base sur un algorithme nommé *Graph Search* et vise à rendre une liste de résultats (nom de villes, photos, etc.) aux utilisateurs au lieu d'une liste de liens hypertextes qui peuvent contenir les résultats⁴³. Graph Search prend en entrée une requête utilisateur en langage naturel, et au-delà des mots-clés, la requête peut utiliser les liens du graphe Facebook (liens d'amitié, lien "J'aime", etc.). L'algorithme effectue la recherche dans les pages qui contiennent les mots clés de la requête, les types des pages (films, personnes, villes, etc.) ainsi que les liens indiqués dans la requête (visiter, aimer, etc.). Les résultats retournés se présentent sous forme d'un ensemble de pages ou d'un ensemble de photos selon ce sur quoi porte la requête. Graph Search peut traiter des requêtes du type⁴⁴ :

- restaurants londoniens où mes amis sont allés ;
- musique que mes amis aiment ;
- villes que ma famille a visitées ;
- photos de mes amis à New York.

Le travail de cette thèse s'intègre dans cette perspective. Nous proposons de combiner le contenu documentaire avec les liens intertextuels dans un modèle de recherche qui retourne des graphes à des requêtes complexes qui portent aussi bien sur le contenu que sur les liens intertextuels. Nous proposons aussi un couplage avec la RI sémantique pour enrichir le contenu des documents. Notre objectif est donc de pouvoir modéliser et interroger des données complexes caractérisées par la richesse de leur contenu sémantique et par la structure du graphe qu'il forment.

3.5 Conclusion

Cette étude de l'état d'art montre que :

- les approches et techniques actuelles de recherche d'information sémantique ne prennent pas en compte l'intertextualité entre les documents pourtant très importante pour retourner des résultats pertinents et complets ;
- les systèmes de RI basés sur des approches qui tiennent compte de la dimension intertextuelle n'intègrent pas les liens dès le début du processus dans l'annotation et l'indexation des documents.

43. Facebook Announces Its Third Pillar "Graph Search" That Gives You Answers, Not Links Like Google. <http://techcrunch.com/2013/01/15/facebook-announces-its-third-pillar-graph-search/> .

44. <https://www.facebook.com/about/graphsearch> .

Notre travail s'intègre dans la cadre de la RI sémantique sur des données complexes représentées sous forme de graphes attribués. Nous cherchons à combler le manque identifié dans les systèmes de RI existants en intégrant les relations intertextuelles dès le début du processus de RI, dans la modélisation d'une collection documentaire. Nous définissons ensuite sur ce modèle des outils d'exploitation et d'accès (interrogation, navigation) pour répondre à des requêtes utilisateurs de plus en plus complexes dans un monde de données inter-reliées.

Nous nous orientons vers une approche qui s'apparente aux approches socio-sémantiques parce qu'elle intègre les deux dimensions sémantique et intertextuelle. Nous nous focalisons sur les approches logiques que nous décrivons dans le chapitre suivant.

Chapitre 4

Méthodes pour la modélisation et l'interrogation de données complexes

Sommaire

4.1	Introduction	43
4.2	AFC et ARC : fondements théoriques	45
4.2.1	Notions de base de la théorie des treillis	45
4.2.2	L'Analyse Formelle de Concepts	46
4.2.3	L'Analyse Relationnelle de Concepts	52
4.3	Applications de l'AFC et ARC	60
4.4	Web sémantique et web de données	62
4.4.1	Les technologies du web sémantique	63
4.4.2	Le web de données et les données liées sur le web	69
4.4.3	Les ontologies	70
4.5	Application à l'analyse documentaire dans le web sémantique	73
4.5.1	Vocabulaires conceptuels et annotation sémantique	73
4.5.2	Ontologies documentaires	74
4.6	Synthèse	75

4.1 Introduction

Pour résoudre la problématique de notre thèse, nous partons sur des approches logiques qui ont travaillé sur des objets documentaires et qui permettraient de rendre compte des deux dimensions intertextuelle et sémantique d'une collection documentaire. La première repose sur l'Analyse Formelle et Relationnelle de Concepts (AFC, ARC), la deuxième sur les techniques du web sémantique pour le traitement de données liées (RDF, OWL et SPARQL).

L'Analyse Formelle de Concepts (AFC), appelée aussi Analyse de Concepts Formels, est un formalisme mathématique basé sur la théorie des ensembles ordonnés (ou la théorie des treillis), qui offre un cadre d'application de ces théories à des problématiques du monde réel telles que l'analyse de données, la découverte et la structuration de connaissances. L'AFC, en tant que méthode d'analyse de données, permet de générer et de représenter graphiquement des regroupements à partir d'un ensemble d'objets décrits par leurs attributs, en s'appuyant sur la notion de partage d'attributs entre objets. Les données sont structurées dans des unités appelées

des concepts formels qui sont partiellement ordonnés et forment une hiérarchie de concepts, appelée le treillis de concepts. L'Analyse Relationnelle de Concepts (ARC) est une extension de l'AFC qui a été définie pour prendre en compte les relations qui peuvent exister entre les objets⁴⁵. Elle permet la construction de concepts relationnels sur plusieurs ensembles d'objets décrits par des attributs et des relations. L'ARC est une version itérative de l'AFC selon laquelle les objets sont structurés non seulement par rapport à leurs attributs communs mais aussi par rapport aux relations qui existent entre eux. Ces relations sont représentées par des tableaux qui lient les données en entrée de l'AFC.

La manipulation de liens entre les objets et plus généralement de données liées a fait l'objet d'une attention particulière ces dernières années parmi les chercheurs du web sémantique. Le mouvement d'ouverture de données par plusieurs gouvernements, institutions et entreprises a accéléré le développement de nouvelles techniques pour la manipulation et l'analyse de ces données, et a donné naissance au web de données. Des efforts sont faits pour rendre ces données compatibles avec les standards et normes définis dans le web sémantique (XML, RDF, OWL, SPARQL, etc.) et définir des modèles sémantiques (ontologies) pour différents domaines. RDF est un modèle de données qui se présente comme un graphe orienté étiqueté, qui se base sur la notion de triplets (sujet, prédicat, objet) et représente la composante principale du web sémantique. OWL est un langage formel d'ontologie utilisé pour modéliser les vocabulaires pour le web sémantique et SPARQL est le langage de requêtes et de mise à jour pour le web de données, utilisé pour interroger des bases de connaissances RDF. Ces efforts ont pour but d'assurer l'interopérabilité des données et de faciliter leur accès et leur gestion par les utilisateurs en ajoutant une couche sémantique aux données et en les liant entre elles.

La première partie de ce chapitre décrit les fondements théoriques de l'approche conceptuelle et passe en revue les applications de ce formalisme pour la recherche d'information. La deuxième partie donne un aperçu des notions de base et des principales fonctionnalités du web sémantique et du web de données. Nous mettons particulièrement l'accent sur la notion d'ontologie et son utilisation pour la modélisation de collections documentaires. Les données sur lesquelles nous illustrons les notions décrites dans ce chapitre représentent un ensemble d'utilisateurs d'un réseau social et les films les mieux notés parmi ces utilisateurs, auxquels ils ont affecté la mention « J'aime ». Les personnes sont décrites par des propriétés telles que l'âge et le lieu d'habitation ; les films sont décrits par leurs catégories.

La suite du chapitre est organisée comme suit. Dans les sections 4.2.1, 4.2.2, 4.2.3 et 4.3 nous dressons un état de l'art qui couvre à la fois les définitions mathématiques relatives à la théorie des treillis, les notions de base de l'AFC et de l'ARC ainsi que les applications de ces formalismes pour la recherche d'information. La description des notions de base des standards et langages du web sémantique et de leurs applications pour la modélisation documentaire fera l'objet des sections 4.4 et 4.5. Les définitions et notions de base de l'AFC/ARC et des langages du web sémantique (RDF/OWL) sont illustrées sur des exemples.

45. Une extension relationnelle de l'AFC pour la prise en compte des relations entre attributs dans un contexte formel est définie dans [Carpineto and Romano, 2004, Priss, 2000]. Cette extension permet d'intégrer des relations sémantiques explicites, extraites à partir des taxonomies, thesaurus ou ontologies, dans les structures conceptuelles de l'AFC. Nous ne nous intéressons pas à cet aspect dans ce travail.

4.2 AFC et ARC : fondements théoriques

4.2.1 Notions de base de la théorie des treillis

Dans cette section nous rappelons les notions relatives à la théorie des treillis [Birkhoff, 1967, Davey and Priestley, 2002] qui servent à la formalisation de notre approche.

Ensemble ordonné

Définition 1 (Relation d'ordre (partiel)) Soient E un ensemble et R est une relation binaire sur E . R est dite **relation d'ordre partiel** (ou simplement relation d'ordre) sur E si elle vérifie les conditions suivantes pour tout $a, b, c \in E$:

1. $(a, a) \in R$ (R est réflexive)
2. si $(a, b) \in R$ et $a \neq b$ alors $(b, a) \notin R$ (R est antisymétrique)
3. si $(a, b) \in R$ et $(b, c) \in R$ alors $(a, c) \in R$ (R est transitive)

On note souvent une relation d'ordre R par " \leq " (R^{-1} est notée par " \geq ") et on dit que " a est plus petit que b " lorsque $a \leq b$.

Définition 2 (Ensemble ordonné) Un couple (E, \leq) – où E est un ensemble et " \leq " est une relation d'ordre sur E – est un **ensemble partiellement ordonné** (ou simplement ensemble ordonné) .

Dans un ensemble ordonné (E, \leq) , deux éléments a et b de E sont dits **comparables** lorsque $a \leq b$ ou $b \leq a$. Autrement ils sont dits **incomparables**. Pour deux éléments comparables et différents, $a \leq b$ et $a \neq b$, on note $a < b$.

Définition 3 (Successeur, prédécesseur, couverture) Soient (E, \leq) un ensemble ordonné et $a, b \in E$. On dit que b est **successeur** de a lorsque $a < b$ et s'il n'existe aucun élément $c \in E$ tel que $a < c < b$ ($a \neq c$ et $b \neq c$). Dans ce cas, a est dit **prédécesseur** de b et on note $a \prec b$. Lorsque a est un **prédécesseur** de b on dit que a **couvre** b (et que b est couvert par a). La **couverture** de a est formée par l'ensemble de ses successeurs.

Tout ensemble ordonné, (E, \leq) , peut être représenté graphiquement par un diagramme appelé "**diagramme de Hasse**" (ou diagramme de couverture) obtenu comme suit :

1. tout élément de E est représenté par un petit cercle dans le plan ;
2. si $a, b \in E$ et $a \prec b$ alors le cercle correspondant à b doit être au-dessus de celui correspondant à a et les deux cercles sont reliés par un segment.

La relation d'ordre se lit à partir de ce diagramme comme suit : $a < b$ si et seulement s'il existe un chemin ascendant qui relie le cercle correspondant à a à celui de b .

Treillis

Définition 4 (Majorant, minorant, supremum, infimum) Soient (E, \leq) un ensemble ordonné et S un sous-ensemble de E . Un élément $x \in E$ est dit **majorant** de S lorsque $x \geq s \forall s \in S$. De façon duale, $x \in E$ est dit **minorant** de S lorsque $x \leq s \forall s \in S$.

Le plus petit majorant (respectivement plus grand minorant) de S , s'il existe, est appelé **supremum** ou borne supérieure (respectivement **infimum** ou borne inférieure) de S . Il est noté $\bigvee S$ (respectivement $\bigwedge S$). Dans le cas où $S = \{s, t\}$, $\bigvee S$ et $\bigwedge S$ sont aussi notés par $s \vee t$ et $s \wedge t$ respectivement.

Lorsque le supremum et l'infimum existent dans un ensemble ordonné, ils sont uniques.

Définition 5 (Treillis, treillis complet) Un *treillis* est un ensemble partiellement ordonné (E, \leq) tel que $a \vee b$ et $a \wedge b$ existent pour tout couple d'éléments $a, b \in E$. Un treillis est dit **complet** si $\bigvee S$ et $\bigwedge S$ existent pour tout sous-ensemble S de E . En particulier, un treillis complet admet un élément maximal (top) noté par \top et un élément minimal (bottom) noté par \perp .

Fermeture et connexion de Galois

Définition 6 (Fermeture) On appelle *opérateur de fermeture* sur un ensemble ordonné, (E, \leq) , toute application $\varphi : E \rightarrow E$ qui vérifie les propriétés suivantes pour tout $a, b \in E$:

- $a \leq \varphi(a)$ (φ est extensive),
- si $a \leq b$ alors $\varphi(a) \leq \varphi(b)$ (φ est monotone croissante),
- $\varphi(a) = \varphi(\varphi(a))$ (φ est idempotente).

Définition 7 (Fermé) Étant donné un opérateur de fermeture φ sur un ensemble ordonné (E, \leq) , un élément $a \in E$ est dit **fermé** pour φ si et seulement si $a = \varphi(a)$.

Définition 8 (Connexion de Galois) Soient $\varphi : E \rightarrow F$ et $\psi : F \rightarrow E$ deux applications entre deux ensembles ordonnés (E, \leq_E) et (F, \leq_F) . φ et ψ forment une **connexion de Galois** entre (E, \leq_E) et (F, \leq_F) si la condition suivante est satisfaite :

$$\forall a \in E, \forall b \in F, \varphi(a) \leq_F b \Leftrightarrow \psi(b) \leq_E a$$

4.2.2 L'Analyse Formelle de Concepts

L'Analyse Formelle de Concepts [Wille, 1982, Ganter and Wille, 1999a], présentée comme un domaine de la mathématique appliquée, repose sur la théorie des treillis et étudie les structures partiellement ordonnées, connues sous le nom de *treillis de Galois* [Barbut and Monjardet, 1970] ou *treillis de concepts*.

C'est une méthode de classification conceptuelle qui construit à partir d'un jeu de données une hiérarchie d'abstractions. Ces abstractions sont représentées par des concepts et chaque concept représente un ensemble maximal d'objets (un regroupement d'individus) ayant en commun un ensemble maximal d'attributs (les propriétés communes de ces individus). L'AFC représente les données sous la forme d'un tableau binaire, appelé *contexte formel*, contenant un ensemble d'individus (*objets*), un ensemble de propriétés (*attributs formels*) et exprimant la relation d'incidence (*objets* \times *attributs*) entre ces individus et ces propriétés. Les contextes formels, qui représentent le point de départ de l'AFC, sont définis dans la section suivante.

Contexte formel

Définition 9 (Contexte formel) Un *contexte formel* est un triplet $\mathcal{K} = (O, A, I)$ où O est un ensemble d'objets, A est un ensemble d'attributs et I est une relation binaire entre O et A appelée relation d'incidence de \mathcal{K} et vérifiant $I \subseteq O \times A$. Un couple $(o, a) \in I$ (noté aussi oIa) signifie que l'objet $o \in O$ possède l'attribut $a \in A$.

Un contexte formel peut être représenté sous la forme d'un tableau binaire à deux dimensions où les lignes correspondent aux objets et les colonnes correspondent aux attributs. Les cases du tableau sont remplies comme suit : si l'objet o_i est en relation I avec l'attribut a_j , alors la case située à l'intersection de la ligne i et de la colonne j contient « \times » ; sinon, la case est vide.

Nous utilisons l'exemple annoncé dans l'introduction (utilisateurs d'un réseau social) pour illustrer les concepts décrits dans cette section. Les tables 4.1 et 4.2 donnent deux contextes formels \mathcal{K}_P et \mathcal{K}_F représentant respectivement les utilisateurs du réseau social (ou personnes) et les films. Dans le contexte \mathcal{K}_P , l'utilisateur **Peter** possède les attributs : adolescent (<18), vit actuellement au Royaume-Uni (UK).

TABLE 4.1 – Contexte formel \mathcal{K}_P décrivant des utilisateurs d'un réseau social.

Objet \ Attribut	Âge			Pays			
	< 18	18 – 30	> 30	UE	UK	US	AU
Kate		×		×			
Peter	×				×		
Tom		×					×
Eva		×		×			
Mark		×				×	
Adam	×						×
Mary			×		×		
John			×	×			

Connexion de Galois dans un contexte formel

Définition 10 Soit $\mathcal{K} = (O, A, I)$ un contexte formel. Pour tout $X \subseteq O$ et $Y \subseteq A$, on définit :

$$X' = \{a \in A \mid \forall o \in X, oIa\}$$

$$Y' = \{o \in O \mid \forall a \in Y, oIa\}$$

Intuitivement, X' est l'ensemble des attributs communs à tous les objets de X et Y' est l'ensemble des objets possédant tous les attributs de Y . Les applications $' : \mathfrak{P}(O) \rightarrow \mathfrak{P}(A)$ et $' : \mathfrak{P}(A) \rightarrow \mathfrak{P}(O)$ sont appelées opérateurs de dérivation entre l'ensemble des objets et l'ensemble des attributs dans un contexte formel. $\mathfrak{P}(O)$ est l'ensemble des parties de O , noté aussi 2^O et $\mathfrak{P}(A)$ est l'ensemble des parties de A , noté aussi 2^A .

La composition de ces opérateurs $'' : \mathfrak{P}(O) \rightarrow \mathfrak{P}(O)$ et $'' : \mathfrak{P}(A) \rightarrow \mathfrak{P}(A)$ produit deux opérateurs de fermeture sur les deux ensembles 2^O et 2^A . Chacun induit une famille d'ensembles *fermés*. Le premier opérateur permet d'associer à un ensemble d'objets X l'ensemble maximal d'objets dans O ayant les attributs communs aux objets de X . Cet ensemble est noté par X'' . De façon duale, le second opérateur permet d'associer à un ensemble d'attributs Y l'ensemble maximal d'attributs dans A communs aux objets ayant les attributs dans Y . Cet ensemble est noté par Y'' . Les opérateurs $'' : \mathfrak{P}(O) \rightarrow \mathfrak{P}(O)$ et $'' : \mathfrak{P}(A) \rightarrow \mathfrak{P}(A)$ définissent deux fermetures respectivement sur l'ensemble des parties de O , $\mathfrak{P}(O)$, et sur l'ensemble des parties de A , $\mathfrak{P}(A)$. Les ensembles X'' et Y'' sont fermés pour ces deux opérateurs respectifs.

L'ensemble des fermés de $\mathfrak{P}(O)$ muni de l'inclusion est un treillis complet. De la même façon, l'ensemble des fermés de $\mathfrak{P}(A)$ muni de l'inclusion est un treillis complet. Les opérateurs de dérivation $' : \mathfrak{P}(O) \rightarrow \mathfrak{P}(A)$ et $' : \mathfrak{P}(A) \rightarrow \mathfrak{P}(O)$ forment une bijection entre les ensembles de fermés de $\mathfrak{P}(O)$ et $\mathfrak{P}(A)$ et définissent un isomorphisme entre les deux treillis respectifs : à chaque fermé X dans $\mathfrak{P}(O)$ correspond un unique fermé Y dans $\mathfrak{P}(A)$ et vice versa.

TABLE 4.2 – Contexte formel \mathcal{K}_F décrivant les films associés à leurs catégories.

	Catégorie				
	Animation	Politique-Historique	Science fiction	Drame	Mystère-Horreur
Star wars			×		
Gravity			×		×
Harry Potter					×
Matrix			×		
12 years a slave		×		×	
Toy story	×				
Les misérables		×		×	
Lincoln		×		×	
Titanic				×	
The princess and the frog	×				

Propriété 1 Les opérateurs de dérivation $'' : \mathfrak{P}(O) \rightarrow \mathfrak{P}(O)$ et $'' : \mathfrak{P}(A) \rightarrow \mathfrak{P}(A)$ forment une connexion de Galois entre $(\mathfrak{P}(O), \subseteq)$ et $(\mathfrak{P}(A), \subseteq)$.

Concept formel

Les couples (X, Y) d'ensembles fermés, où X représente un sous-ensemble d'objets et Y un sous-ensemble d'attributs, reliés par la connexion de Galois détaillée dans la section précédente, forment les concepts formels définis comme suit.

Définition 11 (Concept formel) Soit $\mathcal{K} = (O, A, I)$ un contexte formel. Un **concept formel** est un couple (X, Y) tel que $X \subseteq O$, $Y \subseteq A$, $X' = Y$ et $Y' = X$. X et Y sont respectivement appelées extension (extent) et intension (intent) du concept formel (X, Y) . L'ensemble des concepts formels associés au contexte formel $\mathcal{K} = (O, A, I)$ est noté par $\mathcal{C}(O, A, I)$ ou simplement $\mathcal{C}_{\mathcal{K}}$.

Schématiquement, lorsqu'un contexte formel est décrit par une table binaire, chaque concept formel (X, Y) correspond à une sous-table rectangulaire avec un ensemble de lignes X et un ensemble de colonnes Y non nécessairement contiguës. Le concept formel correspond à un rectangle maximal de la table formée par la relation binaire du contexte : tout objet de l'extension a tous les attributs de l'intension. Ces ensembles maximaux d'objets et d'attributs correspondent à des fermés dans $\mathfrak{P}(O)$ et $\mathfrak{P}(A)$ respectivement. Un sous-ensemble Y de A est l'intension d'un concept formel dans $\mathcal{C}(O, A, I)$ si et seulement si $Y'' = Y$ (Y est fermé pour $''$) et, de façon duale, un sous ensemble X de O est l'extension d'un concept formel dans $\mathcal{C}_{\mathcal{K}}$ si et seulement si $X'' = X$ (X est fermé pour $''$).

La famille $\mathcal{C}_{\mathcal{K}}$ des concepts formels de $\mathcal{K} = (O, A, I)$ est ordonnée par une relation d'ordre hiérarchique entre concepts (appelée aussi relation de subsomption) notée par " \leq " et définie

comme suit.

Définition 12 (Relation de “subsumption”) Soient $(X1, Y1)$ et $(X2, Y2)$ deux concepts formels de \mathcal{C}_K . $(X1, Y1) \leq (X2, Y2)$ si et seulement si $X1 \subseteq X2$ (ou de façon duale $X2 \subseteq X1$). $(X2, Y2)$ est dit **super-concept** de $(X1, Y1)$ et $(X1, Y1)$ est dit **sous-concept** de $(X2, Y2)$. La relation “ \leq ” est dite relation de subsumption.

Un super-concept direct (respectivement un sous-concept direct) d’un concept est aussi appelé “successeur” direct (respectivement “prédécesseur” direct).

La relation “ \leq ” s’appuie sur deux inclusions duales, entre ensembles d’objets et entre ensembles d’attributs et peut ainsi être interprétée comme une relation de généralisation/spécialisation entre les concepts formels. Un concept est plus général qu’un autre concept s’il contient plus d’objets dans son extension avec des attributs partagés par ces objets qui sont réduits. De façon duale, un concept est plus spécifique qu’un autre s’il contient moins d’objets dans son extension. Ces objets ont plus d’attributs en commun.

Treillis de concepts

Définition 13 (Treillis de concepts) La relation “ \leq ” permet d’organiser les concepts formels en un treillis complet (\mathcal{C}_K, \leq) appelé **treillis de concepts** ou encore **treillis de Galois** [Birkhoff, 1967] et noté par $\mathcal{L}(\mathcal{C}_K)$ ou \mathcal{L}_K . L’infimum et le supremum dans \mathcal{L}_K sont donnés par :

$$\bigwedge_{j \in J} (X_j, Y_j) = \left(\bigcap_{j \in J} X_j, \left(\bigcup_{j \in J} Y_j \right)'' \right)$$

$$\bigvee_{j \in J} (X_j, Y_j) = \left(\left(\bigcup_{j \in J} X_j \right)'', \bigcap_{j \in J} Y_j \right)$$

Le treillis de concepts est une représentation équivalente des données contenues dans un contexte formel qui met en avant les groupements possibles entre objets et attributs (ensemble d’objets partageant les mêmes attributs) ainsi que les relations d’inclusion entre ces groupements (entre les objets d’une part et les attributs d’autre part). La représentation graphique du treillis de concepts, sous la forme d’un diagramme de Hasse, facilite la compréhension et l’interprétation de la relation entre les objets et les attributs d’une part (au sein d’un même groupement) et entre objets ou attributs d’autre part (selon la relation d’hierarchie entre groupements). L’avantage de cette représentation est qu’à partir d’un treillis de concepts il est toujours possible de retrouver le contexte formel correspondant et inversement.

Le treillis de concepts \mathcal{L}_P correspondant au contexte formel des utilisateurs du réseau social \mathcal{K}_P de la table 4.1 est donné par la diagramme de Hasse de la figure 4.1 (visualisé grâce au logiciel Galicia⁴⁶). Dans ce treillis, le concept 8 représente dans son extension ($E = \{\text{Eva}, \text{Kate}\}$) le groupement de personnes qui possèdent en commun les propriétés de l’intension ($I = \{18 < \text{age} < 30, \text{UE}\}$). Le treillis de concepts \mathcal{L}_F correspondant au contexte formel des films \mathcal{K}_F de la table 4.2 est donné par la figure 4.2 (les titres des films et les noms des catégories sont affichés en raccourcis dans le treillis pour plus de lisibilité).

46. <http://sourceforge.net/projects/galicia/>

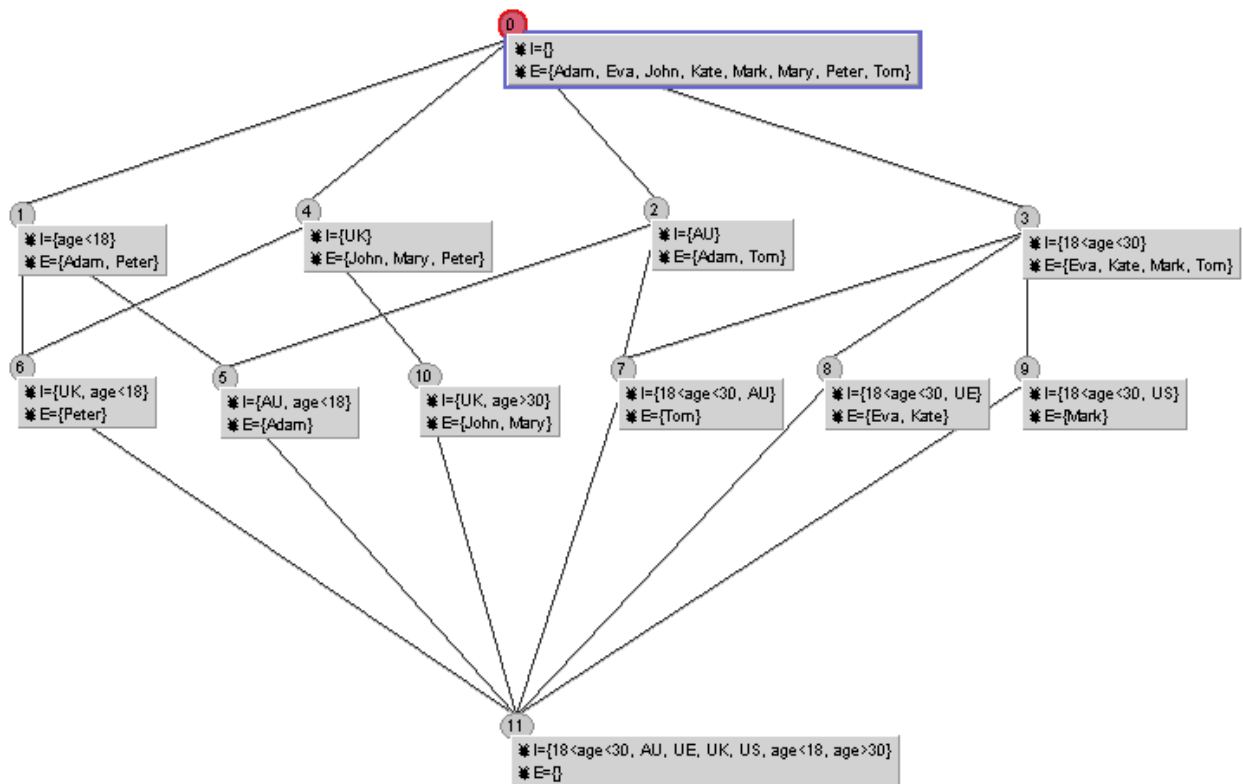


FIGURE 4.1 – Le treillis de concepts \mathcal{L}_P correspondant au contexte formel \mathcal{K}_P donné dans la table 4.1.

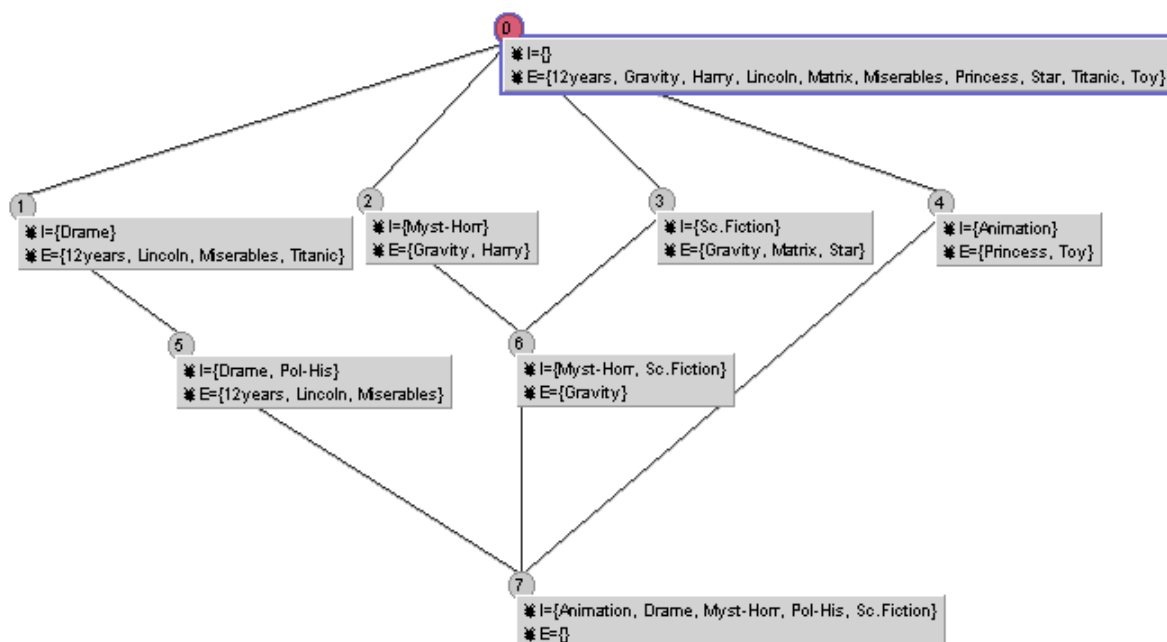


FIGURE 4.2 – Le treillis de concepts \mathcal{L}_F correspondant au contexte formel \mathcal{K}_F donné dans la table 4.2.

Construction du treillis de concepts

La construction du treillis de concepts d'une relation binaire donnée est composée de trois tâches [Guenoche and Mechelen, 1993] : la recherche des concepts (énumération des rectangles maximaux (les fermés)), la recherche de la relation d'ordre partiel entre ces rectangles (calcul de la relation de couverture), la représentation graphique du treillis (construction du diagramme de Hasse correspondant au treillis). Les deux premières tâches constituent le problème du calcul des concepts d'un treillis à partir d'un contexte formel, alors que la troisième relève de la visualisation de graphes. Ces deux problématiques sont souvent traitées indépendamment. Plusieurs travaux de recherche se sont penchés sur le problème de calcul des concepts d'un treillis de concepts à partir d'un contexte formel et ont proposé une grande variété d'algorithmes de plus en plus performants (complexité, temps de calcul, occupation mémoire, passage à l'échelle). Les principaux algorithmes ont fait l'objet d'une étude comparative détaillée dans [Kuznetsov and Obiedkov, 2002] qui a montré que les performances d'un algorithme dépendent fortement des caractéristiques du contexte formel d'entrée. Une étude plus récente comparant et analysant la complexité des algorithmes de l'AFC est donnée dans [Strok and Neznanov, 2010]. Ces algorithmes peuvent être répartis en trois grandes familles selon leurs stratégies d'acquisition de données à partir d'un contexte formel :

Les algorithmes batch prennent en entrée le contexte formel tout entier et calculent les concepts formels et l'ordre entre ces concepts simultanément ou de manière séquentielle. L'un des premiers algorithmes proposés est celui de **Chein** [Chuin, 1969], sa complexité est en $\mathcal{O}(|O|^3|A||L|)$, $|O|$ étant le nombre d'objets dans le contexte, $|A|$ le nombre d'attributs et $|L|$ le nombre de concepts formels dans le treillis obtenu. D'autres algorithmes connus dans

cette catégorie sont **NextClosure** [Ganter, 1984] ($\mathcal{O}(|O|^2|A||L|)$) et **Bordat** [Bordat, 1986] ($\mathcal{O}(|O||A|^2|L|)$).

Les algorithmes incrémentaux considèrent le contexte formel ligne par ligne (ou colonne par colonne) et construisent le treillis de concepts par ajouts successifs de ligne ou de colonne tout en conservant sa structure. Parmi les algorithmes dans cette catégorie, on peut citer celui de **Norris** [Norris, 1978] ($\mathcal{O}(|O|^2|A||L|)$) et celui de **Godin** [Godin et al., 1995a, Godin et al., 1995c].

Les algorithmes d'assemblage permettent de diviser un contexte formel en deux parties verticalement ou horizontalement puis de calculer le treillis de concepts correspondant à chaque partie et enfin d'assembler les treillis obtenus en un seul. Parmi ces algorithmes on peut citer **Divide&Conquer** [Valtchev et al., 2002], **In-Close** [Andrews, 2009], **In-Close2** [Andrews, 2011] et les algorithmes parallèles pour FCA [Kengue et al., 2005, Krajca et al., 2008].

Dans le cas des applications réelles, on estime que la complexité théorique maximale n'est pas atteinte [Carpineto and Romano, 2000]. Des optimisations ont été proposées dans la littérature pour réduire la complexité de la construction des treillis de concepts pour le traitement des applications complexes. Par exemple les *treillis Iceberg* ou *treillis de concepts fréquents* [Waiyamai and Lakhal, 2000, Stumme et al., 2002] minimisent la taille du treillis en limitant la profondeur d'exploration de l'ensemble des concepts ou les *treillis de Galois Alpha* [Ventos and Soldano, 2005] qui filtrent les objets au niveau du contexte.

4.2.3 L'Analyse Relationnelle de Concepts

L'Analyse Relationnelle de Concepts (ARC) [Rouane et al., 2007, Huchard et al., 2007, Rouane et al., 2013] est une extension relationnelle de l'AFC. Elle traite des relations entre des ensembles d'objets décrits par leurs attributs. L'ARC a été introduite pour injecter des liens inter-objets dans le processus de construction des concepts de façon à ce que les descriptions des concepts trouvés renferment une partie relationnelle inférée à partir du partage des liens.

À partir de la notion de partage de liens, les concepts formels créés par l'AFC sont enrichis par des relations vers d'autres concepts formels. L'ARC construit à partir d'un ou plusieurs contextes binaires (*objets* \times *attributs*) et d'un ensemble de relations (*objets* \times *objets*) représentées séparément par des contextes, une Famille de Contextes Relationnels (FCR). Cette famille de contextes relationnels constitue le point de départ du processus itératif de formation des structures conceptuelles correspondantes appelées Famille de Treillis Relationnels (FTR).

Modèle de données de l'ARC

Les données en entrée de l'ARC sont organisées comme une paire constituée d'un ensemble de contextes formels (*objets* \times *attributs*), $\mathbb{K} = (\mathcal{K}_i)_{i=1,\dots,n}$, et un ensemble de relations binaires (*objets* \times *objets*), $\mathbb{R} = (r_k)_{k=1,\dots,m}$, représentant les relations d'incidence entre ensembles d'objets de \mathbb{K} . Une relation $r \in \mathbb{R}$ relie deux ensembles d'objets provenant de deux contextes, à savoir, il existe $i_1, i_2 \in 1, \dots, n$ (éventuellement $i_1 = i_2$) de telle sorte que $r \subseteq O_{i_1} \times O_{i_2}$. Formellement, une FCR est définie de la manière suivante :

Définition 14 (Famille de contextes relationnels) Une FCR est une paire (\mathbb{K}, \mathbb{R}) avec :

- \mathbb{K} est un ensemble de contextes formels $\mathcal{K}_i = (O_i, A_i, I_i)$,
- \mathbb{R} est un ensemble de relations $r_k \subseteq O_i \times O_j$ où O_i et O_j sont des ensembles d'objets de certains contextes de \mathbb{K} .

Reprenons l'exemple de la section 4.2.2. Les utilisateurs du réseau social (décrits dans la table 4.1), leurs relations d'amitié (table 4.3), les films (décrits dans la table 4.2) auxquels ils ont attribué la mention "J'aime" (ou "like") (relation donnée dans la table 4.4) forment une FCR. Dans cette famille de contextes relationnels, l'utilisateur **Peter** possède les attributs : adolescent (<18), masculin (M) et vit actuellement au Royaume-Uni (UK), est ami avec **Adam** et "like" les films "Harry Potter", "Toy story" et "The princess and the frog".

TABLE 4.3 – Contexte relationnel **Amis** décrivant la relation d'amitié entre les utilisateurs du réseau social.

	Kate	Peter	Tom	Eva	Mark	Adam	Mary	John
Kate			×					
Peter						×		
Tom	×							
Eva					×			
Mark				×				
Adam		×						
Mary								×
John							×	

TABLE 4.4 – Contexte relationnel "Like" liant les utilisateurs du réseau social et les films.

	Star wars	Gravity	Harry Potter	Matrix	12 years a slave	Toy story	Les misérables	Lincoln	Titanic	The princess and the frog
Kate	×	×	×	×						
Peter			×			×				×
Tom	×	×	×	×						
Eva					×		×	×		
Mark					×			×		
Adam						×				×
Mary					×		×	×	×	
John					×			×	×	

Dans la définition 14, tous les ensembles d'objets O_i ($i \in \{1, \dots, n\}$) sont deux à deux disjoints. Les relations dans \mathbb{R} sont orientées et représentent des fonctions ensemblistes $r : O_i \rightarrow \mathfrak{P}(O_j)$. De plus, O_i (domaine de r_k) et O_j (co-domaine de r_k) sont les ensembles d'objets des contextes K_i et K_j respectivement.

Définition 15 (Domaine et co-domaine d'une relation) Soit (\mathbb{K}, \mathbb{R}) une FCR. Le domaine et le co-domaine d'une relation $r \subseteq O_i \times O_j$ sont deux applications :

- $dom : \mathbb{R} \rightarrow \mathbb{O}$ avec $dom(r) = O_i$ ssi $\forall (x, y) \in r, x \in O_i$,
- $ran : \mathbb{R} \rightarrow \mathbb{O}$ avec $ran(r) = O_j$ ssi $\forall (x, y) \in r, y \in O_j$.

où \mathbb{O} est l'ensemble de tous les ensembles d'objets dans la FCR, $\mathbb{O} = \{O | \mathcal{K} = (O, A, I) \in \mathbb{K}\}$.

Dans cette définition, la fonction r possède \mathcal{K}_i comme contexte source et \mathcal{K}_j comme contexte cible. La fonction rel permet de définir l'ensemble des relations qui ont pour source le contexte \mathcal{K}_i .

Définition 16 (Fonction de contexte $rel(\mathcal{K})$) La famille des relations qui ont pour domaine un contexte \mathcal{K} est défini par :

$$rel : \mathbb{K} \rightarrow \mathfrak{P}(\mathbb{R}), rel(\mathcal{K} = (O, A, I)) = \{r \in \mathbb{R} | dom(r) = O\}$$

.

Le scaling relationnel

Les instances d'une relation $r_k, r_k(o_i, o_j)$, avec $o_i \in O_i$ et $o_j \in O_j$, sont appelés des liens. Les liens sont traités de façon à ce qu'ils soient introduits comme des attributs binaires dans un contexte formel d'origine. Ce mécanisme s'appelle le « codage relationnel »⁴⁷.

Le codage relationnel s'appuie sur une « convention d'identification »⁴⁸ qui attribue aux éléments de l'ARC un identifiant unique tout au long du processus d'analyse. En fait, l'évolution dans l'ARC est liée à la transformation des liens en descripteurs d'objets formels : les ensembles d'attributs A_i sont enrichis avec de nouveaux éléments mais les ensembles d'objets O_i restent inchangés. Ces derniers forment ainsi une base d'identification des contextes et de leurs versions étendues tout au long du processus. De façon similaire, les concepts des différentes versions d'un contexte gardent la même extension et sont considérés comme versions subséquentes du même concept. Il leur est donc attribué le même identifiant (un numéro unique) dans tous les treillis.

L'ensemble des attributs d'un codage relationnel repose sur les noms des concepts. Étant donnée une relation r de la FCR avec $dom(r) = O_i$ et $ran(r) = O_j$, de nouveaux attributs sont ajoutés au contexte $\mathcal{K}_i = (O_i, A_i, I_i)$ via r . Le codage de \mathcal{K}_i par la relation $r \in rel(\mathcal{K}_i)$ par rapport au treillis \mathcal{L}_j implique une extension de A_i et I_i , mais garde O_i inchangé.

Ainsi, la relation r introduit des abstractions de \mathcal{K}_j dans \mathcal{K}_i . Les attributs résultants, qu'on appelle attributs relationnels, doivent porter clairement une indication de la relation dont ils sont issus. Ils sont ajoutés à A_i sous la forme $r : \mathcal{C}$. Pour qu'un objet o dans \mathcal{K}_i reçoive un attribut relationnel, des conditions sur $r(o)$, image de o par la relation r , doivent être vérifiées.

Intuitivement, le codage associe à un objet un attribut combinant une relation r avec un concept c du treillis \mathcal{L}_j à chaque fois que $r(o)$ est corrélé avec l'extension de c . Une corrélation avec peu de contraintes cherche une intersection non vide et une corrélation forte se traduit par une inclusion entre les deux ensembles. Ces deux schémas de codage relationnel sont appelés respectivement « codage large ou existentiel »⁴⁹ et « codage étroit ou universel »⁵⁰.

47. *Scaling relationnel*

48. *Naming convention*

49. *Wide scaling* ou *Existential scaling*. Nous utilisons ces deux termes dans la suite du manuscrit pour désigner la même notion.

50. *Narrow scaling* ou *Universal scaling*.

Définition 17 (Opérateur de codage existentiel) Soit une relation $r \in \text{rel}(\mathcal{K})$ et un treillis \mathcal{L}_j correspondant à $\mathcal{K}_j = (O_j, A_j, I_j)$ cible de r , l'opérateur de codage existentiel $\mathbb{S}_{(r,\exists),\mathcal{L}_j}$ fait correspondre à \mathcal{K} le contexte $\mathcal{K}^+ = (O^+, A^+, I^+)$ tel que :

- $O^+ = O$,
- $A^+ = \{\exists r : c | c \in \mathcal{L}_j\}$ où tous les $\exists r : c$ sont des attributs relationnels,
- $I^+ = \{(o, \exists r : c) | o \in O, c \in \mathcal{L}_j, r(o) \cap \text{extension}(c) \neq \emptyset\}$.

L'opérateur de codage universel diffère de l'existentiel dans le calcul de I^+ en considérant, au lieu d'une intersection non vide, que l'image de l'objet $r(o)$ doit être complètement incluse dans l'extension du concept c pour que l'objet o ait l'attribut relationnel $\forall r : c$.

Définition 18 (Opérateur de codage universel) Soit une relation $r \in \text{rel}(\mathcal{K})$ et un treillis \mathcal{L}_j correspondant à $\mathcal{K}_j = (O_j, A_j, I_j)$ cible de r , l'opérateur de codage universel $\mathbb{S}_{(r,\forall),\mathcal{L}_j}$ fait correspondre à \mathcal{K} le contexte $\mathcal{K}^+ = (O^+, A^+, I^+)$ tel que :

- $O^+ = O$,
- $A^+ = \{\forall r : c | c \in \mathcal{L}_j\}$ où tous les $\forall r : c$ sont des attributs relationnels,
- $I^+ = \{(o, \forall r : c) | o \in O, c \in \mathcal{L}_j, r(o) \subseteq \text{extension}(c) \text{ et } r(o) \neq \emptyset\}$.

Considérons l'exemple de la FCR des utilisateurs d'un réseau social. Le contexte des utilisateurs \mathcal{K}_P (table 4.1) est enrichi avec la relation "Like" (table 4.4) par rapport au treillis des films \mathcal{L}_F donné par la figure 4.2. Les treillis $\mathcal{L}_{P_F}^{\forall,+}$ et $\mathcal{L}_{P_F}^{\exists,+}$ résultants de ce processus sont donnés respectivement par la figure 4.3 et la figure 4.4.

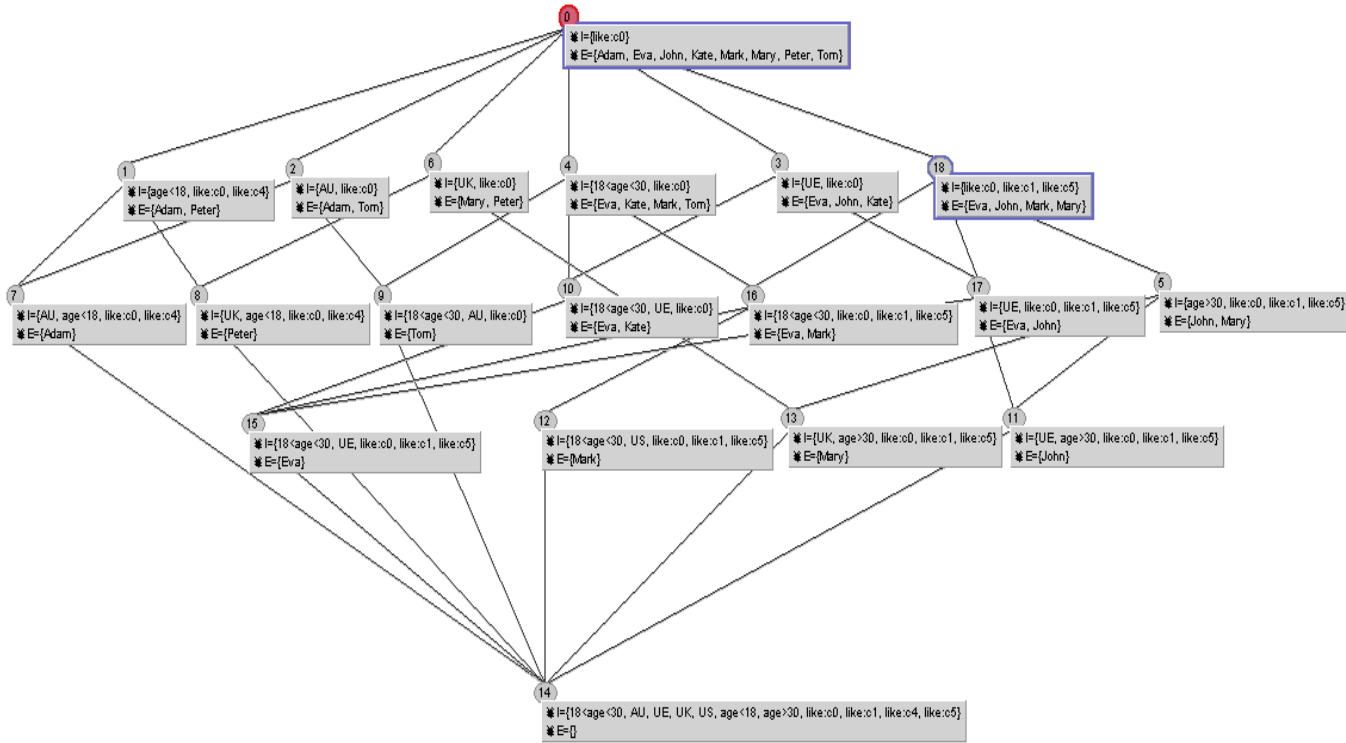


FIGURE 4.3 – Le treillis relationnel $\mathcal{L}_{P_F}^{\forall,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage universel par la relation "Like" par rapport au treillis \mathcal{L}_F .

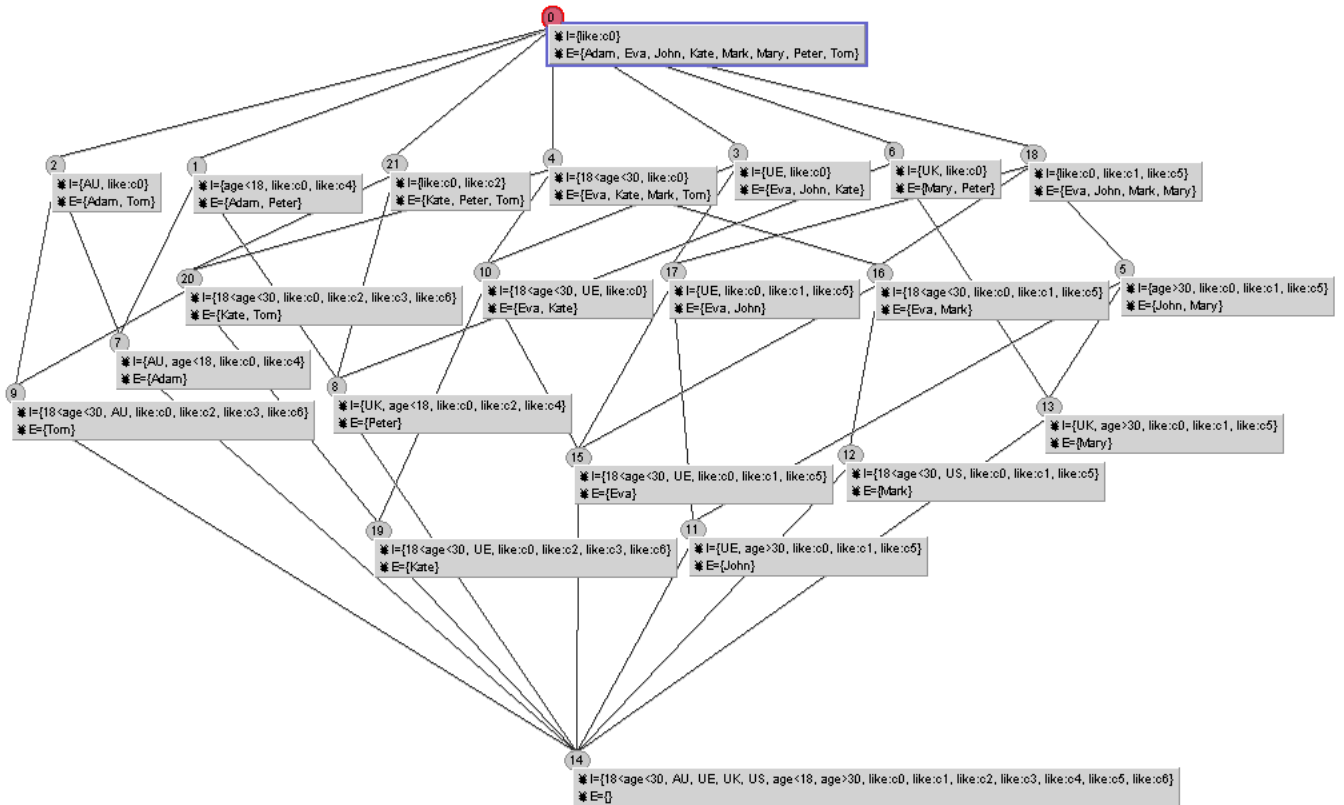


FIGURE 4.4 – Le treillis relationnel $\mathcal{L}_{PF}^{\exists,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage existentiel par la relation "Like" par rapport au treillis \mathcal{L}_F .

Le treillis $\mathcal{L}_{P_F}^{\forall,+}$ est le résultat de l'exécution du codage universel sur le contexte \mathcal{K}_P . Le treillis $\mathcal{L}_{P_F}^{\exists,+}$ est le résultat de l'exécution du codage existentiel sur le même contexte. Si on compare ces deux treillis, nous remarquons à première vue que le nombre de concepts du treillis $\mathcal{L}_{P_F}^{\exists,+}$ est plus grand que celui de $\mathcal{L}_{P_F}^{\forall,+}$. Ceci est le résultat de la contrainte forte imposée par le codage universel, qui impose l'inclusion entre les ensembles $r(o)$ et $extension(c)$, avec o un objet de \mathcal{K}_P et c un concept dans \mathcal{L}_F . Les objets dans le contexte des personnes enrichi par le codage existentiel possèdent plus d'attributs et vérifient donc l'idée intuitive que la structure conceptuelle induite est plus précise. En effet, dès qu'un objet possède une relation avec un autre objet dans l'extension d'un concept du treillis cible, un attribut relationnel lui est affecté.

L'interprétation de certains concepts du treillis $\mathcal{L}_{P_F}^{\forall,+}$ permet de déduire des relations entre les classes d'utilisateurs du réseau social et les classes des films. Par exemple, le concept 1 ($\{\text{Adam, Peter}\}, \{\text{age} < 18, \text{like} : c4\}$) ($\text{like} : c0$ est omis car il possède une intension vide) permet de déduire que les adolescents (âgés de moins de 18 ans) aiment les films d'animation (cf. concept $c4$ ($\{\text{Princess, Toy}\}, \{\text{Animation}\}$) dans le treillis des films). De la même manière, le concept 5 ($\{\text{John, Mary}\}, \{\text{age} > 30, \text{like} : c1, \text{like} : c5\}$) permet de déduire que les adultes (âgés de plus de 30 ans) aiment plutôt les films dramatiques et politiques-historiques (cf. concept $c1$ ($\{\text{12years, Lincoln, Miserables, Titanic}\}, \{\text{Drame}\}$) et concept $c5$ ($\{\text{12years, Lincoln, Miserables}\}, \{\text{Drame}\}$) dans le treillis des films).

Codage d'une relation circulaire (même domaine et co-domaine)

Considérons le scaling du contexte des utilisateurs \mathcal{K}_P de la table 4.1 avec la relation circulaire **Ami** illustrée par la table 4.3 en utilisant le treillis initial \mathcal{L}_P de la figure 4.1 obtenu à partir des attributs binaires du contexte des utilisateurs. Le treillis $\mathcal{L}_{P_P}^{\forall,+}$ de la figure 4.5 représente le résultat de ce scaling.

Les relations inter-concepts induites par des relations inter-objets viennent ajouter une nouvelle dimension à l'interprétation des concepts du treillis initial. Le treillis enrichi $\mathcal{L}_{P_P}^{\forall,+}$ fournit une vue synthétique sur les relations d'amitié dans un réseau social relativement à l'information sur l'âge. En effet, en observant par exemple les deux concepts inter-reliés $c8$ ($\{\text{Eva, Kate}\}, \{\text{18} < \text{age} < 30, \text{UE}, \text{ami} : c1\}$) et $c15$ ($\{\text{Mark, Tom}\}, \{\text{18} < \text{age} < 30, \text{ami} : c3, \text{ami} : c8\}$), nous observons que dans la population considérées les personnes âgées de 18 à 30 choisissent des amis qui ont plus ou moins le même âge. Ceci est confirmé par le concept $c3$ ($\{\text{Eva, Kate, Mark, Tom}\}, \{\text{18} < \text{age} < 30, \text{ami} : c3\}$), avec lequel les deux concepts $c8$ et $c15$ sont en relation. De plus, le concept $c3$ est en relation avec lui même, de la même façon que les concepts $c1$ ($\{\text{Adam, Peter}\}, \{\text{age} < 18, \text{ami} : c1\}$) et $c10$ ($\{\text{John, Mary}\}, \{\text{UK}, \text{age} > 30, \text{ami} : c10, \text{ami} : c14, \text{ami} : c4\}$), ce qui nous permet d'interpréter que les personnes d'une même tranche d'âge sont amies entre elles.

Codage sur toutes les relations partant d'un contexte

Un contexte \mathcal{K} peut être enrichi avec toutes les relations dans $rel(\mathcal{K})$. Ceci est appelé *extension relationnelle complète* de \mathcal{K} et consiste à ajouter à ce contexte tous les attributs relationnels résultants. Formellement, l'extension relationnelle complète est définie comme l'apposition de \mathcal{K} avec le résultat du codage avec chaque relation r dans $rel(\mathcal{K})$. L'apposition de deux contextes exige qu'ils possèdent le même ensemble d'objets et que le contexte résultant possède un ensemble d'attributs et une incidence obtenus par l'union des composants des contextes initiaux. Par exemple, l'extension relationnelle complète du treillis \mathcal{L}_P est donnée par le treillis $\mathcal{L}_{P,F}^{\forall,+}$ de la figure 4.6.

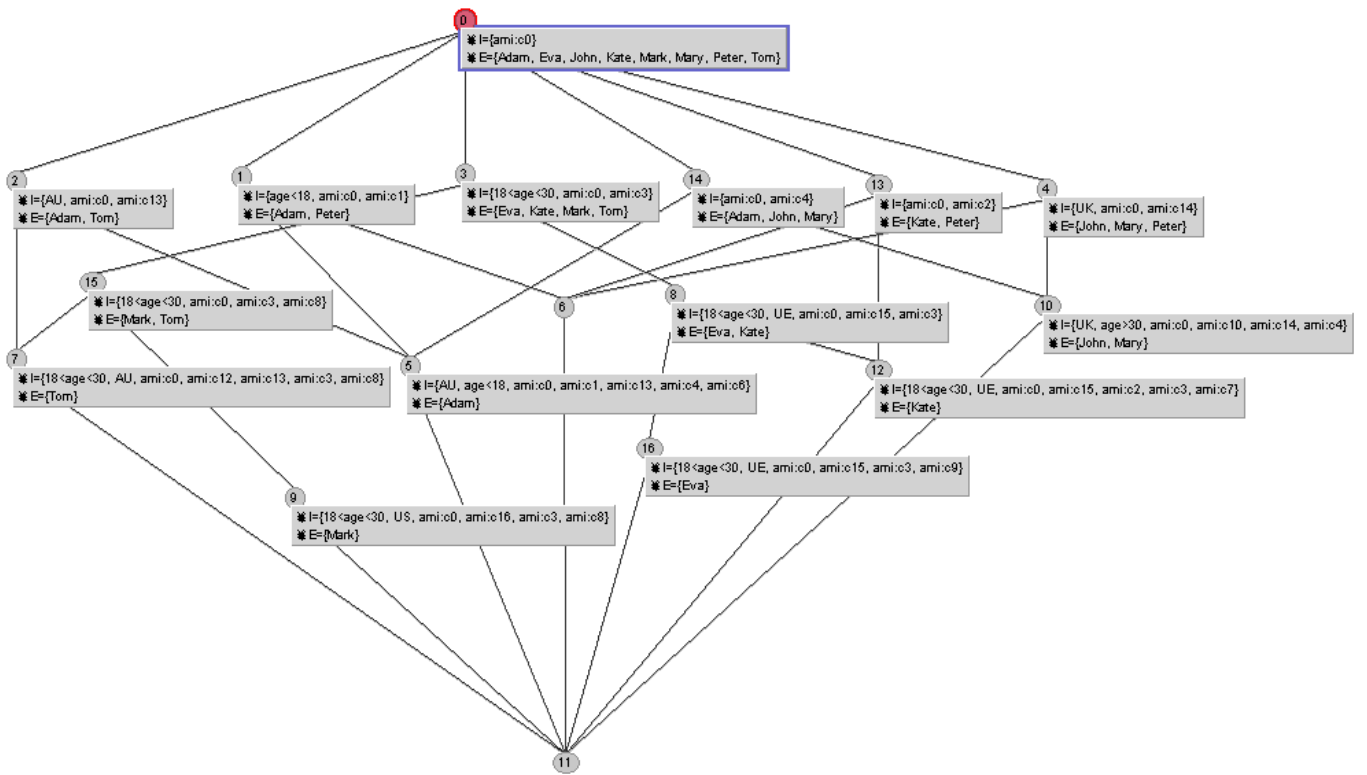


FIGURE 4.5 – Le treillis relationnel $\mathcal{L}_{P_P}^{V,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage existentiel par la relation Ami .

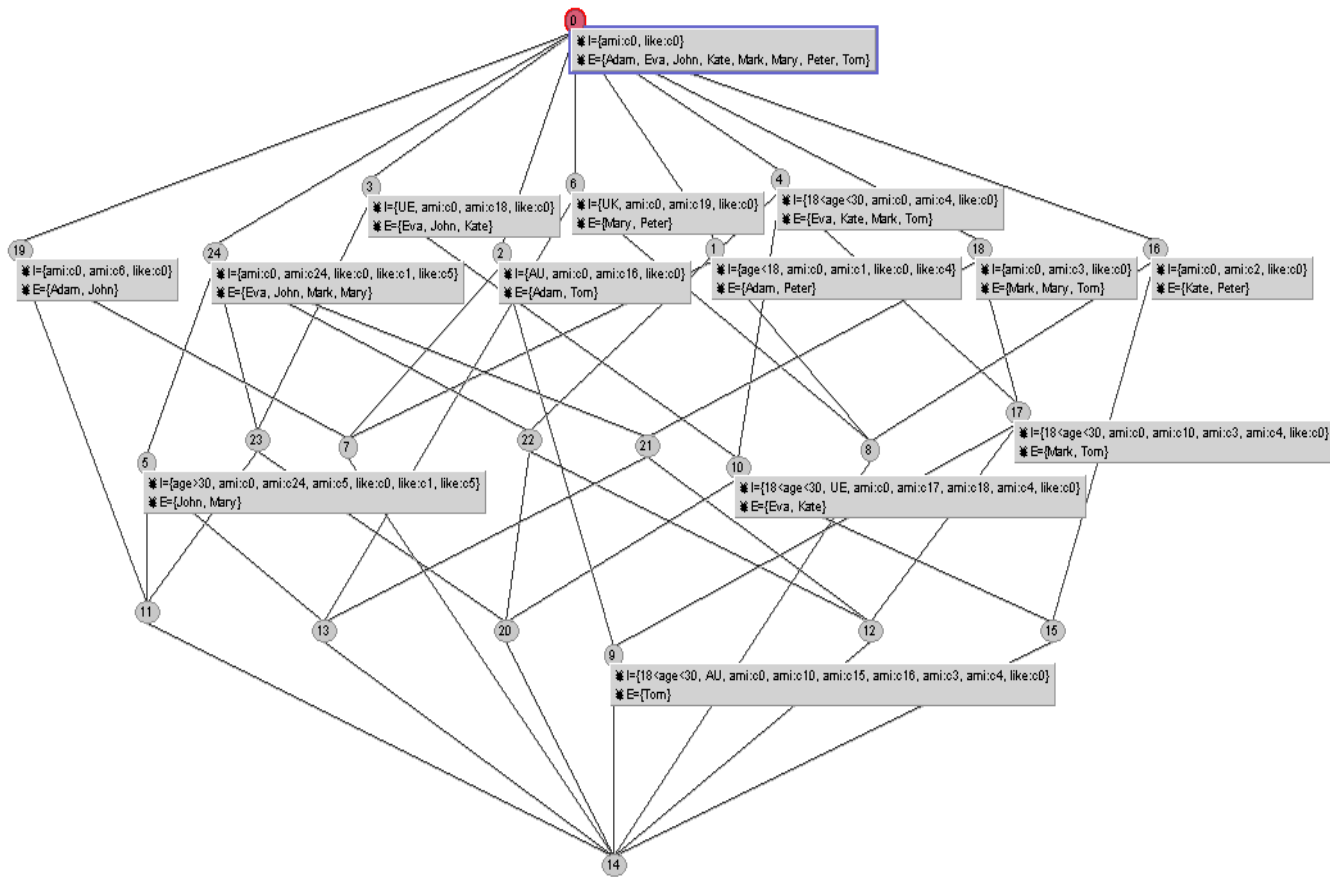


FIGURE 4.6 – Le treillis relationnel $\mathcal{L}_{P,F}^{\vee,+}$ correspondant au contexte formel \mathcal{K}_P enrichi par codage universel par les relations Ami et "Like".

Construction des structures relationnelles

La construction de l'ensemble des treillis associés à une FCR est un processus itératif avec condition d'arrêt qui alterne la pure construction de treillis et l'enrichissement des contextes par codage relationnel. La procédure générale est décrite par l'algorithme 1.

Algorithm 1 Produire un treillis pour chaque contexte d'une FCR : méthode MULTI-FCA

1. **Entrée.** $FCR = (K; R)$: n contextes formels, plusieurs contextes relationnels
 2. **Étape Initialisation.**
 Pour i de 1 à n faire
 $L_i^0 \leftarrow$ construire le treillis de concepts du contexte K_i^0
 3. **Étape Enrichissement.**
 Pour i de 1 à n faire
 – $K_i^p \leftarrow$ enrichir K_i^{p-1} avec les contextes relationnels dans $rel(K_i)$ et les treillis de l'étape précédente L_i^{p-1}
 – $L_i^p \leftarrow$ mise à jour du treillis L_i^{p-1} avec le contexte enrichi K_i^p
 4. Condition d'arrêt $\leftarrow L_i^p$ et L_i^{p-1} sont isomorphes, pour $i = 1, \dots, n$ (point de saturation)
 5. **Sortie.** Une famille de treillis relationnels
-

L'approche prend en entrée une famille de contextes relationnels $FCR = (K; R)$ et donne en sortie une famille de treillis relationnels. Le processus commence (étape d'initialisation) par la construction des treillis initiaux L_i^0 des contextes formels K_i^0 de la FCR en considérant les objets formels avec leurs attributs binaires et en ignorant toute information relationnelle. Ensuite, dans les étapes suivantes (étape enrichissement), un mécanisme de codage relationnel (large ou étroit) traduit pour chaque contexte K_i^{p-1} les liens entre les objets en attributs classiques de l'AFC en partant des treillis construits à l'étape précédente L_i^{p-1} et de l'ensemble de ses relations $rel(K_i)$ (décrites par les contextes relationnels). Les contextes K_i^p sont produits par ajout de ces attributs aux contextes K_i^{p-1} de l'étape précédente puis les treillis enrichis L_i^p sont construits à partir des contextes K_i^p . Une nouvelle étape d'enrichissement relationnel et de construction de treillis est entamée jusqu'à ce que la condition d'arrêt du processus soit vérifiée : les treillis produits à l'étape p sont isomorphes à ceux de l'étape $p - 1$ (il n'y a pas de nouveaux concepts produits). Le processus renvoie alors l'ensemble des treillis relationnels construits (dont les concepts sont liés par les relations de la FCR).

Lorsque la construction des treillis et le codage sont faits par des algorithmes itératifs (voir section 4.2.2), la complexité totale de la méthode MULTI-FCA [Rouane et al., 2013] est en $\mathcal{O}(|L| \times |O| \times (|A| + |O|))$ avec $|L|$ le nombre de concepts du treillis le plus large, $|A|$ le nombre d'attributs du contexte le plus large au point de saturation et $|O|$ le nombre maximal d'objets dans un contexte. Un autre aspect important de la MULTI-FCA c'est sa convergence, c'est-à-dire le nombre d'étapes nécessaires avant le point de saturation. Le processus s'arrête dès qu'il n'y a plus de nouveaux concepts créés.

4.3 Applications de l'AFC et ARC

L'application de l'AFC à la RI et l'utilisation des treillis de concepts à la découverte des ressources et plus précisément dans la recherche documentaire a fait l'objet de plusieurs travaux [Carpineto and Romano, 1993, Godin et al., 1993, Godin et al., 1995a]. Les collections de

documents sont représentées sous la forme de contextes formels, les objets correspondent aux documents et les attributs correspondent aux termes d'indexation. Chaque classe du treillis résultant correspond à un ensemble de documents décrits par les termes d'index communs. Dans une perspective de recherche booléenne, chaque classe peut être vue comme une requête formée par la conjonction des termes d'index de la classe. Le graphe construit représente une relation de généralisation/spécialisation entre les requêtes. Deux modes de recherche par treillis sont définis : la recherche par interrogation, qui consiste à identifier la classe du treillis qui correspond à la requête et la recherche par navigation qui utilise la structure hiérarchique des treillis de concepts pour des fins de généralisation ou de spécialisation. Ces deux modes servent de base aux propositions du chapitre 6.

Dans [Messai et al., 2006] et [Comparot et al., 2010], les auteurs proposent des techniques de raffinement et d'expansion de requêtes en s'appuyant sur des ontologies de domaine, ce qui permet d'améliorer le rappel par généralisation sur la structure du treillis de Galois. Sur des données textuelles, [Carpineto and Romano, 2005] propose une méthode de recherche d'information par treillis de concepts. Une contribution à l'indexation et la recherche d'information sémantique basée sur l'AFC a été proposée dans [Codocedo et al., 2012] et dans [Codocedo et al., 2013] les auteurs utilisent les *pattern structures* pour traiter des données plus complexes. Dans [Messai et al., 2005], les auteurs utilisent les treillis de concepts pour la découverte et l'interrogation de ressources génomiques sur le web et dans [Alam et al., 2013] une approche basée sur les treillis a été proposée pour l'organisation et l'accès aux données liées ouvertes dans le domaine de la biologie.

D'autres travaux ont mis l'accent sur la classification et la structuration des résultats fournis par les algorithmes de RI ce qui influe sur les interfaces de navigation [Nauer and Toussaint, 2008, Poshyvanyk and Marcus, 2007, Carpineto et al., 2006, Koester, 2006]. L'idée principale est de créer un contexte formel à partir des résultats fournis par les moteurs de recherche sur le web, de construire le treillis correspondant à ce contexte, puis de proposer à l'utilisateur un classement des résultats tel que construits par ce treillis. Ce type d'approche est implémenté dans plusieurs systèmes opérationnels tels que CREDINO [Carpineto et al., 2006], FooCA [Koester, 2006] ou CRECHAINDO [Nauer and Toussaint, 2008]. Dans son travail, Nauer [Nauer and Toussaint, 2008] propose de classer les résultats de recherche sur le web pour permettre à l'utilisateur de juger la pertinence des résultats qui lui sont fournis. Poshyvanyk [Poshyvanyk and Marcus, 2007] utilise l'AFC pour classer les résultats de la RI suite à une requête pour localiser des concepts dans un code source. Dans la même direction, les auteurs dans [Chekol and Napoli, 2013] proposent un cadre pour la découverte de connaissances avec l'AFC dans les résultats de requêtes SPARQL. Le système Cordiet-FCA, proposé par [Kuznetsov et al., 2012], est un système de découverte de connaissances dans les grandes collections de textes dynamiques. Il permet à un utilisateur de composer des requêtes contrôlées par une ontologie et retourne un treillis de concepts et des règles d'associations.

Une revue récente étudiant les travaux traitant la problématique de la RI basée sur l'AFC est donnée dans [Poelmans et al., 2011]. L'étude est présentée comme une tâche de fouille de texte sur des communications scientifiques dans ce domaine de recherche.

La navigation conceptuelle basée sur l'AFC prend également en charge la recherche exploratoire en guidant les utilisateurs d'un concept à un autre. Plusieurs travaux ont étudié la contribution de l'AFC pour la recherche par navigation et le parcours de collections de données et ont prouvé son utilité [Carpineto and Romano, 1996, Ducrou and Eklund, 2008, Ferré, 2009]. L'AFC a servi à explorer l'espace d'information du patrimoine culturel et des collections d'art [Wray and Eklund, 2011]. Dans ce travail, les auteurs proposent une approche qui utilise la notion du voisinage conceptuel et de similarité pour la navigation dans un treillis de concepts. Une solution pour le pro-

blème de navigation dirigée par la requête dans des textes non structurés du web à l'aide de l'AFC a été proposée dans [Cole and Eklund, 2001]. En modélisant l'espace de recherche d'une base de donnée par un treillis, les auteurs dans [Demko and Bertet, 2012] proposent une approche de recherche d'information par navigation en-ligne dans cet espace. D'autres travaux s'intéressent aux données multimédia telles que les images. En combinant l'AFC et les vignettes des images, un outil de navigation et de recherche de collections annotées d'images est décrit dans [Ducrou et al., 2006, Ducrou and Eklund, 2008]. L'auteur dans [Ferré, 2009] présente l'outil CAMELIS pour l'organisation et la navigation dans une collection de photos. L'outil est conçu sur le modèle des systèmes d'information logiques (LIS), qui sont fondés sur l'analyse de concept logique (ACL). Le système LIS a été étendu pour permettre une navigation conceptuelle dans des graphes RDF facilitée par des requêtes qui se rapprochent de SPARQL mais qui sont basées sur un langage logique plus compréhensible pour un non expert [Ferré, 2010].

L'AFC a été également utilisée dans plusieurs autres applications pour l'analyse et l'exploitation de données et pour la découverte de ressources comme la gestion de messagerie électronique [Cole et al., 2003], la recherche de séquences vidéo [Mimouni and Slimani, 2006], l'analyse des réseaux sociaux [Missaoui, 2013], etc. Une sélection des approches développées dans ce cadre et une revue récente qui détaille les principaux travaux dans ces domaines sont données dans [Ganter et al., 2005, Andrews and Orphanides, 2013, Poelmans et al., 2013a, Poelmans et al., 2013b].

L'ARC a été utilisée avec succès en ingénierie de connaissances, en génie logiciel et en conception d'ontologies. En génie logiciel, l'ARC a été utilisée dans l'analyse des objets (artefacts) UML [Arévalo et al., 2006, Huchard et al., 2007], la détection et la correction des erreurs de conception [Moha et al., 2008], l'apprentissage de transformations de modèles à partir d'exemples [Saada et al., 2012], la restructuration de diagrammes de cas d'usage UML [Dao et al., 2004] ainsi que la classification et la composition de services web [Azmeh et al., 2011a]. Dans la conception d'ontologies, l'ARC a été appliquée pour la construction et la restructuration des ontologies de domaines [Rouane-Hacene et al., 2010, Shi et al., 2011, Hacene et al., 2011]. Elle a été également utilisée pour la découverte des patrons de connaissances (*knowledge patterns*) [Rouane et al., 2010] et l'exploration de données relationnelles dans les systèmes hydrauliques avec une proposition d'optimisation basée sur les ensembles partiellement ordonnées d'objets-attributs [Dolques et al., 2013].

L'interrogation relationnelle basée sur L'ARC a été introduite dans [Azmeh et al., 2011a] où les auteurs travaillent sur le problème de la sélection des services web appropriés pour l'instantiation d'un workflow abstrait. Ils proposent un algorithme pour naviguer dans la structure relationnelle guidée par la requête de l'utilisateur. L'utilisation de l'ARC pour gérer la structure multi-relationnelle d'une collection de documents et l'application au domaine de la RI pour la recherche relationnelle a reçu moins d'attention. Bien que l'application de l'ARC à un problème aussi complexe n'est pas évidente, c'est une approche prometteuse pour s'attaquer au problème de l'interrogation relationnelle dans la recherche documentaire.

4.4 Web sémantique et web de données

Dans les sections précédentes nous avons étudié la modélisation d'un ensemble d'objets interreliés avec une approche conceptuelle basée sur l'analyse formelle et l'analyse relationnelle de concepts. Cet ensemble d'objets peut aussi naturellement être encodé sous la forme d'un graphe de données faisant appel aux technologies sémantiques.

4.4.1 Les technologies du web sémantique

La figure 4.7 montre les couches de technologies sur lesquelles se base le web sémantique. Dans la suite nous détaillons les technologies utilisées dans le cadre du web de données, notamment celles que nous utilisons dans la suite de notre travail.

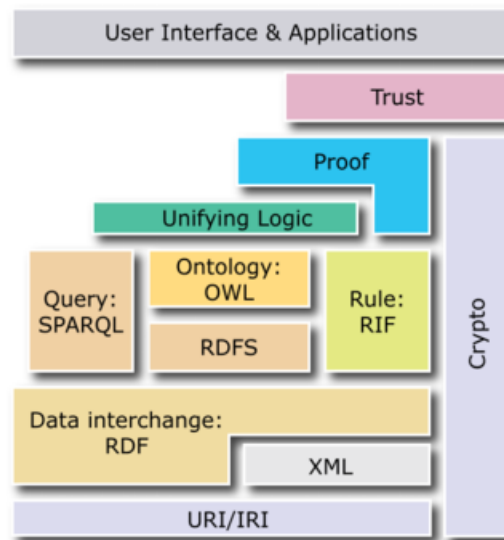


FIGURE 4.7 – Architecture du web sémantique (*semantic web stack*).

Uniform Resource Identifiers (URI)

Les URIs⁵¹ et les IRIs⁵² généralisent les URLs⁵³. Au lieu de faire référence uniquement aux pages web, les IRIs identifient tout type de ressource : une donnée présente sur le web, un objet du monde réel ou aussi une relation. Il s'agit d'un mécanisme d'identification universel qui permet d'identifier de façon unique toutes les ressources.

Resource Description Framework (RDF)

La deuxième couche est consacrée à la représentation syntaxique des données en utilisant RDF⁵⁴. RDF est le format de base pour la représentation de données pour le web sémantique. C'est un modèle qui permet de représenter des informations sur les ressources sous forme de graphes. Il est basé sur des triplets sujet→prédicat→objet qui forment des graphes. Un exemple de triplet RDF (dit aussi graphe RDF) est donné par la figure 4.8.

Dans ce graphe :

- le sujet est **Peter**, une ressource accessible via un URI ;
- le prédicat est "Like", une propriété, possédant un URI, qui définit la relation entre le sujet et l'objet ;

51. <http://www.ietf.org/rfc/rfc2616.txt>

52. *Internationalized Resource Identifiers*

53. *Uniform Resource Locators*

54. <http://www.w3.org/standards/techs/rdf>



FIGURE 4.8 – Graphe de données décrivant la relation "Like" entre un utilisateur d'un réseau social et un film.

- l'objet est `Toy story`, une ressource accessible via un URI. Dans le cas général, l'objet peut être une ressource ou une valeur littérale (entier, caractère, etc.).

RDF est considéré comme la base du web sémantique, qui est vu comme un grand graphe dont les ressources (les noeuds) sont interconnectés via des propriétés (les arcs). Selon les recommandations du W3C, RDF est muni de deux syntaxes : XML et Turtle. Bien qu'il soit la base de la définition des structures de données pour le web sémantique, RDF ne permet pas de décrire la sémantique, ou le sens des données. Pour attribuer de la sémantique aux modèles de données RDF, deux technologies sont utilisées : RDFS (RDF Schema) et OWL (Web Ontology Language).

Resource Description Framework Schema (RDFS)

RDFS⁵⁵ peut être considéré comme un langage d'ontologie simple qui exprime les relations de subsomption entre classes ou propriétés. C'est un vocabulaire utilisé pour exprimer la sémantique qui permet d'interpréter des graphes RDF. Les schémas sont eux-mêmes exprimés par des graphes RDF. RDFS définit la notion de classe et de propriété pour une ressource, ainsi que le domaine et le co-domaine d'une relation. Un vocabulaire RDFS peut contenir des sous-classes et des sous-propriétés. Les spécifications du W3C introduisent deux espaces de noms standards : `http://www.w3.org/1999/02/22-rdf-syntax-ns#` (préfixe *rdf*) et *RDF Schema namespace* `http://www.w3.org/2000/01/rdf-schema#` (préfixe *rdfs*) qui comprennent un ensemble d'URIs ayant un sens prédéfini. Par exemple :

- *rdfs : Class* déclare une ressource comme une classe pour d'autres ressources ;
- les propriétés sont des instances de la classe *rdf : Property* et décrivent une relation entre les ressources sujets et les ressources objets ;
- *rdfs : domain* et *rdfs : range* indiquent les classes domaine et co-domaine d'une propriété ;
- *rdfs : subclassOf* et *rdfs : subPropertyOf* sont utilisés pour décrire une hiérarchie entre les classes et les propriétés respectivement ;
- l'URI prédéfinie *rdf : type* est une propriété utilisée pour indiquer qu'une ressource est une instance d'une classe (définir les types des ressources).

Toutes les entités décrites par RDF sont appelées *des ressources*, et sont des instances de la classe *rdfs : Resource*. La figure 4.9 reprend le graphe RDF de la figure 4.8 en ajoutant un premier niveau de sémantique avec RDFS pour la définition des types des ressources sujet et objet.

55. <http://www.w3.org/TR/rdf-schema/>

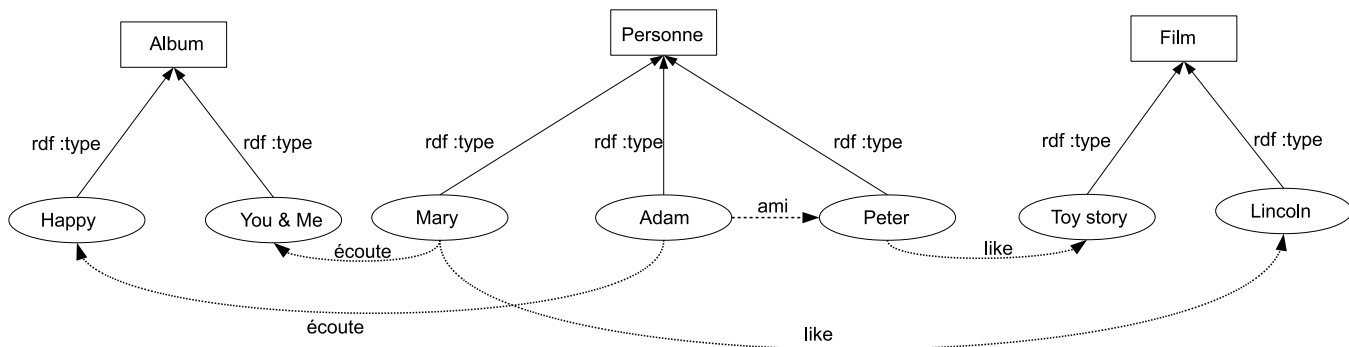


FIGURE 4.9 – Graphe RDF avec types sémantiques des sujets et des objets.

Web Ontology Langage (OWL)

Quand l'expressivité de RDFS n'est pas suffisante, le vocabulaire du web sémantique peut être conçu en OWL⁵⁶. OWL est une recommandation du W3C qui définit une famille de langages de représentation de connaissances pour la création d'ontologies sur le web sémantique. L'espace de noms standard spécifié par W3C pour OWL est <http://www.w3.org/2002/07/owl#> (préfixe *owl*). Grâce à un vocabulaire sémantique plus riche que RDFS, le langage OWL, notamment dans son extension OWL 2, supporte plus de fonctionnalités telles que l'union et l'intersection de classes et la restriction de cardinalité. Il offre trois sous-langages listés ici par ordre d'expressivité croissante : OWL Lite, OWL DL et OWL Full.

- OWL Lite : c'est le langage le plus simple syntaxiquement. Il est conçu pour être utilisé dans les cas où seule une simple hiérarchie de classes et des contraintes simples sont requises. Par exemple, OWL Lite peut suffire pour exprimer des thésaurus ou des structures conceptuelles simples.
- OWL DL offre le plus haut niveau d'expressivité tout en maintenant la décidabilité. Basé sur la logique de description, il permet de calculer automatiquement la classification hiérarchique et de détecter les incohérences dans une ontologie décrite en OWL DL.
- OWL Full : c'est le langage le plus expressif mais sans garantie de décidabilité. Il n'est donc pas possible de faire du raisonnement automatique sur les ontologies OWL Full.

Les données décrites par une ontologie OWL sont interprétées comme des ensembles d'*individus*, appelés *classes*, et un ensemble de *propriétés* qui lient ces individus entre eux (*propriétés d'objets* liant des individus entre eux), ou leur associant des attributs (*propriétés de données* liant les individus à des types prédéfinis (entier, chaîne de caractère, etc.)). L'ontologie se compose d'un ensemble d'axiomes qui placent des contraintes sur des ensembles d'individus et les types de relations autorisées entre eux. Ces axiomes permettent aux systèmes de déduire des informations supplémentaires sur la base des données fournies explicitement.

SPARQL

SPARQL⁵⁷ est un langage d'interrogation de données RDF. Il peut également être utilisé pour interroger directement les ontologies et les bases de connaissances du fait que RDFS et OWL sont construits sur RDF. SPARQL est un langage similaire à SQL, mais il repose sur la

56. <http://www.w3.org/standards/techs/owl>

57. <http://www.w3.org/standards/techs/sparql>

structure des triplets RDF et les ressources pour exprimer des requêtes et retourner des résultats à ces requêtes.

Une requête SPARQL se compose généralement de cinq parties :

1. Déclaration des préfixes : les IRIs des *namespaces* RDF et OWL, souvent écrits en raccourcis *prefix :localname*.
2. Clause de type de requête : SELECT, ASK, CONSTRUCT et DESCRIBE. SPARQL permet d'exprimer des requêtes interrogatives ou constructives. SELECT, requête interrogative, permet de sélectionner des éléments selon le schéma de graphe défini par la requête (*query pattern*) dans la clause WHERE. ASK retourne "TRUE" ou "FALSE" selon que le patron de requête existe ou pas dans la base interrogée. DESCRIBE renvoie une description, sous forme d'un graphe RDF, de la ressource passée en paramètre. CONSTRUCT crée un nouveau sous-graphe RDF, spécifié par le schéma de graphe passé en paramètre, qui complète le graphe interrogé.
3. Jeux de données (*datasets*) : spécifier les collections de graphes RDF interrogés par la requête en utilisant FROM <graph_uri>.
4. Schémas de graphe : ils sont placés dans la clause WHERE et sont à apparier dans les graphes interrogés. Ils sont formés de triplets RDF utilisant la jointure (.), la disjonction (UNION), etc.
5. Modificateurs de solution : ils sont appliqués sur les résultats pour les trier (ORDER BY), les partitionner (HAVING), les grouper (GROUP BY), etc.

SPARQL n'est pas seulement un langage d'interrogation mais aussi un protocole pour accéder aux données RDF. Les services d'interrogation qui adoptent le langage SPARQL sont appelés *SPARQL endpoints* et sont construits au dessus d'une base de connaissance RDF (*a triple store*). SPARUL (SPARQL 1.1 Update), une extension du SPARQL standard, est un langage déclaratif de manipulation de données qui donne la possibilité d'insérer, supprimer ou mettre à jour des données dans une base de connaissance RDF.

Puissance expressive de SPARQL

La puissance expressive du langage de requête SPARQL se détermine par l'ensemble des requêtes exprimables dans ce langage. Une étude exhaustive a été faite dans [Angles and Gutierrez, 2008] pour déterminer la puissance expressive de SPARQL. Les auteurs comparent SPARQL avec l'algèbre relationnelle (AR) et montrent qu'ils possèdent le même pouvoir expressif. En effet, il a été prouvé que l'AR avec les opérateurs SPJUD (Selection, Projection, Join, Union, Difference) [Abiteboul et al., 1995] est équivalente à SPARQL avec SELECT, AND, UNION, OPTIONAL, MINUS et FILTER. Un mapping entre les opérateurs des deux langages est donné par la table 4.5 [Chekol, 2012].

Dans ce travail, nous nous intéressons aux requêtes SPARQL interrogatives décrites par la clause SELECT. La clause SELECT est définie par la syntaxe suivante :

```
SELECT <liste_variables>
```

où <liste_variables> indique la liste des variables à projeter sur le(s) graphe(s) interrogé(s). Les triplets contiennent à la place des IRIs des variables qui peuvent apparaître dans le sujet, l'objet ou le prédicat. Les variables qui apparaissent dans la clause SELECT d'une requête SPARQL forment le focus de la requête. Étudier l'ensemble de requêtes qu'on peut exprimer avec une

TABLE 4.5 – SPARQL vs. Algèbre relationnelle (AR).

	AR	SPARQL
Selection (Restriction)	σ	FILTER
Projection	π	SELECT
Join (Inner join)	\bowtie	AND
Left outer join	\Join	OPTIONAL
Union	\cup	UNION
Set difference	\setminus	MINUS

requête interrogative SPARQL (clause SELECT) revient à étudier les schémas des graphes de requêtes du fait que les variables spécifiées dans la clause SELECT apparaissent aussi dans les schémas de graphes. Ainsi, en combinant ces graphes on combine les requêtes : disjonction de requêtes (UNION), conjonction de requêtes (AND), négation (MINUS), etc.

Les schémas de graphes qui doivent être appariés dans le(s) graphe(s) RDF interrogé(s) sont placés dans la clause WHERE d'une requête SPARQL. Ils sont formés par une liste ou une combinaison de triplets RDF utilisant [Pérez et al., 2009] :

- la jointure/concaténation (\cdot) (*basic graph pattern*),
- la conjonction (AND),
- la disjonction (UNION) (*union graph pattern*),
- left outer join (OPTIONAL) (*optional graph pattern*),
- la restriction (FILTER) : des expressions ajoutant des contraintes (C) sur les variables (*filter graph pattern*),
- la négation (MINUS) (*subtracted graph pattern*).

Par défaut, une liste de triplets est une conjonction. Les accolades, $\{ \text{et} \}$, permettent de combiner différents opérateurs (par ex. les conjonctions, les disjonctions) dans une même requête. Les schémas de graphes de la requête peuvent ainsi correspondre à différents patrons de graphes, comme illustré dans ce qui suit.

Soit A et B deux schémas de graphes, ils peuvent être combinés pour former différents patrons de graphes :

- Graphe de base (*Basic graph patterns*), qui correspond à un ou plusieurs schémas de triplets $A \cdot B$. Le résultat final est calculé en faisant la jointure des résultats de la résolution de A et B en faisant correspondre les valeurs de toutes les variables en commun.
- Graphe optionnel (*Optional graph patterns*) de la forme $A \text{ OPTIONAL } \{B\}$ (left join). Le résultat final est calculé en faisant la jointure des résultats de la résolution de A et B en faisant correspondre les valeurs de toutes les variables en commun, si possible. Garder toutes les solutions de A telles qu'il n'y a pas de solution correspondante pour B.
- Graphe union (*Union graph patterns*) de la forme $\{A\} \text{ UNION } \{B\}$ (disjonction). Le résultat final est calculé en groupant les résultats de la résolution de A et les résultats de la résolution de B.
- Graphe soustrait (*Subtracted graph patterns* (SPARQL 1.1)) de la forme $\{A\} \text{ MINUS } \{B\}$ (négation). Le résultat final est calculé comme suit : résoudre A, résoudre B puis inclure uniquement les résultats de la résolution de A qui ne sont compatibles avec aucun des résultats de B.

Des contraintes peuvent être ajoutées sous la forme $A.B.FILTER(expression)$. Dans *expression*, des opérateurs du type $!$, $\&\&$, $||$, $=!$, $=$, $<$, $<=$, *etc.* sont utilisés.

SPARQL est aussi capable d'exprimer des schémas de graphes cycliques grâce à l'utilisation de variables et de chemins de propriétés (*Property paths*). Les chemins de propriété SPARQL traitent les triplets RDF comme des graphes orientés, éventuellement cycliques, avec des labels sur les arcs. Lorsque la requête est projetée sur un chemin de longueur arbitraire, chaque cycle est considéré au plus une fois. Le graphe interrogé peut aussi contenir des cycles⁵⁸.

Schéma général de requête Dans [Pérez et al., 2009] une syntaxe réduite de SPARQL a été proposée. Les schémas de graphes, constitués par un ensemble de schémas de triplets, sur un tuple t de variables, groupés par les opérateurs AND, UNION et OPT, forment les schémas de requêtes. Un schéma général de requête est composé d'un ensemble de requêtes individuelles et défini récursivement comme suit [Chekol, 2012] :

Définition 19 (Schéma de requête (*Query Pattern*)) $q ::= t|q_1 \text{ AND } q_2|q_1 \text{ UNION } q_2|q_1 \text{ OPT } q_2|q_1 \text{ FILTER } C$

Définition 20 (Requête SELECT) Une requête *SELECT* dans SPARQL est une requête de la forme $q\{\vec{w}\}$ où q est un schéma de requête et \vec{w} est un tuple de variables qui apparaît dans q appelés variables distinguées.

Une réponse à une requête SPARQL peut être une liste de résultats ou un ensemble de graphes RDF. Dans la section suivante nous donnons des exemples de requêtes SPARQL avec les résultats qui leur sont retournés.

Exemples

Considérons les requêtes suivantes qui interrogent le graphe de la figure 4.9 :

1. $q_1\{?personne\}$ interroge sur toutes les personnes qui aiment ou écoutent quelque chose.

```
SELECT  ?personne
WHERE  { {?personne like ?x }
        UNION {?personne ecoute ?x }
      }
```

La réponse à cette requête est :

Personne
Mary
Adam
Peter
Mary

Le résultat contient deux fois la réponse **Mary** du fait que la variable **?personne** a été projetée sur les triplets du graphe RDF décrivant les relations **like** et **écoute**, respectivement entre **Mary** et le film **Lincoln** et entre **Mary** et l'album **You and Me**. Pour éviter cette situation, le modificateur **DISTINCT** peut être utilisé.

58. Pour plus de détails, voir <http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>, section 9.3.

2. $q_2\{?personne\}$ interroge sur toutes les personnes qui aiment des films ou écoutent des albums.

```
SELECT  DISTINCT ?personne
WHERE {  {?personne like ?x . ?x rdf:type Film .}
        UNION {?personne ecoute ?x . ?x rdf:type Album . }
}
```

La réponse à cette requête est :

Personne
Mary
Adam
Peter

3. $q_3\{?personne\}$ interroge sur les personnes qui aiment et écoutent quelque chose.

```
SELECT  ?personne
WHERE {  {?personne like ?x }
        AND {?personne ecoute ?x }
}
```

La réponse à cette requête est :

Personne
Mary

4. $q_4\{?personne, ?film, ?album\}$ interroge à la fois sur les personnes, les films et les albums.

```
SELECT  ?personne, ?film, ?album
WHERE {  {?personne like ?film . ?film rdf:type Film .}
        UNION {?personne ecoute ?album . ?album rdf:type Album . }
}
```

La réponse à cette requête est :

Personne	Film	Album
Peter	Toy story	
Mary	Lincoln	
Adam		Happy
Mary		You and Me

Les requêtes de 1 à 3 sont des requêtes SPARQL unaires puisqu'elles portent sur une seule variable distinguée. La requête q_4 est ternaire (le nombre de variables distinguées est 3).

4.4.2 Le web de données et les données liées sur le web

Les technologies sémantiques permettent de gérer des graphes de données et de les interroger. Le déploiement de ces technologies a accompagné une tendance générale de création et de publication de données de plus en plus liées.

Le terme *Linked data* (données liées) est utilisé pour désigner le mouvement de publication de données liées sur le web. Il vise à publier non seulement des documents, mais aussi des données, et contribue à l'extension du web en un espace global de données basé sur des standards ouverts. Cet espace de données est appelé le web de données (*Web of data*) [Bizer et al., 2009, Berners-Lee, 2007, Heath and Bizer, 2011]. Il a été créé en réponse à deux grandes questions : comment publier des données qui soient réutilisables ? et comment favoriser l'intégration de données venant de sources différentes ? Les solutions proposées consistent à imposer aux données d'être structurées pour faciliter leur partage et leur réutilisation et à suivre un modèle standard pour faciliter leur découverte et leur intégration.

Dans ce cadre, un ensemble de bonnes pratiques pour la publication et l'interconnexion de données structurées sur le web [Berners-Lee, 2006, Heath and Bizer, 2011] sont définies :

1. utiliser les URIs pour donner des noms aux choses ;
2. utiliser les URIs HTTP, de sorte que les utilisateurs peuvent accéder à ces noms ;
3. quand un utilisateur regarde un URI, fournir des informations utiles, en utilisant les standards (RDF, SPARQL) ;
4. inclure des liens vers d'autres URIs, afin que les utilisateurs puissent découvrir plus de choses.

Les trois premiers principes consistent à identifier les entités et à les définir. Le quatrième principe consiste à mettre des liens RDF pointant vers d'autres sources de données sur le web. Ces liens RDF externes sont fondamentaux car ils permettent de relier les données éparpillées sur le web dans un espace global et permettent aux applications de découvrir des sources de données supplémentaires.

Un nombre important d'organismes ont adopté les principes de données liées comme une façon de publier leurs données ce qui a permis de créer un espace global de données interconnectées constitué de plusieurs milliards de triplets RDF provenant de nombreuses sources : données géographiques, statistiques, génétiques, pharmaceutiques, médicales, publications scientifiques, films, musique, etc. (voir figure 4.10⁵⁹).

4.4.3 Les ontologies

Dans la littérature, plusieurs définitions ont été attribuées à la notion d'ontologie. Les définitions les plus utilisées présentent une ontologie comme « une spécification explicite et formelle d'une conceptualisation partagée d'un domaine de connaissance » [Gruber, 1993, Studer et al., 1998]. L'utilisation d'ontologies dans les systèmes d'information est devenue une pratique récurrente du fait de leur capacité à représenter et à organiser les connaissances de différents domaines de façon explicite, non ambiguë et compréhensible à la fois par un utilisateur et par une machine. Les ontologies sont ainsi considérées comme un système fiable pour l'intégration, l'interopérabilité et le partage de données et de connaissances.

Une ontologie est formalisée par un langage de représentation logique. Dans le web sémantique, une famille de langages de représentation, OWL (voir section 4.4.1), est utilisée pour la création d'ontologies. Il existe différents types d'ontologies. Dans [Oberle et al., 2006], les auteurs proposent une classification selon :

- leur but : ontologies d'application, ontologies de référence,
- leur expressivité : ontologies lourdes avec beaucoup d'axiomes pour faire des raisonnements complexes, ontologies légères avec peu ou pas d'axiomes,
- leur spécificité : ontologies génériques ou de haut niveau, ontologies noyaux ou *core* ontologies définissant des concepts communs à un ensemble de domaines, ontologies de domaine définissant les concepts d'un domaine spécifique.

Différents critères ont été définis pour évaluer une ontologie. Selon Gruber [Gruber, 1993], une ontologie doit être :

- claire : elle doit communiquer efficacement le sens des termes définis. Chaque nouveau terme doit être documenté avec des labels et des commentaires en langage naturel. Des propriétés RDFS sont conçues à cet effet : `rdfs:label` et `rdfs:comment` ;
- cohérente : les axiomes qu'elle définit doivent être cohérents avec les définitions, à défaut être logiquement cohérents. Aucune connaissance inférée ne doit contredire la définition d'une classe de l'ontologie ;

59. <http://lod-cloud.net/>

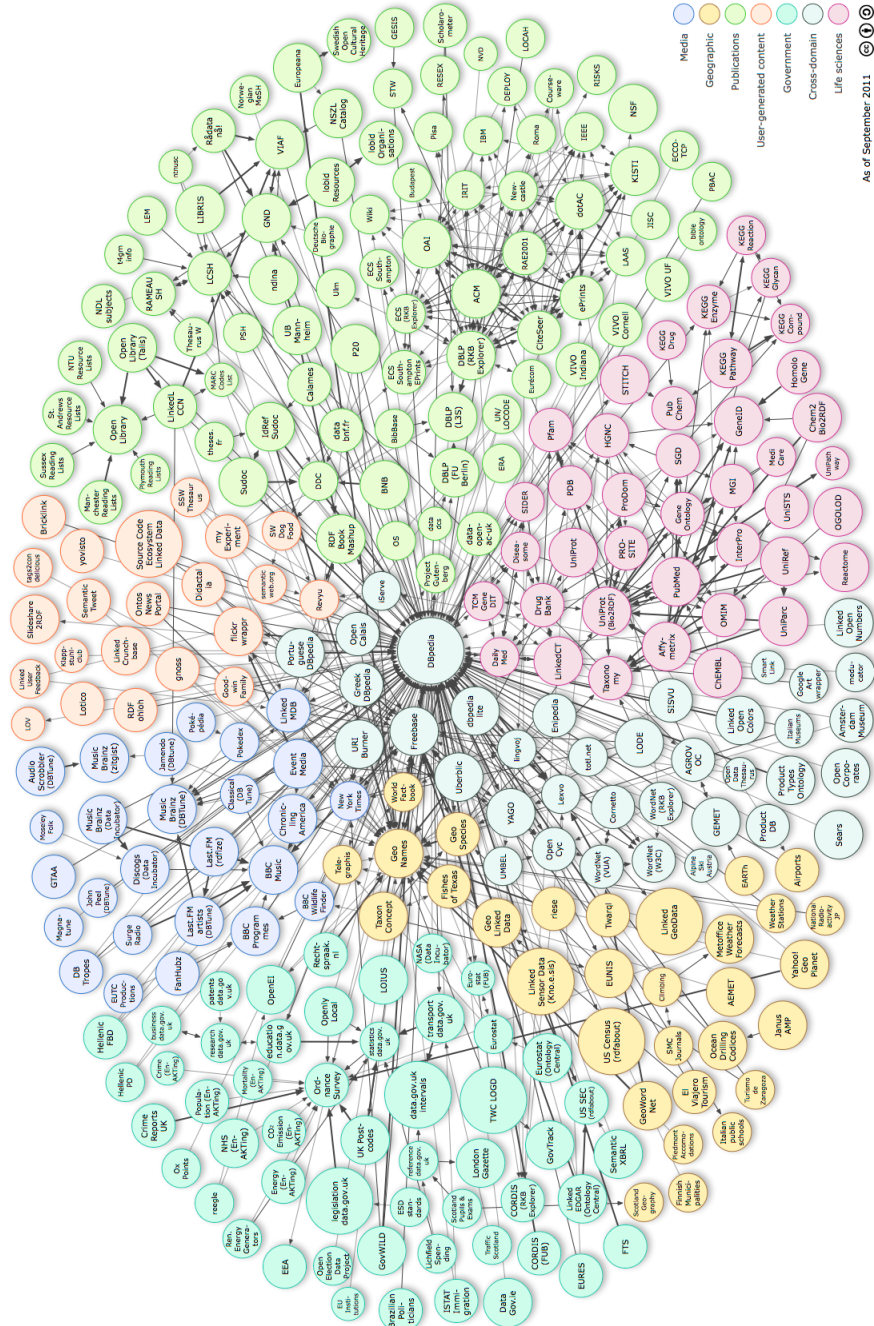


FIGURE 4.10 – Le nuage de données liées (*Linked data cloud diagram*). Chaque cercle représente un ensemble de données publiées selon les principes des données liées. La taille des cercles représente le nombre de triplets qu'ils contiennent. Le jeu de couleurs identifie les domaines. Les flèches indiquent qu'au moins 50 triplets relient les ensembles de données.

- extensible : une ontologie doit être conçue pour anticiper le partage de vocabulaire c'est-à-dire de sorte à pouvoir l'étendre et le spécialiser d'une manière qui ne nécessite pas la révision des définitions existantes ;
- réutilisable : la conceptualisation devrait être spécifiée au niveau de connaissances sans dépendre d'un codage particulier afin de faciliter l'interopérabilité et de permettre le partage de connaissances entre plusieurs applications.

Afin d'approcher de ces critères, la conception d'une ontologie doit respecter certaines règles de bonnes pratiques. Ces règles sont différentes selon la nature de l'ontologie et la technique de construction (automatique, semi-automatique ou manuelle). Plusieurs méthodes sont proposées dans la littérature [Corcho et al., 2003]. La construction manuelle, la technique que nous avons adoptée dans ce travail, présente l'avantage de produire des ontologies cohérentes et réutilisables tout en se basant sur les vocabulaires existants et sur les experts du domaine. Néanmoins, elle nécessite beaucoup de temps pour la conceptualisation et beaucoup de ressources. La méthode de construction que nous avons suivie se rapproche des quatre étapes décrites dans [Wang and Xu, 2000] à savoir : analyser le domaine d'étude et extraire les connaissances utiles, identifier les éléments ontologiques (les concepts, les relations, etc.) et structurer l'ontologie, choisir le langage et formaliser l'ontologie, évaluer et valider l'ontologie (par des experts et des utilisateurs du domaine).

Une série de spécifications et de recommandations sont faites dans le cadre du web de données pour la construction d'un vocabulaire sémantique [Heath and Bizer, 2011] :

- extensibilité : le web de données étant un environnement ouvert, cette spécification rejoint le critère de [Gruber, 1993] qui incite à prendre en compte les éventuelles extensions des applications réutilisant l'ontologie qui doivent être sans aucun impact sur le modèle déjà défini ;
- une ontologie légère (*lightweight*) : les ontologies utilisées dans le web de données sont définies avec le langage RDFS. Les extensions simples de OWL sont acceptées (par exemple owl:equivalentClass, owl:InverseFunctionalProperty), mais l'objectif est toujours de garder des ontologies simples.
- réutilisation de termes existants : favoriser la réutilisation de classes et de propriétés des vocabulaires existants. Si des termes adéquats peuvent être retrouvés dans les vocabulaires existants, ils doivent être réutilisés autant que possible, plutôt qu'être réinventés. Ceci doit permettre aux applications de consommer directement les données exprimées dans un vocabulaire connu sans prétraitement. Plusieurs vocabulaires existants couvrent des données de type commun et sont largement utilisés : le Dublin Core Metadata Initiative (DCMI), le vocabulaire Friend-of-a-Friend (FOAF), le vocabulaire Semantically-Interlinked Online Communities (SIOC), le schéma Creative Commons (CC), l'ontologie bibliographique (BIBO), le vocabulaire Basic Geo (WGS84), etc. Si les besoins de l'application nécessitent la création de nouveaux termes pour décrire des particularités liées à l'ensemble des données manipulées, ces termes doivent être alignés avec les termes qui se rapprochent dans les vocabulaires prédéfinis ;
- disponibilité sur le web : les vocabulaires publiés sur le web sont accessibles. Une page HTML et un document décrivant l'espace de noms (*namespace document*) doivent être associés à l'ontologie. Ce dernier donne une description textuelle des classes et des propriétés avec des exemples.

Une initiative récente, Linked Open Vocabularies (LOV)⁶⁰, vise à rassembler et fournir un seul point d'entrée pour les vocabulaires ouverts liés (ontologies RDFS ou OWL) utilisés dans *Linked Data Cloud*. Les vocabulaires sont listés et décrits individuellement par des métadonnées, organisés dans des classes de vocabulaires et inter-reliés par le vocabulaire dédié VOAF (Vocabulary Of A Friend⁶¹).

Plusieurs outils sont disponibles pour assister le processus de développement de vocabulaires [Heath and Bizer, 2011] :

- Neologism⁶² est un outil web pour créer, gérer et publier des vocabulaires RDFS simples.
- TopBraid Composer⁶³ est un environnement de modélisation (commercial) puissant pour développer des ontologies du web sémantique.
- Protege⁶⁴ un éditeur libre d'ontologies avec un plugin dédié à OWL.
- The NeOn Toolkit⁶⁵ un environnement libre d'ingénierie d'ontologies.
- Terminae⁶⁶ un outil linguistique pour la construction d'une ontologie de domaine.

4.5 Application à l'analyse documentaire dans le web sémantique

L'essor du web sémantique et du web de données repose sur l'évolution des technologies sémantiques qui assurent l'interopérabilité des données mais aussi sur le développement des ressources pour l'annotation sémantique des documents. Dans ce contexte, un effort est fait pour développer des ontologies documentaires mais les modèles existants sous-estiment selon nous la dimension intertextuelle et ne permettent pas de modéliser l'ensemble des propriétés documentaires de manière homogène, ce qui constitue un frein à l'essor des méthodes de recherche d'information sémantique.

4.5.1 Vocabulaires conceptuels et annotation sémantique

L'approche classique de recherche d'information sémantique (comme par exemple dans Aqua-Log [Lopez et al., 2007], KnOWLer [Ciorascu et al., 2003] ou MELISA [Abasolo and Gomez, 2000]) dépasse les méthodes à base de mots-clés en exploitant les annotations sémantiques qui sont apposées sur les documents pour en modéliser le contenu.

Les termes utilisés comme annotations sont définis dans des vocabulaires ou des ontologies qui sont eux-mêmes définis en SKOS ou OWL. Les ontologies de domaine permettent d'associer aux contenus des documents une description sémantique à la fois explicite et formelle, ce qui facilite l'exploitation sémantique des contenus par des outils automatiques et améliore l'interopérabilité des sources. Dans le domaine juridique, des efforts de standardisation et d'annotation s'appuient notamment sur des ontologies comme DOLCE [Gangemi et al., 2005] ou LKIF core [Hoekstra et al., 2009].

Des outils d'annotation sont utilisés pour annoter les documents, c'est-à-dire pour lier certains fragments de textes (des mots, groupes de mots, phrases, etc.) à des entités de l'ontologie, le plus

60. Publiée le 26/04/2013, <http://lov.okfn.org/dataset/lov/>, par Mondeca, Inserm, DataLift project et Open Knowledge Foundation.

61. <http://lov.okfn.org/vocab/voaf/v2.2/index.html>

62. <http://neologism.deri.ie/>

63. http://www.topquadrant.com/products/TB_Composer.html

64. <http://protege.stanford.edu/>

65. <http://neon-toolkit.org/>

66. <http://ontorule-project.eu/news/news/terminae.html>

souvent à des instances [Amardeilh et al., 2005, Uren et al., 2006a], mais aussi, dans certains cas, à des concepts et à des rôles [Ma et al., 2013].

Le contenu d'un document ainsi que les annotations qui lui sont attachées peuvent ainsi être publiés sous forme de triplets RDF. Les annotations permettent d'identifier les entités et les concepts mentionnés dans les documents d'un domaine donné : littérature scientifique dans le domaine biomédical [Croset et al., 2010] ou celui de la biodiversité [Cui et al., 2010], comptes rendus hospitaliers [Minard et al., 2011], etc. Dans [Mokhtari, 2010a], les annotations sémantiques des documents sont stockées sous forme de triplets RDF, qui sont produits selon l'emplacement de leurs propriétés dans le texte. Dans [Croset et al., 2010], la modélisation sous la forme de triplets RDF et d'URIs permet également de lier les articles scientifiques et les bases de connaissances du domaine. [Mrabet et al., 2012] propose à l'inverse d'enrichir des bases de connaissances RDF/OWL en utilisant une base de documents HTML annotés par un ou plusieurs outils d'annotations. Le travail présenté dans [Guissé et al., 2012] traite le problème de la normalisation des règles métiers et leur transformation de langage naturel en langage contrôlé. La structure de données est encodée en RDF, et les liens d'annotation attachés aux unités textuelles des documents (utilisant RDFa⁶⁷) font référence à des ressources qui sont ou bien des entités OWL ou bien des règles candidates.

Une fois publiées sous forme de triplets RDF, les annotations sont interrogeables par des requêtes SPARQL, même si une phase de transformation est nécessaire quand la requête est formulée en langage naturel. Un système de questions réponses basé sur des patrons de requêtes (utilisés par exemple dans [Pradel et al., 2012]) a été proposé comme solution intuitive et expressive au problème d'accès aux données liées publiées en RDF [Unger et al., 2012].

4.5.2 Ontologies documentaires

Au-delà de la modélisation du contenu, des ontologies ont été produites pour modéliser les propriétés documentaires. Elles s'inspirent naturellement des langages de métadonnées définis dans la tradition des documentalistes, comme le Dublin Core. Ces ontologies sont souvent conçues pour des usages particuliers. Dans [Bouzidi et al., 2011] par exemple, la modélisation doit aider la rédaction des documents réglementaires dans le domaine du bâtiment. Ces ontologies mettent l'accent sur différents types de propriétés documentaires.

L'ontologie SDO (*SALT Document Ontology*⁶⁸) décrit la structure d'une publication scientifique, ainsi que ses propriétés identificatoires et les différentes révisions qu'elle comporte. L'ontologie d'annotation SAO (*SALT Annotation Ontology*⁶⁹) permet de lui associer une couche d'annotation sur le contenu en lien avec des ontologies existantes, telles que FOAF, SWRC et l'ontologie bibliographique BIBO. Cette dernière (*Bibliographic Ontology*⁷⁰) décrit en RDF des entités bibliographiques pour le web sémantique.

D'autres ontologies mettent l'accent sur le cycle de vie du document. L'ontologie PDO (*Project Documents Ontology*⁷¹) modélise la structure des documents de projets, en rendant compte de leurs différents statuts (rapports d'étape, rapports finaux, livrables, etc.). De la même manière, dans le domaine juridique, l'ontologie MetaLex [Boer et al., 2002] prend en compte le statut du document (par ex. document de travail) et les relations qu'ils entretiennent `resultOf`,

67. <http://www.w3.org/TR/rdfa-syntax/>

68. <http://salt.semanticauthoring.org/ontologies/sdo>

69. <http://salt.semanticauthoring.org/ontologies/sao>

70. <http://uri.gbv.de/ontology/bibo/>

71. <http://vocab.deri.ie/pdo-Document>

generatedBy, etc.). Le modèle FRBR, sur lequel se base Metalex, propose de distinguer le document en tant qu'oeuvre (*Work*) et les différentes versions qui sont publiées (*Expressions*).

Une ontologie pour les cas juridiques est présentée dans [Wyner and Hoekstra, 2012]. L'ontologie décrit la connaissance du domaine traité, permet le raisonnement sur ce domaine, et peut être utilisée pour annoter les textes qui peuvent à leur tour être utilisés pour peupler l'ontologie. En plus des éléments pour annoter les cas (par ex. les parties, la juridiction et la date), l'ontologie contient des éléments nécessaires pour l'élaboration de décisions comme par exemple des schemas d'arguments.

Un système qui utilise une ontologie OWL pour représenter la structure de l'administration publique et tout type de document qui circule entre les unités administratives, au cours de l'exécution des procédures, est décrit dans [Savvas and Bassiliades, 2009]. Le système adopte une approche orientée processus (unique pour chaque procédure juridique) afin d'aider les organisations publiques produisant chaque jour un grand volume de documents administratifs.

Dans ce travail, nous proposons une ontologie documentaire (dans le chapitre 7) qui intègre les différents types de propriétés (sémantiques, structurelles et temporelles) dans un même modèle. Elle permet aussi de rendre compte de la dimension intertextuelle qui est peu représentée dans les ontologies documentaires existantes. Une fois peuplée, l'ontologie sert de base pour une recherche d'information intégrant ces différents aspects dans les collections documentaires modélisées.

4.6 Synthèse

Les données inter-reliées peuvent être traitées par différentes approches selon leurs natures et les caractéristiques qu'elles présentent. Les données que nous traitons sont des documents qui se présentent sous forme de collections documentaires. Les documents possèdent des propriétés (attributs) et entretiennent des relations. L'analyse formelle et relationnelle de concepts et les techniques du web sémantique sont deux approches différentes mais complémentaires qui permettent d'analyser et d'interroger ce type de données. Les détails des deux approches proposées basées sur ces deux formalismes sont donnés dans les chapitres suivants.

Parallèle entre l'approche conceptuelle et l'approche sémantique Une correspondance entre l'analyse relationnelle de concepts et la logique de description est donné dans [Rouane et al., 2007]. Nous utilisons une partie de cette correspondance que nous modifions légèrement pour correspondre aux besoins de notre application dans le cadre de ce travail. Le tableau 4.6 décrit le parallèle entre une première modélisation basée sur l'AFC et l'ARC et une modélisation, plus riche et allant plus dans les détails des données modélisées, basée sur les langages du web sémantique (RDF et OWL).

Reprenons l'exemple des données décrivant les utilisateurs d'un réseau social avec leurs meilleurs films. Ces données peuvent être modélisées de deux façons différentes selon l'approche choisie. Avec une approche conceptuelle basée sur l'AFC et l'ARC, ces données sont représentées par une famille de contextes relationnels décrivant les objets, leurs attributs et leurs relations (voir section 4.2). Ce même ensemble de données peut être modélisé en suivant une approche sémantique utilisant RDF et OWL (voir section 4.4). La correspondance entre ces deux modélisations est donnée par les relations d'équivalence suivantes :

- Contexte Personne
- Contexte \equiv classe Personne.
- Les objets \equiv instances de la classe Personne.

TABLE 4.6 – Mapping FCA/RCA vers OWL DL.

FCA/RCA	OWL
Objets	Ressources (Instances)
Attributs	Classes (C_1) Types de données
Relation d'incidence	Propriété <code>rdf:type</code> Propriété d'objets (Object property) Propriété de données (Datatype property)
Contextes	Classes (C_2)
Relations FCR (entre contextes)	Propriétés d'objets (Object properties)

- Les attributs \equiv classes Âge ($\text{âge} < 18$, $18 < \text{âge} < 30$, $\text{âge} > 30$) et Pays (EU, UK, US, AU).
Modélisation possible aussi avec des attributs : âge (de type entier) et pays (de type chaîne de caractère).
 - La relation d'incidence \equiv selon le choix de modélisation des attributs : propriété d'objets (âge , habite-à) ou propriété de données.
 - Contexte Film
 - Contexte \equiv classe Film
 - Les objets \equiv instances de la classe Film
 - Les attributs \equiv sous-classes de Film (une classe par type de films)
 - La relation d'incidence \equiv propriété `rdf:type` entre Film et ses sous-classes
 - Relation "Like" \equiv propriété d'objet entre la classe Personne (domaine) et la classe Film (co-domaine)
 - Relation Ami \equiv propriété d'objet sur la classe Personne (même domaine et co-domaine)
- Un extrait du graphe de l'ontologie est donné par la figure 4.11.

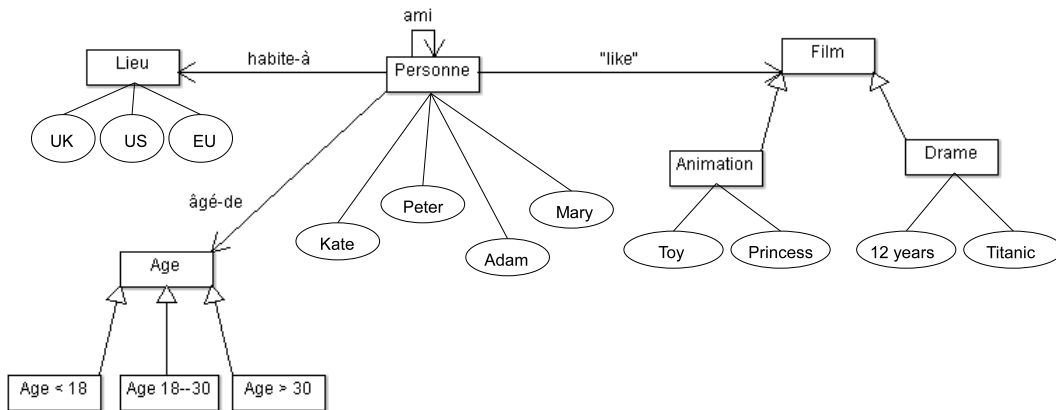


FIGURE 4.11 – Ontologie correspondant aux données (Personne, Film).

Comparaison / Complémentarité des deux approches Les langages du web sémantique sont plus expressifs que l'AFC tant sur le modèle de représentation de données (RDFS / contextes formels) que sur le langage d'interrogation (SPARQL / ensemble d'attributs). L'utilisation de

l'ARC, extension relationnelle de l'AFC, permet de se rapprocher en expressivité des langages du web sémantique mais ne couvre pas (en tout cas pas de manière directe et facile) tout le pouvoir expressif de SPARQL [Ferré, 2010]. D'un autre côté, les bases de données exprimées en RDF ne permettent pas d'avoir suffisamment de retours sur le graphe formé par ces données et donc ne permettent pas de faire une recherche exploratoire dans ce graphe.

L'AFC a été appliquée à divers domaines du web sémantique. Les travaux en relation avec la construction d'ontologies (cités dans la section 4.3) en forment la majeure partie. L'AFC a été aussi utilisée comme base pour une mesure de similarité de concepts pour le web sémantique [Formica, 2008] et pour extraire des questions représentatives sur un ensemble de données RDF [d'Aquin and Motta, 2011]. Dans [Ferré, 2010], l'auteur combine l'utilisation de l'AFC, notamment son extension logique l'ACL (Analyse de Concepts Logique) avec les langages du web sémantique pour proposer une méthode de navigation dans des graphes RDF avec des requêtes qui ressemblent à SPARQL mais qui sont exprimées dans un langage plus proche du langage naturel. Dans [Chekol and Napoli, 2013], les auteurs décrivent un cadre pour la structuration et la découverte de connaissances avec les treillis de concepts dans les résultats de requêtes SPARQL. L'AFC et l'ARC sont utilisées en tant que techniques de fouille de données dans [Shi et al., 2011] pour guider un processus d'amélioration de structure de wikis sémantiques. Une approche pour ajouter une couche de conceptualisation aux données du web utilisant les treillis de concepts a été décrite dans [Kirchberg et al., 2012].

La table 4.7 synthétise l'ensemble des remarques précédentes sur la comparaison des deux approches basées sur l'analyse formelle et relationnelle de concepts et sur les langages du web sémantique selon différents critères et montre leurs complémentarité.

TABLE 4.7 – Tableau comparatif RDF/SPARQL vs AFC/ARC.

	RDF/SPARQL	AFC/ARC	Commentaires
Expressivité	Algèbre relationnelle	Conjonction / Disjonction	SPARQL est plus expressif du fait qu'il est équivalent à l'algèbre relationnelle.
Cycles	Dans les chemins de propriétés des requêtes	Relations de même domaine et co-domaine	Les deux formalismes permettent d'exprimer des cycles.
Requêtes	Vocabulaire des données	Ensemble d'attributs et relations	Pas facile d'exprimer des requêtes en SPARQL sans maîtriser le vocabulaire utilisé pour représenter les données. L'utilisation de formulaires pour AFC/ARC peut aider à la formulation de requêtes.
Navigation	Graphe de données pas affiché	Structure de treillis	La recherche exploratoire n'est pas possible avec RDF/SPARQL. Si la requête ne retourne pas de résultats, il n'est pas possible d'aller directement explorer le voisinage pour retourner une réponse (même approximative) à l'utilisateur sans formuler une nouvelle requête. Un point en faveur des structures conceptuelles c'est qu'elles offrent un espace de navigation structuré et en deux niveaux : groupes de documents dans une même classe et hiérarchie de classes.
Passage à l'échelle	Grande quantité de documents	Nombre de documents limité	Les technologies du web sémantique permettent de manipuler des corpus de grande taille (tirés du web). L'AFC/ARC sont plus adaptées à des corpus spécifiques (petite taille).

Chapitre 5

Interrogation d'un réseau sémantique de documents : application aux sources de droit

Sommaire

5.1	Introduction	79
5.2	L'enjeu de l'intertextualité dans Légilocal	80
5.2.1	Objectif de la thèse	80
5.2.2	Intertextualité dans les sources de droit	81
5.3	Modélisation des collections documentaires	83
5.3.1	Caractéristiques des collections documentaires	83
5.3.2	Les collections comme graphes de documents	83
5.3.3	Exemples de collections juridiques	84
5.4	Interrogation des collections documentaires	88
5.4.1	Langage de requêtes	89
5.4.2	Exemples	90
5.4.3	Analyse des besoins des juristes	91
5.4.4	Jeu de requêtes types	97
5.4.5	Discussion	99
5.5	Conclusion	100

5.1 Introduction

Dans ce chapitre, nous restituons notre problématique dans son contexte applicatif pour analyser les besoins des juristes en matière d'interrogation d'un réseau de documents. Nous présentons les collections de documents et les types de liens sur lesquels nous avons travaillé dans cette thèse. Nous donnons des exemples de requêtes que nous avons collectées auprès de nos partenaires juristes, ce qui nous permet d'identifier un jeu de requêtes types à traiter dans la suite de ce travail.

5.2 L'enjeu de l'intertextualité dans Légilocal

5.2.1 Objectif de la thèse

Nous avons vu dans le chapitre 3 des modèles avancés de RI qui sont proposés pour gérer la recherche sémantique (qui exploite la hiérarchie sémantique de descripteurs de contenu) mais que l'information intertextuelle n'est pas prise en compte par ces modèles. Cette information a été exploitée pour le classement des résultats [Page et al., 1999] ou pour l'analyse des grands graphes (navigation, clustering) mais pas comme un critère de recherche en tant que tel. Nous nous focalisons dans ce travail sur l'interrogation directe des liens (en considérant un lien comme un critère de recherche), une question encore peu explorée par les modèles de RI existants. Nous proposons un modèle de recherche d'information alternatif, centré sur la collection plutôt que sur le document, qui permet d'aborder la complexité des réseaux sémantiques de documents. Ce modèle permet le traitement de la dimension intertextuelle en définissant des requêtes qui portent à la fois sur le contenu sémantique et sur les liens, les requêtes relationnelles, auxquelles on peut répondre par une liste de documents mais aussi par des graphes de documents liés par des relations intertextuelles.

La modélisation de l'intertextualité est particulièrement complexe dans le domaine juridique : elle touche au coeur même de l'activité juridique qui consiste à publier des documents (décision ou jugements, modifications de textes législatifs) qui s'appuient sur des textes existants pour en créer d'autres qui modifient les premiers, s'en justifient, les prolongent dans un contexte différent, les confirment ou les contredisent, etc. Le modèle de recherche d'information proposé est appliqué au domaine juridique qui se caractérise par l'abondance et la diversité des liens entre les documents. Les requêtes recensées à travers divers entretiens avec des juristes montrent l'utilité pour les utilisateurs d'exploiter la complexité des sources juridiques en combinant des critères sémantiques et intertextuels⁷².

Dans certains cas, par exemple lorsqu'un article de code est associé à un concept juridique bien spécifique, utiliser les liens entre les documents pour effectuer la recherche permet d'avoir des réponses plus complètes que d'utiliser les requêtes sémantiques. C'est le cas par exemple de l'article 1382 du code pénal qui parle de la « responsabilité pour faute », si nous cherchons la jurisprudence qui cite cet article, nous trouvons les textes qui parlent de ce concept et de tout son champ sémantique qui peut comprendre plusieurs termes. Cette recherche est plus large que de chercher juste la jurisprudence annotée avec le terme « responsabilité pour faute ».

Selon le public visé, citoyens (simples utilisateurs) ou juristes (experts du domaine), différents types d'applications peuvent être proposés avec des niveaux variables de complexité.

Cas d'usage juridique généraliste Dans le cas d'un simple utilisateur la complexité ne doit pas dépasser celle d'un système d'accès à l'information juridique généraliste (comme dans le cas de Legifrance) avec en plus l'aspect relationnel sur lequel l'utilisateur peut poser des requêtes.

Cas d'usage juridique métier Dans le cas d'un utilisateur averti, plus de fonctionnalités peuvent être proposées et elles peuvent être intégrées dans des dispositifs plus complexes dédiés métier. Dans ce deuxième cas, la complexité des fonctionnalités vis-à-vis de l'utilisateur est filtrée par son domaine de travail. Par exemple :

72. Ce chapitre repose en grande partie sur l'analyse des besoins faite dans le cadre du projet Légilocal. Elle a bénéficié des analyses de Meritxell Fernandez-Barrera et Eve Paul (juristes, partenaires du projet) et a été conduite par Sylvie Salotti et Adeline Nazarenko.

- un agent de mairie qui veut rédiger un document aura besoin de voir comment les agents des mairies voisines ont traité des documents similaires, de savoir quels sont les visas à apposer sur ce document, etc.,
- un législateur, qui a pour métier la création de textes de loi, a des besoins spécifiques différents de ceux d'un agent de mairie.

Dans ce travail, nous proposons des solutions pour de simples utilisateurs et pour les agents de mairies, ce qui correspond aux cas d'usages du projet Légilocal.

Nous ne cherchons pas à construire un système de RI complet, mais plutôt à explorer et à tester la faisabilité et l'intérêt de la prise en compte de l'intertextualité dans un système d'accès à l'information juridique. Nous souhaitons pouvoir interroger une collection de documents pour retrouver des documents décrits par des descripteurs sémantiques et/ou des types de documents ou des graphes de documents liés par des relations intertextuelles.

Comme ce problème est assez nouveau et que nous n'avons pas traité la question de point de vue système réel, nous supposons que des annotations existent sur les documents, que les requêtes à l'entrée du système sont sous forme logique et correspondent à la description de la collection modélisée (nous faisons comme si les utilisateurs étaient capables de créer directement des requêtes logiques). Nous ne définissons pas les interfaces utilisateur de saisie (pour poser des requêtes) et de présentation de résultats (pour analyser les résultats).

Ce chapitre décrit la collection de documents (section 5.3) et le langage de requêtes (section 5.4), après avoir introduit le problème d'intertextualité dans le domaine juridique (section 5.2.2). Ensuite, nous analysons les requêtes recueillies de la part des utilisateurs juristes interviewés et nous dressons la liste des types de requêtes qu'il paraît important de traiter dans un système de recherche d'information juridique (section 5.4.4).

5.2.2 Intertextualité dans les sources de droit

Les systèmes d'accès à l'information juridique existants ne proposent pas de solutions directes pour prendre en compte les liens dans les requêtes. Ils proposent néanmoins de contourner cette difficulté avec des techniques simples comme, par exemple, modéliser les liens comme des attributs qui peuvent être interrogés au même niveau que les types de documents.

Il existe plusieurs types de documents qui sont accessibles par les systèmes juridiques. Ils sont regroupés sous de grandes catégories, par exemple :

- les lois et règlements (textes législatifs) : constitution, codes, loi, décrets, etc. ;
- la jurisprudence : constitutionnelle, administrative et judiciaire ;
- les conventions collectives.

Les actes locaux et les documents éditoriaux sont d'autres types de documents qui ne sont pas traités par les systèmes existants et qui sont visés par le projet Légilocal.

Ces documents sont reliés entre eux par différents types de liens et ces liens prennent souvent des types spécifiques de documents comme domaine et co-domaine, par exemple :

- Codification : entre un texte ou article non codifié et un texte ou article codifié.
- Transposition : entre une directive européenne et un texte national.
- Lien de jurisprudence : entre une jurisprudence et un texte législatif.
- Interprétation : entre un arrêté local et un décret ou une loi.
- Application : entre un décret et une loi.
- Modification (ajout, substitution, etc.) : entre tous types de textes.
- Abrogation.

- Citation.

D'autres relations mériteraient d'être prises en compte. N'étant pas juriste, nous ne prétendons pas lister ici toutes les relations importantes à prendre en compte ni leur donner une définition définitive, d'autant que d'une tradition juridique à l'autre, les pratiques et les définitions varient. Nous nous contentons de lister les relations intertextuelles dont les juristes avec qui nous avons travaillé ont souligné l'importance à travers l'analyse des cas d'usage que nous avons faite et les exemples de requêtes qu'ils nous ont proposées. Par exemple :

- *Quelle est la jurisprudence qui applique-interprète l'article sur la responsabilité pour faute du code civil ?*
- *Quelles conventions implémentent les recommandations qui parlent de licenciement ?*
- *Quelle est la version en vigueur de l'article 1382 du code civil et sa version précédente ?*
- *Est-ce que la loi n 2014-567 du 2 juin 2014 relative à l'interdiction de la mise en culture des variétés de maïs génétiquement modifié a été appliquée (donner des exemples de décrets d'application) ? quel acte local l'applique dans ma commune ?*

Dans ces requêtes, nous remarquons l'intégration de plusieurs caractéristiques de documents en une seule requête : les documents sont décrits par leurs types, par des descripteurs sémantiques de contenu, par des éléments de structure et aussi par les relations intertextuelles qu'ils entretiennent entre eux. Cela impose de modéliser une collection documentaire comme un réseau sémantique :

- en modélisant les types et structures des documents,
- en affinant la typologie des liens,
- en modélisant les liens comme des relations exploitables pour la recherche d'information et pas seulement comme des attributs.

Les documents juridiques possèdent aussi une structure riche comme déjà exposé dans le chapitre 2. Cette structure doit également être prise en compte au moment de la modélisation de la collection. Elle présente un des critères sur lesquels portent les requêtes des experts dans ce domaine, il faut donc la prendre en compte dans un processus de RI sur une collection juridique. Par exemple, une modification d'une loi L peut ne concerner qu'un article A de cette loi et non pas le texte intégral. Dans ce cas, des liens intertextuels de modification ou de citation partent du texte T qui introduit la modification vers l'article A de la loi L. Ainsi, une requête qui porte sur les modifications apportées au texte de la loi L, doit avoir comme réponse l'article modifié A en relation avec le texte T. Ceci nous impose une granularité fine dans la description de la structure des documents.

Un pré-traitement sur les documents est obligatoire afin d'en extraire leur contenu sémantique et les références vers d'autres documents (voir figure 5.1). Le contenu sémantique de documents est représenté comme un ensemble d'annotations sémantiques par rapport à une ressource sémantique (comme détaillé dans le chapitre 3). Les références sont identifiées suite à un processus de résolution de références. La structure des documents est analysée (en s'appuyant sur un standard juridique par exemple) et un identifiant unique est affectée à chaque document ⁷³.

⁷³. Cette étape de pré-traitement est au-delà de la portée de ce travail, elle doit être faite par les partenaires du projet

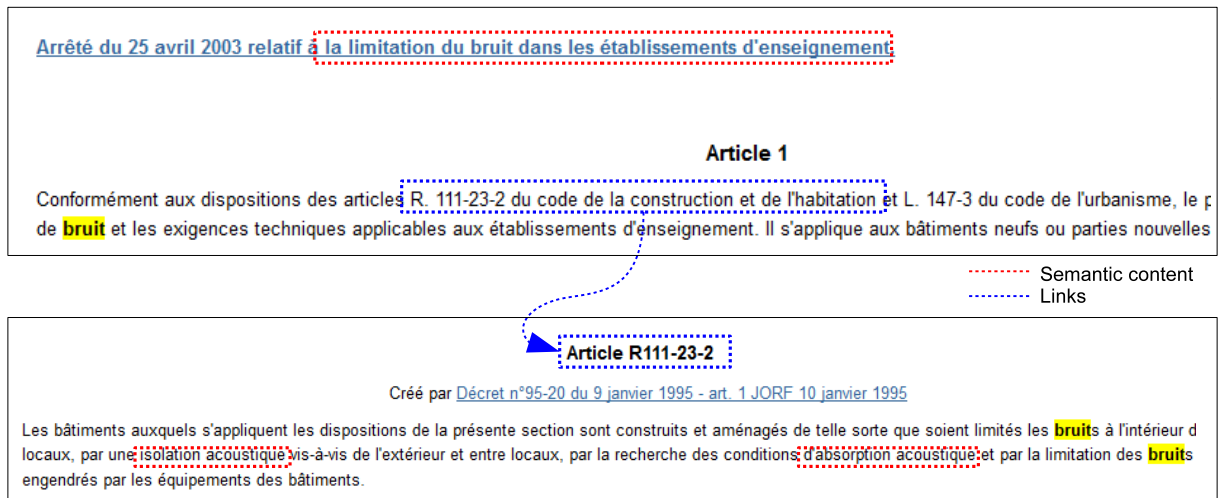


FIGURE 5.1 – Arrêté du 25 Avril 2003 relatif à la limitation du bruit dans les établissements d'enseignement citant l'article R111-23-2 du Code de la construction et de l'habitation.

5.3 Modélisation des collections documentaires

5.3.1 Caractéristiques des collections documentaires

À partir de la description donnée dans la section précédente, nous posons qu'une collection de documents juridiques est caractérisée par l'ensemble des propriétés suivantes :

- une collection est formée d'un ensemble de documents ou, plus généralement, d'unités documentaires ;
- une unité documentaire possède un identifiant unique ;
- toute unité documentaire est typée (*loi*, *code*, *article de loi*, *décret*, etc.) de manière unique : on suppose qu'une unité documentaire ne peut pas relever de deux types à la fois (être une loi et un décret, par ex.) ; en cas d'ambiguïté ou d'indétermination, on suppose qu'on peut caractériser l'unité documentaire par un type plus générique (*loi_ou_décret*) ;
- un ou plusieurs descripteurs sémantiques peuvent être associés à une unité documentaire ;
- les unités documentaires peuvent être liées entre elles par différents types de relations intertextuelles (*appartenance* pour entre deux unités documentaires dont l'une est un fragment de l'autre, *jurisprudence*, etc.) ;
- les types sémantiques et les relations intertextuelles peuvent être structurées en hiérarchie, un type ou une relation étant plus général(e) qu'un(e) autre, mais cette propriété n'est pas prise en compte dans la modélisation qui suit.

5.3.2 Les collections comme graphes de documents

À partir de cette analyse, on peut modéliser une collection documentaire C comme un graphe orienté, étiqueté et attribué $C = \mathcal{G}(D, R, A)$ où

- les noeuds sont des unités documentaires ($d_u \in D$) ;
- les unités documentaires sont décrites par des attributs : $Att(d_u, a_i)$ indique que l'unité documentaire d_u ($d_u \in D$) possède l'attribut a_i ($a_i \in A$) ;
- les arcs sont des relations typées et orientées : $Rel(d_u, r_j, d_v)$ indique que l'unité documentaire d_u ($d_u \in D$) est la source d'une relation r_j ($r_j \in R$) dont la cible est d_v ($d_v \in D$).

$$\begin{aligned}
 \text{graphe}_{coll} &\leftarrow \text{prédictat}_c [\wedge \text{prédictat}_c]^* \\
 \text{prédictat}_c &\leftarrow \text{'Att' } (\text{'id}_{doc}\text{'}, \text{'id}_{att}\text{'}) \mid \text{'Rel' } (\text{'id}_{doc}\text{'}, \text{'id}_{rel}\text{'}, \text{'id}_{doc}\text{'}) \\
 \text{id}_{doc} &\leftarrow \text{'d}_1\text{' } \mid \text{'d}_2\text{' } \mid \text{'d}_3\text{' } \mid \dots \\
 \text{id}_{att} &\leftarrow \text{'a}_1\text{' } \mid \text{'a}_2\text{' } \mid \text{'a}_3\text{' } \mid \dots \\
 \text{id}_{rel} &\leftarrow \text{'r}_1\text{' } \mid \text{'r}_2\text{' } \mid \text{'r}_3\text{' } \mid \dots \\
 &\text{où } (\forall i, j, k,) (d_i \in D, a_j \in A \text{ et } r_k \in R).
 \end{aligned}$$

FIGURE 5.2 – Langage de graphes : description des graphes de collections documentaires. Les éléments du vocabulaire terminal sont notés entre guillemets simples (ex. ‘(’), les non-terminaux sont en italiques (ex. *prédictat*) et les métasymboles utilisés sont la flèche de réécriture (\leftarrow), les crochets pour former les groupes ([]), la barre d’alternative (|) et l’étoile de Kleene pour marquer la répétition de l’élément ou du groupe précédent pour un nombre quelconque d’occurrences (*).

Un graphe de collection documentaire est donc décrit par une formule du langage dont la grammaire est présentée dans la figure 5.2.

Cette modélisation est naturellement simplificatrice et il peut être nécessaire, pour un domaine particulier, de tenir compte de contraintes supplémentaires : pour le domaine juridique, en particulier, il paraît raisonnable par exemple de distinguer deux types d’attributs, les types de documents $t_k \in T$ et les descripteurs sémantiques $s_l \in S$ ($A = T \cup S$) pour exprimer les contraintes supplémentaires suivantes :

- une unité documentaire possède un et un seul type : $(\forall d_u)((\exists t_k, t_l \in T)(Att(d_u, t_k) \wedge Att(d_u, t_l) \Rightarrow t_k = t_l)$;
- une unité documentaire est décrite par un nombre quelconque de descripteurs sémantiques.

En revanche, il n’y a aucune contrainte sur le nombre de noeuds, d’attributs et de relations entrant dans le graphe ou sur les combinaisons d’attributs et de relations.

La figure 5.3 donne un exemple de graphe qui peut être décrit par la formule suivante :

$$\begin{aligned}
 &Att(d_1, t_1) \wedge Att(d_1, s_1) \wedge Att(d_1, s_2) \\
 &\wedge Att(d_2, t_2) \wedge Att(d_2, s_1) \wedge Att(d_2, s_3) \wedge Att(d_2, s_4) \wedge Att(d_2, s_5) \\
 &\wedge Att(d_3, t_1) \wedge Att(d_3, s_2) \wedge Att(d_3, s_3) \wedge Att(d_3, s_4) \\
 &\wedge Att(d_4, t_2) \wedge Att(d_4, s_3) \wedge Att(d_4, s_5) \\
 &\wedge Rel(d_1, r_1, d_2) \wedge Rel(d_2, r_4, d_1) \wedge Rel(d_1, r_1, d_3) \wedge Rel(d_2, r_2, d_3) \\
 &\wedge Rel(d_2, r_3, d_4) \wedge Rel(d_3, r_2, d_4) \wedge Rel(d_4, r_5, d_4) \\
 &\text{où } (\forall i, j, k)(d_i \in D \wedge s_j \in S \wedge t_k \in T)
 \end{aligned}$$

5.3.3 Exemples de collections juridiques

La collection BRUIT

Décrivons tout d’abord la collection qui est représentée de manière schématique dans la figure 5.4. C’est une collection de petite taille rassemblant des documents juridiques traitant du bruit et des nuisances sonores, qui ont été collectés sur Legifrance et sur les sites officiels des mairies de certaines villes.

Ces documents sont de plusieurs types : des arrêtés locaux d’une part (en réalité, des arrêtés municipaux et préfectoraux) et des textes législatifs d’autres part (décrets, lois, codes ou

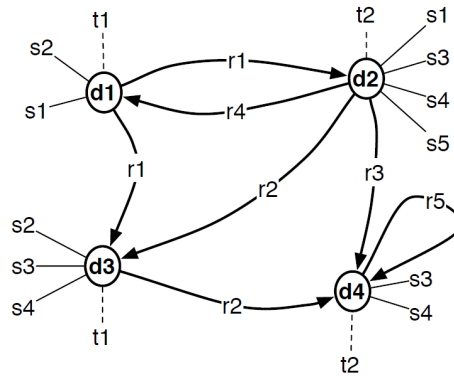


FIGURE 5.3 – Exemple de graphe modélisant une collection documentaire comportant 4 unités documentaires. Pour des questions de lisibilité les attributs et relations partagés par plusieurs documents sont représentés en double. Les unités documentaires sont représentées par des cercles. Les relations sont notées comme des flèches, les attributs sont reliés aux documents par des traits pleins (descripteurs sémantiques) ou pointillés (types de documents).

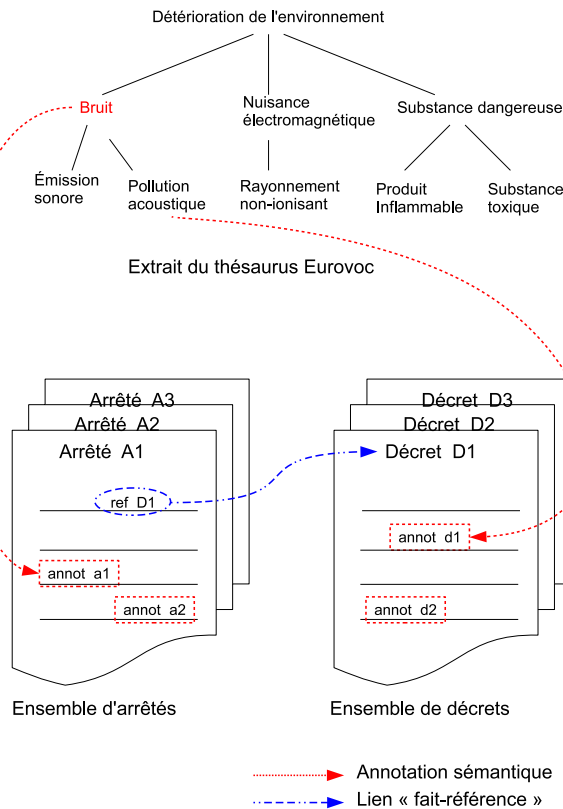


FIGURE 5.4 – Exemple de collection juridique avec annotations sémantiques et lien de référence.

TABLE 5.1 – Composition de la collection BRUIT

Arrêtés locaux			Textes législatifs		
Types	Descriptifs	Identifiants	Types	Descriptifs	Identifiants
Arrêté	Arrêté de Paris	AP	Décret	Décret de 1995	D95
Arrêté	Arrêté de Boulogne	AB	Loi	Loi de 1992	L92
Arrêté	Arrêté des Yvelines	AY	Ordonnance	Ordonnance de 1945	O45
Arrêté	Arrêté de Strasbourg	AS	Code	Code pénal	CPen

ordonnances)⁷⁴. Le tableau 5.1 donne la composition précise de la collection.

Les documents de la collection peuvent se citer les uns les autres. Nous représentons de manière indifférenciée ces citations par la relation **fait-référence**. Celle-ci prend respectivement des arrêtés et des textes législatifs comme sources et comme cibles. Ce sont en effet les actes locaux qui citent la législation nationale, non l'inverse.

Les documents de la collection sont en outre décrits par des descripteurs sémantiques dont le vocabulaire extrait du thésaurus juridique EuroVoc⁷⁵ est résumé dans le tableau 5.2.

TABLE 5.2 – Vocabulaire utilisé pour l'annotation sémantique de la collection BRUIT

Descripteurs	Equivalents terminologiques
bag	« bruit anormalement gênant »
ns	« nuisance sonore »
son	« sonorisation »
lcb	« lutte contre le bruit »
nvs	« niveau sonore »
tv	« tranquillité du voisinage »
ab	« activité bruyante »
ip	« isolation phonique »

Cette collection peut se décrire comme un graphe, selon la formule suivante ou le schéma de la figure 5.5 :

$$\begin{aligned}
 & Att(AP, arrete) \wedge Att(AP, bag) \wedge Att(AP, pa) \\
 & \wedge Att(AB, arrete) \wedge Att(AB, bag) \wedge Att(AB, ns) \wedge Att(AB, son) \\
 & \wedge Att(AY, arrete) \wedge Att(AY, bag) \wedge Att(AY, ns) \wedge Att(AY, nvs) \\
 & \wedge Att(AS, arrete) \wedge Att(AS, pa) \wedge Att(AS, nvs) \\
 & \wedge Att(D95, decret) \wedge Att(D95, lcb) \wedge Att(D95, ab) \\
 & \wedge Att(L92, loi) \wedge Att(L92, tv) \wedge Att(L92, ip) \\
 & \wedge Att(O45, ordonnance) \wedge Att(O45, tv) \wedge Att(O45, ab) \\
 & \wedge Att(CPen, code) \wedge Att(CPen, lcb) \wedge Att(CPen, ip) \\
 & \wedge Rel(AP, fait - reference, D95) \wedge Rel(AB, fait - reference, L92) \\
 & \wedge Rel(AY, fait - reference, CPen) \wedge Rel(AS, fait - reference, O45)
 \end{aligned}$$

74. Ce corpus servira d'exemple jouet pour la suite des chapitres.

75. <http://eurovoc.europa.eu/>

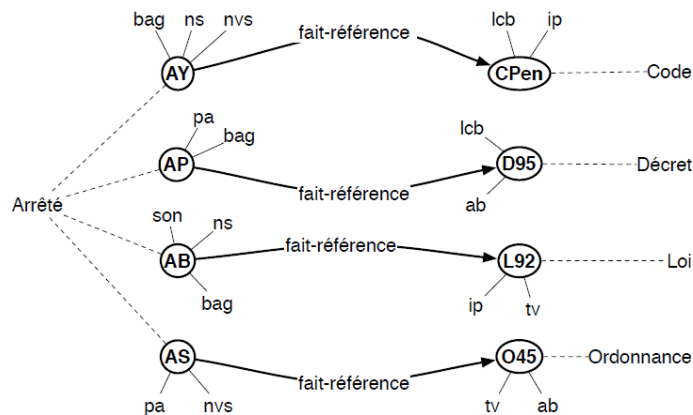


FIGURE 5.5 – Collection BRUIT. Pour des questions de lisibilité les descripteurs sémantiques partagés par plusieurs documents sont représentés en double. Les unités documentaires sont représentées par des cercles. Les relations sont notées comme des flèches. Les attributs sont reliés aux documents par des traits pleins (descripteurs sémantiques) ou pointillés (types de documents).

La collection LÉGILOCAL

Dans le cadre du projet Légilocal, nous avons travaillé sur une petite collection à des fins d'expérimentation et de test. La collection est de taille réduite mais elle est diversifiée. Les documents sont collectés à partir de plusieurs sources : il s'agit de décisions publiées par des collectivités locales, de document éditoriaux fournis par des éditeurs juridiques et ou de textes législatifs (lois, décrets, etc) issus de portails juridiques, principalement Legifrance. La collection est structurée par ailleurs par différents types de relations.

Le projet Légilocal a été conçu pour construire cette base documentaire qui va être étendue au fur et à mesure que de nouvelles collectivités locales souhaiteront mettre en commun leurs actes et que la société Victoires Editions enrichit le réseau Légilocal avec de nouveaux documents éditoriaux (modèles de documents, guides de bonnes pratiques, etc.) et le connecte à d'autres bases documentaires (ex. Legifrance).

Nous présentons ci-dessous les principales relations qui structurent la collection LÉGILOCAL. Il faut d'abord noter que dans le domaine juridique, on raisonne souvent sur plusieurs versions d'un même document. On distingue donc deux types de documents : les versions qui sont publiées et les documents-matrices dont elle dépend ; nous distinguons les *expressions* et les *oeuvres*⁷⁶ pour différencier ces deux types de documents. Sauf mention contraire les différentes relations ci-dessous relient entre eux des documents de type oeuvre.

Appliquer La relation d'application est une relation structurante des collections juridiques. Elle exprime différents types de relations selon le type des arguments qu'elle relie :

- un texte législatif peut *appliquer* un autre texte législatif issu d'une norme supérieure dans la hiérarchie des normes : dans ce cas le texte qui applique la loi ou le décret en explicite les modalités d'application ; cette relation d'application peut aussi relier un texte d'une juridiction locale à un texte d'une juridiction plus globale ;
- quand une décision (ex. arrêté, jugement, etc.) *applique* un texte législatif ou une autre décision, c'est qu'il fait référence à ce document source pour justifier la décision qu'il

⁷⁶. Cette terminologie est empruntée de la description de FRBR donnée par la BnF (Bibliothèque nationale de France) : http://www.bnf.fr/fr/professionnels/modelisation_ontologies/a.modele_FRBR.html .

prend ; la décision interprète alors le texte cité au regard d'un cas particulier ou d'une situation locale et elle est souvent utilisée par la suite comme jurisprudence pour d'autres cas ou situations similaires.

Composer Un document est généralement *composé de* différentes unités documentaires : nous distinguons notamment les différents articles qui composent un document juridique.

Statuer (Confirmer ou Annuler) Certaines décisions (par exemple les arrêts de cour d'appel ou de cassation) statuent sur la légalité ou l'acceptabilité d'autres textes, des textes législatifs ou des décisions dites « antérieures ». Plusieurs cas de figures peuvent se présenter selon que la décision postérieure *confirme* la décision antérieure ou l'*annule* et selon qu'elle porte sur la totalité de la décision antérieure ou seulement une sous-partie de celle-ci.

Modifier Les articles des textes juridiques font souvent l'objet de modifications successives tant qu'ils ne sont pas codifiés. La trace de ces modifications est généralement consignée dans le nouveau document qui cite le texte qu'il *modifie*.

Codifier Les articles de code citent également les articles de lois à partir desquels ils sont créés par *codification* d'articles et qui ne peuvent dès lors plus faire l'objet de modification.

Abroger Un texte juridique peut aussi être *abrogé* par un autre⁷⁷.

Exprimer On dit qu'un document *s'exprime* dans ses différentes versions ou que les documents de types expressions *expriment* le document-oeuvre qui en est la source.

La collection OIT

Nous utilisons également une collection de documents de l'Organisation Internationale du Travail (OIT)⁷⁸. Il s'agit d'une collection de plus grande taille mais qui ne comporte que deux types de documents (188 conventions et 199 recommandations) et un seul type de relation, les conventions *implémentant* les recommandations⁷⁹.

Autres collections

D'autres bases de données juridiques dans d'autres pays peuvent également être utilisés avec notre modélisation. Plusieurs données et documents sont disponibles en ligne tels que :

- Les données de l'initiative *UK Opening Up Government*⁸⁰.
- Les données de l'initiative *Dutch Regulations as Linked Data*⁸¹.

5.4 Interrogation des collections documentaires

Les collections étant modélisées comme des graphes, la recherche documentaire s'apparente à l'interrogation de graphes : les requêtes et les réponses qui y sont apportées se modélisent également sous le forme de graphes.

Nous posons qu'une requête s'exprime comme un graphe du même type qu'une collection mais il peut comporter

- des éléments variables à la place des identifiants de documents ou de relations ;
- des contraintes d'inégalité ou de type sur ces variables ;

77. A noter qu'un texte juridique reste en vigueur tant qu'il n'est pas modifié par un autre document ou abrogé.

78. www.ilo.org

79. Nous remercions Thibault Mondary qui nous a aidé à la construction ce corpus.

80. <http://data.gov.uk/data/search>

81. <http://doc.metalex.eu/>

```

'graphereq' ← [cible ':' ]? grapher ['avec' contrainte ['∧' contrainte ]*]?
cible ← '(' variable [',' variable ]* ')'
grapher ← prédicatr ['∧' prédicatr]*
prédicatr ← 'Att' '(' document ',' attributtype ')' | 'Att' '(' document ',' attributsem ')' | 'Rel'
 '(' document ',' relation ',' document ')'
document ← iddoc | variable
attributsem ← idsem | variable
attributtype ← idtype | variable
relation ← idrel | variable
iddoc ← 'd1' | 'd2' | 'd3' | ...
idsem ← 's1' | 's2' | 's3' | ...
idtype ← 't1' | 't2' | 't3' | ...
idrel ← 'r1' | 'r2' | 'r3' | ...
contrainte ← variable '≠' variable | variable '∈' ensemble
ensemble ← 'D' | 'A' | 'S' | 'T' | 'R'
où
variable ∈ D ∪ C ∪ T ∪ R
et (∀ i, j, k, l) (di ∈ D ∧ sj ∈ S ∧ tk ∈ T ∧ rl ∈ R)

```

FIGURE 5.6 – Langage de requêtes. Les éléments du vocabulaire terminal sont notés entre guillemets simples (ex. '('), les non-terminaux sont en italiques (ex. *prédicat*) et les métasymboles utilisés sont la flèche de réécriture (\leftarrow), les crochets pour former les groupes ([]), la barre d'alternative (|) et l'étoile de Kleene pour marquer la répétition de l'élément ou du groupe précédent pour un nombre quelconque d'occurrences (*).

- une cible qui permet de focaliser la requête sur certains éléments, la cible étant un sous-ensemble des éléments variables de la requête.

Les réponses retournées sont également des graphes.

5.4.1 Langage de requêtes

Un graphe requête est donc décrit par une formule du langage donné par la grammaire de la figure 5.6.

Répondre à un graphe requête revient à chercher à l'instancier sur une collection. La requête n'est satisfiable que si le graphe requête est instanciable sur la collection :

- si le graphe requête n'est pas instanciable, la requête retourne un graphe vide ;
- si le graphe requête comporte des éléments variables, le résultat de la requête est donné par l'ensemble des sous-graphes de la collection instanciant le graphe requête, ou l'ensemble des n-uplets instanciant la cible de la requête et vérifiant les propriétés exprimées par l'ensemble du graphe requête, si ce dernier comporte une cible ;
- si le graphe requête ne comporte aucun élément variable, le résultat de la requête est booléen : le graphe requête est retourné si c'est un sous-graphe du graphe de la collection.

5.4.2 Exemples

A titre d'illustration, considérons la liste des requêtes suivantes et les réponses obtenues en projetant ces requêtes sur le graphe de la collection exemple de la figure 5.3 :

Requêtes non ciblées

1. $Att(x, s_1)$
Traduction : *Trouver tous les sous-graphes composés d'un document décrit par s_1 .*
Résultat : $\{d_1, d_2\}$ (2 graphes réduits à un seul document)
2. $Rel(x, r_1, d_2)$
Traduction : *Trouver tous les sous-graphes composés d'un document ayant pour cible d_2 par la relation r_1 .*
Résultat : $\{Rel(d_1, r_1, d_2)\}$ (1 graphe)
3. $Rel(d_1, r_1, d_2)$
Traduction : *Y a-t-il une relation r_1 entre les documents d_1 et d_2 ?*
Résultat : $\{Rel(d_1, r_1, d_2)\}$ (le graphe requête, ce qui équivaut à VRAI)
4. $Rel(x, y, x)$
Traduction : *Trouver tous les sous-graphes composés d'un document en relation avec lui-même quel que soit le type de la relation .*
Résultat : $\{Rel(d_4, r_5, d_4)\}$ (1 graphe)
5. $Att(x, s_1) \wedge Rel(x, r_1, d_2)$
Traduction : *Trouver tous les sous-graphes composés d'un document décrit par s_1 et ayant pour cible d_2 par la relation r_1 .*
Résultat : $\{Att(d_1, s_1) \wedge Rel(d_1, r_1, d_2)\}$ (1 graphe)
6. $Att(x, s_1) \wedge Rel(y, r_1, d_2)$
Traduction : *Trouver les sous-graphes composés de documents décrits par s_1 et de documents ayant pour cible d_2 par la relation r_1 .*
Résultat : $\{Att(d_1, s_1) \wedge Rel(d_1, r_1, d_2), Att(d_2, s_1) \wedge Rel(d_1, r_1, d_2)\}$ (2 graphes : cette requête comportant deux variables indépendantes x et y , il s'agit de deux requêtes indépendantes)

Requêtes ciblées

1. $(x) : Rel(x, y, x)$
Traduction : *Trouver tous les documents en relation avec eux-mêmes quel que soit le type de la relation.*
Résultat : $\{d_4\}$ (1 document)
2. $(y) : Rel(x, y, x)$
Traduction : *Trouver tous les types de relation liant un document à lui-même.*
Résultat : $\{r_5\}$ (1 relation)
3. $(x, y) : Rel(x, y, x)$
Traduction : *Trouver tous les couples composés d'un document lié à lui-même et du type de relation qui les lie.*
Résultat : $\{(d_4, r_5)\}$ (1 couple composé d'un document et d'une relation)
4. $(x, y) : Att(x, z) \wedge Rel(x, r_1, y) \wedge Att(y, z)$
Traduction : *Trouver tous les couples de documents décrits par un même descripteur sémantique et tels que le second est la cible du premier par la relation r_1 .*
Résultat : $\{(d_1, d_2), (d_1, d_3)\}$ (2 couples)

5. $(x, y, z) : Att(x, z) \wedge Rel(x, r_1, y) \wedge Att(y, z)$

Traduction : *Trouver tous les triplets composés de deux documents décrits par un même descripteur sémantique et tels que le second est la cible du premier par la relation r_1 et du descripteur sémantique que ces documents partagent.*

Résultat : $\{(d_1, d_2, s_1), (d_1, d_3, s_2)\}$ (2 triplets)

Requêtes avec contraintes

1. $Att(x, s_2) \wedge Att(x, y)$ avec $y \neq s_2 \wedge y \in S$

Traduction : *Trouver tous les sous-graphes composés d'un document décrit par s_2 et un autre descripteur sémantique différent.*

Résultat : $\{Att(d_1, s_1) \wedge Att(d_1, s_2), Att(d_3, s_2) \wedge Att(d_3, s_3), Att(d_3, s_2) \wedge Att(d_3, s_4)\}$
(3 graphes)

2. $Att(x, y) \wedge Att(x, z)$ avec $y \neq z \wedge y \in T \wedge z \in T$

Traduction : *Trouver tous les sous-graphes composés d'un document associé à deux types différents.*

Résultat : \emptyset (la requête n'est pas satisfiable)

3. $(x) : Rel(x, r_5, y)$ avec $x \neq y$

Traduction : *Trouver tous documents avant yn autre document pour cible par r_5 .*

Résultat : \emptyset (la requête n'est pas satisfiable)

5.4.3 Analyse des besoins des juristes

La formalisation ci-dessous

- ne fixe pas de limite à la taille des graphes requêtes, à la cible des requêtes ciblées ou au nombre de contraintes à prendre en compte,
- ne fixe aucune contrainte sur la structure des graphes requêtes : elle autorise notamment toute forme de cycles,
- n'impose aucune contrainte sur la combinaison d'attributs ou de relations pour un document.

La prise en compte des besoins des utilisateurs sur des domaines d'application particuliers et des collections particulières permet cependant de cerner le type de requêtes auxquelles il est important de pouvoir répondre.

Nous listons ci-dessous les requêtes que nous avons recueillies auprès des juristes que nous avons interrogés et nous montrons les réponses qu'il faudrait leur apporter à partir des collections documentaires présentées dans la section 5.3.3.

L'analyse qui suit montre que le langage de requête défini plus haut comporte certaines limitations :

1. Certains opérateurs logiques ne sont pas pris en compte. :
 - la quantification : les présupposés d'unicité⁸² ne sont pas exprimés dans la formalisation : les requêtes « Quelle(s) recommandation(s) implémente(nt) la/une convention ? » sont formalisées de la même manière ; les présupposés de non-univocité ne sont pas davantage exploités : les requêtes « Quelles sont les recommandations qui implémentent des/les/une convention(s) » sont considérées comme équivalentes ;
 - l'absence d'autres opérateurs logiques – la négation et la disjonction notamment – limite de fait l'expressivité du langage de requête ; nous avons conscience ici de simplifier le

⁸². (Comparer pa exemple « Quelle recommandation ... » et « Quelles recommandations... » ou « implémente la recommandation » et « implémente une recommandation »).

problème du traitement de l'intertextualité : nous considérons qu'en l'état actuel des systèmes de recherche d'information sémantique, le langage proposé ci-dessus répond à l'essentiel des besoins exprimés par les utilisateurs ; l'extension du langage de requêtes sera peut-être nécessaire à terme mais elle est laissée en perspective de ce travail.

2. Par ailleurs, certaines requêtes (par ex. de la forme « Quelles sont les conventions qui . . . ») sont ambiguës en français car on ne sait pas si elles comportent une cible unaire ou une cible plus complexe, c'est-à-dire si elles attendent comme réponse une liste de documents ou une liste de graphes de documents.

Organisation Internationale du Travail

Un premier ensemble de requêtes portant sur le corpus de l'Organisation Internationale du Travail (droit européen, voir section 5.3.3) qui a été recueilli auprès de Meritxell Fernandez⁸³. Rappelons que ce corpus comporte deux types de documents, des conventions (**conv**) et des recommandations (**recom**), liés par la relation d'implémentation (**impl**), les conventions implémentant les recommandations.

Nous avons annoté les documents avec des concepts du domaine du travail. La liste des descripteurs sémantiques utilisés dans les requêtes est donnée dans le tableau 5.3 avec leurs pendants terminologiques.

OIT1-1 Quelle convention implémente la recommandation 113 sur la consultation aux échelons industriel et national ?

$$(x) : Att(x, Conv) \wedge Rel(x, implémenter, Recom_{113}) \wedge Att(Recom_{113}, consultation)$$

Résultat attendu L'ensemble des documents de type « convention » qui ont la recommandation 113 pour cible par la relation d'implémentation si la recommandation 113 porte bien le descripteur **consultation**, sinon le graphe requête n'est pas satisfiable.

OIT1-2 Quelle convention implémente la recommandation qui parle des accidents du travail des marins ?

$$(x) : Att(x, Conv) \wedge Att(y, Recom) \wedge Rel(x, implémenter, y) \wedge Att(y, accT) \wedge Att(y, marin)$$

Résultat attendu L'ensemble des documents de types convention ayant pour cible par la relation d'implémentation au moins un document de type recommandation et portant les descripteurs **accT** et **marin**.

OIT1-3 Quelle recommandation est implémentée par la convention qui parle de l'exposition à l'amiante ?

$$(x) : Att(x, Recom) \wedge (y, implémenter, x) \wedge Att(y, Conv) \wedge Att(y, expoAmiante)$$

Résultat attendu L'ensemble des documents de type recommandation qui sont la cible par la relation d'implémentation d'au moins un document de type convention et portant le descripteur **expoAmiante**.

OIT1-4 Quelles sont les recommandations implémentées par les conventions qui parlent de la pollution de l'air ?

$$(x) : Att(x, Recom) \wedge Rel(y, implémenter, x) \wedge Att(y, Conv) \wedge Att(y, pollAir)$$

Résultat attendu L'ensemble des documents de type recommandation qui sont la cible par la relation d'implémentation d'au moins un document de type convention portant le descripteur **pollAir**.

⁸³. Juriste chez CERSA (Centre d'Études et de Recherches de Sciences Administratives et Politiques, <http://www.cersa.cnrs.fr/>), partenaire du projet Légilocal.

TABLE 5.3 – Vocabulaire utilisé pour la modélisation de la collection OIT et les requêtes associées. Les types et les identifiants de documents ont une majuscule à l’initiale ; les identifiants comportent en outre un indice ; les noms de relations et les descripteurs sémantiques ont une initiale minuscule mais les noms de relations sont des verbes.

Types	Descriptifs
Recom	Recommandation
Conv	Convention
Relations	Descriptifs
implémenter	Implémentation (les conventions implémentent les recommandations)
Descripteurs	Equivalent terminologique
consultation	« consultation aux échelons industriels et national »
accT	« accidents du travail »
navire	« navire »
expoAmiante	« exposition à l’amiante »
pollAir	« pollution de l’air »
convColl	« convention collective »
negoColl	« la négociation collective »
bruit	« bruit »
vibration	« vibrations »
benzene	« benzène »
cancerP	« cancer professionnel »
Identifiants	Référents
Recom ₁₁₃	Recommandation 113
Conv ₁₃₉	Convention 139

OIT1-5 Quelles sont les recommandations implémentées par des conventions qui parlent de la convention collective et de la négociation collective ?

$$(x) : Att(x, Recom) \wedge Redl(y, implementer, x) \wedge Att(y, Conv) \wedge Att(y, convColl) \wedge Att(y, negoColl)$$

Résultat attendu L'ensemble des documents de type recommandation qui sont la cible par la relation d'implémentation d'au moins un document de type convention portant à la fois le descripteur `convColl` et le descripteur `negoColl`.

OIT1-6 Quelles conventions implémentent les recommandations qui parlent de bruit et vibrations ?

$$(x) : Att(x, Conv) \wedge Rel(x, implementer, y) \wedge Att(y, Recom) \wedge Att(y, bruit) \wedge Att(y, vibration)$$

OIT1-7 Quelle recommandation, qui parle du benzène, est implémentée par la convention 139 sur le cancer professionnel ?

$$(x) : Att(x, Recom) \wedge Att(x, benzene) \wedge Rel(Conv_{139}, implementer, x) \wedge Att(Conv_{139}, cancerP)$$

Résultat attendu L'ensemble des documents de type recommandation et portant le descripteur `benzene` qui sont la cible par la relation d'implémentation d'au moins un document de type convention portant le descripteur `cancerP`.

Autres propositions de requêtes qui sont plus génériques :

OIT2-1 Quelles sont les recommandations qui sont implémentées ?

$$(x) : Att(x, Recom) \wedge Rel(y, implementer, x)$$

OIT2-2 Quels sont les couples de conventions et de recommandations (en relation d'implémentation) ?

$$(x, y) : Att(x, Conv) \wedge Att(y, Recom) \wedge Rel(x, implementer, y)$$

OIT2-3 Quels sont les couples de conventions et de recommandations (en relation d'implémentation) qui parlent de sujets différents ?

Analyse Cette requête ne peut être formalisée sans opérateur de négation.

OIT2-4 Quelles sont les conventions qui implémentent la même recommandation ?

$$(y, z) : Att(x, Recom) \wedge Rel(y, implementer, x) \wedge Rel(z, implementer, x) \text{ avec } y \neq z$$

Analyse On analyse cette requête comme une recherche de couples, parce qu'on est obligé de fixer la taille de la cible. On ne peut pas retrouver un ensemble de conventions de taille quelconque qui implémenteraient une même recommandation : il faut rechercher des couples ou des triplets ou etc.

OIT2-5 Quelles sont les conventions qui implémentent la même recommandation et la recommandation qu'elles implémentent ?

$$(y, z, x) : Att(x, Recom) \wedge Rel(y, implementer, x) \wedge Rel(z, implementer, x) \text{ avec } y \neq z$$

Analyse Cette requête est interprétée comme une variante de la précédente. Il est difficile de préciser la taille de la cible dans les requêtes en langage naturel.

OIT2-6 Quelles sont les recommandations qui sont implémentées de deux manières différentes (c'est-à-dire par au moins deux conventions différentes) ?

$$(x) : Att(x, Recom) \wedge Rel(y, implementer, x) \wedge Rel(z, implementer, x) \text{ avec } y \neq z$$

OIT2-7 Existe-t-il des conventions qui implémentent deux recommandations différentes ?

$$Att(x, Conv) \wedge Rel(x, implementer, y) \wedge Rel(x, implementer, z) \wedge Att(y, Recom) \wedge Att(z, Recom) \text{ avec } y \neq z$$

Légilocal

Les requêtes portant sur la collection Légilocal sont plus diverses du fait de l'hétérogénéité de la collection de départ. Ces requêtes concernent des relations existant entre les arrêtés locaux, les textes législatifs et des décisions de jurisprudence relatives au droit français.

Un premier ensemble de requêtes a été recueilli auprès de Eve Paul⁸⁴.

TABLE 5.4 – Vocabulaire utilisé pour la formation de la collection Légilocal et des requêtes associées

Types	Descriptifs
Décision	décision
ArrêtéComC	Arrêté de la commune C (« ma commune »)
Décret	décret
Arrêté	arrêté
ArrêtéMun	arrêté municipal
ArrêtCcass	Arrêt de Cour de cassation
ArrêtCappel	Arrêt de Cour d'appel
TexteLégislatif	Texte législatif
Code	Code
ArticleCode	Article de code
Relations	Descriptifs (voir p. 87)
appliquer	un texte législatif en applique un autre ou une décision applique une autre décision ou un texte législatif
exprimer	un document <i>est exprimé par</i> dans ses différentes versions
∈	composer
Descripteurs	Equivalents terminologiques
cheminR	« chemins rural »
véhiculeAMoteur	« véhicule à moteur »
Identifiants	Référents
Code _{CV} _Article ₁₃₈₂	Article 1382 du Code Civil
Loi _{Machin} _Article _X	Article X de la Loi Machin
ArrêtCcass _A	Arrêté A de la Cour de Cassation
ArrêtCappel _X	Arrêté X de la Cour d'appel
Décret _X	Décret X
Code _Y _Article _X	Article X du code Y
Décision _D	Décision D

L1-1 Quelles sont les décisions de jurisprudence qui citent l'article 1382 du code civil ?

$$(x) : Att(x, Decision) \wedge Rel(x, appliquer, Code_{CV_Article1382})$$

Analyse Le terme générique « cite » est interprété ici comme une relation d'application du fait du type des documents reliés et du fait que la décision est donnée comme jurisprudentielle.

84. Juriste chez Victoires Éditions, partenaire du projet Légilocal.

L1-2 Je voudrais tous les textes d'application de l'article X de la loi Machin.

$$(x) : Rel(x, appliquer, L_{Machin_ArticleX})$$

L1-3 Je voudrais la décision qui fait l'objet de l'arrêt A de la Cour de cassation.

$$(x) : Att(x, Decision) \wedge Rel(ArretCass_A, statuer, x)$$

L1-4 Je voudrais les décisions qui ont fait l'objet d'un arrêt de la Cour de cassation.

$$(x) : Att(x, Decision) \wedge Rel(y, statuer, x) \wedge Att(y, ArretCass)$$

L1-5 Je voudrais savoir si l'arrêt X de la cour d'appel a fait l'objet d'un pourvoi en cassation.

$$Rel(x, statuer, ArretCappel_X) \wedge Att(x, ArretCass)$$

L1-6 Je voudrais savoir ce sur quoi portait l'arrêt X de la cour d'appel ?

$$(x) : Rel(ArretCappel_X, statuer, x)$$

L1-7 Je voudrais savoir si ma commune a pris un arrêté d'application du décret X.

$$Att(x, Arret_ComC) \wedge Rel(x, appliquer, Decret_X)$$

Pour compléter ces exemples, nous proposons les requêtes plus complexes suivantes :

L2-1 Je cherche des arrêtés municipaux concernant les chemins ruraux qui ont fait l'objet d'un recours et ont été annulés par une décision de jurisprudence.

$$(x) : Att(x, ArreteMun) \wedge Att(x, cheminR) \wedge Rel(y, annuler, x)$$

Remarque Le recours n'est pas modélisé en tant que tel. Comme la relation **annuler** est une relation plus précise que **statuer**, seule la première est prise en compte dans la formalisation.

L2-2 Quels sont les textes législatifs sur lesquels s'appuient les décisions de jurisprudence qui ont annulé des arrêtés municipaux concernant les chemins ruraux ?

$$(x) : Att(x, texteLegislatif) \wedge Att(y, Decision) \wedge Rel(y, appliquer, x) \wedge Rel(y, annuler, z) \wedge Att(z, ArreteMun) \wedge Att(z, cheminR)$$

L2-3 Y a-t-il des décisions de jurisprudence qui ont annulé un arrêté municipal concernant les chemins ruraux en s'appuyant sur l'article X du code Y ?

$$Att(x, Decision) \wedge Rel(x, annuler, y) \wedge Att(y, ArreteMun) \wedge Att(y, cheminR) \wedge Rel(x, appliquer, CodeY_ArticleX)$$

L2-4 Quels sont les articles de code cités par les arrêtés municipaux parlant de chemins ruraux qui n'ont pas été annulés par une décision de jurisprudence ?

$$(x) : Att(x, ArticleCode) \wedge Att(y, ArreteMun) \wedge Att(y, cheminR) \wedge Rel(y, appliquer, x) \wedge Rel(z, confirmer, y) \wedge Att(z, Decision)$$

Remarque En l'absence d'opérateur de négation, « ne pas être annulé » est interprété comme « être confirmé », qui est plus fort.

L2-5 Quels sont les articles de code cités par les arrêtés municipaux parlant de chemins ruraux qui ont été annulés par une décision de jurisprudence ?

$$(x) : Att(x, ArticleCode) \wedge Rel(y, appliquer, x) \wedge Att(y, ArreteMun) \wedge Att(y, cheminR) \wedge Rel(z, annuler, y) \wedge Att(z, Decision)$$

L2-6 Quelles sont toutes les décisions antérieures à la décision D ?

$$(x) : Att(x, Decision) \wedge Rel(decision_D, statuer, x) \text{ (version 1)}$$

$$(x) : Att(x, Decision) \wedge Rel(decision_D, statuer, y) \wedge Rel(y, statuer, x) \text{ (version 2)}$$

Remarque En l'absence d'opérateur de disjonction, on doit fixer le degré d'antériorité des décisions recherchées par rapport à la décision D (décisions immédiatement antérieures dans la formalisation 1, décisions ayant donné lieu à deux décisions enchaînées dans la formalisation 2).

L2-7 Je voudrais des exemples d'annulation d'arrêtés municipaux concernant les chemins ruraux par des décisions de jurisprudence.

$$(x, y)Att(x, ArreteMun) \wedge Att(x, cheminR) \wedge Att(y, Decision) \wedge (y, annuler, x)$$

L2-8 Je voudrais des arrêtés parlant de chemins et de véhicules à moteur avec tous les documents visés.

$$(x, z) : Att(x, Arrete) \wedge Att(x, cheminR) \wedge Att(x, vehiculeMoteur) \wedge (x, y, z)$$

Exemples de requêtes portant sur l'historique des documents avec cible :

L3-1 Je voudrais savoir quel texte a codifié l'article L362-1 du code de l'environnement.

$$(x) : Rel(x, codifier, Code_{CE_Article_{L362-1}})$$

L3-2 Je voudrais la dernière version (ou la version en vigueur) de l'article L362-1 du code de l'environnement.

$$(x) : Rel(x, exprimer, Code_{CE_Article_{L362-1}})$$

Remarque En l'absence de négation, on ne peut pas formaliser cette requête parce qu'il faudrait retrouver la version de l'article L262-1 du code de l'environnement qui n'a pas été modifiée ou abrogée ; la formalisation proposée retourne toutes les version de l'article L362-1 du code de l'environnement.

L3-3 Je voudrais savoir si des textes visés par l'arrêté municipal 97-17 de Champigné ont été modifiés, et, si oui, quelles sont les nouvelles versions de ces textes.

$$(x, z) : Rel(ArreteMunicipalChampigne_{97-17}, appliquer, x) \wedge Rel(y, exprimer, x) \wedge Rel(z, modifier, y)$$

Remarque La formalisation de la requête doit fournir toutes les versions des textes cités ayant fait l'objet de modifications associées aux textes qui les ont modifiés mais ces dernières ont pu elle-mêmes été modifiées.

L3-4 Je voudrais savoir si des textes visés par l'arrêté municipal 97-17 de Champigné ont été abrogés, et le cas échéant, quels sont les nouveaux textes qui les ont remplacés.

$$(y) : Rel(ArreteMunicipalChampigne_{97-17}, appliquer, x) \wedge Rel(y, exprimer, x) \wedge Rel(z, abroger, y)$$

5.4.4 Jeu de requêtes types

A partir des exemples ci-dessus, on peut identifier des requêtes types qui méritent d'être prises en compte dans un système d'accès à l'information légale. Ces types varient selon quatre paramètres principaux :

- la complexité structurelle de la requête : requêtes simples ou requêtes relationnelles qui varient elles mêmes selon le nombre de relations, la réflexivité et présence de cycles ;
- l'utilisation des variables ou des identifiants pour désigner les documents, les relations, les types et les attributs ;
- la cible de requête ;
- les contraintes.

Nous exprimons ci-dessous ces requêtes-types de manière formelle, indépendamment des collections sur lesquelles elles peuvent être instanciées. Les descripteurs, types et variables utilisés sont listés dans le tableau 5.5.

RT1-1 Requête sans prédicat relationnel et sans variable

$$Att(i_1, t_1) \wedge Att(i_1, d_1)$$

TABLE 5.5 – Vocabulaire utilisé dans le jeu de requêtes-types

Types	t_1, t_2, t_3, \dots
Descripteurs	d_1, d_2, d_3, \dots
Relations	r_1, r_2, r_3, \dots
Identifiants	i_1, i_2, i_3, \dots
Variables	x, y, z, \dots

RT1-2 Requête sans prédicat relationnel avec document variable

$$Att(x, t_1) \wedge Att(x, d_1)$$

RT1-3 Requête sans prédicat relationnel avec type variable

$$Att(i_1, x) \wedge Att(i_1, d_1)$$

RT1-4 Requête sans prédicat relationnel avec descripteur variable

$$Att(i_1, t_1) \wedge Att(i_1, x)$$

RT1-5 Requête sans prédicat relationnel avec variables et contrainte

$$Att(i_1, t_1) \wedge Att(i_1, x) \wedge Att(i_1, y) \text{ avec } x \neq y$$

RT2-1 Requête avec prédicat relationnel et sans variable

$$Att(i_1, t_1) \wedge Att(i_1, d_1) \wedge Rel(i_1, r_1, i_2)$$

RT2-2 Requête avec prédicat relationnel et document variable

$$Att(x, t_1) \wedge Att(x, d_1) \wedge Rel(x, r_1, i_2)$$

RT2-3 Requête avec prédicat relationnel et type variable

$$Att(i_1, x) \wedge Att(i_1, d_1) \wedge Rel(i_1, r_1, i_2)$$

RT2-4 Requête avec prédicat relationnel et descripteur variable

$$Att(i_1, t_1) \wedge Att(i_1, x) \wedge Rel(i_1, r_1, i_2)$$

RT2-5 Requête avec prédicat relationnel et variable de relation

$$Att(i_1, t_1) \wedge Att(i_1, d_1) \wedge Rel(i_1, u, i_2)$$

RT2-6 Requête avec prédicat relationnel, variables et contrainte

$$Rel(x, u, y) \wedge Att(x, m) \wedge Att(y, n) \text{ avec } m \neq n$$

RT2-7 Requête avec prédicat relationnel reflexive

$$Rel(x, u, x)$$

RT3-1 Requête avec chaînage et sans variable

$$Rel(i_1, r_1, i_2) \wedge Rel(i_2, r_2, i_3)$$

RT3-2 Requête avec chaînage et variables de documents (avec et sans cible)

$$Rel(x, r_1, y) \wedge Rel(y, r_2, z)$$

$$(x) : Rel(x, r_1, y) \wedge Rel(y, r_2, z)$$

$$(x, z) : Rel(x, r_1, y) \wedge Rel(y, r_2, z)$$

$$(x, y, z) : Rel(x, r_1, y) \wedge Rel(y, r_2, z)$$

RT3-3 Requête avec chaînage et variables de relations

$$Rel(i_1, x, i_2) \wedge Rel(i_2, y, i_3)$$

RT3-4 Requête avec chaînage et variables de documents et de relations

$$Rel(x, u, y) \wedge Rel(y, v, z)$$

RT4-1 Requête en étoile et sans variable

$$Rel(i_1, r_1, i_2) \wedge Rel(i_1, r_2, i_3)$$

RT4-2 Requête en étoile avec variables de documents (avec et sans cible)

$$Rel(x, r_1, y) \wedge Rel(x, r_2, z)$$

$$(x) : Rel(x, r_1, y) \wedge Rel(x, r_2, z)$$

$$(x, z) : Rel(x, r_1, y) \wedge Rel(x, r_2, z)$$

$$(x, y, z) : Rel(x, r_1, y) \wedge Rel(x, r_2, z)$$

RT4-3 Requête en étoile avec variables de relations

$$Rel(i_1, x, i_2) \wedge Rel(i_1, y, i_3)$$

RT4-4 Requête en étoile avec variables de documents et de relations

$$Rel(x, u, y) \wedge Rel(x, v, z)$$

RT5-1 Requête avec cycle et sans variable

$$Rel(i_1, r_1, i_2) \wedge Rel(i_2, r_2, i_3) \wedge Rel(i_3, r_3, i_1)$$

RT5-2 Requête avec cycle et variables de documents (avec et sans cible)

$$Rel(x, r_1, y) \wedge Rel(y, r_2, z) \wedge Rel(z, r_3, x)$$

$$(x) : Rel(x, r_1, y) \wedge Rel(y, r_2, z) \wedge Rel(z, r_3, x)$$

$$(x, z) : Rel(x, r_1, y) \wedge Rel(y, r_2, z) \wedge Rel(z, r_3, x)$$

$$(x, y, z) : Rel(x, r_1, y) \wedge Rel(y, r_2, z) \wedge Rel(z, r_3, x)$$

RT5-3 Requête avec cycle et variables de relations

$$Rel(i_1, x, i_2) \wedge Rel(i_2, y, i_3) \wedge Rel(i_3, z, i_1)$$

RT5-4 Requête avec cycle et variables de documents et de relations

$$Rel(x, u, y) \wedge Rel(y, v, z) \wedge Rel(z, w, x)$$

5.4.5 Discussion

Le langage de requêtes ci-dessus permet de traiter l'intertextualité mais présente des limites.

Quantification Les requêtes en langage naturel impliquent des hypothèses de (non-)unicité qui ne sont pas exprimables dans le langage proposé. Par exemple, les variantes de requêtes suivantes sont considérées comme équivalentes dans notre langage de requête, où les variables sont quantifiées de manière existentielle : « Quels sont les jugements qui confirment une/des/plusieurs décision(s) . . . ». Même si la quantification universelle permettrait d'exprimer des requêtes telles que « Y a-t-il un article de code cité par tous les arrêtés portant sur les routes rurales ? », nous avons choisi de ne pas l'inclure dans un premier temps, car elle est difficile à maîtriser pour les utilisateurs et qu'elle n'apparaissait pas dans les requêtes recueillies.

Négation et disjonction Pour préserver la simplicité de la langue pour les utilisateurs, nous avons choisi de ne pas inclure la négation ou la disjonction des opérateurs dans la spécification du langage de requête, ce qui est une limitation en ce qui concerne les besoins des praticiens. Dans les exemples ci-dessus, deux formules différentes sont proposées pour la requête **L2-6**, tandis que la traduction adéquate impliquerait un opérateur de disjonction :

(x) :

$Att(x, Decision) \wedge (Rel(decision_D, statuer, x) \vee (Rel(decision_D, statuer, y) \wedge Rel(y, statuer, x)))$

pour prendre en compte différentes longueurs de chaînes de décision. Aussi, sans opérateur de négation, une requête comme « Quels sont les articles qui ne sont pas annulés ? » ne peut être formalisée que comme « Quels sont les articles qui ont été confirmés ? », qui est plus restrictive.

Cible de requête Il est souvent difficile d'identifier si une requête en langage naturel est ciblée ou non. Même si on est habitué à avoir des listes de documents, nous nous attendons à ce que les utilisateurs spécialisés apprécient un large éventail de types de réponses. Les graphes réponses donnent plus de contexte et peuvent être affinés grâce à une interface interactive. La différence ne réside pas dans la mise en correspondance du graphe de la requête et de la collection, mais dans la présentation des résultats.

Opérateur de comptage Jusqu'à présent, nous n'avons recueilli aucune requête nécessitant un opérateur de comptage, mais ce point doit être étudié plus avant.

Topologie de graphe Nous n'avons mis aucune contrainte sur la taille des graphes de requêtes ni sur la présence de cycles. Même si les exemples ci-dessus de graphes de requêtes sont simples, nous nous attendons à ce que les utilisateurs spécialisés entrent progressivement des requêtes plus complexes.

5.5 Conclusion

Dans ce chapitre nous avons analysé les besoins des experts dans le domaine d'application de notre travail. Nous n'avons pas pu revenir auprès de nos juristes pour valider notre proposition et la complexité de la matière juridique nous inspire beaucoup de prudence mais la diversité sémantique des relations observée entre les documents juridiques convainc aisément de l'enjeu que représente la prise en compte de l'intertextualité dans ce domaine et l'intérêt de la traiter sous forme de requêtes. Dans ce travail nous ne proposons pas de développer un outil de RI intertextuelle mais plutôt de tester la faisabilité d'une approche qui tient compte de cette dimension. Nous montrons respectivement dans le chapitre 6 et le chapitre 7 comment l'AFC/ARC et les techniques du web sémantique permettent de représenter une collection documentaire et les possibilités d'interrogation que cela ouvre. Ces modèles documentaires sont exploités par des outils de recherche et de navigation, ce qui permet, entre autres, de répondre aux requêtes relationnelles exposées dans ce chapitre.

Chapitre 6

RI et intertextualité : approche conceptuelle

Sommaire

6.1	Introduction	101
6.2	Collection documentaire et choix de modélisation	102
6.3	Modélisation du contenu sémantique par l’AFC	103
6.3.1	Construction des treillis formels	104
6.3.2	Interprétation des structures conceptuelles	105
6.4	Modélisation des liens intertextuels par l’ARC	107
6.4.1	Modèle de données	107
6.4.2	Construction des treillis relationnels	108
6.4.3	Interprétation de la structure relationnelle	109
6.4.4	Modèle de la collection documentaire	111
6.5	Interrogation du modèle documentaire	111
6.5.1	Stratégie de recherche dans le modèle documentaire	112
6.5.2	Requêtes simples	113
6.5.3	Requêtes relationnelles	114
6.5.4	Déroulement sur un exemple	117
6.6	Navigation dans la structure conceptuelle	119
6.6.1	Raffinement et expansion des résultats	120
6.6.2	Recherche par exemple de documents	123
6.6.3	Recherche de réponses approchées	127
6.7	Algorithmes d’interrogation et de navigation	129
6.8	Requêtes exprimables par le modèle	132
6.9	Conclusion	136

6.1 Introduction

Dans ce chapitre nous présentons une première approche pour la modélisation d’une collection documentaire. Ce modèle permet de représenter et d’interroger de manière unifiée les descripteurs de contenu des documents et les relations intertextuelles que ces derniers entretiennent. La méthode que nous proposons repose sur l’Analyse Formelle de Concepts (AFC) et

l'Analyse Relationnelle de Concepts (ARC). Ce modèle documentaire est exploité à des fins de recherche (répondre à des requêtes qui portent sur les relations entre documents) et de navigation dans le graphe des documents. Nous utilisons ces techniques pour formaliser un processus de RI et de navigation qui exploite à la fois le contenu sémantique des documents et leurs relations intertextuelles.

L'AFC avec son extension relationnelle, l'ARC, est une méthode de classification conceptuelle qui, à partir d'un jeu de données décrit par des objets, des attributs et des relations, construit une structure hiérarchique de concepts. Ces concepts représentent des ensembles d'objets groupés en fonction des attributs et relations qu'ils partagent. La structure ainsi construite sert alors d'espace de recherche et de navigation pour répondre aux besoins d'un utilisateur. Ces besoins peuvent être exprimés par des requêtes simples ou relationnelles auxquels il faut retourner un ensemble de réponses. Exploiter la structure construite pour naviguer entre les groupes des documents similaires permet de satisfaire ces besoins d'une autre manière.

Nous montrons dans la suite comment l'AFC et l'ARC permettent de représenter une collection documentaire et les possibilités d'interrogation et de navigation que cela ouvre. Nous déroulons les étapes de notre méthode sur un exemple réel de collection juridique : la collection bruit (décrite dans la section 5.3.3). Dans la section 6.3, nous présentons la manière dont nous proposons de modéliser le contenu sémantique d'une collection documentaire sur l'exemple détaillé. Cette modélisation est étendue pour prendre en compte les liens intertextuels entre les documents de la collection dans la section 6.4. Nous définissons deux types de requêtes pour interroger la collection dans la section 6.5. Les différentes possibilités de navigation et de recherche de documents similaires sont décrites dans la section 6.6. L'algorithme de recherche et de navigation fait l'objet de la section 6.7. Nous étudions l'expressivité du modèle présenté et montrons l'intérêt de cette modélisation pour la recherche d'information dans la section 6.8.

6.2 Collection documentaire et choix de modélisation

Sur la figure 6.1, nous présentons une collection de documents juridiques que nous proposons de modéliser avec l'approche AFC/ARC. Cette collection est composée d'un ensemble de documents de différents types (lois, décrets, arrêtés, jurisprudence) et de relations orientées (arrêtés \rightarrow décrets, décrets \rightarrow lois, jurisprudence \rightarrow lois et jurisprudence \rightarrow jurisprudence). Notons ici qu'un lien peut être défini sur un même type de document (comme dans le cas de la relation entre jurisprudences sur la figure 6.1).

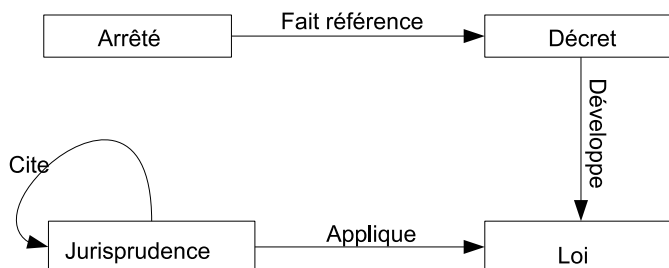


FIGURE 6.1 – Schéma d'un exemple de collection de documents juridiques.

Le contenu des documents est décrit par un ensemble de descripteurs sémantiques de contenu résultant d'un processus d'annotation sémantique au regard d'une ressource termino-ontologique.

En nous appuyant sur cette représentation riche de la collection, nous nous proposons d'exploiter les caractéristiques des documents dans cette collection pour créer un modèle qui articule la dimension sémantique et la dimension intertextuelle à des fins de recherche d'information.

Dans l'approche que nous présentons dans ce chapitre, nous avons fait un certain nombre de choix de modélisation afin d'optimiser la représentation de la collection et d'en faciliter l'exploitation. Ces choix ont été faits sur la base des caractéristiques des collections juridiques (types des documents, relations orientées, etc.) et seront explicités dans les sections suivantes. Cependant, l'approche proposée reste générale et permet de traiter des données différentes de celles qui sont manipulées dans le cadre de ce travail.

Nous montrons dans les sections suivantes comment l'AFC permet de construire une première structure modélisant le contenu des documents, qui est ensuite enrichie par la prise en compte, avec l'ARC, d'informations sur les liens intertextuels entre ces documents.

6.3 Modélisation du contenu sémantique par l'AFC

Dans cette section nous montrons comment l'approche AFC est appliquée pour la formalisation du contenu de notre collection documentaire. Les définitions des notions que nous utilisons dans ce qui suit sont données dans le chapitre 4.

Le contenu des documents est d'abord modélisé sous la forme d'un contexte formel qui décrit une relation binaire entre un ensemble d'objets et un ensemble d'attributs (*objets* \times *attributs*). Les objets correspondent aux documents de notre collection juridique. Les attributs sont des descripteurs sémantiques qui annotent le contenu de ces documents.

Nous définissons un contexte formel par type de document. Différents treillis correspondent donc aux différents types de documents.

La division de la collection initiale en plusieurs contextes formels et respectivement plusieurs treillis présente l'avantage de réduire le coût de calcul d'un grand treillis. Cela donne aussi une vision plus proche de la réalité des collections juridiques généralement regroupées par types de documents.

Considérons la collection documentaire de la figure 6.1. Pour modéliser cette collection, nous construisons quatre contextes formels (documents \times descripteurs sémantiques) pour les quatre types de documents : arrêtés, décrets, lois et jurisprudence. L'ensemble des contextes formels qui modélisent cette collection est donné par la figure 6.2.

Reprenons cette modélisation en détail sur la collection BRUIT. Deux contextes formels (documents \times descripteurs sémantiques) sont construits pour les deux types de documents : arrêtés d'un côté, décrets et lois de l'autre. Dans la suite, nous utilisons le terme « décrets » pour désigner l'ensemble de documents de types décrets et lois.

La formalisation du contenu des documents de type arrêtés est donnée par le contexte formel $\mathcal{K}_{arr} = (A, S, Inc)$, où A est un ensemble de documents (Arrêté préfectoral Paris, Arrêté municipal Strasbourg, etc.), S est un ensemble de descripteurs sémantiques du domaine (par exemple **nuisance sonore**) et Inc une relation binaire entre A et S appelée incidence de \mathcal{K}_{arr} et vérifiant les propriétés : $Inc \subseteq A \times S$ et $(a, s) \in Inc$ ou $(a \text{ Inc } s)$ où a et s sont tels que $a \in A$ et où $s \in S$ signifie que le document a est caractérisé sémantiquement par le descripteur s . De la même façon, la formalisation du contenu des documents de type décrets est donnée par le contexte formel $\mathcal{K}_{dec} = (D, S', Inc)$, où D est un ensemble de documents (Décret 95, Code Penal, Loi 1992, etc.),

Arrêté	a1	a2	a3	a4	a5
AP	X		X		
AB	X	X		X	
AY	X	X			X
AS			X		X

Contexte formel - arrêtés

Décret	d1	d2	d3	d4
D1	X		X	
D2		X		X
D3		X	X	
D4	X			X

Contexte formel - décrets

Jurisprudence	j1	j2	j3	j4	j5	j6	j7
J1	X			X			X
J2	X	X				X	X
J3			X	X	X		X
J4		X				X	
J5		X			X	X	
J6		X		X	X	X	

Contexte formel - jurisprudence

Loi	l1	l2	l3	l4	l5	l6
L1		X		X		X
L2	X		X		X	X
L3	X	X	X			X
L4	X		X		X	
L5	X	X		X		X

Contexte formel - lois

FIGURE 6.2 – Ensemble de contextes correspondant à la collection juridique de la figure 6.1.

S' est un ensemble de descripteurs sémantiques du domaine (par exemple **activité bruyante**, **isolation phonique**) et Inc une relation binaire entre D et S' appelée incidence de \mathcal{K}_{dec} .

Les contextes formels correspondant aux deux types de documents de notre collection juridique sont donnés dans la table 6.1 (arrêtés) et la table 6.2 (décrets).

TABLE 6.1 – Le contexte formel des arrêtés \mathcal{K}_{arr} .

	Bruit anormalement gênant (bag)	Nuisance sonore (ns)	Pollution acoustique (pa)	Sonorisation (son)	Niveau sonore (nvs)
Arrêté Paris (AP)	X		X		
Arrêté Boulogne Billancourt (AB)	X	X		X	
Arrêté Yvelines (AY)	X	X			X
Arrêté Strasbourg (AS)			X		X

6.3.1 Construction des treillis formels

Un concept formel dans la formalisation des documents de notre collection \mathcal{K}_{arr} est un ensemble de documents partageant un ensemble de descripteurs sémantiques.

TABLE 6.2 – Le contexte formel des décrets \mathcal{K}_{dec} .

	Lutte contre le bruit (lcb)	Tranquillité du voisinage (tv)	Activité bruyante (ab)	Isolation phonique (ip)
Décret 95 (D95)	x		x	
Code Pénal (CPen)		x		x
Ordonnance 1945 (O45)		x	x	
Loi 1992 (L92)	x			x

La figure 6.3 montre le treillis de concepts \mathcal{L}_{arr} correspondant au contexte formel des arrêtés \mathcal{K}_{arr} donné par la table 6.1. La figure 6.4 montre le treillis de concepts \mathcal{L}_{dec} construit à partir du contexte formel des décrets \mathcal{K}_{dec} donné par la table 6.2.

6.3.2 Interprétation des structures conceptuelles

Dans ces treillis, les documents sont structurés sous forme de concepts. Un concept représente une classe de documents (l'extension) caractérisée ou décrite par un ensemble de descripteurs (l'intension). Pour plus de clarté nous notons dans la suite a_i les concepts du treillis des arrêtés \mathcal{L}_{arr} et d_j les concepts du treillis des décrets \mathcal{L}_{dec} .

Par exemple, le concept a_4 dans le treillis des arrêtés (figure 6.3) représente l'ensemble des documents qui partagent les descripteurs *bag* (bruit anormalement gênant) et *ns* (nuisance sonore). Cela correspond dans notre exemple aux documents *AB* (arrêté de Boulogne) et *AY* (arrêté des Yvelines). Le lien entre les concepts a_3 et a_4 peut être interprété comme un lien de généralisation/spécialisation entre les classes représentées par ces concepts. Le concept a_4 contient dans son extension l'ensemble des documents décrits par *bag* et *ns*, une description plus générale que celle des documents contenus dans l'extension du concept a_3 qui sont décrits par plus de propriétés à savoir *bag*, *ns* et *son*.

Les documents *CPen* (Code Penal) et *L92* (loi 1992) sont tous les deux décrits par le descripteur sémantique *ip* (isolation phonique) dans le treillis des décrets (figure 6.4). Ces deux documents sont classés dans le même concept d_8 qui est donc la classe de documents juridiques partageant la même propriété. Le concept d_3 contient le seul document *CPen*, décrit par les deux propriétés *ip* et *tv*, puisqu'aucun autre document de la collection ne partage les mêmes propriétés. Le concept d_3 est subsumé par le concept d_8 qui est plus général.

Les treillis construits par l'AFC présentent les regroupements de documents correspondant à toutes les combinaisons possibles des attributs des documents. Dans le contexte de la RI, on peut interpréter les intensions de concepts comme des requêtes (combinaison de descripteurs) et les extensions comme les documents satisfaisant ces requêtes. Construire le treillis revient donc à pré-calculer les réponses à toutes les requêtes satisfaisables qui peuvent être posées sur cette collection étant donné un ensemble S de descripteurs.

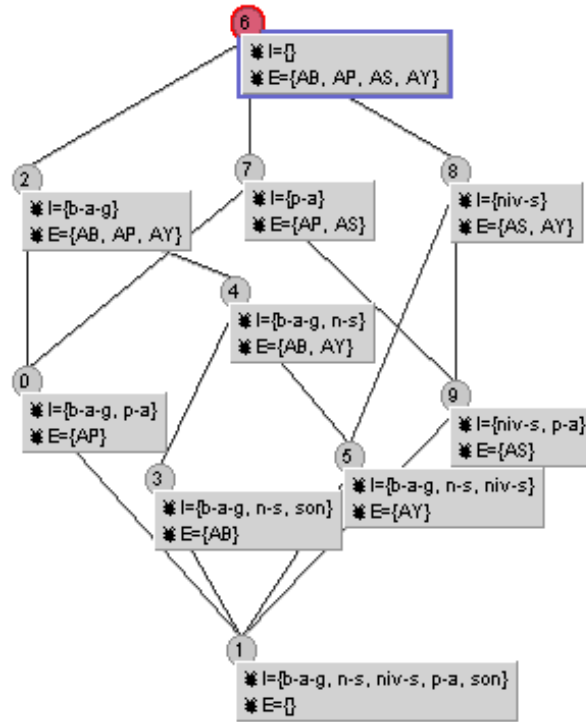


FIGURE 6.3 – Le treillis de concepts \mathcal{L}_{arr} correspondant au contexte formel des arrêtés \mathcal{K}_{arr}

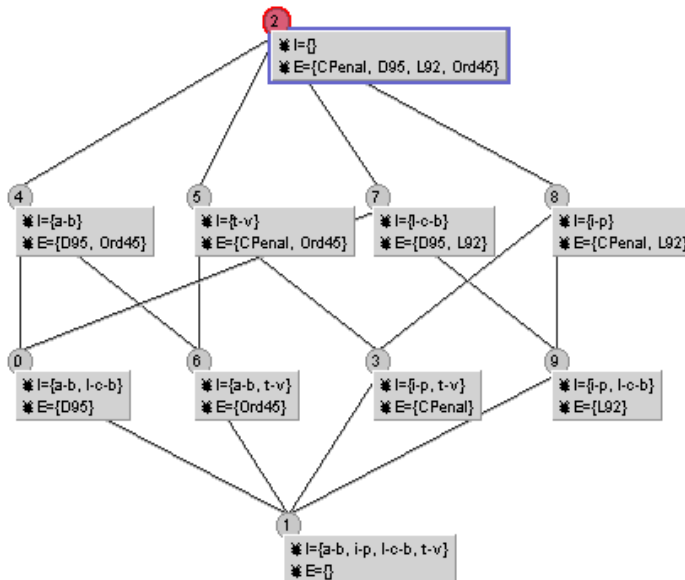


FIGURE 6.4 – Le treillis de concepts \mathcal{L}_{dec} correspondant au contexte formel des décrets \mathcal{K}_{dec}

6.4 Modélisation des liens intertextuels par l'ARC

La complexité des données juridiques tient en premier lieu à son facteur d'intertextualité (plusieurs liens entre les documents) et de la diversification des types de ces liens comme détaillé dans le chapitre 2. Nous utilisons l'ARC, extension relationnelle de l'AFC, pour prendre en compte la dimension intertextuelle dans la modélisation de la collection documentaire⁸⁵.

6.4.1 Modèle de données

Les données d'entrée à l'ARC sont organisées comme une paire constituée d'un ensemble de contextes formels (*objets* \times *attributs*), $\mathbb{K} = \mathcal{K}_{ii=1,\dots,n}$ et un ensemble de relations binaires (*objets* \times *objets*) $\mathbb{R} = r_{kk=1,\dots,m}$. Une relation $r \in \mathbb{R}$ relie deux ensembles d'objets provenant de deux contextes, à savoir, il existe $i_1, i_2 \in 1, \dots, n$ (éventuellement $i_1 = i_2$) de telle sorte que $r \subseteq O_{i_1} \times O_{i_2}$.

Les contextes de la figure 6.5 montrent un exemple de données en entrée pour la collection de la figure 6.1. Les relations sont représentées séparément par des tables qui lient les objets de contextes formels, appelés contextes relationnels. Sur cette collection, quatre contextes relationnels sont créés : **fait référence** (arrêtés \rightarrow décrets), **développe** (décrets \rightarrow lois), **applique** (jurisprudence \rightarrow lois) et **cite** (jurisprudence \rightarrow jurisprudence).

Fait référence	D1	D2	D3	D4
A1	X			
A2				X
A3		X		
A4			X	

Contexte relationnel - fait référence

Développe	L1	L2	L3	L4	L5
D1			X		X
D2				X	
D3		X	X		
D4	X				

Contexte relationnel - développe

Applique	L1	L2	L3	L4	L5
J1			X		X
J2	X	X			
J3		X			X
J4	X		X		
J5		X		X	
J6	X			X	

Contexte relationnel - applique

Cite	J1	J2	J3	J4	J5	J6
J1		X				
J2				X		X
J3	X					
J4						X
J5	X		X			
J6				X		

Contexte relationnel - cite

FIGURE 6.5 – Ensemble de contextes correspondant à la collection juridique de la figure 6.1.

Dans notre exemple, nous disposons de deux contextes binaires (*documents* \times *descripteurs-sémantiques*) \mathcal{K}_{arr} et \mathcal{K}_{dec} représentant respectivement l'ensemble des arrêtés et l'ensemble des décrets. Nous définissons une relation r représentant un lien direct entre les deux contextes formels \mathcal{K}_{arr} et \mathcal{K}_{dec} tel que $domaine(r) = A$ (ensemble des objets du contexte \mathcal{K}_{arr}) et $co-domaine(r) = D$ (ensemble des objets du contexte \mathcal{K}_{dec}). Cette relation décrit le lien *fait-référence* qui part des arrêtés vers les décrets. Elle est représentée séparément dans un contexte relationnel. La relation de référence est représentée par les couples $\{(AP,D95), (AB,L92), (AY,CPen)$ et $(AS,O45)\}$ sur la table 6.3.

L'ensemble de contextes résultants forme une Famille de Contextes Relationnels, $FCR = (\mathbb{K}, \mathbb{R})$ où

⁸⁵. L'ARC permet de modéliser deux types de relations : relations entre objets et relations entre attributs (propriétés). Nous nous focalisons dans ce travail sur l'étude du premier type, qui exprime les relations qui existent entre nos documents.

TABLE 6.3 – Relation : fait_référence

	D95	CPen	O45	L92
AP	×			
AB				×
AY		×		
AS			×	

- \mathbb{K} est un ensemble des contextes formels qui contient deux éléments \mathcal{K}_{arr} et \mathcal{K}_{dec} ,
- \mathbb{R} est un ensemble de contextes relationnels qui contient un seul élément $r_{arr-dec} \subseteq A \times D$, où A , domaine de la relation $r_{arr-dec}$, est l'ensemble des arrêts (objets du contexte \mathcal{K}_{arr}) et D , co-domaine de $r_{arr-dec}$, est l'ensemble des décrets (objets du contexte \mathcal{K}_{dec}).

Cette famille constitue le point de départ du processus de formation des structures conceptuelles correspondantes appelées Famille de Treillis Relationnels (FTR) [Rouane et al., 2007].

6.4.2 Construction des treillis relationnels

L'approche ARC construit, à partir du contexte source d'une relation, un treillis unique unifiant les informations provenant des contextes formels initiaux (*documents* \times *descripteurs-sémantiques*) et du contexte relationnel (*documents* \times *documents*). Le mécanisme du scaling relationnel (détaillé dans la section 4.2.3) d'un contexte avec une relation permet d'intégrer cette relation dans le contexte sous la forme d'attributs d'objets et de calculer le treillis résultant après cet enrichissement. Dans la suite de ce travail nous utilisons le codage existentiel⁸⁶.

Le processus consiste à construire d'abord les treillis initiaux à partir des contextes formels. Ensuite, dans les étapes suivantes, le mécanisme d'enrichissement (*codage*) relationnel traduit les liens entre les objets en attributs classiques de l'AFC et produit un ensemble de treillis dont les concepts sont liés par les relations décrites par les contextes relationnels. Ces étapes sont répétées jusqu'à atteindre un point de stabilité des treillis (lorsque aucun nouveau concept n'est produit).

Dans notre exemple, le treillis des arrêts est enrichi par l'information sur les relations de ses objets vers les objets du treillis des décrets. Le processus comporte plusieurs étapes.

1. La première consiste à construire les treillis de concepts initiaux correspondant aux contextes formels \mathcal{K}_{arr} et \mathcal{K}_{dec} , \mathcal{L}_{arr} et \mathcal{L}_{dec} .
2. La deuxième étape enrichit le contexte des arrêts à partir du treillis des décrets obtenu \mathcal{L}_{dec} et de la relation *fait-référence*. L'étape d'enrichissement du contexte \mathcal{K}_{arr} consiste à ajouter les relations vers les concepts du treillis des décrets \mathcal{L}_{dec} comme nouveaux attributs dans le contexte des arrêts. L'ajout d'un attribut au contexte \mathcal{K}_{arr} est effectué lorsqu'un document de ce contexte (un arrêt) est en relation avec au moins un document dans l'extension du concept considéré dans \mathcal{L}_{dec} .

⁸⁶. car dans le cas où $r(o)$, l'image par r de o dans O_i (les objets dans O_j qui sont en relation avec un objet o dans O_i) n'ont aucun attribut commun, aucun attribut relationnel ne sera ajouté à o . Ce qui résulte en une situation tout ou rien : ou bien un attribut relationnel pour tous les objets en relation ou bien aucune relation, ce qui fait perdre les relations vers les objets qui sont dans des concepts séparés.

Suite à cette étape, le contexte des arrêtés est modifié comme le montre la table 6.4 :
 $\mathcal{K}_{arr}^1 = \mathcal{K}_{arr}^0 + \mathcal{K}_{arr}^\Delta$.

TABLE 6.4 – Le contexte formel des arrêtés \mathcal{K}_{arr}^1 à l'itération 1 du processus d'enrichissement relationnel (dans les attributs $rf : ci$, les ci correspondent aux concepts du treillis des décrets).

	bag	ns	pa	son	nvs	rf : c0	rf : c2	rf : c3	rf : c4	rf : c5	rf : c6	rf : c7	rf : c8	rf : c9
AP	x		x			x	x		x			x		
AB	x	x		x			x					x	x	x
AY	x	x			x		x	x		x			x	
AS			x		x		x		x	x	x			

3. À la troisième étape, un nouveau treillis des arrêtés est construit à partir du contexte \mathcal{K}_{arr}^1 enrichi de l'étape 2 de la première itération.

Le processus poursuit en itérant les étapes 2 et 3, et s'arrête lorsqu'aucune nouvelle relation ne peut être déduite à partir du treillis obtenu à l'étape précédente. Dans notre exemple, le treillis des décrets \mathcal{L}_{dec} , qui sert à enrichir le treillis des arrêtés, reste inchangé au cours du processus après sa première construction, donc le processus itératif s'arrête après la première itération.

Le déroulement de l'algorithme 1 (algorithme de l'ARC) sur notre exemple est donné par les étapes suivantes :

1. **Entrée** : $FCR = (\mathcal{K}_{arr}; \mathcal{K}_{dec}; r_{arr-dec})$: contexte formel des arrêtés, contexte formel des décrets, une relation *fait-référence* représentée par un contexte relationnel ;
2. **Étape d'initialisation** Construire \mathcal{L}_{arr}^0 le treillis de concepts du contexte \mathcal{K}_{arr}^0 , et \mathcal{L}_{dec} le treillis de concepts du contexte \mathcal{K}_{dec} ;
3. **Étape d'enrichissement**
 - Calculer l'extension relationnelle \mathcal{K}_{arr}^Δ du contexte \mathcal{K}_{arr}^0 avec l'unique relation $r_{arr-dec}$ et en utilisant \mathcal{C}_{dec} qui contient tous les concepts du treillis \mathcal{L}_{dec} ;
 - Créer le contexte étendu \mathcal{K}_{arr}^1 par la fusion de cette extension \mathcal{K}_{arr}^Δ avec \mathcal{K}_{arr}^0 ;
 - Construire le treillis \mathcal{L}_{arr}^+ du contexte \mathcal{K}_{arr}^1 enrichi ;
 - Critère d'arrêt : pas de nouveaux concepts qui s'ajoutent lors d'une itération d'enrichissement.
4. **Sortie** Retourner le treillis final enrichi \mathcal{L}_{arr}^+ .

Le treillis obtenu pour l'exemple de la famille de contextes relationnels décrite par les tables 6.1, 6.2 et 6.3 est donné dans la figure 6.6.

6.4.3 Interprétation de la structure relationnelle

Le treillis \mathcal{L}_{arr}^+ de la figure 6.6 est obtenu en intégrant au treillis initial de la figure 6.3 l'information sur les relations que les arrêtés entretiennent avec les décrets.

Si on compare ces deux treillis, la plupart des concepts ont une extension inchangée mais leur intension est enrichie d'attributs relationnels. C'est le cas par exemple du concept a_4 qui a la

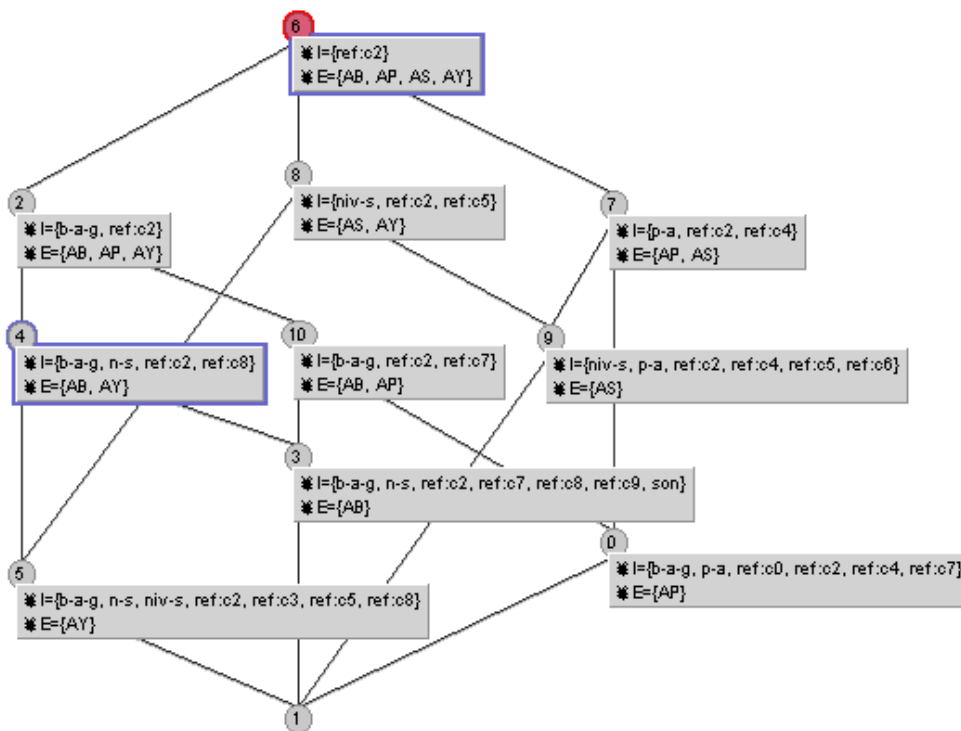


FIGURE 6.6 – Treillis relationnel \mathcal{L}_{arr}^+ résultant de l'enrichissement relationnel entre les objets du contexte des arrêts et du contexte des décrets.

même extension $E = \{ABoulogne, AYvelines\}$ dans les deux treillis mais dont l'intension finale combine les descripteurs de contenu de départ $\{bag, ns\}$ avec deux descripteurs relationnels $\{ref : c2, ref : c8\}$. La relation *fait-référence* est traduite (par enrichissement relationnel) en attributs relationnels ajoutés à l'intension de ce concept. Ceci indique que le nouveau concept a_4 est lié à deux autres concepts formels, d_2 et d_8 . Ces deux derniers correspondent à des classes de décrets dans le treillis \mathcal{L}_{dec} .

6.4.4 Modèle de la collection documentaire

Modélisée à l'aide de l'analyse formelle et relationnelle de concepts, la collection documentaire est représentée par un ensemble de classes de documents qui sont caractérisées à la fois par des descripteurs de contenus et par les relations que les documents entretiennent les uns avec les autres. Formellement, la collection documentaire est représentée par une famille de treillis de concepts relationnels, dont les extensions sont des classes de documents et les intensions sont des conjonctions d'attributs qui sont des descripteurs de contenu et/ou des relations vers d'autres classes de documents.

Dans le cas des contextes des arrêtés et des décrets, l'ajout des attributs relationnels s'interprète comme l'introduction de relations entre différentes classes de documents. L'interprétation de certains concepts du treillis \mathcal{L}_{arr}^+ permet de déduire des relations entre les classes des arrêtés et celui des décrets. Par exemple, la classe a_4 des arrêtés est ainsi liée aux classes de décrets d_2 et d_8 . La classe des arrêtés qui parlent de nuisances sonores et de bruits anormalement gênants ($\{ABoulogne, AYvelines\}$) est liée à la classe de décrets sur l'isolation phonique ($\{CPenal, L92\}$).

L'introduction des relations fait aussi apparaître de nouveaux concepts. Ceci est dû au fait que l'enrichissement existentiel (que nous utilisons dans ce travail) fait correspondre à un arrêté a_i en relation avec un décret d_i des attributs relationnels associés à des concepts (dans le treillis des décrets) qui regroupent d_i avec d'autres décrets conduisant ainsi à la formation de nouveaux regroupements dans le treillis des arrêtés après enrichissement relationnel. Sur l'exemple présenté, c'est le cas du concept a_{10} qui apparaît dans le treillis \mathcal{L}_{arr}^+ de la figure 6.6 mais qui n'était pas dans le treillis initial \mathcal{L}_{arr} . Le concept a_{10} regroupe les arrêtés qui font référence à au moins un décret (ou loi) sur la lutte contre le bruit (*lcb*) (ces décrets et lois sont regroupés dans le concept d_7). L'information relationnelle a donc conduit à créer un nouveau regroupement intermédiaire ($\{ABoulogne, AParis\}$) entre ceux des concepts a_2 ($\{ABoulogne, AParis, AYvelines\}$) et a_0 ($\{AParis\}$) du treillis initial.

6.5 Interrogation du modèle documentaire

Comme montré dans l'état de l'art, un treillis de concepts représente un moyen efficace de navigation et d'interrogation dans le contexte qui lui correspond, et donc dans la base documentaire qu'il représente. Le treillis relationnel ajoute une nouvelle dimension en introduisant des relations inter-concepts induites à partir des liens inter-objets [Rouane et al., 2007].

La famille de treillis relationnels que nous obtenons suite au processus d'enrichissement relationnel représente ainsi une structure riche qui permet de prendre en compte l'intertextualité dans la recherche d'information.

Cette structure nous permet, dans une perspective de RI, de sélectionner une classe (ou groupe) de documents qui sont pertinents par rapport à une requête décrite par des descripteurs de contenu et/ou par des liens vers d'autres documents. Deux types de requêtes peuvent être exprimées : les requêtes simples et les requêtes relationnelles. Dans ce qui suit nous donnons les

définitions de ces deux types de requêtes et nous présentons l'approche générale de recherche dans un treillis de concepts ou dans une famille de treillis relationnels.

6.5.1 Stratégie de recherche dans le modèle documentaire

Le principe général de l'interrogation du treillis de concepts représentant les documents de la collection est similaire à celui des méthodes introduites dans la section 4.3 dans la mesure où un objet représentant la requête doit être positionné dans le treillis puis le concept correspondant est identifié pour construire la réponse. Cette méthode est étendue au cas des treillis relationnels où la requête concerne deux ou plusieurs treillis avec des relations entre leurs concepts.

Une fois que le treillis de concepts ou la famille de treillis relationnels est construit à partir des documents de la collection, la stratégie de recherche de documents pertinents consiste à appliquer la suite des étapes suivantes :

1. Définition d'une requête simple ou relationnelle : il s'agit de donner les ensembles de descripteurs sémantiques et/ou relationnels des documents recherchés. Autrement dit, il s'agit de donner les attributs qui reflètent les propriétés des documents à identifier. Une requête se présente sous la forme d'un ensemble d'attributs formels et/ou relationnels.
2. Insertion de la requête dans le(s) treillis de concepts/relationnels :

Dans le cas de requêtes simples (sur un seul treillis) cette étape est facilitée par l'existence d'algorithmes performants pour la construction incrémentale des treillis de concepts (section 4.2.2). Disposant initialement du treillis de concepts construit à partir des documents de la collection et d'une requête qui consiste en un ensemble de descripteurs de contenu (les attributs), la construction incrémentale se fait par ajout d'objet en considérant que la requête correspond à un objet fictif qui possède tous les attributs indiqués. La requête est donc présentée sous la forme d'un couple (objet, attributs) qui peut être inséré dans le treillis de concepts.

Dans le cas de requêtes relationnelles (sur une FTR), il n'existe pas d'algorithmes pour la construction incrémentale d'une FTR. Disposant initialement d'une FCR représentant les documents de la collection et les liens qui existent entre eux, et d'une requête décrite par un ensemble de descripteurs de contenus (attributs formels) et de descripteurs de relations (attributs relationnels), l'insertion de la requête dans les treillis relationnels peut être faite dès le début dans les contextes de la FCR qui sont modifiés pour prendre en compte la requête. Les contextes formels sont modifiés par ajout d'objets (possédant les attributs formels de la requête) ce qui correspond au traitement d'une requête simple par contexte formel. Les contextes relationnels sont modifiés par ajout d'une relation entre les objets (des requêtes simples) ajoutés dans les contextes formels. La requête relationnelle est ainsi présentée sous la forme d'un graphe de couples (objets, attributs) liés par des relations (objets, objets) qui peut être inséré dans la FTR.

3. Localisation de la requête dans le(s) treillis de concepts obtenu(s) : cette étape consiste à localiser, dans le(s) treillis de concepts modifié(s) (suite à l'insertion de la requête simple ou relationnelle), le concept le plus général incorporant toutes les propriétés de la requête. La recherche d'un tel concept est facilitée par l'identification, dans l'ensemble des concepts, de ceux qui contiennent les objets fictifs de la requête. Le concept représentant la requête dans le treillis est le concept qui vérifie les deux conditions suivantes : (1) il contient les attributs de la requête dans son intension et (2) il n'a pas de super-concept qui vérifie la condition (1).

4. Présentation de la réponse sous la forme d'un ensemble de documents (ou graphes de documents) pertinents pour la requête.

La stratégie de recherche de réponse pertinente à une requête commence par la définition de la requête à passer en entrée à un algorithme d'interrogation de la structure conceptuelle pour retourner l'ensemble des résultats. La formalisation des requêtes simples et relationnelles est donnée dans ce qui suit (la description formelle de l'algorithme fait l'objet de la section 6.7).

6.5.2 Requêtes simples

Une requête simple est traditionnellement interprétée comme une combinaison d'attributs. Une requête simple est satisfiable s'il existe un concept dans le treillis formel interrogé dont l'intension correspond à cet ensemble d'attributs et dont l'extension n'est pas vide. La réponse à la requête est l'ensemble des documents qui composent l'extension de ce concept formel.

Nous rappelons dans ce qui suit la définition de requête simple pour l'interrogation de treillis de concepts [Messai et al., 2005]. Nous gardons cette définition pour l'interrogation de notre structure relationnelle.

Définition 21 (Requête simple) *Une requête simple sur un treillis de concepts \mathcal{L} correspondant à un contexte formel $\mathcal{K} = (O, A, Inc)$ est un concept requête $\mathcal{Q}_s = (Q_E, Q_I)$, avec*

- l'extension, Q_E , contient un unique objet virtuel Q_{vo} , qui représente l'objet cible de la requête (supposé satisfaire les attributs de la requête $Q'_E = Q_I$),
- l'intension, Q_I , contient un ensemble d'attributs a_i de la requête ($Q_I = \{a_1, a_2, \dots, a_i\} \subseteq A$) décrivant les objets à chercher.

Dans la définition 21, la requête se présente sous la forme d'un couple comme motivé en section 6.5.1. Cette forme facilite l'insertion de la requête dans le treillis de concepts en utilisant un algorithme de construction incrémentale de treillis de concepts. Une telle insertion peut être considérée comme l'ajout d'une nouvelle entrée (un nouvel objet et ses attributs) dans le contexte formel considéré comme décrit dans la définition ci-dessous [Messai et al., 2005].

Définition 22 (\oplus) *Pour un contexte formel $\mathcal{K} = (O, A, Inc)$ et une requête $\mathcal{Q}_s = (Q_E, Q_I)$, nous définissons l'opérateur d'addition \oplus comme suit :*

$$\begin{aligned} \mathcal{K}_Q &= \mathcal{K} \oplus \mathcal{Q}_s \\ &= (O, A, Inc) \oplus (Q_E, Q_I) \\ &= (O \cup Q_E, A \cup Q_I, Inc_Q) \\ &= (O_Q, A_Q, Inc_Q) \end{aligned}$$

L'utilisation de $A \cup Q_I$ couvre le cas où la requête est définie avec des descripteurs de contenu qui ne sont pas forcément dans l'ensemble initial A . L'utilisation de ces attributs est possible en ayant recours au raffinement de requête en utilisant une ressource sémantique [Messai et al., 2006] ce qui permet de répondre à des requêtes plus riches sémantiquement (nous n'étudions pas ce cas dans le cadre de ce travail). La relation Inc_Q désigne la relation Inc à laquelle s'ajoute la relation entre Q_E et Q_I .

L'insertion de la requête $\mathcal{Q}_s = (Q_E, Q_I)$ dans le treillis de concepts \mathcal{L} produit un nouveau treillis \mathcal{L}_Q . Le concept représentant la requête dans \mathcal{L}_Q est le concept formel $C_Q = (Q'_I, Q_I)$ avec Q'_I est l'ensemble de tous les objets qui possèdent les attributs Q_I . Différents cas se présentent pour le concept C_Q dans le treillis \mathcal{L}_Q :

- S’il n’existe pas de concept dans le treillis qui contienne tous les attributs de la requête, alors l’ajout de la nouvelle entrée dans le contexte formel produit un nouveau concept $C_Q = (Q'_I, Q_I)$ avec $Q'_I = Q_E$ et transforme les concepts qui contiennent une partie de ces attributs en ajoutant dans leurs extensions l’objet de la requête Q_{vo} (ces concepts vont paraître comme super-concepts de C_Q dans le treillis \mathcal{L}_Q).
- S’il existe un concept $C = (A, B)$ qui contient tous les attributs de la requête alors deux cas sont possibles :
 - Si le concept du treillis contient les attributs de la requête avec d’autres attributs c.à.d $Q_I \subset B$, alors l’insertion crée un nouveau concept $C_Q = (Q'_I, Q_I)$ avec $Q'_I = A \cup Q_{vo}$ et transforme les concepts qui contiennent une partie de ces attributs en ajoutant dans leurs extensions l’objet de la requête Q_{vo} .
 - Si le concept du treillis possède exactement les attributs de la requête c.à.d $Q_I = B$, alors l’insertion ne produit pas de nouveau concept. Le concept $C = (A, B)$ est transformé en $C_Q = (Q'_I, B)$ avec $Q'_I = A \cup Q_{vo}$ et de même les concepts qui contiennent une partie de ces attributs sont transformés en ajoutant dans leurs extensions l’objet de la requête Q_{vo} .

Répondre à cette requête consiste à trouver tous les objets qui sont pertinents par rapport à la requête. Une réponse pertinente par rapport à la requête $\mathcal{Q}_s = (Q_E, Q_I)$ est contenue dans l’extension du concept $C_Q = (Q'_I, Q_I)$. Tous les objets dans Q'_I sont pertinents pour $\mathcal{Q}_s = (Q_E, Q_I)$ puisqu’ils partagent tous les attributs de la requête (l’ensemble Q_I). Si l’extension $Q'_I = Q_{vo}$, c.à.d qu’après insertion dans le treillis, l’extension du concept C_Q ne contient que l’objet requête Q_{vo} , alors aucun objet ne possède les mêmes attributs que la requête. Ceci signifie que la requête ne possède pas de réponse exacte.

Des réponses approchées sont néanmoins possibles. Elles sont contenues dans les super-concepts de C_Q . Les super-concepts de C_Q contiennent dans leurs extensions des objets qui possèdent au moins un attribut de la requête (par définition de la relation d’ordre entre les concepts dans le treillis de concepts). Ce cas de réponses approchées sera développé lors de la présentation de la navigation dans le treillis dans la section 6.6.

6.5.3 Requêtes relationnelles

Les besoins des experts peuvent aussi être exprimés, en plus des descripteurs sémantiques de contenu, par des descripteurs de liens entre les documents : c’est le cas de requêtes relationnelles. Une requête relationnelle se représente alors comme un ensemble de requêtes simples (portant chacune sur le contenu d’un document) et un ensemble de relations entre les requêtes simples. Nous schématisons une requête relationnelle par un graphe où les noeuds sont les documents (objets), décrits par leurs descripteurs (attributs) et les arcs sont les différents types de relations spécifiés par la requête.

Nous donnons dans ce qui suit la définition d’une requête relationnelle qui s’apparente de celle proposée dans [Azmeah et al., 2011b] même si nous formalisons les choses un peu différemment.

Définition 23 (Requête relationnelle) *Étant donné une famille de contextes relationnels, $FCR = (\mathbb{K}, \mathbb{R})$, composée d’un ensemble de contextes formels \mathbb{K} et d’un ensemble de relations \mathbb{R} , une requête relationnelle sur une famille de treillis relationnels (FTR) correspondant à la FCR est un graphe dont les noeuds sont des concepts et les arcs sont des relations entre les concepts. Elle est désignée par $\mathcal{Q}_r = (\mathcal{C}, \mathcal{R})$ avec :*

- \mathcal{C} est l'ensemble de concepts $\mathcal{Q}_{s,i}$ correspondant aux sous-requêtes simples de \mathcal{Q}_r , telles que $\mathcal{Q}_{s,i} = (Q_{E,i}, Q_{I,i})$ et $Q_{E,i} = \{Q_{vo,i}\}$
- \mathcal{R} est l'ensemble des contraintes relationnelles entre les objets virtuels de \mathcal{C} . $\forall \mathcal{R}_k \in \mathcal{R}, \exists rel_k \in \mathbb{R}, \exists i, j \mid \mathcal{R}_k = rel_k(Q_{vo,i}, o_{Q,j})$ tel que $o_{Q,j} \in O_{Q,j} \cup \{Q_{vo,j}\}$, $Q_{vo,i}$ et $Q_{vo,j} \in Q_{E,i}$.

Dans la définition 23, la requête \mathcal{Q}_r se présente sous la forme d'un graphe. Les noeuds du graphe correspondent aux sous-requêtes simples représentées par des concepts requêtes simples $\mathcal{Q}_{s,i} = (Q_{E,i}, Q_{I,i})$, une par contexte formel (dans lequel nous souhaitons trouver un objet). Les arcs du graphe sont les relations qui peuvent exister entre les contextes qui expriment les contraintes relationnelles ajoutées aux requêtes simples. Les contraintes relationnelles sont exprimées par des relations \mathcal{R}_k entre les objets virtuels $Q_{vo,i}$ des sous-requêtes simples. Un objet virtuel $Q_{vo,i}$ peut avoir des relations (selon les contextes relationnels) avec un objet $o_j \in O_j$ d'un contexte formel $\mathcal{K}_j = (O, A, Inc)$ ou avec un autre objet virtuel $Q_{vo,j}$ d'une requête simple.

Par définition, le graphe de la requête suit le schéma des données en entrée. Prenons un exemple d'une collection avec quatre types de documents ($D1, D2, D3, D4$) reliés entre eux par trois relations de la façon suivante : $\langle D1 \rightarrow_{R1} D3 \rangle, \langle D2 \rightarrow_{R2} D3 \rangle, \langle D3 \rightarrow_{R3} D4 \rangle$.

Nous pouvons définir un graphe requête relationnelle sur cette collection comme le montre la figure 6.7. Les noeuds de ce graphe correspondent aux sous-requêtes simples chacune relative à un contexte formel et les arcs sont les relations entre ces documents.

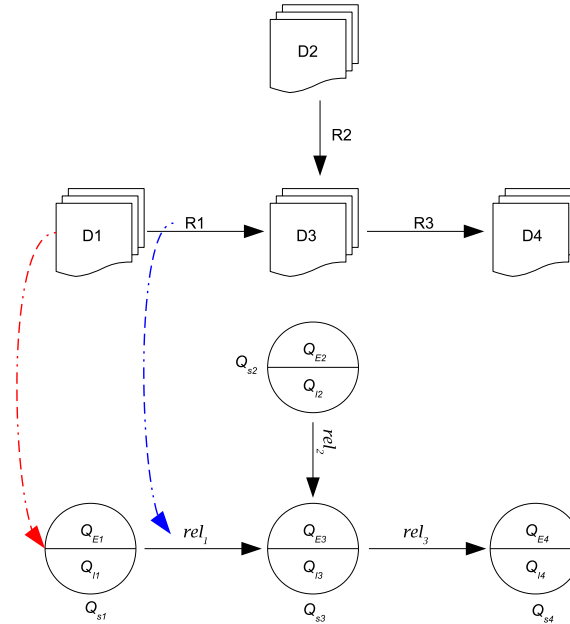


FIGURE 6.7 – Correspondance entre le schéma des données (documents dans la collection) et le graphe de la requête relationnelle

L'insertion de la requête dans la FTR est considérée comme l'ajout d'une nouvelle entrée dans chaque contexte formel de la FCR (un nouvel objet et ses attributs) impliqué dans la requête et d'une nouvelle relation dans chaque contexte relationnel de la FCR (une relation entre deux objets) impliqué dans la requête [Azmeah et al., 2011b].

Définition 24 (\oplus_R) *Pour une famille de contextes relationnels FCR = (\mathbb{K}, \mathbb{R}) et une requête relationnelle $\mathcal{Q}_r = (\mathcal{C}, \mathcal{R})$, nous définissons l'opérateur d'addition \oplus_R en s'appuyant sur la définition \oplus (Définition 22) comme suit :*

$$\begin{aligned} (\mathbb{K}_Q, \mathbb{R}_Q) &= (\mathbb{K}, \mathbb{R}) \oplus_R \mathcal{Q}_r \\ &= (\{\mathcal{K} \oplus \mathcal{C} \mid \mathcal{K} \in \mathbb{K}\}, \{\mathcal{R}_k \oplus \mathcal{R} \mid \mathcal{R}_k \in \mathbb{R}\}) \end{aligned}$$

Dans la définition précédente :

- l'ajout de nouvelles entrées correspondant aux objets virtuels des concepts sous-requêtes simples \mathcal{C} dans les contextes formels de \mathbb{K} impliqués par la requête \mathcal{Q}_r se fait comme expliqué dans la section 6.5.2 pour les requêtes simples ;
- l'ajout de nouveaux liens correspondant aux contraintes relationnelles \mathcal{R}_k dans les contextes relationnels de \mathbb{R} impliqués par la requête \mathcal{Q}_r consiste à ajouter dans chaque contexte un lien rel_k entre un objet virtuel (de \mathcal{C}) et un autre objet virtuel, $\mathcal{R}_k = rel_k(Q_{vo,i}, Q_{vo,j})$, ou entre un objet virtuel (de \mathcal{C}) et un objet d'un contexte formel, $\mathcal{R}_k = rel_k(Q_{vo,i}, o_j)$.

Une nouvelle FTR est produite à partir de la FCR $(\mathbb{K}_Q, \mathbb{R}_Q)$. Les treillis relationnels de la FTR \mathcal{L}_Q^+ sont modifiés après insertion de la requête. Cela s'explique par le même raisonnement décrit pour les requêtes simples dans la section 6.5.2 en considérant que l'ensemble des attributs peut contenir des attributs formels et aussi relationnels. Les différents cas qui se présentent pour le concept C_Q d'une requête simple dans le treillis \mathcal{L}_Q sont aussi valables dans le cas d'un concept qui contient un objet virtuel requête, un ensemble d'attributs formels et relationnels.

Répondre à cette requête consiste à trouver tous les objets qui sont pertinents par rapport à cette requête, c'est-à-dire qui répondent aux sous-requêtes simples et vérifient les contraintes relationnelles. Une réponse pertinente à la requête $\mathcal{Q}_r = (\mathcal{C}, \mathcal{R}) = (\mathcal{Q}_{s,i}, \mathcal{R}) = ((Q_{E,i}, Q_{I,i}), \mathcal{R})$ est contenue dans les réponses des requêtes simples $\mathcal{Q}_{s,i}$. Puisque $\mathcal{Q}_{s,i}$ sont insérés dans leurs treillis correspondant comme décrit dans la section 6.5.2, les réponses à ces requêtes sont incluses dans l'extension des concepts qui contiennent les objets $Q_{vo,i}$. Et la réponse à la requête \mathcal{Q}_r est donnée par le graphe qui lie les objets réponses aux requêtes simples par les relations de \mathcal{R} .

Définition 25 (Réponse à une requête relationnelle) *Une réponse à une requête relationnelle $\mathcal{Q}_r = (\mathcal{C}, \mathcal{R})$ est un graphe \mathcal{G} dont les noeuds sont des objets et les arcs sont des relations entre ces objets. Il est désigné par $\mathcal{G} = (\mathcal{O}_G, \mathcal{R}_G)$ avec :*

- \mathcal{O}_G est l'ensemble d'objets des contextes \mathcal{K}_i impliqués dans la requête (au moins un objet par contexte) : $\forall Q_{vo,i} \in Q_{E,i}, \exists o_i \in \mathcal{O}_G$;
- les noeuds du graphe sont décrits par les attributs de la requête : $\forall o_i \in \mathcal{O}_G, o'_i \subset Q_{I,i}$;
- les noeuds du graphe sont reliés par les relations de la requête : $\forall \mathcal{R}_k \in \mathcal{R}_G, \exists rel_k, \exists i, j \mid \mathcal{R}_k = rel_k(o_i, o_j)$.

Si l'extension des concepts qui contiennent les objets $Q_{vo,i}$ est vide, c.à.d qu'après insertion dans les treillis, l'extension ne contient que l'objet requête $Q_{vo,i}$, alors pour le treillis correspondant au contexte \mathcal{K}_i aucun objet ne possède les mêmes attributs de la sous-requête. Ceci signifie que la requête ne possède pas de réponse exacte. Des réponses approchées peuvent être calculées et retournées. Ce cas sera développé lors de la présentation de la navigation dans la structure conceptuelle dans la section 6.6.

6.5.4 Déroutement sur un exemple

Reprenons l'exemple de la collection documentaire des arrêtés et des décrets représentée par les treillis \mathcal{L}_{arr} et \mathcal{L}_{dec} . On peut considérer que le treillis initial (des arrêtés ou des décrets) représente l'ensemble des requêtes simples (ou combinaisons de descripteurs) qui peuvent être faites sur la collection documentaire et qui sont satisfiables, c'est-à-dire qui permettent de retourner des documents (toutes les combinaisons de descripteurs associées à une extension non nulle). Si la requête correspond à l'intension d'un concept qui a une extension, ce sont les documents de cette extension qui sont retournés en réponse à la requête ; si la requête correspond à une intension sans extension propre, on peut proposer des spécialisations ou au contraire généralisations de la requête (détails dans les sections suivantes).

Le treillis \mathcal{L}_{arr}^+ représente le résultat de l'enrichissement relationnel de \mathcal{L}_{arr} par la relation *fait-référence*. Les extensions des concepts de \mathcal{L}_{arr} contiennent, en plus des descripteurs de contenu, des descripteurs de liens. Notons que tous les concepts formels du treillis initial \mathcal{L}_{arr} sont conservés dans le treillis résultant après enrichissement relationnel \mathcal{L}_{arr}^+ . Ceci implique que toutes les requêtes satisfiables sur le treillis initial le restent sur le treillis final. On peut répondre à davantage de requêtes puisqu'il y a plus de concepts avec une extension propre dans le treillis (l'information relationnelle affine la catégorisation de l'ensemble des documents).

Exemple de requête simple

Considérons la requête suivante sur la collection des décrets :

"Quels sont les décrets qui parlent d'activités bruyantes (*ab*) ?".

Le mot clé *activités bruyantes* (*ab*) est considéré comme un descripteur sémantique annotant les documents de type *décrets*. Un concept requête simple $Q_s^{dec} = (Q_E^{dec}, Q_I^{dec})$ est créé tel que :

- $Q_E^{dec} = Q_{vo}^d$, l'objet virtuel de la requête (extension),
- $Q_I^{dec} = \{ab\}$, l'ensemble des attributs de la requête contenant un seul élément *ab* (intension).

Une telle requête est traitée en insérant le concept Q_s^{dec} dans le treillis des décrets $\mathcal{L}_{Q,dec}$ comme le montre la figure 6.9. Q_{vo}^d apparaît dans l'extension d'un concept qui existe déjà, donc l'insertion ne produit pas de nouveaux concepts dans le treillis \mathcal{L}_{dec} .

La réponse à la requête se trouve dans le concept le plus spécifique contenant l'objet Q_{vo}^d (qui correspond au concept le plus général qui contient tous les attributs de Q_{vo}^d dans le treillis), soit le concept *c*. Si *c* ne contient que Q_{vo}^d dans son extension, la requête initiale n'est pas satisfiable et aucun document n'est retourné. Si d'autres documents appartiennent à l'extension de *c* avec Q_{vo}^d , la requête est satisfiable et ces documents sont retournés.

L'algorithme d'interrogation localise l'objet Q_{vo}^d dans le treillis $\mathcal{L}_{Q,dec}$, la réponse est donnée par le concept d_4 qui a dans son extension Q_{vo}^d avec d'autres documents, *O45* et *D95*, qui sont retournés comme réponses pertinentes.

Exemple de requête relationnelle

Considérons l'exemple suivant de requête relationnelle sur la collection des décrets et des arrêtés :

"Quels sont les arrêtés qui parlent de niveau sonore (*nvs*) et qui font référence (*rf*) aux décrets

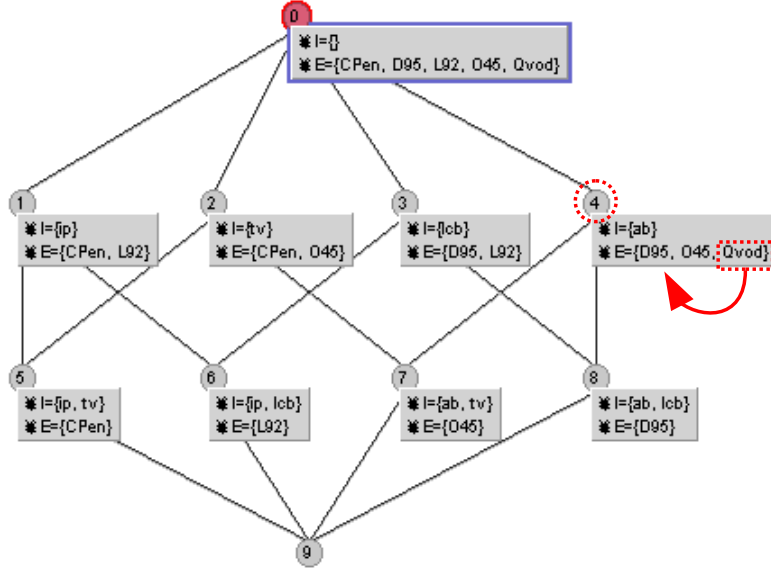


FIGURE 6.8 – Requête simple Q_s^{dec} sur le treillis des décrets $\mathcal{L}_{Q,dec}$.

sur les activités bruyantes (*ab*) ?".

Le mot clé *niveau sonore* (*nvs*) est considéré comme un descripteur sémantique annotant les documents qui sont de type *arrêtés*. Le mot clé *activités bruyantes* (*ab*) est le descripteur sémantique annotant les documents de type *décrets*. Le lien entre ces deux types de documents dans la requête est donné par la relation *fait référence* (*rf*).

La requête relationnelle est représentée par $\mathcal{Q}_r = (\mathcal{C}, \mathcal{R})$ tel que :

$$\begin{aligned}
 \text{Sous-requêtes simples } \mathcal{C} &= \{ \mathcal{Q}_{s,arr}, \mathcal{Q}_{s,dec} \} \\
 &= \{ (Q_{E,arr}, Q_{I,arr}), (Q_{E,dec}, Q_{I,dec}) \} \\
 &= \{ (\{Q_{vo}^a\}, \{nvs\}), (\{Q_{vo}^d\}, \{ab\}) \} \\
 &= \{ (Q_{vo}^a, nvs), (Q_{vo}^d, ab) \}. \\
 \text{Contrainte relationnelle } \mathcal{R} &= \{ \mathcal{R}_k \} \\
 &= \{ rel_k(Q_{vo,i}, Q_{vo,j}) \} \\
 &= \{ rf_{arr-dec}(Q_{vo}^a, Q_{vo}^d) \}.
 \end{aligned}$$

La requête relationnelle correspond à un graphe qui contient deux noeuds et un arc. Le premier noeud du graphe correspond au concept requête simple $\mathcal{Q}_{s,arr}$, qui représente la requête "*arrêtés parlant de niveau sonore*", sur le treillis des arrêtés après enrichissement relationnel \mathcal{L}_{arr}^+ . Le deuxième noeud du graphe correspond au concept requête simple $\mathcal{Q}_{s,dec}$, qui représente la requête "*décrets parlant d'activités bruyantes*", sur le treillis de décrets \mathcal{L}_{dec} . L'arc du graphe correspond à la relation $rf_{arr-dec}$ entre les objets virtuels Q_{vo}^a et Q_{vo}^d des deux concepts requêtes simples.

Une telle requête est traitée en l'insérant dans la FTR $(\mathcal{L}_{arr}^+, \mathcal{L}_{dec})$ comme suit :

- ajouter une nouvelle entrée dans les deux contextes formels \mathcal{K}_{arr} et \mathcal{K}_{dec} pour les objets Q_{vo}^a et Q_{vo}^d avec leurs attributs ;
- ajouter dans le contexte relationnel *fait référence* (*rf*) une relation entre l’objet virtuel Q_{vo}^a (dans une nouvelle ligne) et l’objet virtuel Q_{vo}^d (dans une nouvelle colonne).

L’algorithme construit ensuite la nouvelle FTR après insertion de la requête et le résultat est donné par la figure 6.9. Q_{vo}^a et Q_{vo}^d apparaissent dans les extensions de deux concepts qui existent déjà dans \mathcal{L}_{arr}^+ et dans \mathcal{L}_{dec} et donc l’insertion ne génère pas de nouveaux concepts dans les treillis $\mathcal{L}_{Q,arr}^+$ et dans $\mathcal{L}_{Q,dec}$ de la nouvelle FTR.

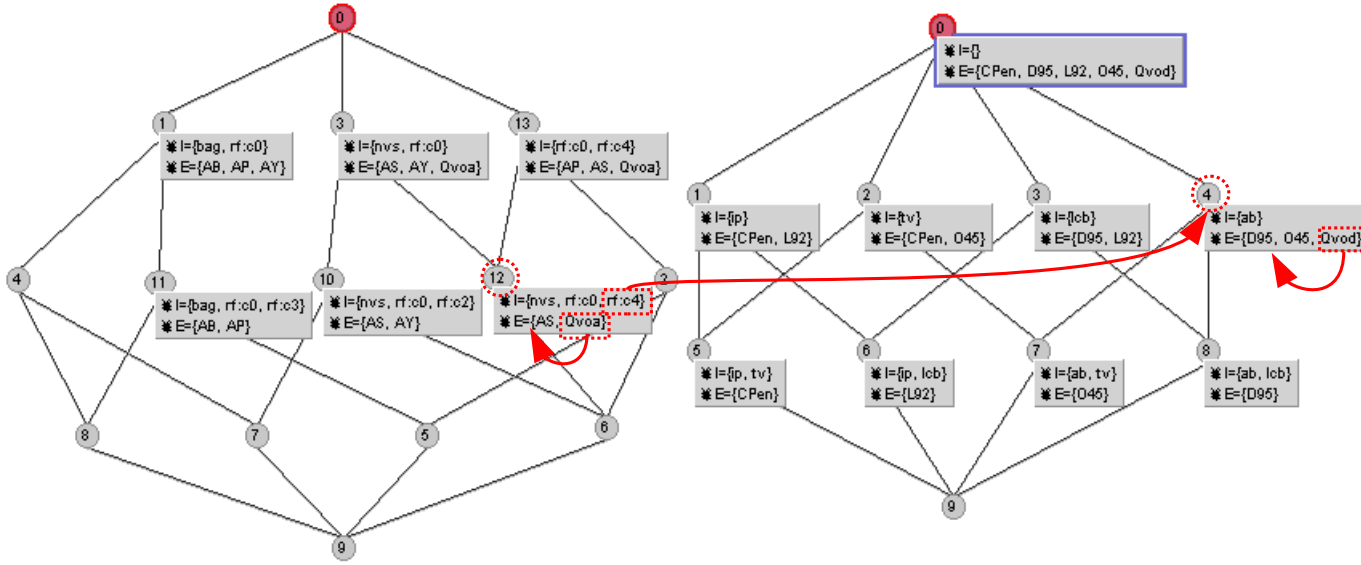


FIGURE 6.9 – Requête relationnelle \mathcal{Q}_r sur la FTR $(\mathcal{L}_{Q,arr}^+, \mathcal{L}_{Q,dec})$.

Localiser Q_{vo}^a et Q_{vo}^d dans les concepts de ces treillis donne la réponse à la requête. Comme le montre la figure 6.9, la réponse est donnée par les objets du graphe $\langle a_{12} \rightarrow_{rf} d_4 \rangle$. Le concept a_{12} contient dans son extension l’objet Q_{vo}^a et le document *AS* (Arrêté Strasbourg) qui possèdent tous les deux les attributs dans l’intension à savoir le descripteur sémantique *nvs* et une relation *rf* vers le concept d_4 . Le concept d_4 appartient au treillis des décrets $\mathcal{L}_{Q,dec}$ et contient dans son extension l’objet Q_{vo}^d avec les documents *O45* et *D95* qui partagent le descripteur sémantique *ab*. Selon le contexte relationnel, un graphe réponse exacte existe, il est composé du document *AS* lié au document *O45* ($\mathcal{G} = \langle AS \rightarrow_{rf} O45 \rangle$).

6.6 Navigation dans la structure conceptuelle

Les treillis de concepts ont l’avantage d’offrir une vue structurée des collections d’objets et de proposer une classification de l’ensemble de l’espace de recherche (classification des objets des solutions potentielles) dans une structure navigable (on peut naviguer de proche en proche en suivant les liens de généralisation ou de spécialisation entre les concepts du treillis). Les treillis relationnels ajoutent une nouvelle dimension à cette structure en introduisant des relations inter-concepts déduites des relations entre les objets.

La structure construite utilisant l'AFC et l'ARC à partir d'un graphe de documents organise les documents dans des groupes qui partagent les mêmes propriétés (descripteurs sémantiques de contenu et attributs relationnels), qui sont classés dans une hiérarchie de documents de types homogènes qui sont liés par différents types de relations intertextuelles. Cette hiérarchie organise les classes de documents selon une relation de généralisation/spécialisation⁸⁷. Les documents sont classés de manière à ce que nous ayons toujours la possibilité de généraliser, spécifier ou retourner une réponse approximative à l'utilisateur s'il n'existe pas de réponse exacte en naviguant dans la structure sans calcul supplémentaire.

La navigation dans cette structure offre de nouvelles fonctionnalités sémantiques aux systèmes d'accès documentaires. En effet, outre la fonctionnalité d'interrogation, l'utilisateur peut utiliser la structure pour explorer l'ensemble des documents. Deux stratégies de navigation sont possibles :

- par généralisation ou raffinement de requête ;
- par calcul de similarité.

Il y a plusieurs scénarios dans lesquels ces stratégies de navigation sont utiles :

- Dans certains cas, après avoir présenté une requête au système et obtenu un résultat, l'utilisateur veut élargir ou affiner la requête afin d'élargir ou de restreindre l'ensemble de documents retournés. Ce processus d'expansion/raffinement de résultats est obtenu en accédant à partir de la réponse retournée à un concept plus général ou plus spécifique dans la structure.
- L'utilisateur peut disposer au départ d'un échantillon, un document ou un ensemble de documents. Il s'agit dans ce cas d'identifier leurs caractéristiques communes et de trouver dans la structure relationnelle tous les autres documents qui possèdent ces mêmes attributs (ou une partie).
- Dans d'autres cas, la requête de l'utilisateur ne correspond pas à une réponse exacte. Cela signifie qu'il n'y a pas un concept qui contient l'ensemble des attributs de la requête mais une stratégie de navigation par similarité dans la structure relationnelle permet de calculer des résultats approchés.

Nous détaillons dans les sections suivantes les possibilités de navigation offertes par la structure des treillis de concepts formels et relationnels.

6.6.1 Raffinement et expansion des résultats

L'utilisateur n'est pas toujours satisfait par les résultats de la recherche par interrogation (voir section 6.5). Il a souvent en retour trop de documents ou trop peu de documents. Dans ces cas, il doit formuler une nouvelle requête, avec plus ou moins d'attributs ou de contraintes sémantiques et intertextuelles.

Grâce à la structure du treillis, de tels résultats affinés ou généralisés peuvent être obtenus sans avoir besoin de relancer une nouvelle requête – ce qui est efficace d'un point de vue opérationnel – en aidant l'utilisateur à reformuler sa requête initiale. Si le système renvoie trop (respectivement trop peu) de documents, l'utilisateur peut choisir de naviguer à partir du concept représentant la requête vers ses voisins supérieurs (ou inférieurs) pour généraliser (ou affiner) sa requête. Pour obtenir un accès aux voisins supérieurs (les super-concepts du concept requête), l'utilisateur relâche une ou plusieurs contraintes de sa requête, par la suppression d'un ensemble d'attributs de l'intension de la requête initiale. Pour les voisins inférieurs (les sous-concepts), l'utilisateur

87. Naviguer de bas en haut correspond à une généralisation : on passe d'un concept à un concept supérieur qui a une extension plus large (plus de documents), mais une intension plus petite (moins de propriétés). Inversement, la navigation du haut en bas permet de spécialiser.

peut affiner sa requête en ajoutant un ou plusieurs attributs à l'intension de la requête initiale. Dans les deux cas, la structure du treillis indique quel(s) est (sont) l'attribut (les attributs) pertinents à enlever ou à ajouter afin d'élargir ou affiner la requête, et le nombre de documents qu'un tel élargissement ou raffinement pourrait donner.

Prenons un exemple de requête simple :

$Q_s^{dec} = \text{"Quels sont les lois et les décrets qui parlent d'activités bruyantes (ab) et de tranquillité du voisinage (tv)?"}$.

L'objet requête Q_{vo}^d est classé dans le concept d_6 qui a pour intension $I = (ab, tv)$ et pour extension $E = (O45, Q_{vo}^d)$ comme illustré sur la figure 6.10. Une réponse exacte à Q_s^{dec} est donc le document $O45$. Si l'utilisateur a besoin d'avoir plus de résultats, il a la possibilité de parcourir le voisinage du document retourné en relâchant une contrainte sémantique de sa requête initiale. Le retrait du descripteur tv implique de rechercher des lois et décrets sur les activités bruyantes (ab), auxquelles on peut répondre par le concept d_4 où $I = (ab)$ et $E = (D95, O45, Q_{vo}^d)$. Le retrait du descripteur ab implique de rechercher des lois et décrets sur la tranquillité du voisinage (tv), qui retourne le concept d_5 où $I = (tv)$ et $E = (CPen, O45, Q_{vo}^d)$. Deux documents supplémentaires ($D95$ et $CPen$) sont retournés à l'utilisateur : ils représentent des réponses approchées à la requête initiale Q_s^{dec} .

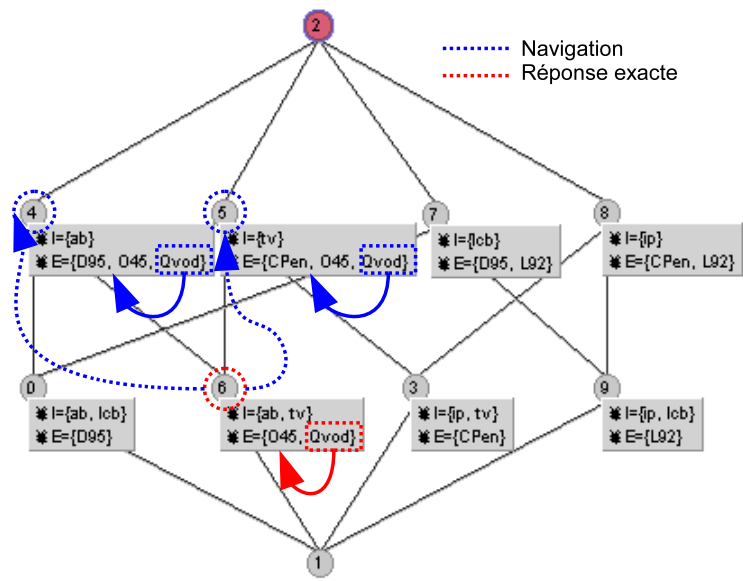


FIGURE 6.10 – Exemple de navigation par généralisation basée sur une requête simple

Nous adaptons la technique proposée par [Wray and Eklund, 2011] au cadre de l'ARC et au cas des requêtes relationnelles. L'approche de voisinage conceptuel permet de naviguer dans l'espace de recherche fourni par la famille de treillis relationnels. Telle que définie dans la section 6.5.3, une requête relationnelle est un graphe dans lequel chaque noeud correspond à une sous-requête simple sur un treillis (un type de documents) et les arêtes correspondent aux contraintes relationnelles entre ces sous-requêtes simples. L'utilisateur peut relâcher soit les contraintes sémantiques soit les contraintes relationnelles. La même chose s'applique pour l'ajout de contraintes pour le raffinement de requêtes.

Si les contraintes relationnelles sont considérées comme étant obligatoires, seules les contraintes sémantiques sont relâchées ou ajoutées. L'utilisateur doit choisir quelle sous-requête simple doit

être généralisée ou affinée, *i.e.* quel treillis ou quel type de documents doit être parcouru. En fait, dans une requête relationnelle, il existe un treillis principal qui correspond au type de documents qui doivent être retournés. Par défaut, l'utilisateur navigue dans le treillis principal enrichi selon la relation de la requête.

L'utilisateur peut également relâcher une contrainte relationnelle, ce qui équivaut à laisser tomber une partie du graphe requête. L'ajout d'une contrainte relationnelle est plus complexe car la nouvelle relation n'est généralement pas déjà modélisée dans la famille des treillis relationnels.

Prenons un exemple de requête relationnelle (voir section 6.5.3) :

$Q_r^{arr} = \text{"Quels sont les arrêtés qui parlent du niveau sonore (nvs) et qui font référence à des décrets sur les activités bruyantes (ab) ?"}$.

Les objets requête Q_{vo}^a et Q_{vo}^d sont respectivement classés dans les concepts a_{12} et d_4 comme illustré sur la figure 6.11. Le treillis des arrêtés est le treillis principal. Le graphe-réponse exact est $\mathcal{G} = \langle AS \rightarrow_{rf} O45 \rangle$. Si l'utilisateur relâche la contrainte relationnelle, il se retrouve dans le cas d'une requête simple. Sinon, il peut relâcher une contrainte sémantique sur le treillis des arrêtés. En retirant le descripteur *nvs*, l'utilisateur cherche tous les arrêtés faisant référence à des lois et décrets sur les activités bruyantes (*ab*) qui sont regroupés dans le concept a_{13} . Un graphe-réponse approché est donné par le document *AP* lié au document *D95* ($\mathcal{G} = \langle AP \rightarrow_{rf} D95 \rangle$). En retirant l'attribut relationnel *rf* : *c4*, l'utilisateur cherche les arrêtés qui parlent de niveau sonore *nvs* qui sont regroupés dans le concept a_3 . Une réponse approchée est donc donnée par les documents *AS* et *AY*.

L'utilisateur peut également relâcher des contraintes sémantiques sur le treillis des décrets si le concept requête simple sur ce treillis ne contient pas de réponse exacte.

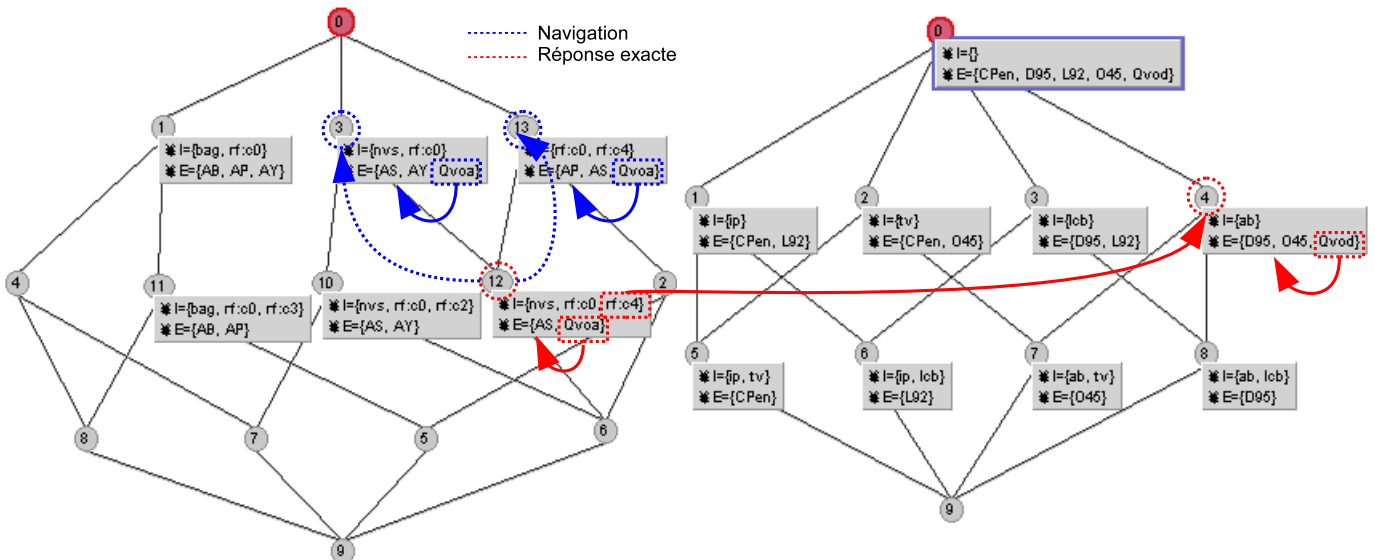


FIGURE 6.11 – Exemple de navigation par généralisation à partir d'une requête relationnelle

6.6.2 Recherche par exemple de documents

Le modèle permet aussi de parcourir la collection de documents en spécifiant un document ou un ensemble de documents plutôt qu'un ensemble d'attributs comme dans les requêtes ci-dessus. Dans ce cas, le document (ou l'ensemble de documents) est (sont) utilisé(s) comme un point de départ du processus de navigation. Cette fonction de requête par l'exemple a été définie dans le cadre de l'AFC en utilisant les attributs formels [Wray and Eklund, 2011], nous en proposons une extension dans le cadre de l'ARC pour prendre en compte les contraintes intertextuelles.

Cas1 : recherche par équivalence entre attributs de documents

Partant d'un document ou d'un ensemble de documents comme échantillon, il est facile de déterminer leurs caractéristiques communes et de retourner le concept correspondant à cet ensemble d'attributs.

L'utilisateur utilise un document ou un ensemble de documents de même type comme un échantillon et cherche les documents similaires. Ceci est équivalent à avoir une requête simple, sur le treillis correspondant au type de documents de l'échantillon, définie par les attributs communs des documents. Si ce treillis est enrichi avec des attributs relationnels, ils sont pris en compte dans l'ensemble des attributs communs. Ainsi, la similarité concerne à la fois les descripteurs sémantiques et les relations entre les documents.

Cette fonctionnalité est très utile dans plusieurs situations où l'utilisateur ne possède qu'un ensemble de documents au départ. Prenons par exemple le cas d'un secrétaire de mairie de la ville de Paris qui doit rédiger un arrêté local sur le bruit anormalement gênant. Pour commencer, le secrétaire de mairie cherche quelques arrêtés similaires, AB et AY (arrêtés de Boulogne-Billancourt et des Yvelines), issus de villes voisines (qui parlent du même thème). Il veut maintenant savoir quels sont les lois et les décrets qu'il doit citer dans le nouvel arrêté qu'il doit publier et s'il existe d'autres documents similaires à ceux qu'il possède déjà. Les documents qu'il possède comme échantillon appartiennent à l'extension du concept a_4 dans le treillis enrichi des arrêtés (figure 6.6). Puisque l'extension du concept a_4 ne contient pas de documents autres que AB et AY , le secrétaire de mairie sait qu'il n'y a pas de documents similaires disponibles. Cependant, outre les descripteurs sémantiques identifiés pour AB et AY (bag et ns : ces arrêtés parlent de bruit anormalement gênant et de nuisance sonore), l'intension de ce concept contient certains attributs relationnels ($ref : c2$ et $ref : c8$). Le secrétaire de mairie comprend donc qu'une référence similaire aux lois et décrets sur l'isolation phonique (ip) – le concept d_8 du treillis des décrets (figure 6.4) – peut être pertinente pour le nouvel arrêté.

Cas2 : recherche par mesure de similarité entre documents

Pour avoir une approche plus générale de recherche de documents similaires, nous définissons des mesures de distance et de similarité. Nous nous basons sur les mesures décrites dans [Ducrou et al., 2006] que nous étendons au cas de l'ARC. Deux objets o_1, o_2 sont ainsi similaires si leurs ensembles d'attributs o'_1 et o'_2 sont similaires. Un premier niveau de similarité est donné par le regroupement d'objets dans les concepts formels. La similarité est alors calculée sur des concepts pour permettre de retourner une liste triée de concepts similaires selon les mesures de distance et de similarité.

Notons par c le concept (E, I) tel que E est l'ensemble des documents de l'échantillon et I est l'ensemble des attributs formels et relationnels communs aux documents de l'échantillon. La distance entre un concept $c_i = (E_i, I_i)$ et le concept $c = (E, I)$ est défini comme suit :

Définition 26 (Distance entre concepts formels) [Ducrou et al., 2006] La distance entre un concept c_i et un concept donné c dans \mathcal{C} , ensemble des concepts d'un treillis \mathcal{L} correspondant à un contexte $\mathcal{K} = (O, A, Inc)$ est :

$$\begin{aligned} dist_f & : \mathcal{C} \times \mathcal{C} \longrightarrow [0, 1] \\ dist(c, c_i) = dist_f((E, I), (E_i, I_i)) & = \frac{1}{2} \left(\frac{|E - E_i| + |E_i - E|}{|O|} + \frac{|I - I_i| + |I_i - I|}{|A|} \right) \end{aligned}$$

Notons par $sim_f = 1 - dist_f$ la similarité déduite de cette mesure de distance. Nous étendons cette formule pour prendre en compte les attributs relationnels et nous définissons la similarité entre concepts d'un treillis relationnel comme suit :

Définition 27 (Similarité entre concepts relationnels) La similarité d'un concept c_i avec un concept donné c d'un treillis \mathcal{L}^+ correspondant à un contexte $\mathcal{K} = (O, A, Inc)$ enrichi avec la relation $r \in \mathbb{R}$ est :

$$\begin{aligned} sim_r & : \mathcal{C}^+ \times \mathcal{C}^+ \longrightarrow [0, 1] \\ sim_r(c, c_i) & = 1 - dist_r(c, c_i) \\ & = 1 - dist_r((E, I), (E_i, I_i)) \\ & = 1 - \frac{1}{2} \left(\frac{|E - E_i| + |E_i - E|}{|O|} + \frac{|I - I_i| + |I_i - I|}{|A| + |\mathcal{R}|} \right) \end{aligned}$$

Avec

\mathcal{L}^+ est le treillis résultant du scaling relationnel de \mathcal{L} avec la relation $r \in \mathbb{R}$ de la $FCR = (\mathbb{K}, \mathbb{R})$. \mathcal{C}^+ est l'ensemble des concepts de \mathcal{L}^+ .

$|O|$ est la cardinalité de l'ensemble des objets du contexte \mathcal{K} (objets dans les extensions de \mathcal{L}^+).

$|A|$ est la cardinalité de l'ensemble des attributs du contexte \mathcal{K} (attributs formels dans les intensions de \mathcal{L}^+).

$|\mathcal{R}|$ est la cardinalité de l'ensemble des attributs relationnels ajoutés au contexte \mathcal{K} après enrichissement relationnel avec la relation r (attributs relationnels dans les intensions des concepts de \mathcal{L}^+).

$E - E_i$ est la différence entre l'ensemble E et l'ensemble E_i : les objets appartenant à l'extension E du concept c et n'appartenant pas à l'extension E_i du concept c_i .

$I - I_i$ est la différence entre l'ensemble I et l'ensemble I_i : les attributs formels et relationnels appartenant à l'intension I du concept c et n'appartenant pas à l'intension I_i du concept c_i .

Dans la définition, l'intension d'un concept regroupe les attributs formels et relationnels qui sont tous les deux représentés par l'ensemble I . Nous désignons dans la suite par F l'ensemble les attributs formels, par R l'ensemble des attributs relationnels et par R_c l'ensemble des concepts référencés par les attributs relationnels.

Définition 28 (Concepts similaires) Considérons deux concepts c_1 et c_2 dans un treillis formel \mathcal{L} (resp. dans un treillis relationnel \mathcal{L}^+). Le concept c_1 est dit similaire à c_2 , $c_1 \sim_f c_2$ (resp. $c_1 \sim_r c_2$), si $sim_f(c_1, c_2) > v$ (resp. $sim_r(c_1, c_2) > v$), $v \in [0, 1]$.

Dans la suite, nous considérons que $c_1 \sim c_2$ (dans le cas général) si $sim(c_1, c_2) \geq 0.5$ et $c_1 \approx c_2$ sinon.

Exemples de calcul de similarité entre deux concepts relationnels : Soit le treillis enrichi des arrêtés \mathcal{L}_{arr}^+ de la figure 6.11.

Exemple 1 Similarité entre les concept a_{12} et a_{10}

- $a_{12} = (E, F, R) = (\{AS\}, \{nvs\}, \{rf : c4\})$
- $a_{10} = (E_i, F_i, R_i) = (\{AS, AY\}, \{nvs\}, \{rf : c2\})$
- $|O| = 4$: nombre total d'objets
- $|A| = 5$: nombre total d'attributs
- $|\mathcal{R}| = 9 - 1 = 8$: nombre total d'attributs relationnels - attribut relationnel vers le concept Top du treillis \mathcal{L}_{dec} ($rf : c0$)
- $sim_r(a_{12}, a_{10}) = 1 - \frac{1}{2} \left(\frac{0+1}{4} + \frac{0+0+1+1}{5+8} \right) = 0,79$
- $a_{12} \sim a_{10}$

Exemple 2 Similarité entre les concept a_{12} et a_{11}

- $a_{12} = (E, F, R) = (AS, nvs, rf : c4)$
- $a_{11} = (E_i, F_i, R_i) = (AB, AP, bag, rf : c3)$
- $sim_r(a_{12}, a_{11}) = 1 - \frac{1}{2} \left(\frac{1+2}{4} + \frac{1+1+1+1}{5+8} \right) = 0,47$
- $a_{12} \approx a_{11}$

La distance considère le nombre d'objets et d'attributs formels et relationnels appartenant exclusivement à chacun des concepts comparés, normalisé par le nombre total d'objets et d'attributs formels et relationnels. La similarité est égale à $1 - distance$.

Telle qu'elle est définie, cette mesure ne prend pas en compte les deux noeuds du graphe liés par la relation r . Le calcul est basé sur un treillis (\mathcal{L}^+) que nous considérons comme source de la relation, dans lequel nous pouvons naviguer pour trouver des concepts similaires au concept c . Le treillis \mathcal{L}' , correspondant au contexte co-domaine de la relation r utilisée pour l'enrichissement relationnel, n'est pas utilisé pour naviguer ou pour le calcul de similarité.

Le calcul de distance sur les attributs relationnels ($|R - R_i| + |R_i - R|$), considère les attributs relationnels au même niveau que les attributs formels, c.à.d que si deux attributs relationnels, sur une même relation, ne font pas référence au même concept, ils sont considérés comme différents. Ils sont donc comptés comme attributs exclusifs ce qui donnera une mesure de distance plus grande.

Sur l'exemple 1 :

- $R = \{rf : c4\}$
- $R_i = \{rf : c2\}$
- $c_4 \neq c_2 \Rightarrow$
- $R - R_i = \{rf : c4\}, |R - R_i| = 1$
- $R_i - R = \{rf : c2\}, |R_i - R| = 1$

Dans le cas où les deux concepts référencés par les attributs relationnels, c_4 et c_2 , sont similaires dans le treillis \mathcal{L}' (les enrichissements relationnels de \mathcal{L}' ne sont pas considérés), leur distance ($dist_f$) est plus petite (voir définition 26). Afin de prendre en compte la similarité sim_f des concepts référencés par les attributs relationnels de R et de R_i dans le treillis \mathcal{L}' , nous définissons une nouvelle mesure dans laquelle nous introduisons cette dimension pour le calcul de $|R - R_i|$.

Définition 29 (Différence entre ensembles de concepts) Soit \mathcal{C}' l'ensemble des concepts de \mathcal{L}' , $\mathfrak{P}(\mathcal{C}')$ est l'ensemble des parties de \mathcal{C}' . Soit A et B deux sous-ensembles de $\mathfrak{P}(\mathcal{C}')$, $A \subset$

$\mathfrak{P}(\mathcal{C}')$, $B \subset \mathfrak{P}(\mathcal{C}')$. La fonction \ominus calcule la différence entre les deux ensembles A et B en fonction de la similarité entre leurs concepts. \ominus est définie comme suit :

$$\begin{aligned} \ominus & : \mathfrak{P}(\mathcal{C}') \times \mathfrak{P}(\mathcal{C}') \longrightarrow \mathfrak{P}(\mathcal{C}') \\ A \ominus B & = A \setminus \{c_i \in A \mid \exists c_j \in B, \text{ avec } : c_i \sim_f c_j\} \end{aligned}$$

Pour chaque concept c_i de A , calculer sa similarité (\sim_f) avec tous les concepts c_j de B et ne garder que le c_i qui n'a aucun concept similaire c_j .

Reprenons l'exemple 1 :

- $R = \{rf : c_4\}, R_c = \{c_4\}$
- $R_i = \{rf : c_2\}, R_{c_i} = \{c_2\}$
- $sim_f(c_4, c_2) = 1 - \frac{1}{2} \left(\frac{1+1}{4} + \frac{1+1}{4} \right) = 0,5$
- $\Rightarrow c_4 \sim_f c_2$
- $R_c \ominus R_{c_i} = \emptyset, |R_c - R_{c_i}| = 0$
- $R_{c_i} \ominus R_c = \emptyset, |R_{c_i} - R_c| = 0$

Prendre en compte cette similarité entre concepts référencés permet de donner des valeurs de distance ($dist_r$) et de similarité (sim_r) qui sont plus précises. Nous modifions alors le calcul de la similarité sim_r (de la définition 27) entre deux concepts c et c_i comme suit :

Définition 30 La similarité d'un concept c_i avec un concept donné c d'un treillis \mathcal{L}^+ correspondant à un contexte $\mathcal{K} = (O, A, Inc)$ enrichi avec la relation $r \in \mathbb{R}$ est :

$$\begin{aligned} sim_r & : \mathcal{C}^+ \times \mathcal{C}^+ \longrightarrow [0, 1] \\ sim_r(c, c_i) & = 1 - \frac{1}{2} \left(\frac{|E - E_i| + |E_i - E|}{|O|} + \frac{|F - F_i| + |F_i - F| + |R_c \ominus R_{c_i}| + |R_{c_i} \ominus R_c|}{|A| + |\mathcal{R}|} \right) \end{aligned}$$

Modification sur les exemples de calcul de similarité entre deux concepts relationnels : Reprenons les exemples présentés ci-dessus :

Exemple 1 Similarité entre les concept a_{12} et a_{10}

- $a_{12} = (E, F, R) = (\{AS\}, \{nvs\}, \{rf : c_4\})$
- $a_{10} = (E_i, F_i, R_i) = (\{AS, AY\}, \{nvs\}, \{rf : c_2\})$
- $sim_r(a_{12}, a_{10}) = 1 - \frac{1}{2} \left(\frac{0+1}{4} + \frac{0+0+0+0}{5+8} \right) = 0,88$
- $a_{12} \sim a_{10}$

Exemple 2 Similarité entre les concept a_{12} et a_{11}

- $a_{12} = (E, F, R) = (\{AS\}, \{nvs\}, \{rf : c_4\})$
- $a_{11} = (E_i, F_i, R_i) = (\{AB, AP\}, \{bag\}, \{rf : c_3\})$
- $sim_r(a_{12}, a_{11}) = 1 - \frac{1}{2} \left(\frac{1+2}{4} + \frac{1+1+0+0}{5+8} \right) = 0,54$
- $a_{12} \sim a_{11}$ ($a_{12} \approx a_{11}$ avec la première mesure)

Synthèse

La recherche de concepts similaires a l'avantage de trouver des concepts proches. Outre la fonction de voisinage des concepts qui propose une méthode de navigation utile dans l'ensemble de données et le regroupement des objets similaires dans le même concept, la fonction de recherche par l'exemple d'objets en utilisant la similarité entre les concepts renvoie une liste de concepts

similaires, plutôt qu'une liste de documents. Les attributs des documents des concepts retournés permettent de montrer de quelle manière ces documents se rapportent à l'ensemble des documents dans l'échantillon.

Avec cette fonctionnalité nous avons la possibilité d'analyser le contexte d'interprétation d'un document donné par la visualisation de documents voisins. Cette fonctionnalité consiste à retourner la liste triée des documents plus ou moins similaires au document source. Les documents voisins sont groupés par classes présentées par des concepts formels dans la famille des treillis relationnels. Ces concepts sont classés par ordre décroissant de similarité par rapport au concept contenant le document initial. Les documents les plus similaires sont ceux contenus dans l'extension du concept contenant le document source et la similarité diminue à mesure que les documents partagent moins d'attributs sémantiques et relationnels avec la source.

La fonction de recherche par l'exemple de documents peut être considérée comme la fonction duale de la recherche avec un ensemble d'attributs (interrogation simple et relationnelle décrites dans les sections 6.5.2 et 6.5.3). Si la recherche par un ensemble défini d'attributs (A) ne renvoie pas d'objet (cela signifie que le concept (A', A) possède une extension vide) nous pouvons calculer les concepts similaires (avec une petite distance) et les retourner comme réponse approchée à l'utilisateur. Cette fonction est décrite dans la section suivante (section 6.6.3).

6.6.3 Recherche de réponses approchées

Dans certains cas, la requête de l'utilisateur n'a pas de réponse exacte. Cela se produit lorsqu'aucun document ne correspond exactement à toutes les propriétés spécifiées dans la requête et donc le concept requête possède une extension vide. Il est intéressant dans ce cas de retourner une réponse approchée à l'utilisateur ce qui est possible à partir de la structure des treillis sans avoir besoin de faire des calculs supplémentaires (avantage majeur de l'utilisation des treillis).

Ayant défini la distance et la similarité entre les concepts d'un treillis enrichi, nous pouvons utiliser ces mesures pour naviguer dans le treillis principal de la requête pour rechercher des concepts similaires et les classer en conséquence. Nous utilisons l'algorithme défini dans [Ducrou et al., 2006] que nous étendons à l'ARC.

Le processus de navigation commence par trouver les voisins possibles du concept de la requête et ensuite il parcourt le treillis pour les trier par ordre de pertinence. Le parcours du treillis est limité à une certaine largeur. Pour chaque concept visité, une condition de test est calculée pour vérifier si ce concept doit être utilisé pour élargir la navigation.

La condition de test dépend d'un paramètre de largeur de recherche (*SearchWidth*, qu'on notera σ), qui est spécifié par l'utilisateur pour rendre la recherche plus large ou plus étroite dans le treillis, et la distance entre le concept visité et le concept de la requête. Cette condition de test permet de ne garder dans l'ensemble des résultats que les concepts qui sont à une certaine distance du concept de la requête. À la fin, une liste triée des concepts pertinents est retournée au lieu d'un ensemble résultat initialement vide.

La condition de test prenant en compte les attributs relationnels est définie comme suit :

$$dist_r((E, I), (E_i, I_i)) \times \sigma < \frac{1}{2} \left(\frac{|E|}{|O|} + \frac{|F| + |R|}{|A| + |\mathcal{R}|} \right)$$

où E et I (respectivement E_i et I_i) sont l'extension et l'intension du concept de la requête (respectivement du concept visité), F est l'ensemble des attributs formels et R est l'ensemble des attributs relationnels du concept de la requête, O est l'ensemble total des objets, A est l'ensemble total des attributs formels et \mathcal{R} est l'ensemble total des attributs relationnels.

Exemple de réponse approchée à une requête simple

Considérons un exemple de requête simple sur le treillis des décrets :

$Q_s^{dec} =$ "Quels sont les lois et décrets qui parlent d'activités bruyantes (*ab*) et d'isolation phonique (*ip*) ?".

La requête ne possède pas de réponse exacte. Un nouveau concept d_{10} avec une extension vide (qui contient uniquement l'objet virtuel de la requête Q_{vod}) est ajouté au treillis. Dans ce cas, l'algorithme de recherche et de navigation effectue un parcours des concepts du treillis des décrets (comme le montre la figure 6.12) afin d'identifier les concepts similaires contenant les réponses approchées potentielles.

Pour $\sigma = 0.8$, les étapes de parcours du treillis et les concepts similaires ajoutés à chaque étape pour calculer une réponse approchée à $Q_s^{dec} = (Q_{vod}, \{ab, ip\})$ sont comme suit :

- 1- d_{10} contient le seul objet virtuel Q_s^{dec}
- 2- d_4 $d_f(d_{10}, d_4) = 1/2((0 + 2)/5 + (1 + 0)/4) = 0.325$
 $d_f(d_{10}, d_4) \times \sigma < 1/2(1/5 + 2/4)(= 0.35)$
 d_4 est utilisé pour étendre la navigation
 $sim_f(d_{10}, d_4) = 0.675$
- d_8 $d_r(d_{10}, d_8) = 0.325$
 $d_r(d_{10}, d_8) \times \sigma < 0.35$
 d_8 est utilisé pour étendre la navigation
 $sim_f(d_{10}, d_8) = 0.675$
- 3- d_0 $d_f(d_{10}, d_3) = 1/2((1 + 1)/5 + (1 + 1)/4) = 0.45$
 $d_f(d_{10}, d_3) \times \sigma > 0.35$
 d_3 n'est pas utilisé pour étendre la navigation
 $sim_f(d_{10}, d_0) = 0.55$
- d_6 $d_f(d_{10}, d_6) = 0.45$
 $d_f(d_{10}, d_6) \times \sigma > 0.35$
 d_6 n'est pas utilisé pour étendre la navigation
 $sim_f(d_{10}, d_6) = 0.55$
- d_3 $d_f(d_{10}, d_3) = 0.45$
 $d_f(d_{10}, d_3) \times \sigma > 0.35$
 d_3 n'est pas utilisé pour étendre la navigation
 $sim_f(d_{10}, d_0) = 0.55$
- d_9 $d_f(d_{10}, d_9) = 0.45$
 $d_f(d_{10}, d_9) \times \sigma > 0.35$
 d_9 n'est pas utilisé pour étendre la navigation
 $sim_f(d_{10}, d_9) = 0.55$

Au lieu de trier des documents, des classes de documents (représentées par des concepts) sont triées en utilisant la mesure de similarité détaillée dans la définition 26 après avoir effectué un parcours du treillis.

Par exemple, sur la figure 6.12, le concept d_4 (contenant des documents sur les *activités bruyantes*) a un score de similarité égal à 0.675 et est classé avant d_3 (contenant des documents sur *l'isolation phonique* et *tranquillité du voisinage*) qui a un score de similarité 0.55.

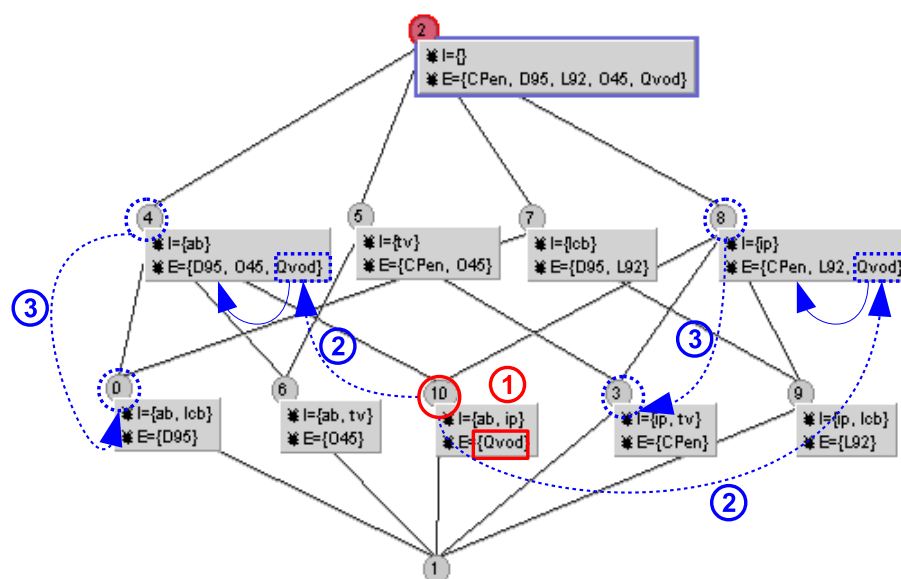


FIGURE 6.12 – Un exemple de navigation pour retourner des réponses approchées dans le cas d'une requête simple

Exemple de réponse approchée à une requête relationnelle

On peut faire le même raisonnement pour les requêtes relationnelles, en ajoutant les attributs relationnels dans le calcul de distance et similarité.

Considérons l'exemple suivant de requête relationnelle sur les treillis des arrêtés et des décrets :

$Q_r^{arr} = \text{"Quels sont les arrêtés qui parlent de bruit anormalement gênant (bag) et de nuisance sonore (ns) et qui font référence à des décrets sur les activités bruyantes (ab) ?"}$.

L'objet requête Q_{vod} sur le treillis de décrets est classé dans le concept d_4 (comme le montre la figure 6.13). Cette sous-requête possède comme réponse exacte les documents $D95$ et $O45$. La sous-requête sur le treillis des arrêtés ne possède pas de réponse exacte. Un nouveau concept, a_{12} , avec une extension vide (contenant uniquement l'objet virtuel de la requête Q_{voo}) est ajouté au treillis enrichi des arrêtés. Dans ce cas, l'algorithme effectue un parcours du treillis. Le treillis concerné par la navigation est celui des arrêtés.

6.7 Algorithmes d'interrogation et de navigation

Dans cette section, nous présentons les algorithmes d'interrogation et de navigation qui ont été utilisés dans les exemples des sections précédentes et qui sont testés dans le chapitre 8. L'algorithme 2 décrit l'algorithme général de recherche relationnelle qui se compose d'un ensemble de procédures : enrichissement des contextes (algorithme 3), construction des treillis (algorithme MultiFCA [Rouane et al., 2007]) et construction de réponses (algorithme 4). Les algorithmes 5 et 6 décrivent les procédures de construction de réponses exactes et approchées.

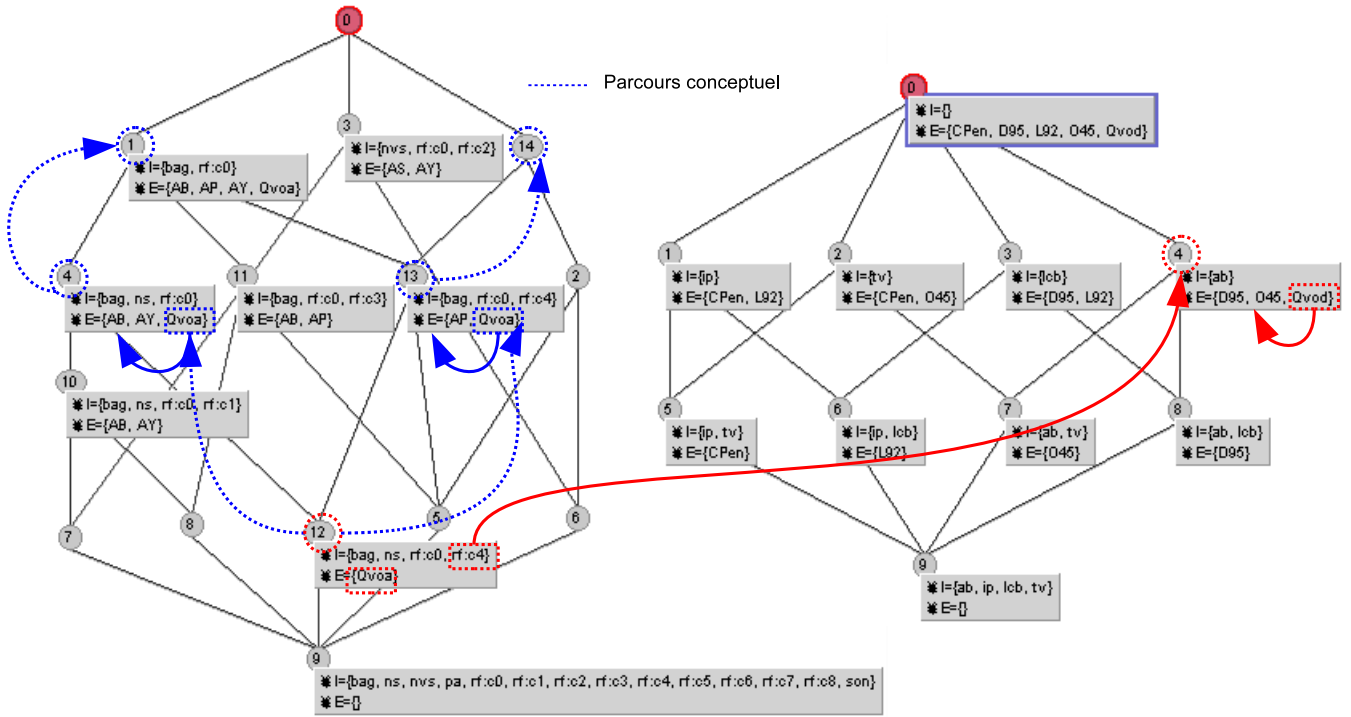


FIGURE 6.13 – Un exemple de navigation pour retourner des réponses approchées dans le cas d’une requête relationnelle

Algorithm 2 Recherche relationnelle

Require: $FCR = (\mathbb{K}, \mathbb{R})$ // famille de contextes relationnels

$Q_r = (\mathcal{C}, \mathcal{R})$ // requête relationnelle (voir définition 23)

Ensure: $FTR = \{\mathcal{L}_Q^+\}$ // ensemble de treillis relationnels correspondant à FCR

$G = (\mathcal{O}_G, \mathcal{R}_G)$ // graphe résultat (voir définition 25)

- 1: Enrichir (\mathbb{K}, \mathbb{R}) par $Q_r = (\mathcal{C}, \mathcal{R})$: $(\mathbb{K}_Q, \mathbb{R}_Q) := (\mathbb{K}, \mathbb{R}) \oplus_R Q_r$
 - 2: Construire $\{\mathcal{L}_Q^+\}$ correspondant à $(\mathbb{K}_Q, \mathbb{R}_Q)$ // utilisation de l’algorithme MultiFCA
 - 3: Construire $G = (\mathcal{O}_G, \mathcal{R}_G)$, le graphe réponse à Q_r
-

Algorithm 3 Enrichir FCR avec $Q_r : "\oplus_R"$ **Require:** $FCR = (\mathbb{K}, \mathbb{R})$ // famille de contextes relationnels $Q_r = (\mathcal{C}, \mathcal{R})$ // requête relationnelle (voir définition 23)**Ensure:** $(\mathbb{K}_Q, \mathbb{R}_Q)$ // famille de contextes enrichies par la requête

// Enrichissement des contextes formels

- 1: **for** $Q_{s,i} \in \mathcal{C}$ **do**
- 2: // ajouter $Q_{s,i}$ à \mathbb{K}_i correspondant
- 3: $O_i := O_i \cup \{Q_{vo,i}\}$ // ajouter $Q_{vo,i}$ à l'ensemble des objets du contexte
- 4: $A_i := A_i \cup \{a_i\}$ // ajouter les attributs de la requête à l'ensemble d'attributs du contexte
- 5: $Inc := Inc \cup \{Q_{vo,i}\} \times \{a_i\}$ // ajouter la relation d'incidence de la requête à celle du contexte
- 6: **end for** // Enrichissement des contextes relationnels
- 7: **for** $R_k \in \mathcal{R}$ **do**
- 8: // ajouter les relations de la requête au contexte relationnel $r_k \in \mathbb{R}$ correspondant
- 9: $O_i := O_i \cup \{Q_{vo,i}\}$
- 10: $O_j := O_j \cup \{o_j\}$
- 11: $Inc := Inc \cup \{Q_{vo,i}\} \times \{o_j\}$ // ajouter la relation d'incidence de la requête à celle du contexte relationnel
- 12: **end for**

Algorithm 4 Construire $G = (\mathcal{O}_G, \mathcal{R}_G)$, le graphe réponse à Q_r **Require:** $FTR = \{\mathcal{L}_Q^+\}$ // ensemble de treillis relationnels correspondant à $(\mathbb{K}_Q, \mathbb{R}_Q)$ $Q_r = (\mathcal{C}, \mathcal{R})$ // requête relationnelle (voir définition 23)**Ensure:** $G = (\mathcal{O}_G, \mathcal{R}_G)$ // graphe résultat (voir définition 25)

- 1: **for** $Q_{s,i} \in \mathcal{C}$ **do**
- 2: $C_Q = (Q'_I, Q_I) := \text{Localiser } Q_{s,i} \text{ dans } \{\mathcal{L}_{Q,i}^+\}$
- 3: **if** $Q'_I \setminus \{Q_{vo,i}\} \neq \emptyset$ **then**
- 4: Construire réponse exacte à partir de C_Q
- 5: **else**
- 6: Construire réponse approchée à partir de C_Q
- 7: **end if**
- 8: **end for**

Algorithm 5 Construire réponse exacte à partir d'un concept $C_Q = (Q'_I, Q_I)$ **Require:** $FTR = \{\mathcal{L}_Q^+\}$ // ensemble de treillis relationnels correspondant à $(\mathbb{K}_Q, \mathbb{R}_Q)$ $C_Q = (Q'_I, Q_I)$ // Concept identifié pertinent**Ensure:** $G = (\mathcal{O}_G, \mathcal{R}_G)$ // graphe résultat (voir définition 25)

- 1: $\mathcal{O}_G := \mathcal{O}_G \cup Q'_I \setminus \{Q_{vo,i}\}$
- 2: **for** $rel_k \in Q_I \cap \mathcal{R}$ **do**
- 3: $C_j :=$ Le concept référencé par rel_k
- 4: $C_j = (E_j, I_j) := \text{Localiser } C_j \text{ dans } \text{mathcal{L}}_j$
- 5: $\mathcal{O}_G := \mathcal{O}_G \cup E_j \setminus \{Q_{vo,j}\}$
- 6: $\mathcal{R}_G := \mathcal{R}_G \cup rel_k$
- 7: **end for**

Algorithm 6 Construire réponse approchée à partir d'un concept $C_Q = (Q'_I, Q_I)$

Require: $FTR = \{\mathcal{L}_Q^+\}$ // ensemble de treillis relationnels correspondant à $(\mathbb{K}_Q, \mathbb{R}_Q)$

$C_Q = (Q'_I, Q_I)$ // Concept identifié pertinent

Ensure: $G = (\mathcal{O}_G, \mathcal{R}_G)$ // graphe résultat (voir définition 25)

- 1: $\mathcal{V} := \text{Voisins}(C_Q)$
 - 2: **for** $V_i \in \mathcal{V}$ **do**
 - 3: $G := G \cup \text{Construire réponse exacte à partir de } V_i$
 - 4: **if** V_i vérifie la contrainte de distance (Equation 6.6.3) **then**
 - 5: Construire réponse approchée à partir de V_i
 - 6: **end if**
 - 7: **end for**
-

6.8 Requêtes exprimables par le modèle

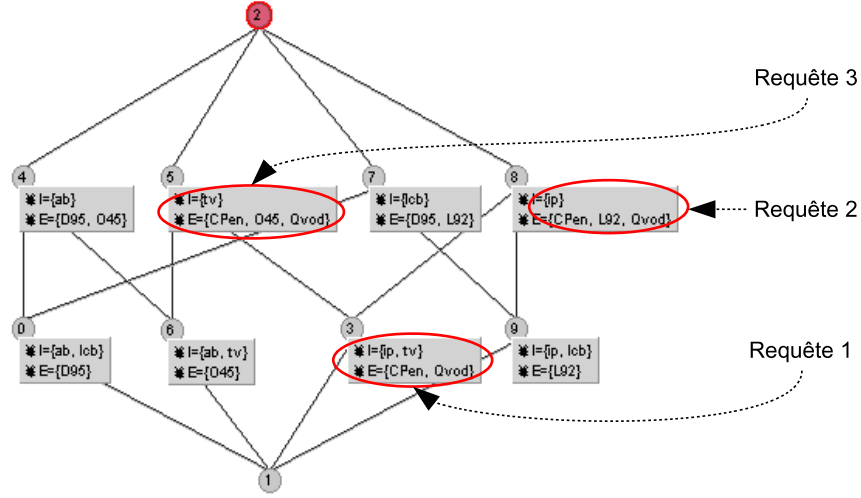
Le modèle relationnel que nous proposons permet de répondre, en plus des requêtes simples (RS) décrites par un ensemble de descripteurs de contenu, à de nouvelles formes de requêtes, les requêtes relationnelles (RR). Ce modèle permet de retrouver à la fois des documents qui portent sur un sujet donné et qui sont liés à d'autres documents ou classes de documents avec des types spécifiques de relations. Le modèle permet aussi de retourner des réponses approchées dans le cas où la requête ne possède pas de réponses exactes. Ceci est possible grâce à la structure conceptuelle hiérarchique construite sur la collection de documents.

Dans ces deux cas, les objets retrouvés dans les deux étapes partagent un attribut (formel et/ou relationnel) ou une conjonction d'attributs (formels et/ou relationnels) avec la requête. De cette manière, le traitement d'une requête simple ($\mathcal{Q}_s = (Q_{vo}; A_q)$) ou relationnelle (avec des relations \mathcal{R}_k) est équivalent au traitement d'une suite de requêtes conjonctives. La première requête est formée par la conjonction de tous les attributs dans A_q plus les attributs relationnels de \mathcal{R}_k . Les requêtes suivantes sont formées par la conjonction des sous ensembles de A_q et de \mathcal{R}_k jusqu'aux requêtes formées par un seul attribut de A_q ou de \mathcal{R}_k . Le résultat final est formé par l'union des résultats de chaque requête.

Exemple Pour illustrer la décomposition de la requête en un ensemble de requêtes conjonctives, prenons l'exemple d'une requête simple sur le treillis des décrets : *décrets sur l'isolation phonique (ip) et la tranquillité du voisinage (tv)*. $Q_s^{dec} = (Q_E^{dec}, Q_I^{dec}) = (Q_{vo}^d, \{ip, tv\})$. Les requêtes conjonctives et les résultats qui leur correspondent sont donnés par la figure 6.14 et sont détaillés comme suit :

Requête1	$ip \wedge tv$	-	$CPen$
Requête2	ip	-	$CPen$ et $L92$
Requête3	tv	-	$CPen$ et $O45$

Formalisation Dans notre modélisation, les requêtes simples ou relationnelles peuvent porter, en plus des attributs formels ou relationnels, sur un ou plusieurs types de documents. Comme décrit dans la section 6.2, nous créons un contexte formel par type de document ce qui permet de considérer le type de document comme paramètre dans la formalisation des requêtes. Après enrichissement, les relations sont exprimées par des attributs relationnels qui sont représentés de la même manière que les attributs formels. Nous exprimons ci-dessous ces requêtes de manière

FIGURE 6.14 – Conjonction de requêtes simples sur le treillis des décrets \mathcal{L}_{dec} .

formelle en utilisant le vocabulaire décrit dans le tableau suivant :

Types	T_1, T_2, \dots
Descripteurs	D_1, D_2, \dots
Attributs relationnels	R_1, R_2, \dots

- Une RS sur un treillis de concepts \mathcal{L} correspondant à un contexte formel $\mathcal{K} = (O, A, Inc)$ représentant un type de document (T) et portant sur un ou plusieurs descripteurs sémantiques ($D_i, i = 1..N$ avec $N = |A|$) est décrite comme suit.

$$(D_1 \wedge D_2 \wedge \dots \wedge D_i) \wedge T = \left(\bigwedge_{i \in N} D_i \right) \wedge T$$

$$i = 1..N, N = |A|$$

La réponse à cette requête est localisée dans un concept du treillis et dans ses super-concepts (comme montré dans les sections précédentes). Chaque concept contient dans son extension un ensemble d'objets qui représente la réponse à une requête caractérisée par tout ou une partie des attributs de la requête. Ainsi, répondre à la requête initiale revient à donner une réponse à une conjonction de RS (voir exemple figure 6.14). Ce cas exprime le *ET* logique.

- Étant donné un treillis de concepts \mathcal{L} , il est aussi possible de répondre à plusieurs requêtes disjonctives par navigation dans le treillis. En effet, les concepts fermés du treillis (qui ont des intensions exclusives) représentent pour chacun une requête décrite par un ensemble propre de descripteurs (qui n'est pas partagé même en partie avec une autre requête). Ce cas exprime le *OU* logique. Si on considère que chaque requête est décrite par un seul descripteur sur l'ensemble des concepts fermés de \mathcal{L} , soit \mathcal{F} , ce cas se présente formellement comme suit.

$$(D_1 \vee D_2 \vee \dots \vee D_j) \wedge T = \left(\bigvee_{j \in M} D_j \right) \wedge T$$

$$j = 1..M, M = |\mathcal{F}|$$

- Le cas général est donné par la combinaison des cas précédents :

$$((D_1 \wedge D_2 \wedge \dots \wedge D_i)_{i=1} \vee \dots \vee (D_1 \wedge D_2 \wedge \dots \wedge D_i)_{i=j}) \wedge T = \left(\bigvee_{j \in M} \left(\bigwedge_{i \in N} D_{ij} \right) \right) \wedge T$$

$$i = 1..N, N = |A|$$

$$j = 1..M, M = |\mathcal{F}|$$

Dans le cas général, l'ensemble des RS que nous pouvons exprimer sur une collection de documents représentée par un treillis de concepts est décrit par une disjonction de conjonction d'un ensemble ou d'une partie d'un ensemble de descripteurs de contenu de ces documents.

- Les RR exprimables sur une FTR correspondant à une $FCR = (\mathbb{K}, \mathbb{R})$ représentant une collection documentaire avec plusieurs types de documents ($T_l, l = 1..S$ avec $S = |\mathbb{K}|$) reliés entre eux par différentes relations ($R_k, k = 1..P$ avec $P = |\mathbb{R}|$) et portant sur un ou plusieurs descripteurs sémantiques ($D_i, i = 0..N$ avec $N = |A|$ sur un contexte dans \mathbb{K}) sont décrites formellement comme suit.

$$\begin{aligned} & ((D_1 \wedge \dots \wedge D_i \wedge R_1 \wedge \dots \wedge R_k) \wedge T_1) \wedge \dots \wedge ((D_1 \wedge \dots \wedge D_i \wedge R_1 \wedge \dots \wedge R_k) \wedge T_l) \\ = & \left(\left(\bigwedge_{i \in N} D_i \wedge \bigwedge_{k \in P} R_k \right) \wedge T_1 \right) \wedge \dots \wedge \left(\left(\bigwedge_{i \in N} D_i \wedge \bigwedge_{k \in P} R_k \right) \wedge T_l \right) \\ = & \bigwedge_{l \in S} \left(\bigwedge_{i \in N} D_i \wedge \bigwedge_{k \in P} R_k \wedge T_l \right) \\ & i = 0..N, N = |A| \\ & k = 1..P, P = |\mathbb{R}| \\ & l = 1..S, S = |\mathbb{K}| \end{aligned}$$

La réponse à cette requête est donnée par un (ou plusieurs) graphe(s) dont les noeuds sont les concepts (ou les super-concepts) des treillis de la FTR et les arêtes sont les relations définies par les attributs relationnels. Les noeuds de chaque graphe contiennent dans leurs extensions un ensemble d'objets qui représente la réponse à une requête caractérisée par tout ou partie des attributs formels (si $i \neq 0$) ou relationnels de la requête (que des attributs relationnels si $i = 0$). Ainsi, répondre à la requête initiale revient à donner une réponse à une conjonction de RR. Ce cas exprime le *ET* logique.

- Étant donné une FTR, il est aussi possible de répondre à plusieurs RR disjonctives par navigation dans le treillis (comme décrit pour le cas des RS). Si on considère que chaque requête est décrite par un seul descripteur sémantique et/ou un seul attribut relationnel,

ce cas se présente formellement comme suit.

$$\begin{aligned}
& ((D_1 \vee \dots \vee D_j \wedge R_1 \vee \dots \vee R_k) \wedge T_1) \wedge \dots \wedge ((D_1 \vee \dots \vee D_j \wedge R_1 \vee \dots \vee R_k) \wedge T_l) \\
= & \bigwedge_{l \in S} \left(\bigvee_{j \in M} D_j \wedge \bigvee_{k \in P} R_k \wedge T_l \right) \\
& j = 1..M, M = |\mathcal{F}| \\
& k = 1..P, P = |\mathbb{R}| \\
& l = 1..S, S = |\mathbb{K}|
\end{aligned}$$

– Le cas général est donné par la combinaison des deux cas relatifs aux RR :

$$\bigwedge_{l \in S} \left(\bigvee_{j=1..M} \left(\bigwedge_{i=0..N} D_{ij} \right) \wedge \bigvee_{k=1..P} \left(\bigwedge_{k=0..P} R_k \right) \wedge T_l \right)$$

Ce cas est le plus général puisqu'il combine les RS et les RR.

- | | |
|--------------|--|
| Si $i=0$, | la RR ne porte pas sur les descripteurs D_i , mais sur le type T_l et sur les relations R_k (i.e. on cherche tout les documents d'un type donné et qui possèdent des relations vers d'autres documents ; |
| Si $k=0$, | la RR ne porte pas sur les relations R_k , mais sur les D_i et le type T_l . On est donc dans le cas d'une RS qui porte sur le contenu d'un document de type donné ; |
| Si $i=k=0$, | la requête n'est pas valide (une requête concerne au moins un aspect : contenu ou relation). |

Le tableau 6.5 donne un récapitulatif des différents types de requêtes simples et relationnelles que nous pouvons exprimer dans le modèle conceptuel de la collection documentaire et la compare avec la typologie des requêtes décrite dans le chapitre 5.

L'étude de l'expressivité que nous venons d'effectuer montre que le modèle tel qu'il est présenté dans ce travail ne permet de satisfaire qu'une partie des requêtes identifiées dans l'analyse des besoins. Au vue des propriétés des requêtes types listées dans le chapitre 5, le modèle FCA/RCA :

- permet de gérer la complexité structurelle des requêtes (nombre de relations, réflexivité, cycles) au moment de l'enrichissement relationnel,
- permet l'utilisation des variables pour désigner les documents et les attributs mais n'autorise pas la variabilisation des relations (définies par les contextes relationnels) et des types (définis par les contextes formels),
- traite la cible et les contraintes par filtrage des résultats retournées,
- ne permet pas de traiter les présupposés d'unicité dans les requêtes en langage naturel. S'il existe plus d'une réponse, elles sont toutes retournées.

L'étude de l'expressivité montre également que le modèle permet de répondre à d'autres types de requêtes, les requêtes disjonctives, qui ne sont pas prises en compte dans la typologie du chapitre 5. Ceci est possible grâce à la navigation dans la structure des treillis sans effectuer de calculs supplémentaires.

TABLE 6.5 – Tableau récapitulatif de la typologie des requêtes exprimables par l’AFC et l’ARC et leur correspondance avec les requêtes-types issues de l’analyse des besoins.

	Formalisation	Correspondance requêtes-types
Requêtes simples		
RS (ET)	$(\bigwedge_{i \in N} D_i) \wedge T$	RT1-1, RT1-2, RT1-4, RT1-5
RS (OU)	$(\bigvee_{j \in M} D_j) \wedge T$	non exprimable par le langage de requêtes
RS (OU(ET))	$(\bigvee_{j \in M} (\bigwedge_{i \in N} D_{ij})) \wedge T$	non exprimable par le langage de requêtes
Requêtes relationnelles		
RR (ET)	$\bigwedge_{l \in S} (\bigwedge_{i \in N} D_i \wedge \bigwedge_{k \in P} R_k \wedge T_l)$	RT2-1, RT2-2, RT2-4, RT3-1, RT3-2, RT4-1, RT4-2
RR (OU)	$\bigwedge_{l \in S} (\bigvee_{j \in M} D_j \wedge \bigvee_{k \in P} R_k \wedge T_l)$	non exprimable par le langage de requêtes
RR (OU(ET))	$\bigwedge_{l \in S} \bigvee_{j=1..M} (\bigwedge_{i=0..N} D_{ij}) \wedge \bigvee_{k=1..P} (\bigwedge_{k=0..P} R_k) \wedge T_l$	non exprimable par le langage de requêtes

6.9 Conclusion

Nous avons présenté une modélisation qui donne une représentation unifiée des descripteurs de contenus et des relations intertextuelles caractérisant une collection documentaire. Cette modélisation est basée sur l’AFC et l’ARC que nous avons appliqué à des collections documentaires. Nous avons étendu les propositions d’interrogation et de navigation de l’AFC à l’ARC et défini un algorithme pour l’exploitation des structures relationnelles construites.

La figure 6.15 donne une vue globale de l’approche décrite dans ce chapitre qui se compose de quatre étapes principales :

1. Modélisation du contenu sémantique : le contenu sémantique des documents est annoté et les contextes formels sont extraits en fonction de ces annotations permettant la construction des treillis formels.
2. Modélisation de la structure intertextuelle : les liens entre les documents sont identifiés et les contextes relationnels sont extraits sur la base de ces liens permettant la construction de treillis relationnels enrichis.
3. Interrogation : l’utilisateur crée une requête, qui peut être une combinaison de descripteurs sémantiques de contenu et de contraintes sur les liens intertextuels.
4. Construction des résultats : l’algorithme de recherche analyse la requête et cherche des réponses pertinentes dans les treillis. L’utilisateur peut avoir en réponse une liste de documents ou de graphes de documents. Il peut également naviguer dans la structure relationnelle construite.

Par rapport à l’analyse des besoins, cette approche permet de répondre à l’ensemble des requêtes (simples et relationnelles) exprimées dans le chapitre 5, de répondre à d’autres types de requêtes (requêtes disjonctives) grâce à la navigation, de retourner des réponses plus riches (sous forme de graphes et pas que des documents isolés) et de fournir des réponses approchées à l’utilisateur en l’absence de réponses exactes. Cependant, elle ne permet pas de traiter tous

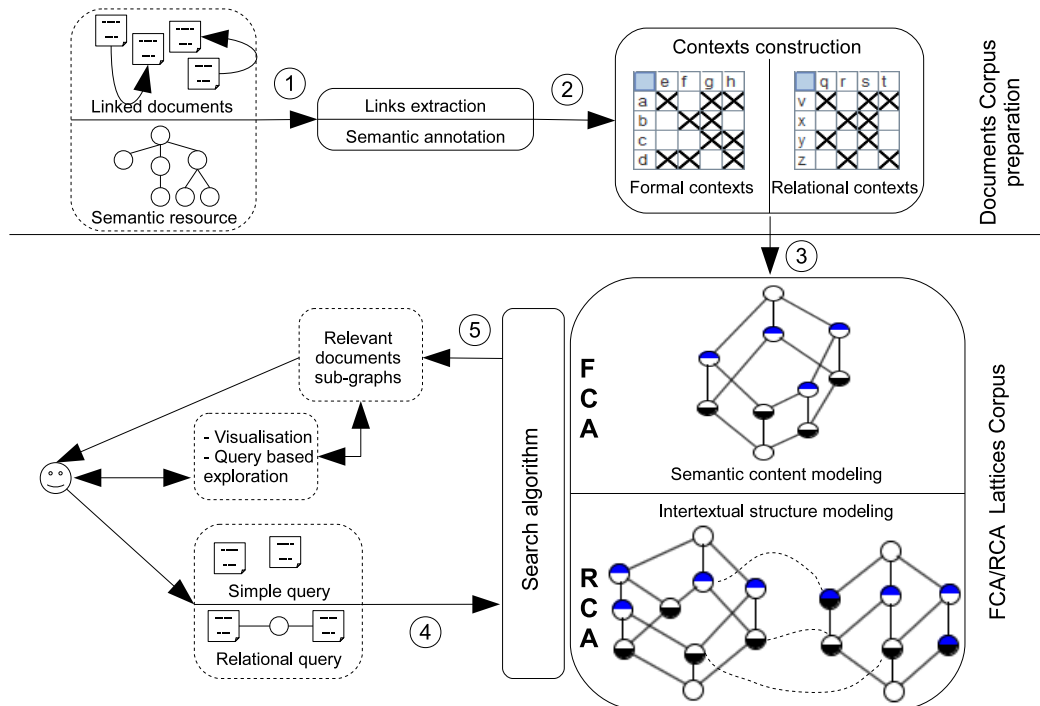


FIGURE 6.15 – Aperçu de l'approche conceptuelle de RI relationnelle.

les types de requêtes ni d'avoir un processus homogène pour le traitement et ne permet pas de travailler sur des collections de grande taille. Nous proposons des solutions à ces points avec l'approche sémantique que nous décrivons dans le chapitre suivant.

Chapitre 7

RI et intertextualité : approche sémantique

Sommaire

7.1	Introduction	139
7.2	Bonnes pratiques pour la construction de vocabulaires	140
7.3	Première ontologie documentaire	141
7.3.1	Structure globale de l'ontologie	142
7.3.2	Modélisation de la collection documentaire	144
7.3.3	Modélisation des documents	147
7.3.4	Modélisation sémantique des contenus textuels	152
7.4	Deuxième ontologie documentaire	154
7.4.1	Gestion des versions d'un document	156
7.4.2	Gestion des références	156
7.4.3	Structure globale de l'ontologie	163
7.4.4	Positionnement par rapport au standard juridique Metalex	165
7.5	Mise en œuvre des ontologies documentaires	166
7.5.1	Instanciation et interrogation dans la première ontologie	166
7.5.2	Instanciation et interrogation dans la deuxième ontologie	173
7.6	Conclusion	179

7.1 Introduction

Dans ce chapitre nous décrivons une approche différente dans le but de modéliser la collection puis de l'interroger. Nous proposons un modèle basé sur les technologies du web sémantique, qui permet de passer à l'échelle. Dans cette approche, l'effort ne porte pas tant sur l'interrogation de la collection documentaire que sur sa modélisation. De ce point de vue, cette approche diffère clairement de la précédente qui reposait sur un modèle <objets × attributs> simple.

Dans l'approche sémantique, on peut proposer un modèle documentaire beaucoup plus riche et l'essentiel de notre effort a consisté à intégrer la dimension intertextuelle dans une ontologie documentaire adaptée aux documents juridiques. Une telle ontologie permet de représenter le contenu sémantique du document (ce dont parle le document), sa structure logique, ses différentes versions et son cycle de vie, ainsi que la structure de la collection documentaire qui organise différents types de documents dans un vaste réseau de liens intertextuels.

Nous proposons de représenter les relations entre les documents de deux manières différentes. Dans une première ontologie, les relations entre les documents sont représentées comme des liens directs entre les classes (des propriétés d'objets). Ce choix de modélisation représente une première vision naïve des liens intertextuels dans le domaine juridique. Dans un deuxième temps, nous présentons une ébauche de deuxième ontologie où les relations sont décrites comme de véritables opérations documentaires. Nous verrons que ce deuxième modèle rend même compte de la dynamique des collections juridiques. Une fois une collection documentaire modélisée comme une instanciation de cette ontologie, les requêtes relationnelles peuvent se traduire facilement sous la forme de requêtes SPARQL.

La suite du chapitre est organisée comme suit. La section 7.2 décrit les bonnes pratiques et les règles que nous avons suivies pour la création des ontologies. La section 7.3 présente l'ontologie documentaire traitant les relations comme des liens directs, avec les différents modules la composant et leurs dépendances. La section 7.4 présente l'ontologie documentaire où les relations sont modélisées comme des opérations. Des exemples d'utilisation de ces ontologies pour la modélisation d'une collection juridique et l'interrogation avec des requêtes relationnelles sont décrits dans la section 7.5.

7.2 Bonnes pratiques pour la construction de vocabulaires

Dans la conception des deux ontologies documentaires, nous avons essayé autant que possible de suivre les recommandations et les bonnes pratiques pour construire un vocabulaire dans le cadre du web de données⁸⁸. Les ontologies documentaires créées réutilisent des vocabulaires largement déployés dans le web de données. Elles suivent une approche légère (*lightweight*) puisqu'elles se basent essentiellement sur des assertions de base en RDFS et OWL. Nous avons également veillé à documenter chaque nouveau terme avec des étiquettes et des commentaires. Pour faciliter la manipulation du vocabulaire, nous avons aussi défini des propriétés inverses pour les principales propriétés d'objets.

Pour la réutilisation de vocabulaires, nous avons adopté la stratégie de conception recommandée par la communauté de web de données qui consiste en premier lieu à rechercher des termes de vocabulaires largement utilisés qui pourraient être réutilisés pour représenter les données ; si ces vocabulaires ne fournissent pas tous les termes qui sont nécessaires pour décrire le contenu complet d'un ensemble de données, les termes requis doivent être définis comme un vocabulaire propriétaire (avec un espace de noms (*namespace*) contrôlé par le concepteur) et utilisés en complément des termes de ces vocabulaires [Heath and Bizer, 2011]. Nous avons aussi utilisé des propriétés de RDFS et OWL pour relier les nouveaux termes à ceux de vocabulaires existants.

Les ontologies documentaires développées réutilisent les vocabulaires de référence suivants :

Le schéma Dublin Core : un schéma de métadonnées générique qui permet de décrire des ressources numériques ou physiques et d'établir des relations avec d'autres ressources⁸⁹.
Préfixe : `dc` (ou aussi `dce`).

L'ontologie DCMI terms : une spécification mise à jour de tous les termes de métadonnées gérés par la *Dublin Core Metadata Initiative* (DCMI), y compris les propriétés, les schémas de codage de vocabulaire, des schémas de codage de syntaxe, et les classes⁹⁰. Préfixe : `dct` (ou aussi `dcterms`).

⁸⁸. Le vocabulaire définit dans le cadre de ce travail ne peut pas être publié sur le web (selon les recommandations du web de données) puisqu'il est développé dans le cadre d'un projet avec des partenaires industriels.

⁸⁹. <http://purl.org/dc/elements/1.1/>

⁹⁰. <http://purl.org/dc/terms/>

L'ontologie WGS84 Geo Positioning : représente tout objet avec étendue spatiale (position, taille, etc.) comme par exemple les personnes et les lieux⁹¹. Préfixe : `geo`.

L'ontologie Metalex : un schéma OWL de l'*Open XML Interchange Format for Legal and Legislative Resources*⁹². Préfixe : `metalex`. Nous réutilisons plusieurs termes de cette ontologie qui sont des termes propres à Metalex ou des termes empruntés à d'autres vocabulaires très déployés, tels que :

L'ontologie FRBR (*Functional Requirements for Bibliographic Records* ou Fonctionnalités requises des notices bibliographiques) de l'IFLA⁹³ : un modèle conceptuel de données bibliographiques qui schématise le processus intellectuel du catalogage (modéliser les documents en tant qu'entités qui vont du plus concret au plus abstrait)⁹⁴. Préfixe : `frbr`.

L'ontologie Event : traite la notion d'événement réifiée. Elle définit un concept principal `Event`. Un événement peut avoir un lieu, une date, des agents actifs, des facteurs et des produits⁹⁵. Préfixe : `event`.

L'ontologie `bibo` définit les termes `bibo:Document`, `bibo:Collection` et `bibo:LegalDocument` mais dans un contexte différent de notre objectif (contexte bibliographique). Nous n'utilisons pas ces termes dans les ontologies documentaires que nous créons.

Nous avons utilisé Protégé et TopBraid Composer pour le développement des deux ontologies documentaires que nous détaillons dans ce chapitre. Nous avons utilisé Corese⁹⁶ dans un premier temps pour l'interrogation avec SPARQL d'un premier ensemble de données représentées par un schéma RDFS simple décrivant les documents et leurs annotations sémantiques.

Les choix de modélisation et les ontologies créées résultent du travail de recherche effectué dans le cadre de cette thèse en s'appuyant sur l'expertise métier de juristes du projet Légilocal (notamment Ève Paul de Victoires Éditions) et les discussions avec les partenaires du projet. Les modèles présentés dans ce travail ne sont pas ceux adoptés dans le cadre du projet. Ils ont servi de base à un modèle simplifié qui rejoint les objectifs du projet dans sa première phase. Une deuxième phase du projet est en cours de préparation dans laquelle des aspects de modélisation plus avancés peuvent être pris en compte.

7.3 Première ontologie documentaire

L'ontologie que nous proposons a été conçue sur la base de l'analyse des besoins présentée dans le chapitre 5 et en suivant les recommandations décrites dans la section 7.2. Elle permet de représenter de manière homogène toutes les informations relatives aux documents juridiques :

1. la structure d'un document (sections, paragraphes, etc.),
2. le cadre temporel dans lequel il s'inscrit (dates, versions),
3. la caractérisation sémantique de son contenu à l'aide de concepts ou d'entités du domaine considéré,

91. http://www.w3.org/2003/01/geo/wgs84_pos

92. <http://justinian.leibnizcenter.org/MetaLex/metalex-cen.owl>

93. International Federation of Library Associations and Institutions.

94. <http://vocab.org/frbr/core.html>

95. <http://purl.org/NET/c4dm/event.owl>

96. Corese [Corby et al., 2004] : un moteur d'interrogation du Web Sémantique développé en utilisant les graphes conceptuels et implémentant les langages RDF, RDFS, SPARQL 1.1 Query et Update ainsi que des règles d'inférence.

4. son type (loi, décret, etc.),
5. les relations qu'il entretient avec d'autres documents (modification, abrogation, jurisprudence, transposition, etc.).

7.3.1 Structure globale de l'ontologie

L'ontologie documentaire que nous proposons intègre les différents types de propriétés (sémantiques, structurelles et temporelles) dans un même modèle. Elle permet aussi de rendre compte de la dimension intertextuelle qui est peu représentée dans les ontologies documentaires existantes. Cette ontologie est structurée en trois grands modules qui permettent de modéliser les propriétés ci-dessus :

- le module document (propriétés 1 et 2 : structure et cadre temporel),
- le module collection (propriétés 4 et 5 : types de documents et liens),
- le module sémantique (propriété 3 : contenu sémantique).

La figure 7.1 donne une vue globale de l'ontologie documentaire et montre la dépendance entre ses différents modules⁹⁷. La granularité de la description a été adaptée au cas d'usage Légilocal pour lequel cette ontologie a été initialement développée.

97. Nous avons utilisé des noms en anglais pour exprimer les classes, les propriétés et les attributs de l'ontologie afin d'homogénéiser avec les vocabulaires de tiers utilisés dans la conception (qui sont exprimés en anglais).

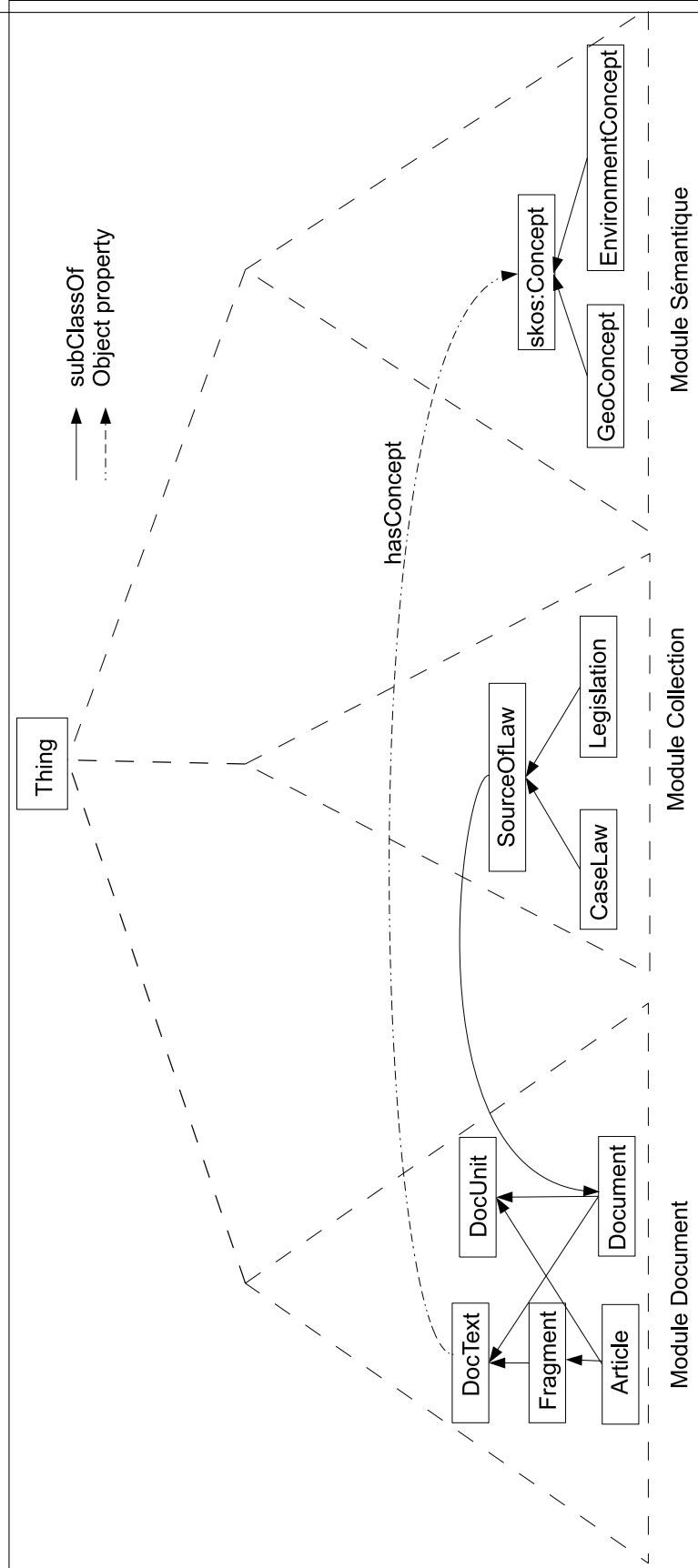


FIGURE 7.1 – Ontologie de collection documentaire : modules et dépendances

Le module document est représenté par les classes `DocumentText` et `DocumentaryUnit`. Le module sémantique est modélisé par un thésaurus en SKOS⁹⁸ qui est représenté dans l'ontologie par la classe `skos:Concept` et ses sous-classes et est lié au module document *via* les propriétés `hasConcept` et `isAssignedToDocText`. Le module collection est représenté par l'ensemble des types de documents (`Législation` est une sous classe de `Document`, par exemple) et un ensemble des relations intertextuelles (par ex. `modifies` pour une relation de modification, `isCodifiedBy` pour exprimer une relation de codification, etc.).

Les relations intertextuelles peuvent porter sur n'importe quelle unité documentaire, que ce soit un document juridique complet ou un de ses articles. À la différence des relations intertextuelles, les annotations sémantiques portent sur des composants de documents (paragraphe, par ex.).

La figure 7.2 présente le haut niveau de l'ontologie avec les concepts représentant ces différents aspects.

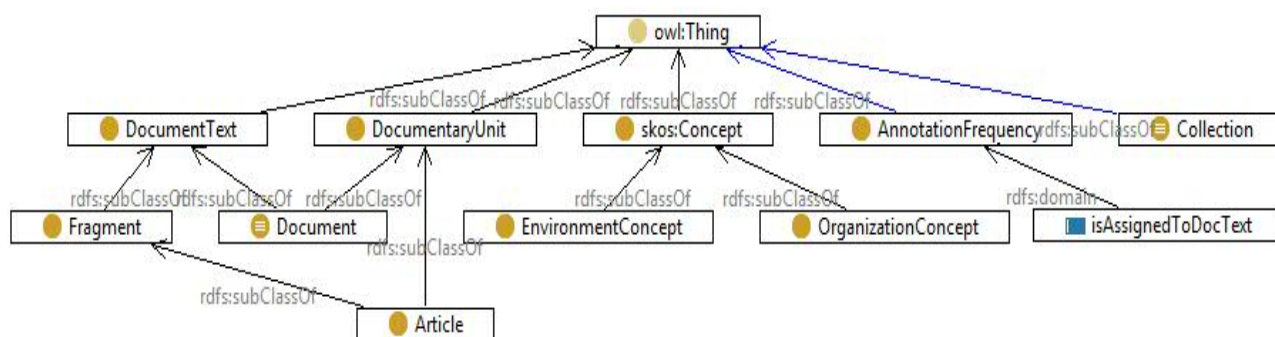


FIGURE 7.2 – Les concepts de haut niveau de l'ontologie documentaire.

7.3.2 Modélisation de la collection documentaire

Types de documents

Dans le domaine juridique, plusieurs types de documents sont créés et doivent être manipulés. Ceci est particulièrement le cas pour la collection de documents Légilocal qui comporte des documents de différents types : législation, décisions de justice (figure 7.3), actes locaux (figure 7.4) ainsi que les documents éditoriaux (figure 7.5).

En effet, pour préparer un acte municipal sur un sujet particulier, les agents des administrations locales (agents de mairie) doivent examiner la législation nationale et la jurisprudence sur le même sujet. Afin de les aider, le projet Légilocal fournit des fonctionnalités de recherche sémantique dans la législation nationale et la jurisprudence, ainsi que dans les actes locaux d'autres communes sur le même sujet et même dans certains documents éditoriaux. Ces fonctionnalités de recherche sémantique nécessitent l'annotation de ces documents pour faire ressortir leurs contenu sémantique, leurs dépendances ainsi que leurs structures. En effet, selon son type (par ex. acte local ou document législatif) un document possède une structure particulière qui est importante à préciser. Ceci permet d'annoter le document avec des propriétés connexes telles que l'organisation locale ou la personne en charge du document (qui sont spécifiques pour chaque acte local)

98. SKOS : Simple Knowledge Organization System, <http://www.w3.org/2004/02/skos/>.

Le : 17/06/2013

Cour administrative d'appel de Bordeaux

N° 99BX00597

Inédit au recueil Lebon

2E CHAMBRE

Mme Merlin-Desmartis, rapporteur
M. Rey, commissaire du gouvernement

lecture du mardi 28 mai 2002


REPUBLIQUE FRANCAISE
AU NOM DU PEUPLE FRANCAIS

Vu la requête, enregistrée le 24 mars 1999, présentée pour M. Y..., demeurant... (Pyrénées-Atlantiques), par Maître Lacaze ;

M. Y... demande à la cour :

1) d'annuler le jugement en date du 19 janvier 1999 par lequel le tribunal administratif de Pau a rejeté sa demande dirigée contre l'arrêté du 4 juillet 1997 du maire d'Ance interdisant la circulation de tout véhicule à moteur sur toute l'étendue du territoire de la commune non desservi par une route bitumée ainsi que sur la totalité des terrains communaux à vocation pastorale ou sylvicole ;

FIGURE 7.3 – Une décision de justice



**MAIRIE
DE VILLECRESNES**
Place Charles de Gaulle
94440 Villecresnes

ARRETE N°2012-17

**ARRETE PERMANENT INTERDISANT LA CIRCULATION DE TOUS VEHICULES A
MOTEUR SUR LE CHEMIN DU PONT DE PARIS ET LE CHEMIN DES PLANTES**

Le Maire,

Vu le Code Général des Collectivités Territoriales, articles L. 2122-28 et L. 2213.2,

Vu le Code de la Route,

Vu le Code de la Voirie Routière, Vu l'instruction interministérielle sur la signalisation routière (Livre I – huitième partie de la signalisation temporaire) approuvée par arrêté interministériel du 06 novembre 1992 et notamment son article 135 ;

Vu le décret n°2008-754 du 30 juillet 2008 portant diverses dispositions de sécurité routière ;

Considérant que le chemin du Pont de Paris est un chemin forestier sur lequel il convient d'interdire la circulation de tous véhicules à moteur ;

Considérant que le chemin des Plantes, situé entre le chemin de Vaux et le chemin du Pont de Paris est un chemin forestier sur lequel il convient d'interdire la circulation de tous véhicules à moteur ;

ARRÊTE

Article 1 : La circulation de tous véhicules à moteur est interdite chemin du Pont de Paris, sauf aux véhicules de services publics et de secours.

FIGURE 7.4 – Un acte local



FIGURE 7.5 – Un document éditorial

ou vérifier leur conformité. Nous distinguons ici un acte local d'un document législatif par le fait que nous n'avons pas besoin d'aller à un niveau fin de décomposition lorsque nous traitons un acte local, contrairement à la législation où habituellement l'unité de base manipulée est l'article.

Les différents types de documents sont modélisés par une hiérarchie de classes dont le haut niveau est présenté par la figure 7.6. Les trois catégories principales permettent de distinguer :

- les documents des collectivités territoriales (dont ceux des mairies) : classe `LocalAuthorityAct` ;
- les documents éditoriaux (revues, guides, modèles) : classe `EditorialDocument`. Ces documents sont principalement des guides pratiques et des modèles de documents qui aident les administrateurs locaux à créer leurs propres actes et qui font généralement référence à la législation et à la jurisprudence ;
- les documents correspondant aux sources du droit (classe `SourceOfLaw`) parmi lesquels on distingue la législation (`Legislation`) et la jurisprudence (`CaseLaw`).

À chaque type de document sont attachés des attributs et des propriétés spécifiques. Par exemple, la classe jurisprudence (`CaseLaw`) a une propriété (*object property*) `applique-législation` (`appliesLegislation`) qui la relie à la classe législation.

Liens entre documents

Dans notre ontologie, l'intertextualité est modélisée par des relations (*object properties*) qui ont pour sujet une unité documentaire (document ou article) et pour objet une autre unité documentaire. Par exemple, la propriété `creates` définit la relation de création entre un article non codifié⁹⁹ (`UncodifiedArticle`) et un article de texte codifié¹⁰⁰ (`CodifiedArticle`). Dans notre modèle, chaque type de relation est associé à une source et une cible spécifiques, ce qui permet de spécifier non seulement à quels types et parties de textes il réfère, mais aussi dans quels types de textes et parties de textes le lien modélisé peut apparaître.

La figure 7.7 donne un aperçu des types de relations que nous avons codés dans ce module et leur organisation hiérarchique. La relation `references` représente le sommet de cette hiérarchie et donc le type de lien le plus générique. Une requête qui porte sur les liens entre documents

99. Un article non codifié est un article qui appartient à un texte réglementaire autre que les codes : articles de loi, etc.

100. Un article codifié est un article qui appartient à un code : code civil, code de l'environnement.

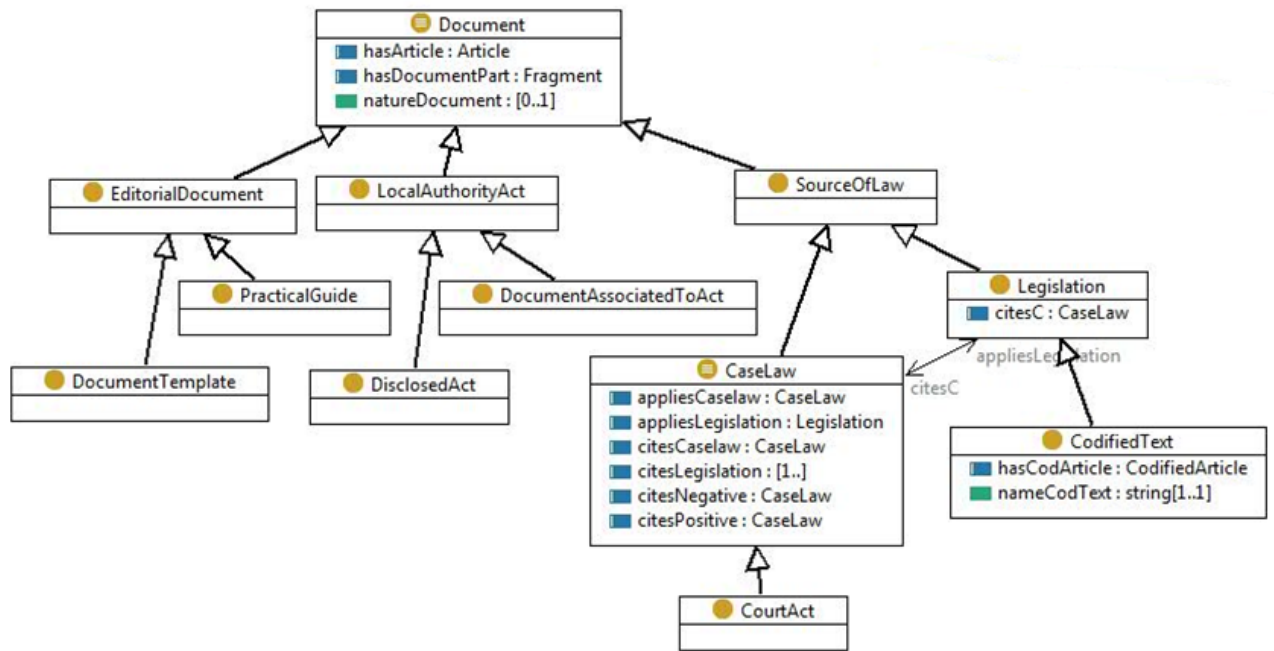


FIGURE 7.6 – Hiérarchie des types de documents.

exprimée par la relation **references** aura comme réponse tous les documents possédant les relations filles.

Nous distinguons deux grands types de relations : la relation de référence **references** et la relation de citation **cites**. La relation de référence exprime tout type de relation agissant ou pas sur le document source : modification, codification, abrogation, etc. La relation citation, sous-type de la relation référence, avec ses sous-classes (cite jurisprudence, visas législation) exprime le cas particulier d'un simple lien de citation partant d'un document vers un autre.

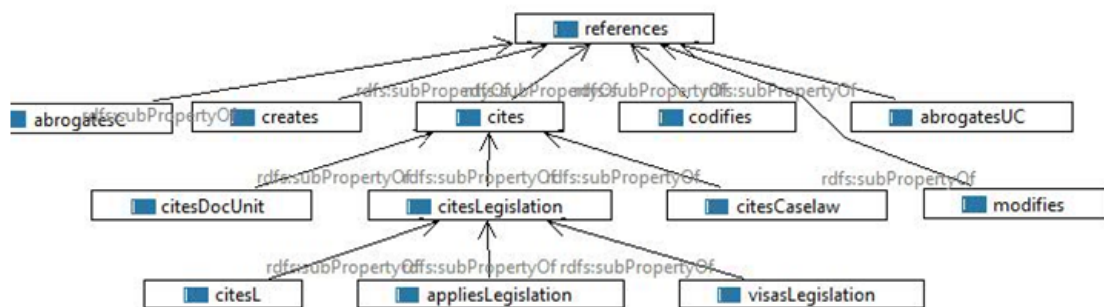


FIGURE 7.7 – Types de liens entre les documents et leur hiérarchie.

7.3.3 Modélisation des documents

Les documents juridiques possèdent une structure riche dont la sémantique est importante à prendre en compte. Les parties d'un document n'ont pas toutes la même importance : le

préambule est généralement peu utile alors que les articles qui composent le texte font l'objet de requêtes particulières. L'intérêt de l'utilisateur (citoyen, personnel administratif ou juriste dans le cas de Légilocal) porte souvent sur une partie du texte plutôt que sur le texte dans son ensemble. Cela suppose que les métadonnées d'identification et les annotations sémantiques soient attachées non pas au texte globalement mais à ses sous-parties [Hoekstra, 2011]. Les mêmes besoins sont valables pour les références entre les textes permettant une analyse fine des interdépendances entre eux.

C'est pourquoi nous représentons une collection documentaire comme un ensemble d'unités documentaires (classe Unité documentaire : `DocumentaryUnit`) et de fragments de documents (classe Texte de document : `DocumentText`) plutôt que comme un ensemble de documents. L'ensemble des documents (unités documentaires et fragments de documents) représentent des parties d'une même collection documentaire (représentée par la classe `Collection`) et ils y sont attachés par des propriétés d'appartenance (`hasCollectionPart` et son inverse `isPartOfCollection`). Nous pouvons de cette façon modéliser plusieurs collections qui ne sont pas forcément homogènes dans un même modèle.

Une unité documentaire correspond à un document ou à un élément de document qui a un cycle de vie propre comme par exemple un article de la législation. Un texte correspond à un document entier ou un fragment de document (élément de structure). C'est au niveau de chaque unité documentaire que seront définies les relations de référence (propriétés `references`, `cites`, etc.). Les annotations sémantiques sont attachées à un fragment de texte. Nous présentons dans cette section la modélisation de la structure du document ainsi que la gestion de son cycle de vie et ses différentes versions.

Structure de document

La structure d'un document est modélisée dans l'ontologie par la classe `DocumentText` et ses sous-classes `Fragment` et `Document`. La figure 7.8 montre les détails des classes modélisant la structure d'un document.

La classe `DocumentText` : représente le texte d'un document pris dans sa globalité (`Document`) ou en partie (`Fragment`).

La classe `Document` : représente un document dans le sens général du mot. Il correspond à l'unité de texte globale qui contient des fragments de texte. Les deux classes `Document` et `Fragment` sont reliées par des relations de composition `isPartOfDocument` et `hasDocumentPart`. Ci-dessous une description¹⁰¹ de la classe document :

```
1      @prefix : <http://www-lipn.univ-paris13.fr/~mimouni/owl/2014/06/DocModelOntology.owl#> .
2      @prefix owl: <http://www.w3.org/2002/07/owl#> .
3      @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4
5      :Document
6          a owl:Class ;
7          rdfs:label "Document"@en ;
8          rdfs:subClassOf :DocumentaryUnit , :DocumentText ;
9          owl:disjointWith :Fragment , :Article ;
10         owl:equivalentClass
11             [ a owl:Restriction ;
12               owl:allValuesFrom :Fragment ;
13               owl:onProperty :hasDocumentPart
14             ] .
```

Listing 7.1 – Classe Document.

101. En TURTLE : Terse RDF Triple Language, <http://www.w3.org/TR/turtle/>

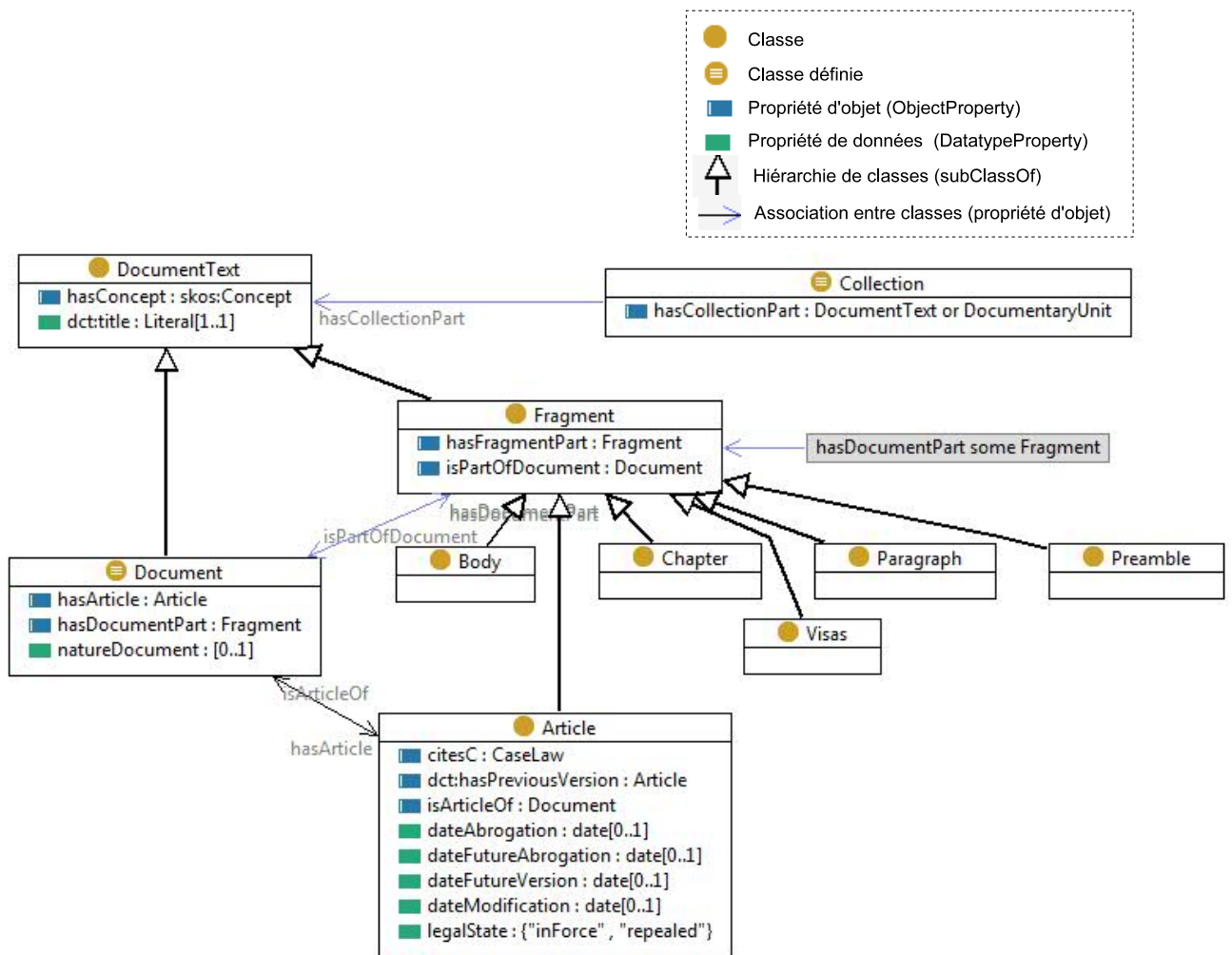


FIGURE 7.8 – Les classes modélisant la structure d'un document.

La classe `Fragment` : représente un élément de structure d'un document : préambule, visas, corps, article, paragraphe et chapitre.

Les classes `Body`, `Chapter`, `Paragraph`, `Preamble` : représentent les éléments de structure d'un document. Une relation de composition (`hasFragmentPart`) peut être définie entre certains éléments. Cette liste peut être étendue avec d'autres éléments en cas de besoin.

La classe `Article` : représente un élément de structure particulier contenu dans un document juridique. Nous justifions ci-dessous l'intérêt de faire la distinction entre cet élément de structure et le reste des composants ainsi que les détails des propriétés de cette classe.

La représentation des éléments de structure dans l'ontologie a pour principal but de contextualiser les annotations sémantiques et les relations (références et citations entre documents). Nous nous sommes limitée à ce niveau de représentation de la structure dans l'ontologie et nous représentons la structure détaillée, correspondant à chaque type de document (acte local, texte de législation, etc.) dans un schéma XML. Un schéma XML a été défini dans le cadre du projet Légilocal¹⁰² pour préciser, pour chaque type de document, les différentes métadonnées et la structure attendue. Nous avons fait ce choix pour plusieurs raisons : ne pas alourdir l'ontologie et ainsi la base RDF créée après instantiation, profiter de tout le potentiel du standard XML pour la description de la structure d'un document et, principalement, permettre un contrôle de conformité des documents (au moment de leur création) selon leurs types en se référant au schéma.

Gestion de l'aspect temporel/de cycle de vie

Dans notre conceptualisation, nous introduisons une opposition entre les fragments de texte et les unités documentaires. Nous jugeons important de distinguer les parties du document qui sont susceptibles d'être citées (unités documentaires) de celles qui ne le sont pas (simples fragments). Par exemple, dans une loi donnée, nous considérons le document en entier et ses articles comme des unités documentaires mais pas le préambule. De la même manière, seules les unités documentaires peuvent être retournées en réponse aux requêtes des utilisateurs.

Dans le cadre du projet Légilocal, nous considérons en effet que l'unité documentaire de base possédant un cycle de vie indépendant (suite à un processus de modification, etc.) est l'article. Ainsi, la deuxième partie du module document de l'ontologie est représentée par la classe `DocumentaryUnit` et ses sous classes `Document` et `Article` comme le montre la figure 7.9.

La complexité du cycle de vie du document juridique (et des articles qui le composent) provient du fait qu'il subit, à des dates différentes, des processus de modification, de consolidation, etc. La modification ou la consolidation, à une date précise, d'un texte implique la création de nouvelles versions identifiées par leurs dates. Il est important de gérer toutes ces versions avec leurs dates respectives (date de modification, date de mise en vigueur, etc.).

Plusieurs dates sont de ce fait associées à un document (date de publication `datePublication`, d'entrée en vigueur `dateInForce`, etc.). Nous représentons ces dates par des attributs (*datatype properties*) dont les valeurs sont de type `dct:date`¹⁰³ comme sur la figure 7.10.

La classe `DocumentaryUnit` : un ensemble étendu d'attributs et de propriétés est défini au niveau de chaque unité documentaire : titre (`dct:title`), sujet (`dct:subject`), identifiant (`dct:identifiant`), etc. Sont définies aussi les dates de signature (`dateSignature`), de publication (`datePublication`) et de mise en vigueur (`dateInForce`).

102. Ce schéma ne rentre pas dans le cadre de notre travail et ne sera pas décrit dans ce mémoire.

103. <http://purl.org/dc/terms/date>

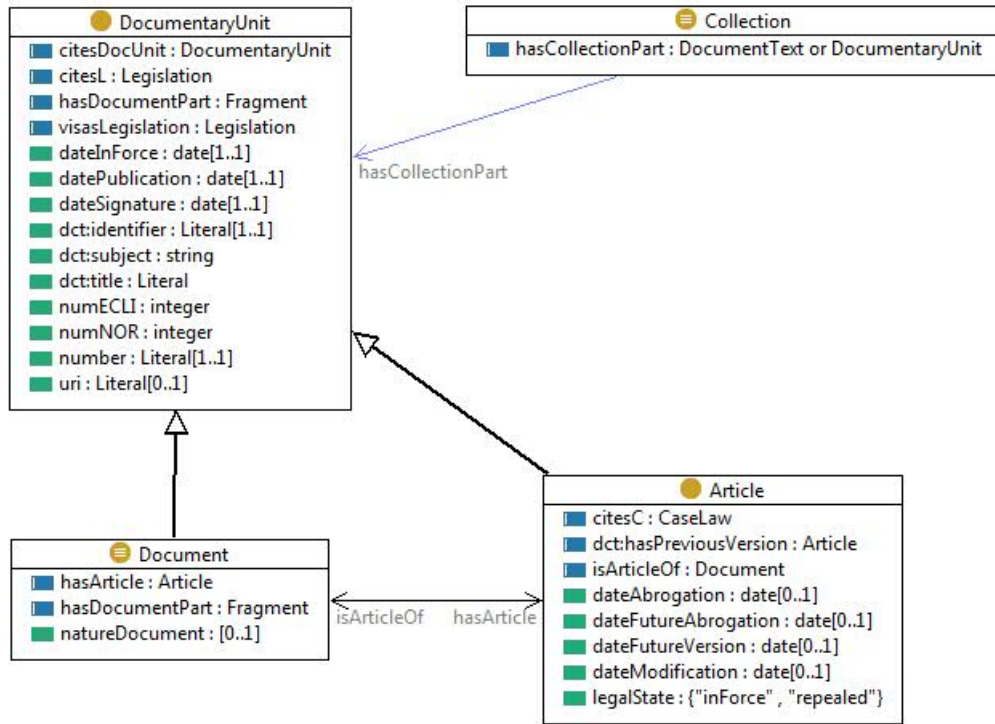


FIGURE 7.9 – Gestion du cycle de vie d’une unité documentaire (document ou article).

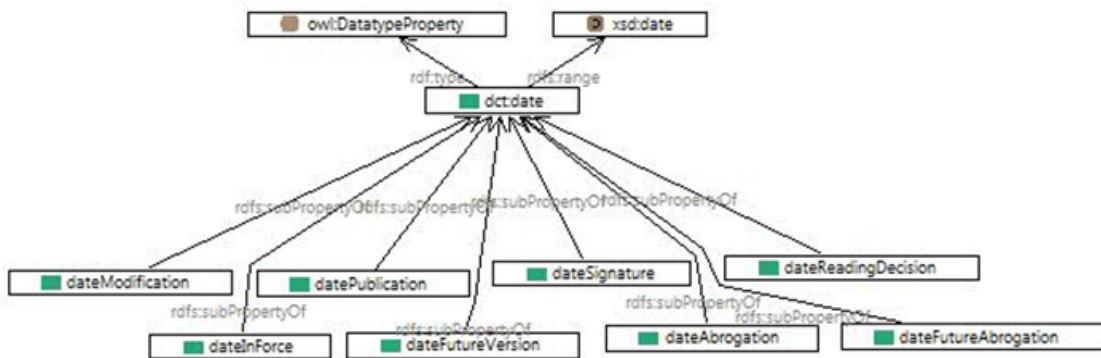


FIGURE 7.10 – Dates associées à une unité documentaire ou un article.

La classe Article : représente l'unité documentaire de base qui peut être identifiée, qui subit des modifications et qui a un cycle de vie propre. Un article est lié au document qui le contient (classe **Document**) par les propriétés **hasArticle** et **isArticleOf**. Au niveau de chaque article sont définies les dates d'abrogation (**dateAbrogation**) et de modification (**dateModification**). Un attribut **legalState** permet d'indiquer l'état d'un article : en vigueur ou abrogé.

Nous considérons toutes les versions d'articles comme des unités documentaires différentes et la modification d'un article est représentée par un lien entre l'article modificateur et l'article modifié, la date de modification étant celle de l'entrée en vigueur de l'article modificateur. Nous représentons le chaînage entre les versions des articles par la propriété **hasPreviousVersion** (figure 7.11). De cette manière, le cycle de vie d'un document est traité comme une relation entre des unités documentaires correspondant chacune à une version (la gestion des relations est faite au niveau de la collection).

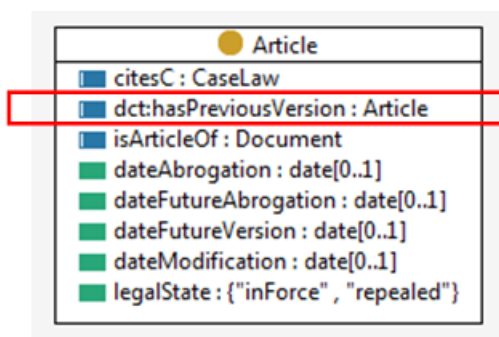


FIGURE 7.11 – Gestion de versions d'un article.

7.3.4 Modélisation sémantique des contenus textuels

Le module sémantique est classique. En pratique, on cherche généralement à réutiliser une ontologie ou un thésaurus existants. Dans le projet Légilocal, des ressources terminologiques ont été développées pour définir les termes utilisés pour l'annotation sémantique. Elles sont structurées en trois grands sous-modules ou vocabulaires :

- le sous-module organisation décrit les différentes juridictions françaises et les entités administratives (par exemple, la cour d'appel, le tribunal de district, le ministère) ;
- le sous-module géographique décrit les différentes entités géographiques françaises (par exemple, les régions, les communes) ;
- le sous-module juridique décrit les notions juridiques de base du droit français ;
- le sous-module randonnée décrit le vocabulaire couramment utilisé pour administrer les routes, chemins et sentiers, ce domaine relevant de la compétence de la municipalité ; ce module représente le cas d'utilisation de test choisi par le projet.

Il est prévu dans Légilocal d'enrichir ces ressources sémantiques avec des éléments de vocabulaire utilisateurs acquis à partir des mots-clés de recherche des citoyens et ensuite l'alignement de ce vocabulaire avec la terminologie juridique et administrative des documents.

Dans notre modélisation, nous regroupons toutes ces ressources sémantiques dans un seul module de l'ontologie documentaire : le module sémantique. Ce module est décrit dans la figure 7.12.

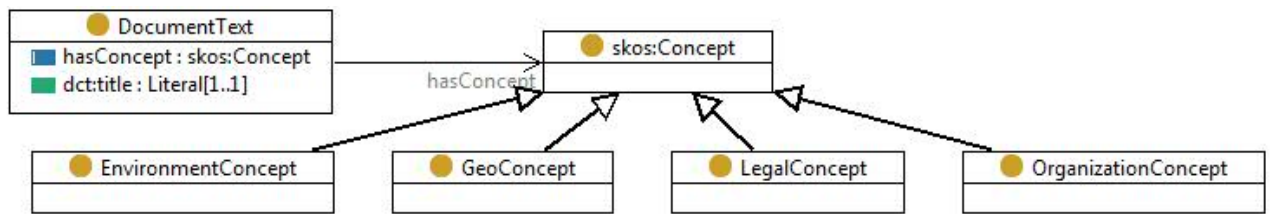


FIGURE 7.12 – Ressources et annotation sémantique.

Chaque ressource sémantique développée dans le projet correspond à une branche d'une même racine représentée par un concept SKOS. Les ressources sémantiques correspondent de ce fait à différents thésaurus (en SKOS). À chaque ressource nous faisons correspondre un concept terminologique qui représente la classe de termes de cette ressource. Par exemple, `EnvironmentConcept` est un concept terminologique (en relation d'héritage (`rdfs:subClassOf`) avec `skos:Concept`), qui représente la classe de tous les termes du domaine de l'environnement (polluants, bruit, etc.) comme décrit sur la figure 7.13.

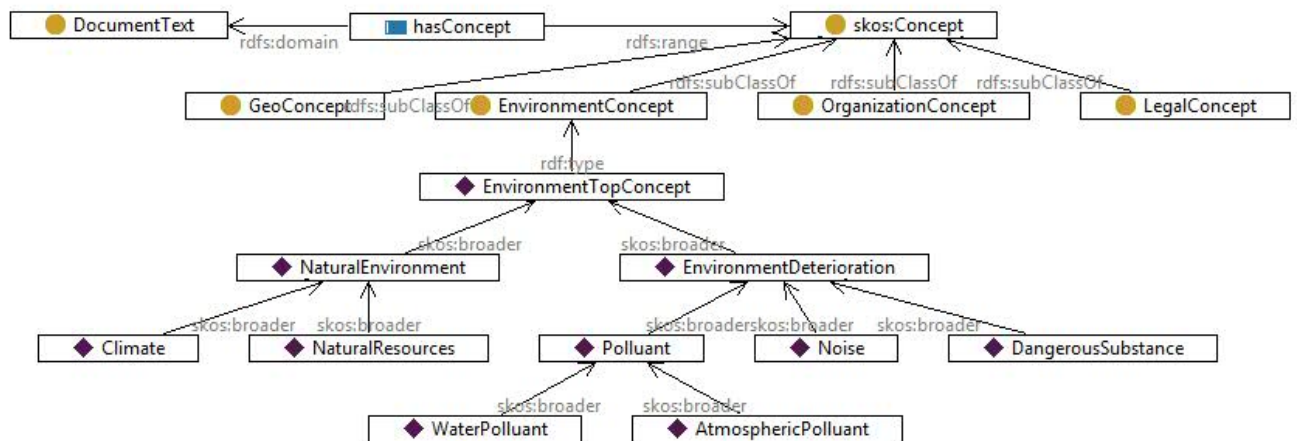


FIGURE 7.13 – Concepts terminologiques représentant les ressources sémantiques. Hiérarchie entre concepts de la ressource Environnement.

Cette modélisation est particulièrement utile lorsqu'il s'agit de données provenant de sources externes et hétérogènes. Cette modélisation permet d'étendre facilement l'ontologie avec de nouvelles ressources, de modifier les ressources existantes ou de les supprimer sans affecter le reste des modules de l'ontologie (bien que les vocabulaires SKOS et OWL soient utilisés dans le même graphe, les deux flux de données restent séparés).

Tous les concepts du module sémantique sont des `skos:Concept` qui possèdent au moins l'attribut `skos:prefLabel` et éventuellement l'attribut `skos:altLabel`. Une hiérarchie peut être créée entre les concepts d'une ressource sémantique. Elle est définie par les relations de généralisation ou de spécification créées par les propriétés `skos:broader` et `skos:narrower`. La figure 7.13 montre un exemple de hiérarchie entre les concepts de la ressource sémantique `Environnement`. En RIS, la hiérarchie des concepts permet de répondre à des requêtes auxquelles des réponses exactes n'ont pas pu être retrouvées en retournant des réponses approchées (en spécialisant ou

en généralisant la requête) et éviter ainsi de retourner un ensemble vide.

Le module sémantique est relié au module document par la propriété `hasConcept` définie entre un texte juridique `DocumentText` et un concept du module sémantique (du concept terminologique `EnvironmentConcept` par exemple) comme décrit sur la figure 7.13. Cette relation directe permet notamment de retrouver des documents qui possèdent les annotations sémantiques indiquées dans les requêtes.

En conclusion, dans le cadre de Légilocal, ces annotations sémantiques sont à la base des fonctionnalités de recherche d'information sémantique offertes par la plateforme du projet aux agents de collectivités locales et aux citoyens. En liant les fragments des textes aux ressources sémantiques, ces derniers associent un contexte sémantique aux documents retournés et permettent une navigation à facettes basée sur un thésaurus et offrent des fonctionnalités de généralisation ou de spécialisation des requêtes.

7.4 Deuxième ontologie documentaire

La première ontologie permet de représenter assez simplement une collection documentaire comme un réseau de documents, une unité documentaire pouvant être liée à une autre par différents types de relations. Cette approche – assez intuitive – suffit sans doute à modéliser l'intertextualité dans certains domaines et pour certaines applications (par ex. les articles de presse, les oeuvres littéraires, les brevets) mais elle s'avère trop limitée pour rendre compte de la complexité de l'intertextualité juridique. Cette complexité ressort de l'analyse des besoins faite dans le cadre du projet Légilocal montrant que certaines caractéristiques spécifiques aux collections juridiques (versions d'un même document, actions juridiques) ne peuvent être modélisées.

L'intertextualité est au coeur de l'activité juridique où les actions (décisions, jugements, recours, régulation, etc.) se réalisent au travers de la publication de documents qui font référence à d'autres actions, c'est-à-dire d'autres documents qu'ils modifient ou dont ils s'inspirent. Modéliser la complexité de cette activité documentaire par des relations binaires comme proposé ci-dessus est impossible parce qu'on a des cas de relations ternaires, notamment lorsqu'un document crée un nouveau document en agissant sur un document antérieur. Cela nous incite à modéliser explicitement les opérations documentaires sous-jacentes aux relations intertextuelles (pour prendre en compte le cas de relations qui prennent plus de deux unités documentaires comme arguments) plutôt que sous la forme de relations directes.

Par ailleurs, il est essentiel de prendre en compte l'historique des différentes versions des documents juridiques qui résultent de ces opérations. Même si ces différentes versions se remplacent les unes les autres, elles coexistent au sein du système juridique puisque chacune est la version de référence pour une période donnée. Or, d'un point de vue général, la première ontologie représente les documents comme de simples artefacts sans tenir compte des différentes natures d'objets informationnels que la science de l'information a mis au jour depuis longtemps et sans lesquelles on ne peut pas rattacher les différentes versions d'un document à une source commune. Vu l'importance de cette notion dans le domaine juridique, nous introduisons donc dans l'ontologie la distinction entre le document maître, l'oeuvre, et les différentes versions qui en sont données. Nous suivons en cela l'approche proposée par Metalex qui repose sur la distinction classique introduite par FRBR¹⁰⁴ [IFLA, 1998]. Cela implique de spécifier à quel niveau

104. FRBR introduit la distinction entre l'oeuvre (*work*), ses différentes expressions (*expression*), les manifestations (*manifestation*) de ces dernières et les différents exemplaires (*item*) qui en résultent. Cette classification permet d'expliquer que le livre écrit par Marcel Proust (l'oeuvre) n'est pas le même que celui que je viens de déchirer (l'exemplaire), que les différentes éditions (des manifestations) reprennent le texte dit de « la pléiade » (une

se situent les relations d'intertextualité introduites et cela permet de factoriser une partie des propriétés documentaires sur l'œuvre sans les dupliquer sur chacune de ses versions.

Nous montrons dans ce qui suit comment nous proposons de prendre en compte ces deux aspects – les relations d'intertextualité à plus de deux arguments et la distinction œuvre-version – avant de présenter globalement une structure d'ontologie documentaire qui les intègre, l'ontologie LIDO (Legal Interlinked Documents Ontology).

Nous avons le souci de suivre les recommandations du web de données et de nous aligner avec les vocabulaires ouverts du web. Nous avons réutilisé des vocabulaires généralistes tels que Dublin Core et FOAF. Nous avons aussi étudié les standards juridiques et plus spécifiquement Metalex, et nous nous sommes alignée avec l'ontologie définie dans ce standard (voir chapitre 2 pour une présentation détaillée de ce standard).

Les termes des vocabulaires tiers qui sont réutilisés par le vocabulaire LIDO sont listés dans le tableau 7.1. Plus de détails sur leur utilisation sont donnés dans les sections qui suivent.

TABLE 7.1 – Classes et propriétés réutilisés par le vocabulaire LIDO.

Classes	
<code>metalex:Legislative</code>	Opération qui résulte en la création d'une source de loi.
<code>Creation</code>	
<code>metalex:Author</code>	Auteur d'un document, agent participant à toute création bibliographique.
<code>metalex:Legislator</code>	Législateur, un type d'auteur, agent d'une création législative.
<code>metalex:Editor</code>	Éditeur, un type d'auteur, agent d'une édition.
<code>metalex:Date</code>	Date d'une opération.
<code>metalex:Matter</code>	Instrument participant à une opération.
<code>metalex:Result</code>	Objet résultant d'une opération.
<code>foaf:Agent</code>	Tout agent (personne ou groupe) participant à une opération. Un auteur est un agent, un législateur ou un éditeur (qui sont des auteurs) sont des agents.
<code>geo:SpatialThing</code>	Désigne toute instance juridique (mairie, tribunal, etc.).
Propriétés	
<code>event:place</code>	Relier une opération à un lieu.
<code>dct:hasPart</code>	Exprimer un lien d'appartenance.
<code>dct:identifiant</code>	Affecter un identifiant à un document.
<code>dct:title</code>	Décrire le titre du document.
<code>dct:date</code>	Exprimer une date comme attribut de document.
<code>metalex:participant</code>	Relier un participant à une opération.
<code>metalex:realizes</code>	Relier une œuvre à une de ces expressions.

Dans la suite nous décrivons les modules de la nouvelle ontologie documentaire conçue pour satisfaire les deux critères décrits plus haut : la gestion des versions des documents (distinction œuvre-version) et la gestion des relations d'intertextualité à plus de deux arguments (propriétés liées aux opérations à l'origine de ces relations).

expression) mais que l'édition de la Pléiade (une manifestation) n'est pas ce que j'ai acheté hier (un exemplaire).

7.4.1 Gestion des versions d'un document

La gestion des versions des documents repose sur le modèle FRBR, utilisé par Metalex, qui introduit quatre niveaux d'abstraction d'un document. Une œuvre peut naturellement donner lieu à plusieurs expressions. Une expression peut se manifester de différentes manières et chaque manifestation peut être produite en plusieurs exemplaires.

Dans notre modèle, nous utilisons les deux niveaux supérieurs, œuvre et expression, afin de représenter les différentes versions des articles et des documents. Parmi ces quatre niveaux, seules les œuvres et les expressions correspondent à des unités documentaires auxquelles on peut faire référence ou qu'on peut citer. Les documents que nous modélisons sont de plusieurs types, comme décrits dans la section 7.3.2, et sont considérés comme étant des œuvres.

Nous maintenons l'opposition proposée dans la première ontologie entre une unité documentaire « citable » et tous les textes ou fragments de document qui sont « annotables ». Nous créons une nouvelle entité documentaire (classe `DocumentObject`) subsumant la classe des fragments de documents (classe `DocumentText`) et celles des unités documentaires (`CitableDocumentObject`) que nous avons créée pour désigner tout objet documentaire (œuvre ou expression) possédant un cycle de vie indépendant, pouvant être cité, modifié, etc. La classe `DocumentText` reste inchangée par rapport à la première modélisation et garde les mêmes attributs et propriétés. Les propriétés d'appartenance entre une collection de documents (classe `Collection`) et un objet documentaire (classe `DocumentObject`), entre un document (classe `Document`) et un fragment de document (classe `Fragment`) sont désormais des sous-propriétés de la propriété `dct:hasPart`.

Nous avons distingué dans l'ontologie les classes `DocumentaryUnitWork` et `DocumentaryUnitExpression`. La première classe correspond à l'unité documentaire en tant qu'œuvre comme par exemple l'article *L2213 – 2 du code général des collectivités territoriales*, alors que la seconde classe correspond à chaque version de cet article (des expressions différentes). Une œuvre est un objet documentaire réalisé par une ou plusieurs expressions et une expression est un objet documentaire qui réalise une œuvre. Toute modification dans une expression produit une nouvelle expression. Les liens entre les documents sont attachés aux expressions du fait qu'une nouvelle expression (version) peut faire référence à un ensemble de documents différent de celui référencé par la version précédente et que, dans l'autre sens, pour les liens entrants, ce n'est pas l'œuvre qui est citée mais plutôt la version en vigueur à une date donnée.

La figure 7.14 présente la hiérarchie des classes permettant de modéliser les unités documentaires en tant qu'œuvres et versions pour les différents types de documents.

Les propriétés `metalex:realizes` et son inverse `metalex:realizedBy` permettent de relier une unité documentaire en tant qu'œuvre à ses différentes versions (ou expressions). La propriété `metalex:realizes` (figure 7.14), associant une `DocumentaryUnitExpression` et un `DocumentaryUnitWork`, est définie comme propriété fonctionnelle (*Functional Property*) : une version ne pouvant être reliée qu'à une seule œuvre, ce qui signifie qu'une version donnée (individu de la classe `DocumentaryUnitExpression`) ne peut être reliée qu'à un individu au plus par cette propriété. En d'autres termes, une version ne peut correspondre qu'à un seul document.

Lorsqu'une version est créée, elle est reliée à la précédente et à l'unité documentaire qui l'a créée via une action (de modification ou codification par exemple). Ces relations et les classes qui les gèrent sont présentées dans la section qui suit (section 7.4.2).

7.4.2 Gestion des références

Nous proposons de modéliser l'intertextualité sous la forme de concepts intertextuels pour prendre en compte le cas des relations qui prennent plus de deux unités documentaires comme

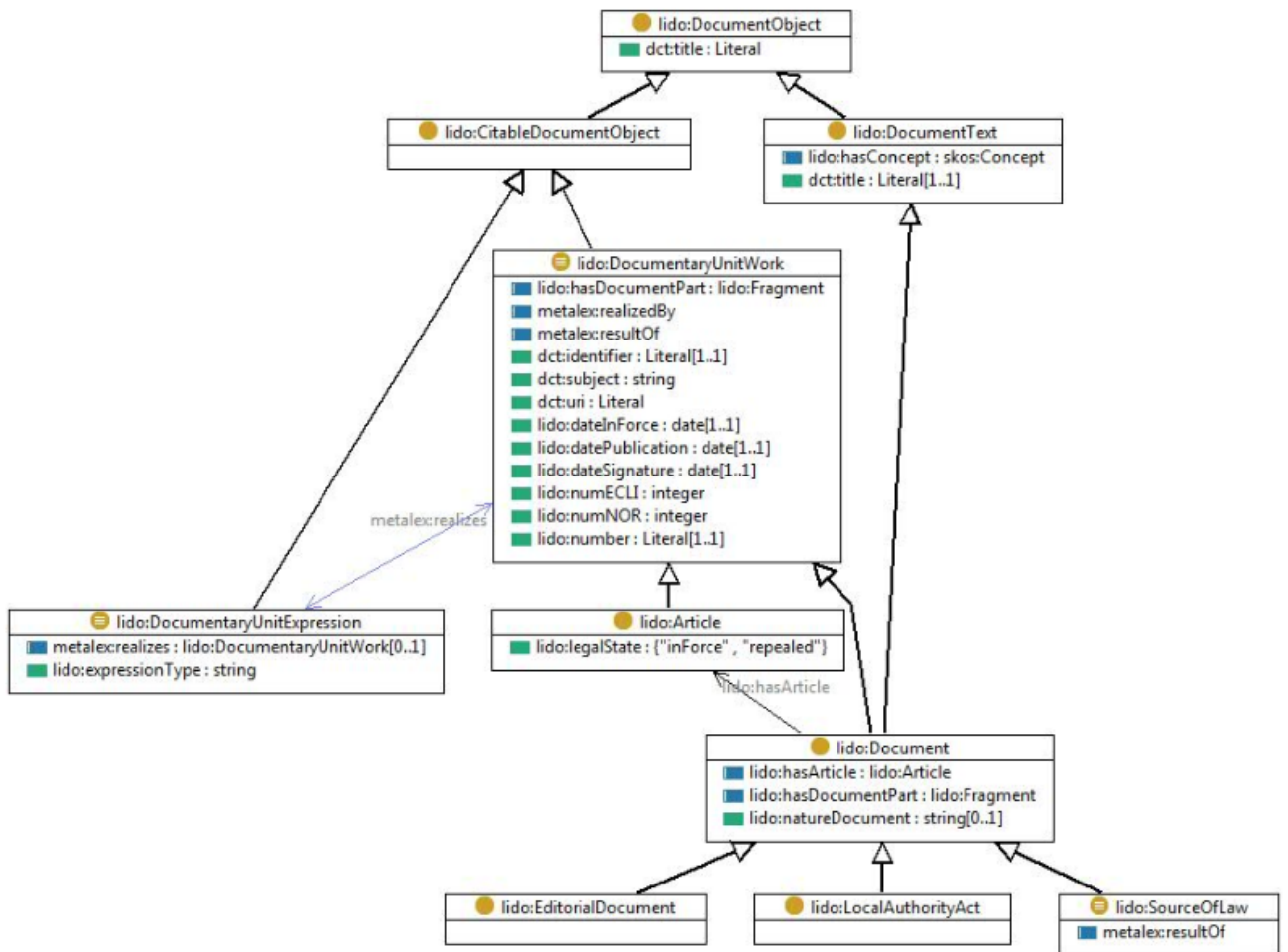


FIGURE 7.14 – Gestion des versions des documents d’une collection et la relation de réalisation entre un document (œuvre) et ses versions (expression). La classe `DocumentText` représente les fragments de documents qui peuvent être annotés et la classe `CitableDocumentObject` représente les unités documentaires qui peuvent en outre être citées.

arguments. Ces concepts intertextuels modélisent des citations ou des opérations documentaires qui peuvent faire intervenir un nombre variable d'unités documentaires selon le type de relation qu'ils représentent :

- opérations documentaires à un argument : création (d'une œuvre) ;
- opérations documentaires à deux arguments : abrogation ;
- opérations documentaires à trois arguments : codification, transposition, modification.

Les textes jouent un rôle dans ces opérations documentaires, ils portent parfois la trace de ces opérations : un lien existe toujours dans le document initiant l'opération documentaire (par exemple du document modifieur vers le document modifié, lien **modifie**) mais pas nécessairement dans le document résultat de l'opération (lien **modifié par**). Ces liens sont ajoutés *a-posteriori* par des systèmes de RI juridique comme Legifrance. Dans notre modélisation, ils peuvent être déduits à partir des rôles des documents participants (source, cible et résultat) aux opérations documentaires.

Nous distinguons deux types de liens entre les unités documentaires, les liens de citation et de référence. Une citation est un lien qui possède une source et une cible. À chaque type de citation correspondent des types de documents particuliers. Une citation apparaît dans le texte d'un document source pour indiquer une information complémentaire qui pourrait être utile à la compréhension du contenu, elle n'agit pas (en modification ou codification par exemple) sur un autre document. Contrairement aux citations, les références concernent l'évolution des textes juridiques (création, codification, modification, transposition et abrogation). Un lien de référence est la trace d'une opération documentaire qui modifie la collection documentaire. Dans le cas de relations ternaires, elle fait intervenir en plus du document source et du document cible, un document qui est le résultat d'une opération comme par exemple la transposition (par exemple, transposition de la directive européenne 2004/114/CE¹⁰⁵ comme le montre le graphe de la figure 7.15).

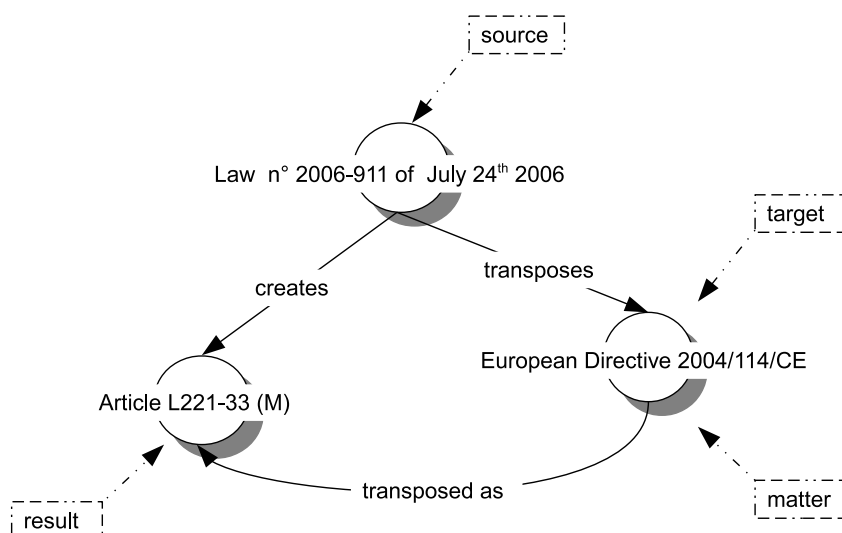


FIGURE 7.15 – Transposition de la directive 2004/114/CE, cible de la relation de transposition (la source de la relation est le texte de loi Loi n° 2006 – 911 du 24 Juillet 2006) et objet de l'opération de transposition (le résultat est l'article Art. L221 – 33 (M) du Code monétaire et financier).

105. <http://eur-lex.europa.eu/legal-content/FR/TXT/HTML/?uri=CELEX:32004L0114>

Dans l'ontologie LIDO, l'intertextualité est modélisée par la classe `IntertextualLink` et ses sous-classes `Citation` et `DocumentaryOperation` comme décrit dans la figure 7.16. Les objets de la classe `IntertextualLink` sont attachés aux fragments de texte qui contiennent le lien (trace), représentés par la classe `ReferenceText`, par la propriété `textOf`.

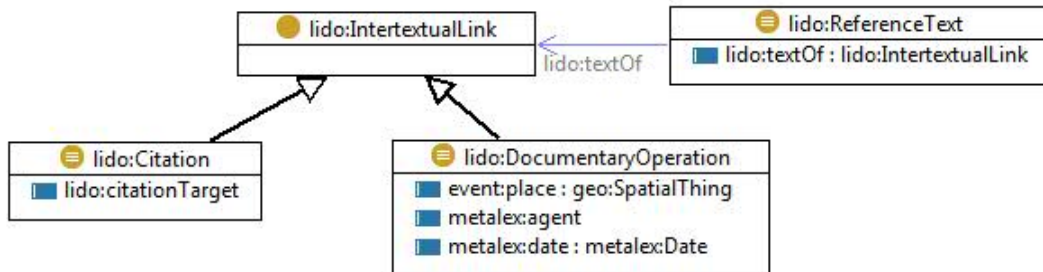


FIGURE 7.16 – Gestion des liens intertextuels.

Les citations

La classe `Citation` représente tout type de lien de citation, dans le sens expliqué plus haut, vers une unité documentaire « citable » (`CitableDocumentObject`). Pour nous caler sur l'usage, nous introduisons différents types de citations en fonction des types des documents qui citent (`citationSource`) et sont cités (`citationTarget`) (voir figure 7.17). Par exemple, le lien d'application (`lido:Application`) correspond à une citation reliant une jurisprudence (document source) à une législation (document cible). La description de cette classe est donnée en Listing 7.2. Les citations sont définies sur les versions des documents (`DocumentaryUnitExpression`). On impose à ces versions de documents la relation de réalisation (`lido:realizes`) avec une œuvre (`DocumentaryUnitWork`) de même type (ceci est valable aussi dans le cas où une seule version existe pour un document, chaque œuvre possède au moins une expression).

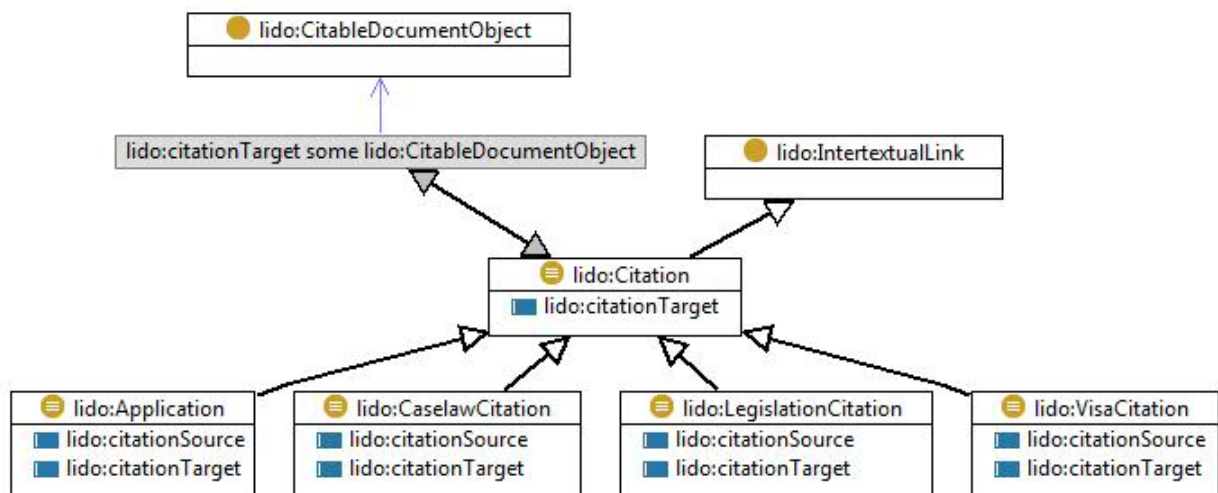


FIGURE 7.17 – Classe Citation.

```

1 @prefix lido : <http://www-lipn.univ-paris13.fr/~mimouni/owl/2013/12/docOntology#> .
2 @prefix metalex: <http://www.metalex.eu/metalex/2008-05-02#> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5
6 lido:Application
7   a owl:Class ;
8   rdfs:label "Application"^^xsd:string ;
9   rdfs:subClassOf lido:Citation ;
10  rdfs:subClassOf
11    [ a owl:Restriction ;
12      owl:onProperty lido:citationSource ;
13      owl:someValuesFrom
14        [ a owl:Class ;
15          owl:intersectionOf (lido:DocumentaryUnitExpression [ a owl:Restriction ;
16            owl:onProperty metalex:realizes ;
17            owl:someValuesFrom lido:CaseLaw
18          ])
19        ]
20    ] ;
21  owl:equivalentClass
22    [ a owl:Restriction ;
23      owl:onProperty lido:citationTarget ;
24      owl:someValuesFrom
25        [ a owl:Class ;
26          owl:intersectionOf (lido:DocumentaryUnitExpression [ a owl:Restriction ;
27            owl:onProperty metalex:realizes ;
28            owl:someValuesFrom lido:SourceOfLaw
29          ])
30        ]
31    ] .

```

Listing 7.2 – Classe Application.

Les références - opérations documentaires

La classe `DocumentaryOperation` représente les différents types d'opérations documentaires qui font intervenir des documents en tant que source, cible et résultat. Des liens de référence, qui lient les documents participants deux à deux, sont la trace de ces opérations. Dans le cas général, seul le lien entre source et cible existe effectivement ; il apparaît dans le texte du document source. Les autres liens de référence sont déduits à partir des documents participants.

Prenons l'exemple d'une opération de modification. La figure 7.18 décrit les participants à cette opération (en noir) et les liens de référence qui en découlent (en gris) : seul le lien `modifie` (en trait continu) a une trace dans les textes des documents participants. Les liens `version suivante`, `version précédente`, `créé par modification` et `créé par modification par` peuvent être ajoutés à la collection au moment de l'annotation.

En plus des documents source, cible et résultat, nous ajoutons aux opérations les informations concernant l'agent (en tant que personne responsable de la création du document source), la date et le lieu de l'opération documentaire. Ils sont représentés dans l'ontologie comme des entités filles de la classe `OperationParticipant` (qui sont souvent liées au document source). La classe `DocumentaryOperation` est décrite dans la figure 7.19. Ces différentes propriétés et les classes reliées sont détaillées dans le tableau 7.2.

Dans le tableau, le co-domaine de la propriété `matter` correspond en général au document cible. Les propriétés `matter` et `result` peuvent ne pas être utilisées dans certains cas. Par exemple, dans le cas d'une création législative, il n'existe pas de document objet (`matter`), et dans le cas d'une abrogation il n'y a pas de document résultat (un attribut décrivant le statut du document doit être modifié).

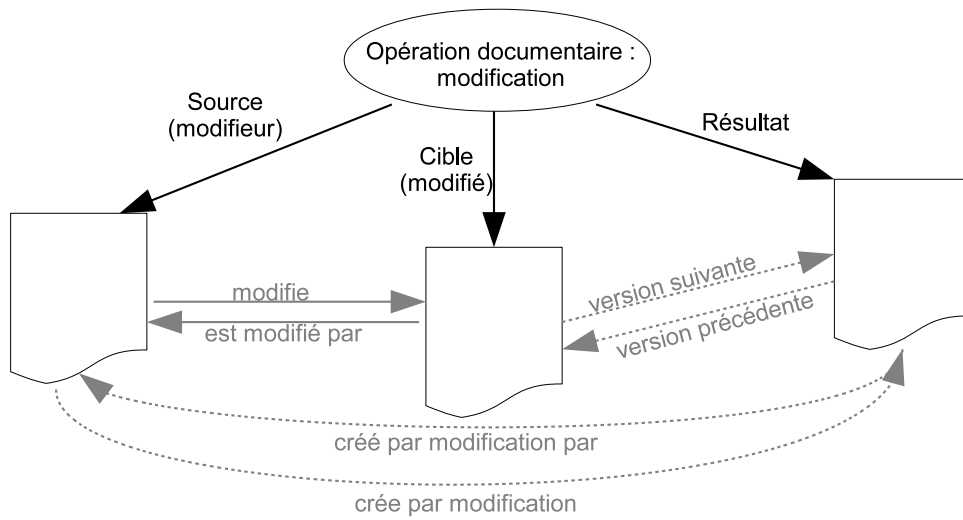


FIGURE 7.18 – Opération documentaire de modification : participants et liens de référence et citation résultants.

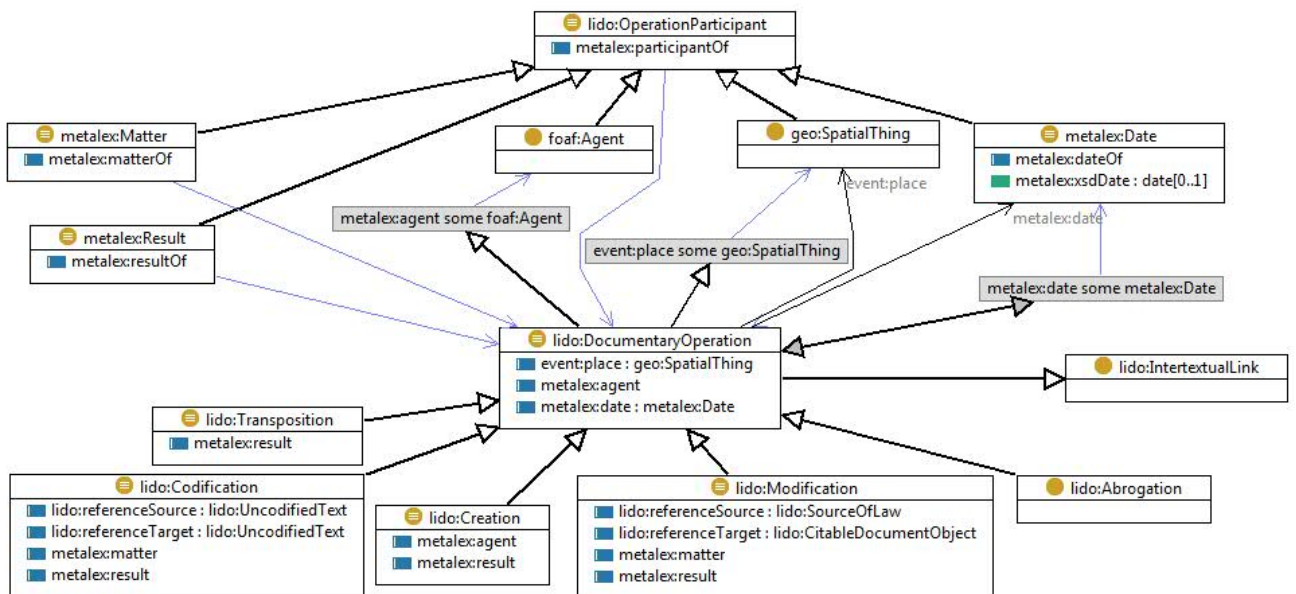


FIGURE 7.19 – Classe DocumentaryOperation.

TABLE 7.2 – Classes et propriétés reliées à la classe `DocumentaryOperation`.

Classe <code>metalex:Date</code> ¹⁰⁶	:	<i>Date</i> - Une date d'une opération de référence.
Classe : <code>foaf:Agent</code> ¹⁰⁷	:	<i>Agent</i> - La classe des agents (par exemple, une personne, un groupe ou une organization).
Classe <code>goe:SpatialThing</code> ¹⁰⁸	:	<i>Spatial thing</i> - Tout objet ayant des dimensions dans l'espace, c'est-à-dire une taille, une forme ou une position (par exemple, personnes, places).
Propriété <code>metalex:date</code>	:	<i>date</i> - Relie une opération de référence à une date. Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>metalex:Date</code> .
Propriété <code>metalex:agent</code>	:	<i>agent</i> - L'agent responsable de l'opération à l'origine de la référence juridique (la propriété <code>foaf:maker</code> peut également être utilisée). Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>foaf:Agent</code> .
Property : <code>event:place</code>	:	<i>place</i> - Relie une opération à un lieu. Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>goe:SpatialThing</code> .
Property <code>lido:referenceSource</code>	:	<i>referenceSource</i> - Le document source de la référence juridique. Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>LegalDocument</code> .
Property <code>lido:referenceTarget</code>	:	<i>referenceTarget</i> - Le document cible de la référence juridique. Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>LegalDocument</code> .
Property <code>metalex:matter</code>	:	<i>matter</i> - Le document sur lequel un agent agit (via le document source). Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>LegalDocument</code> .
Property <code>metalex:result</code>	:	<i>result</i> - Le document qui résulte de l'action de l'agent. Domaine : <code>DocumentaryOperation</code> , co-domaine : <code>LegalDocument</code> .

Exemple d'utilisation Prenons l'exemple suivant d'une opération de codification d'un article du Code général des impôts (décrit sur Legifrance¹⁰⁹). Cet événement implique :

- comme source : le décret (classe **Decree**) *Décret n°92-836 du 27 août 1992*,
- comme cible : l'article non codifié (classe **UncodifiedArticle**) *Article 46 quater-00 A bis - 4 juillet 1992*,
- comme résultat : l'article codifié (classe **CodifiedArticle**) *Article 46 quater-00 A bis- 29 août 1992*,
- comme lieu l'Assemblée nationale,
- comme date le 27 août 1992,
- comme signataire le ministre de budget, *Michel Charasse*.

L'opération de codification (classe **Codification**) fait intervenir un document de type décret (**Decree**), un article non codifié (**UncodifiedArticle**) et un article codifié (**CodifiedArticle**). Dans cette opération, l'article non-codifié est à la fois cible (du lien de référence) et objet (de l'opération). Le digramme des instances est décrit dans la figure 7.20.

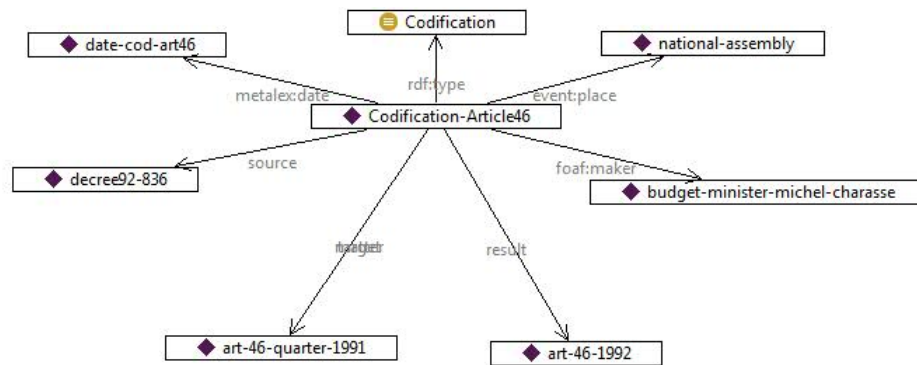


FIGURE 7.20 – Codification de l'Article 46 quater-00 A bis du 4 juillet 1992.

7.4.3 Structure globale de l'ontologie

La deuxième ontologie proposée permet de prendre en compte les deux aspects décrits ci-dessus : les relations d'intertextualité à plus de deux arguments et la distinction œuvre-version. La structure globale de cette ontologie est donnée dans la figure 7.21.

109. <http://www.legifrance.gouv.fr/affichCode.do?cidTexte=LEGITEXT000006069577&dateTexte=20140623>

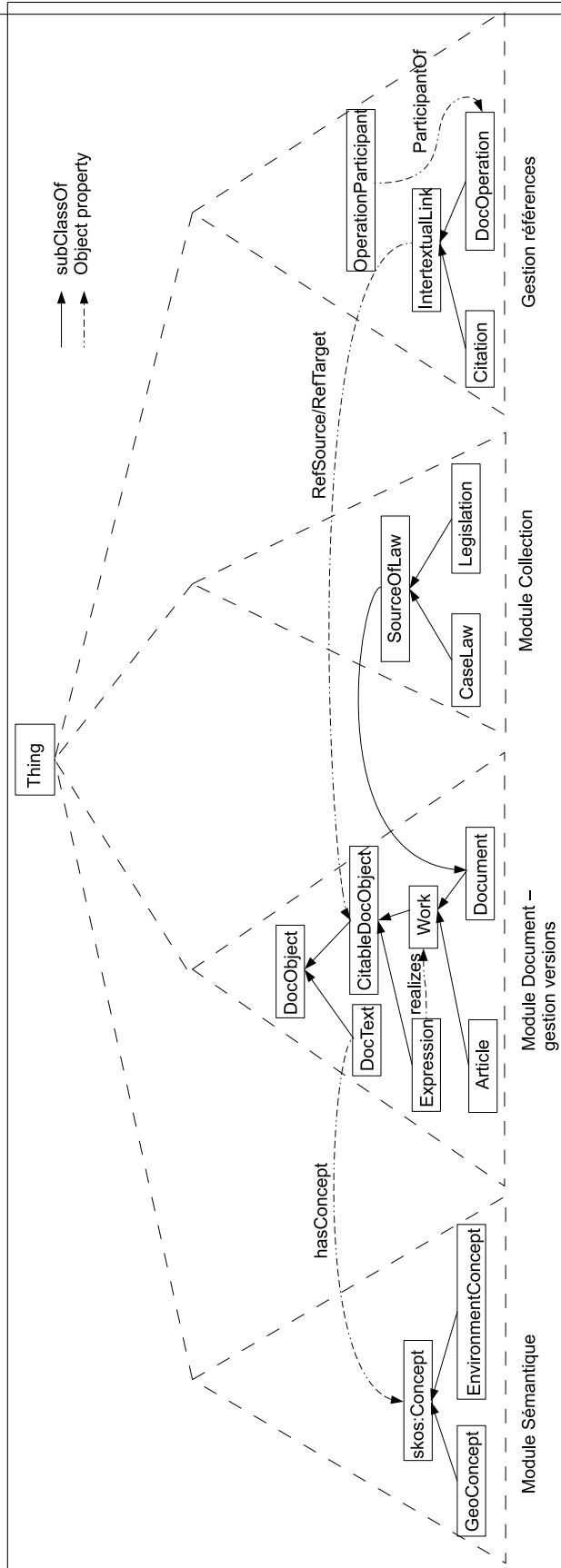


FIGURE 7.21 – Ontologie de collection documentaire avec gestion des versions et des références (relations ternaires).

7.4.4 Positionnement par rapport au standard juridique Metalex

Document vs. collection La vision de l'ontologie Metalex est centrée autour du document : les classes et propriétés sont décrites relativement au document qui est annoté avec le standard, alors que dans notre modélisation, nous nous focalisons sur la collection et pas sur un document. L'ontologie Metalex a d'abord été conçue pour modéliser la législation, nous élargissons la typologie des documents pour prendre en compte et décrire davantage de documents : les textes produits par les collectivités locales et la jurisprudence. Pour les besoins du projet Légilocal, les sources de droit sont traitées comme une collection de documents qui sont de différents types, reliés par différents types de citations.

Liens intertextuels (citation vs. référence) et annotations sémantiques Les deux notions, référence et citation, sont définies dans l'ontologie Metalex. Elles permettent de faire la distinction entre les liens vers des objets textuels (*cites*) ou des objets non-textuels (*refersTo*). Une citation a comme cible un objet bibliographique (par exemple l'article 1, le premier article, l'article précédent), tandis que la référence est un élément qui se réfère à tout type d'entité intéressante mais non-bibliographique (par exemple, le ministre, le Président de la République, l'accusé).

Dans Metalex, la modification, par exemple, est une action bibliographique (sous classe de *Action*), *BibliographicModification*. Elle a comme résultat un objet bibliographique qui peut être un document (expression manifestation ou item) ou une citation (ce type de résultat n'est pas indiqué explicitement mais il n'est pas exclu, nous considérons qu'il est possible). La source de la modification est le document courant (celui qui contient le lien) et la cible est le document relié par la relation *matter*. Une citation (*metalex:BibliographicCitation*) relie les ressources (objets bibliographiques) au niveau des articles (plutôt qu'au niveau des éléments dans le texte portant la référence) aux ressources citées. C'est ce que nous exprimons dans notre modèle par la dualité texte de document / unité documentaire (*DocumentText/DocumentaryUnit*) où l'objet référencé ou cité est une unité documentaire (article ou document) et l'objet annoté est tout fragment de texte.

Notre modèle de gestion de références diffère de celui de Metalex sur deux points. D'une part, nous affinons la notion générique citation/référence en introduisant divers sous-types de références (modification, codification, etc.) et de citations (application, visas, etc.). D'autre part nous distinguons les annotations sémantiques des liens intertextuels. En effet, une large distinction oppose les citations et les références qui font référence à un objet textuel (liens intertextuels) et les annotations sémantiques qui font référence à des objets non textuels (concepts d'une ressource sémantique).

Gestion des évènements vs. opérations documentaires Dans l'ontologie Metalex, les références et les citations (sous-types de référence) sont représentées comme le résultat d'un évènement. Les références (*metalex:BibliographicReference*) et les évènements *metalex:Event* sont représentés par des entités différentes reliées par la propriété *metalex:resultOf*. Il existe plusieurs types d'évènements (creation, modification, etc.) mais un seul type générique de référence.

Dans notre modèle, nous proposons une représentation compacte d'un évènement et d'une référence dans une seule entité qui décrit à la fois les liens (traces dans les textes) et les opérations génératrices de ces liens. La modélisation proposée permet de gérer les liens intertextuels comme des opérations : une unique opération documentaire est créée, bien que trois liens puissent être produits, ce qui permet de réduire les efforts d'instanciation. Elle a également l'avantage de

garder une cohérence des données lors de la manipulation des références entre les documents source, cible et résultat impliqués dans une opération documentaire.

7.5 Mise en œuvre des ontologies documentaires

La modélisation d'une collection juridique revient à instancier l'ontologie documentaire en produisant un ensemble de triplets RDF. Sont ainsi modélisés les documents et leurs types (**Legislation**, **Jurisprudence**, etc), les articles (**CodifiedArticle**, **UncodifiedArticle**), les concepts terminologiques (**Environment**, **Organisation**, etc.), les annotations sémantiques (**hasConcept**), les liens entre documents (**cites**, **modifies**, etc). La collection de documents est ainsi représentée comme une base de connaissances qui peut ensuite être interrogée à l'aide de requêtes SPARQL.

Les modèles ontologiques représentés ci-dessus offrent des fonctionnalités avancées de recherche et permettent de répondre à des requêtes relationnelles par des graphes de documents (figure 7.22) :

1. *Est-ce que ce texte de loi a été modifié ? à quelle date ? quelle est la nouvelle version (résultante après la modification) ?*
2. *Par quel texte juridique l'articles 46 a-t-il été codifié ? quel est l'agent (ou l'institution juridique) qui a effectué cette codification ?*
3. *À quelle date la loi 1994 a-t-elle été abrogée ? et par quel texte juridique ?*

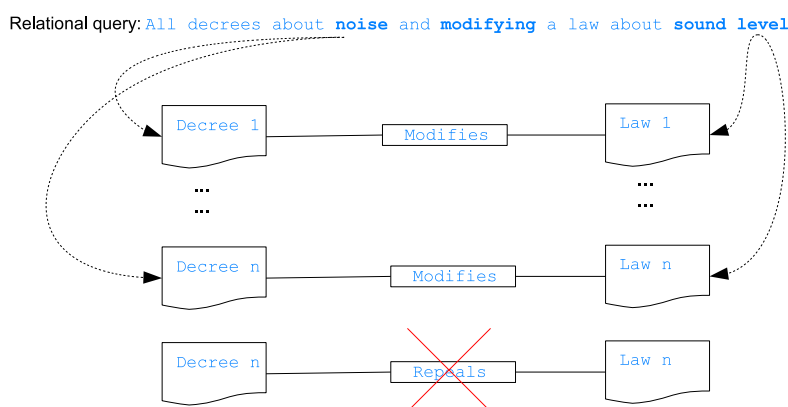
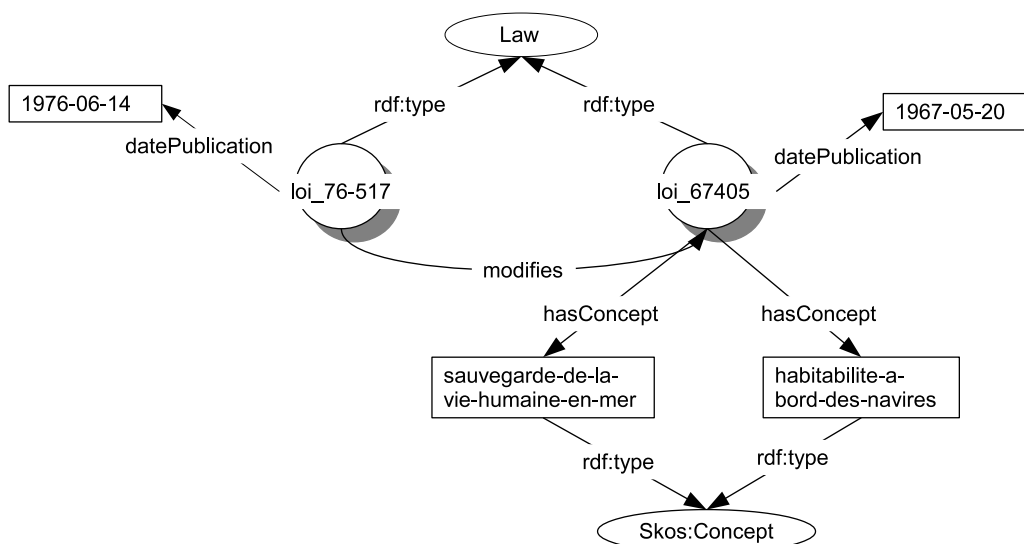


FIGURE 7.22 – Graphes réponses à une requête relationnelle.

7.5.1 Instanciation et interrogation dans la première ontologie

Les figures 7.23, 7.24 et 7.25 montrent trois extraits de collections juridiques modélisées à l'aide du modèle documentaire de la première ontologie :

1. Loi 76 – 517 du 14 juin 1976 qui modifie la loi 67405 du 20 – 05 – 1967 sur la sauvegarde de la vie humaine en mer et l'habitabilité à bord des navires (figure 7.23).
2. Loi n° 57 – 362 du 23 mars 1957 RUR. qui modifie l'article 402 du code rural sur la pêche fluviale (figure 7.24).
3. Décret n° 2004 – 62 du 14 janvier 2004 modifiant le décret n° 99 – 508 du 17 juin 1999 qui cite l'article 266 du code des douanes et instituant une taxe générale sur les activités polluantes (figure 7.25).

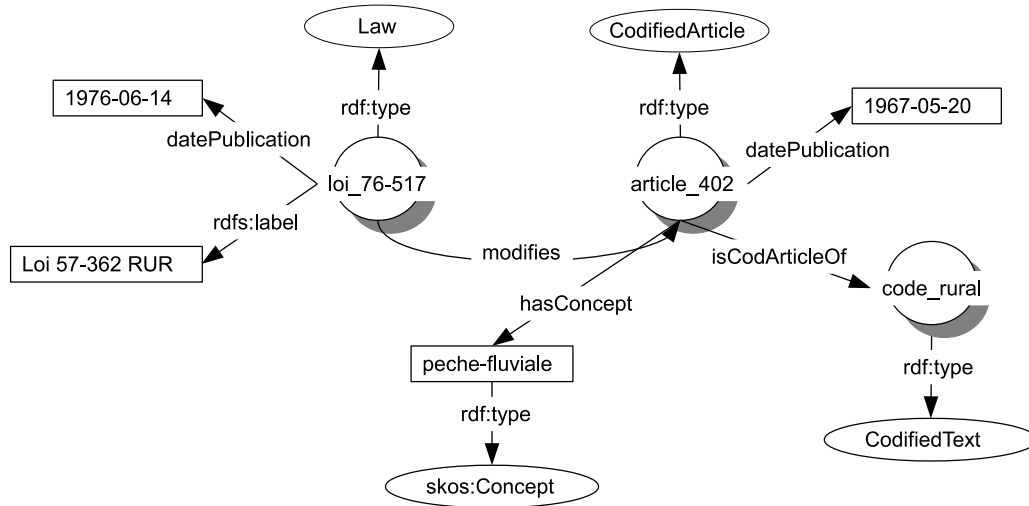


```

1 @prefix : <http://example.org#>
2 @prefix dmo: <http://www-lipn.univ-paris13.fr/~mimouni/owl/2014/06/DocModelOntology.owl#> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5 @prefix dct: <http://purl.org/dc/terms/> .
6 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
7
8
9 :loi_76-517
10   a dmo:Law ;
11   rdfs:label "loi 76-517"^^xsd:string ;
12   dmo:datePublication "1976-06-14"^^xsd:date ;
13   dmo:modifies :loi_67405 ;
14   skos:prefLabel "loi 76-517"^^xsd:string .
15
16 :loi_67405
17   a dmo:Law ;
18   rdfs:label "loi 67405"^^xsd:string ;
19   dmo:datePublication "1967-05-20"^^xsd:date ;
20   dmo:hasConcept :habitabilite_à_bord_des_navires , :sauvegarde_de_la_vie_humaine_en_mer ;
21   skos:prefLabel "loi 67405"^^xsd:string .
22
23 :habitabilite_à_bord_des_navires
24   a skos:Concept ;
25   rdfs:label "habitabilité à bord des navires"^^xsd:string ;
26   skos:prefLabel "habitabilité à bord des navires"^^xsd:string .
27
28 :sauvegarde_de_la_vie_humaine_en_mer
29   a dmo:EnvironmentConcept ;
30   rdfs:label "sauvegarde de la vie humaine en mer"^^xsd:string ;
31   skos:prefLabel "sauvegarde de la vie humaine en mer"^^xsd:string .

```

FIGURE 7.23 – Exemple 1

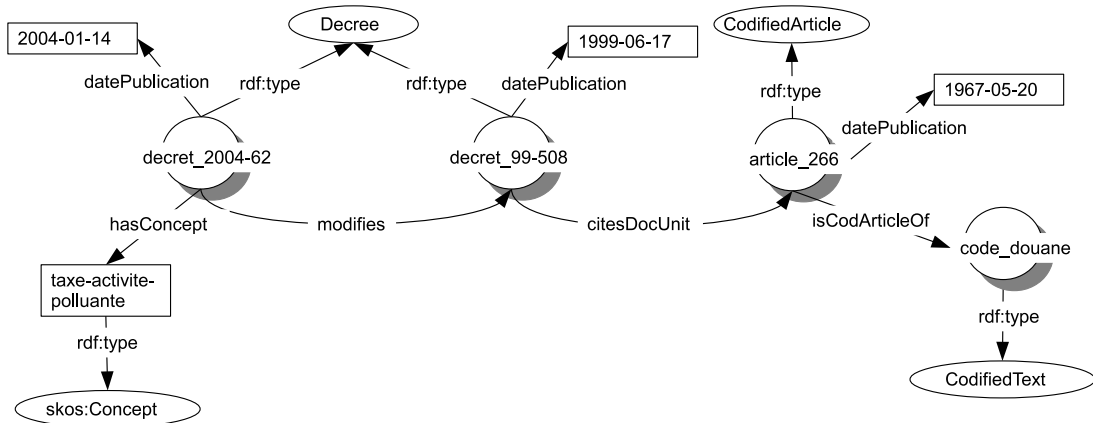


```

1 @prefix : <http://example.org#>
2 @prefix dmo: <http://www-lipn.univ-paris13.fr/~mimouni/owl/2014/06/DocModelOntology.owl#> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5 @prefix dct: <http://purl.org/dc/terms/> .
6 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
7
8
9 :loi_57-362
10   a dmo:Law ;
11   rdfs:label "loi 57-362 RUR"^^xsd:string ;
12   dmo:datePublication "1957-03-23"^^xsd:date ;
13   dmo:modifies :article_402 ;
14   skos:prefLabel "loi 57-362 RUR"^^xsd:string .
15
16 :article_402
17   a dmo:CodifiedArticle ;
18   rdfs:label "article 402"^^xsd:string ;
19   dmo:hasConcept :peche_fluviale ;
20   dmo:isCodArticleOf :code_rural ;
21   skos:prefLabel "article 402"^^xsd:string .
22
23
24 :code_rural
25   a dmo:CodifiedText ;
26   rdfs:label "code rural"^^xsd:string ;
27   skos:prefLabel "code rural"^^xsd:string .

```

FIGURE 7.24 – Exemple 2



```

1 @prefix : <http://example.org#>
2 @prefix dmo: <http://www-lipn.univ-paris13.fr/~mimouni/owl/2014/06/DocModelOntology.owl#> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5 @prefix dct: <http://purl.org/dc/terms/> .
6 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
7
8
9 :decret_2004-62
10   a dmo:Decree ;
11   rdfs:label "decret 2004-62"^^xsd:string ;
12   dmo:datePublication "2004-01-14"^^xsd:date ;
13   dmo:hasConcept :taxe_activite_polluante ;
14   dmo:modifies :decret_99-508 ;
15   skos:prefLabel "decret 2004-62"^^xsd:string .
16
17 :decret_99-508
18   a dmo:Decree ;
19   rdfs:label "decret 99-508"^^xsd:string ;
20   dmo:citesDocUnit :article_266 ;
21   dmo:datePublication "1999-06-17"^^xsd:date ;
22   skos:prefLabel "decret 99-508"^^xsd:string .
23
24 :article_266
25   a dmo:CodifiedArticle ;
26   rdfs:label "article 266"^^xsd:string ;
27   dmo:isCodArticleOf :codes_des_douanes ;
28   skos:prefLabel "article 266"^^xsd:string .
29
30 :codes_des_douanes
31   a dmo:CodifiedText ;
32   rdfs:label "codes des douanes"^^xsd:string ;
33   skos:prefLabel "codes des douanes"^^xsd:string .

```

FIGURE 7.25 – Exemple 3

L'adoption d'un modèle documentaire unifié pour coder la structure des documents, leurs annotations sémantiques et la structure sémantique de la collection permet de traiter des requêtes complexes combinant des critères de recherche structurels, intertextuels et de contenu. Par exemple,

- si un administrateur local veut trouver des exemples d'actes locaux qui traitent des « routes rurales » et qui citent un décret particulier *d*, il peut exprimer une requête combinant des contraintes sur l'annotation sémantique (utilisant la propriété `hasConcept` vers un concept de la classe des termes `cheminRural`) et sur les liens entre documents (citer (`cites`) le décret *d*).
- De même, un secrétaire de mairie peut rechercher des arrêtés existants (`orders`) parlant de la circulation de camions électriques (`camionElectrique`) dans la région parisienne (`regionParisienne`) pendant les pics de pollution (`pics de pollution`) et citant l'article `article-R.221-1` du code de l'environnement (`codeEnvironnement`).
- Un agent de collectivité territoriale (`collectiviteTerritoriale`) peut être amené à créer un arrêté local (`arreteLocal`) en application (`applies`) de l'arrêté inter-préfectoral relatif à la procédure d'information et d'alerte du public (`procedureInformation`, `alertePublic` en cas de pointe de pollution atmosphérique (`pollutionAtmospherique`) en région d'Île-de-France (`region-Ile-de-France`)¹¹⁰. Pour ce faire, il peut rechercher des exemples d'actes locaux de communes voisines sur le même sujet pour s'en inspirer.

Les requêtes peuvent porter sur différents aspects de la collection :

1. Le contenu sémantique d'un type donné de documents :
 - *Quels sont les textes qui parlent de la préservation de l'environnement ?*
 - *Quels articles traitent de la responsabilité pour faute ou responsabilité pour négligence ?*Ces requêtes portent sur :
 - les classes `DocumentText` et `Article`,
 - les concepts terminologiques `Environnement` et `Juridique`,
 - la propriété `hasConcept`.
2. L'historique d'une unité documentaire (versions), qu'il s'agisse d'un document ou de l'un de ses articles :
 - *Comment a été abrogé l'article 22 de la loi sur l'enseignement obligatoire ?*
 - *Quelles sont les différentes versions de l'Article 1382 du Code Civil ?*
 - *Trouver la version en vigueur de l'Article 1328 avec sa date de modification.*Ces requêtes font appel aux :
 - classes `Article`, `CodifiedText` (Code Civil) et `UncodifiedText` (la loi sur l'enseignement obligatoire),
 - attribut `legalState` (`inForce`),
 - propriétés `hasPreviousVersion`, `dateModification` et `abrogates`. La requête sur l'abrogation doit retourner tous les documents qui ont abrogé l'article 22 en question (si la version de l'article 22 considérée n'est pas précisée, tous les textes abrogatifs doivent être retournés).
3. Les types de documents et les types des liens :
 - *Donnez moi les jurisprudences qui ont appliqué l'article 4 actuellement en vigueur de la loi Sapin.*

110. http://www.driee.ile-de-france.developpement-durable.gouv.fr/IMG/pdf/20111027-arrete_interprefectoral_pointe_de_pollution_cle7a15da.pdf

- Par quel texte l'article 1382 du Code Civil a-t-il été créé ?

Ces requêtes portent sur :

- les classes `CaseLaw` (jurisprudence), `Article` et `UncodifiedText`,
- les propriété `appliesLegislation` (lien d'application entre jurisprudence et législation), `isCreatedBy`.

Nous pouvons aussi poser des requêtes sur la consolidation d'un texte à une date donnée. Cela suppose un calcul un peu plus compliqué, puisqu'il faut partir de la structure du texte, identifier la liste des articles qui le composent et retrouver pour chacun la version en vigueur à la date considérée.

Exemple collection Bruit Reprenons l'exemple illustratif étudié dans le chapitre 6.

Cette collection est modélisée par une ontologie documentaire comme le montre la figure 7.26. Les documents de types arrêtés et décrets correspondent à des instances des classes `Order` et `Decree`. Les annotations sémantiques des documents représentent des instances du concept terminologique `EnvironmentConcept` et spécifient le concept `Noise`, ils sont décrits dans la figure 7.27. Le lien de référence entre les documents est représenté par la propriété `makesReference` de la classe des arrêtés vers la classe des décrets.

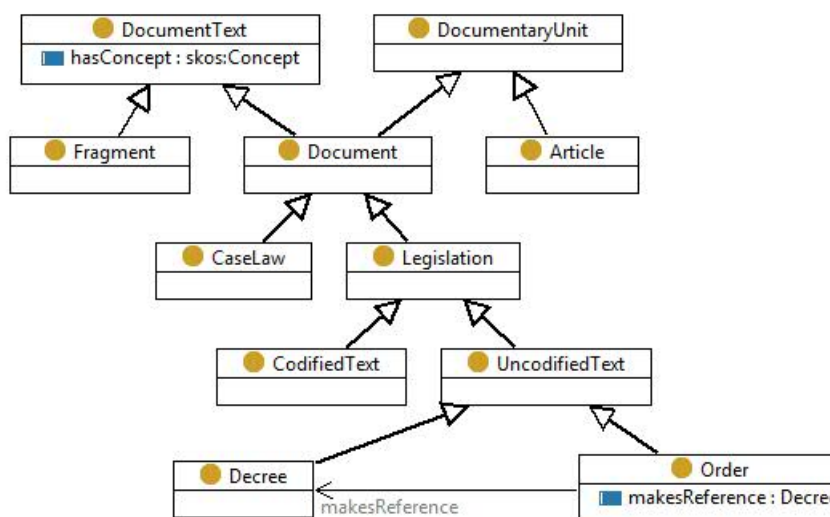


FIGURE 7.26 – Modélisation de la collection arrêtés-décrets.

Requête 1 : *Quels sont les documents qui parlent de nuisance sonore ?* Cette requête vise à retrouver tous les documents (arrêtés ou décrets) qui sont annotés sémantiquement par le concept `nuisance sonore (ns)`.

```

1 SELECT *
2 WHERE {
3 ?subject :hasConcept ?concept .
4 ?concept skos:prefLabel "ns" .
5 }
  
```

La réponse à cette requête est formée des deux arrêtés de Boulogne et des Yvelines :

Subject	Concept
AB	ns
AY	ns

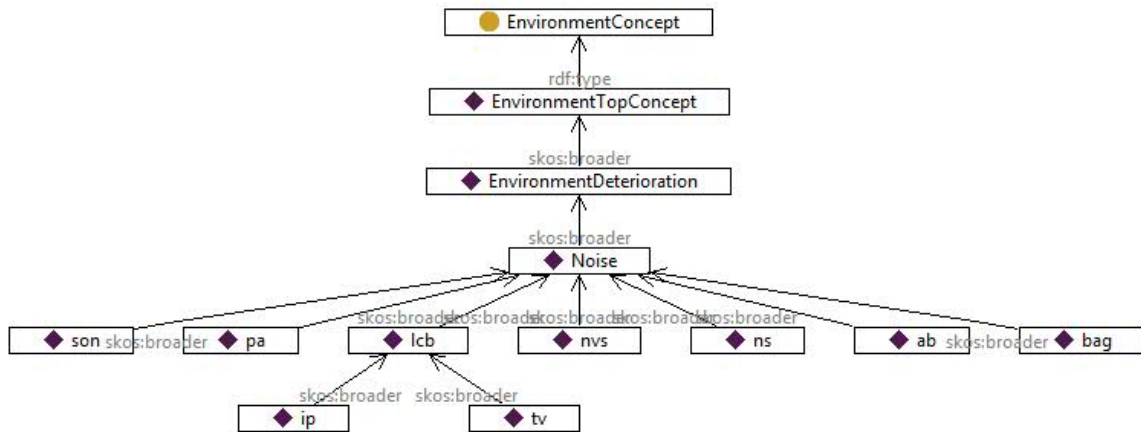


FIGURE 7.27 – Annotations sémantiques des arrêtés et des décrets.

Requête 2 : *Quels sont les documents (arrêtés ou décrets) qui parlent de nuisance sonore ou de lutte contre le bruit ?* Cette requête vise à retrouver tous les documents (arrêtés ou décrets) qui sont annotés sémantiquement par le concept nuisance sonore (ns) ou le concept lutte contre le bruit (lcb).

```

1 SELECT *
2 WHERE {
3   ?subject :hasConcept ?concept .
4   {?concept skos:prefLabel "ns"}
5   UNION {?concept skos:prefLabel "lcb"} .
6 }
    
```

La réponse à cette requête est formée par les deux arrêtés de Boulogne et des Yvelines annotés par le concept ns auxquels s'ajoute la loi 1992 et le décret 1995 tous les deux annotés par le concept lcb :

Subject	Concept
AB	ns
AY	ns
L92	lcb
D95	lcb

Requête 3 : *Quels sont les arrêtés qui font référence à des décrets qui parlent de lutte contre le bruit ?* Cette requête vise à retrouver les documents de type arrêté qui ont une relation fait-référence vers des documents de type décret annotés par le concept lutte contre le bruit (lcb).

```

1 SELECT *
2 WHERE {
3   ?subject :makesReference ?object .
4   ?object :hasConcept ?concept .
5   ?concept skos:prefLabel "lcb" .
6 }
    
```

La réponse à cette requête est donnée par deux graphes réponse dont les noeuds sont reliés par la relation fait-référence : arrêté de Boulogne - loi 1992 et arrêté de Paris - décret 1995.

Subject	Object	Concept
AB	L92	lcb
AP	D95	lcb

7.5.2 Instanciation et interrogation dans la deuxième ontologie

Considérons les exemples suivants extraits de la collection Légilocal (décrite dans le chapitre 5).

L'article L362-1 du code l'environnement possède trois versions (expressions) :

1. La 1ère version en vigueur le 21 septembre 2000 créée par un évènement de codification : (dont elle est le résultat (*result*))
 - par l'Ordonnance n°2000-914 du 18 septembre 2000 (publiée au JORF le 21/09/2000) relative à la partie législative du code de l'environnement (source)
 - à partir de l'ancien texte : Loi n°91-2 du 3 janvier 1991 relative à la circulation des véhicules terrestres dans les espaces naturels et portant modification du code des communes (objet (*matter*) et cible (*target*)).
2. La 2ème version en vigueur le 15 avril 2006 créée par un évènement de modification : (dont elle est le résultat)
 - par la Loi n°2006-436 du 14 avril 2006 (JORF du 15/04/2006) relative aux parcs nationaux, aux parcs naturels marins et aux parcs naturels régionaux (source)
 - à partir de la 1ère version ci-dessus (objet et cible).
3. La 3ème version en vigueur le 1^{er} juillet 2013 créée par un évènement de modification (dont elle est le résultat) :
 - par l'Ordonnance n°2012-34 du 11 janvier 2012 (source)
 - à partir de la 2ème version ci-dessus (objet et cible).

L'Ordonnance n°2000-914 du 18 septembre 2000 abroge la Loi n°91-2 du 3 janvier 1991 relative à la circulation des véhicules terrestres dans les espaces naturels et portant modification du code des communes. Un évènement d'abrogation a lieu avec :

- comme source : l'Ordonnance n°2000-914 du 18 septembre 2000,
- comme objet et cible : la Loi n°91-2 du 3 janvier 1991.

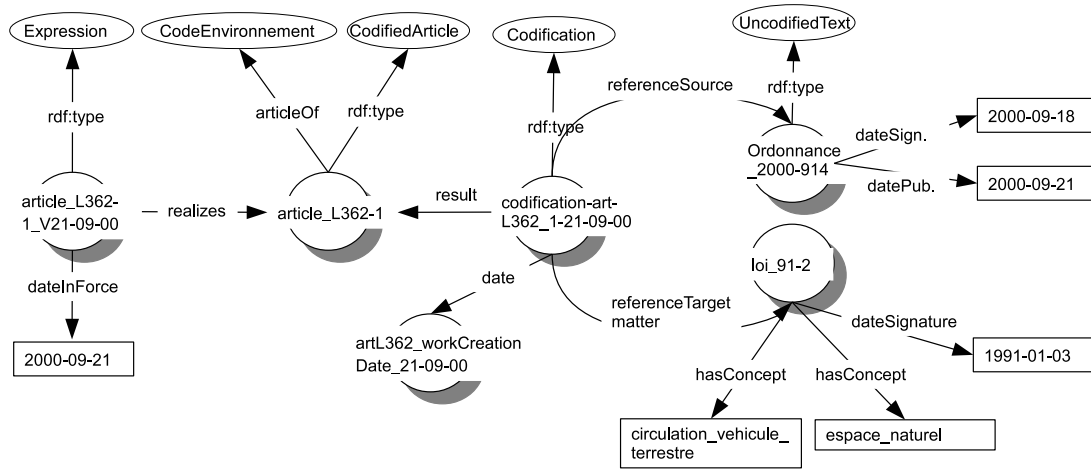
La modélisation de cet extrait de collection utilisant le modèle documentaire de la deuxième ontologie est donnée par les figures 7.28 (codification), 7.29 (première modification), 7.30 (deuxième modification) et 7.31 (abrogation).

Les requêtes peuvent porter sur différents aspects de la collection, par exemple la version en vigueur d'un article à une date donnée ou le texte qui modifie une version d'un article en vigueur à une date donnée. Sur l'exemple de collection décrit ci-dessous, nous pouvons répondre aux requêtes suivantes :

Requête 1 : *Quelle est la version en vigueur de l'article 362-1 du code de l'environnement au 26/09/2007 ?*

Cette requête fait appel à :

- la classe `DocumentaryUnitExpression` : le type de l'objet recherché est une expression (version d'un article),
- la propriété `realizes` : l'expression réalise l'œuvre "article L362-1" ,
- la propriété `dateInForce`.



```

1 @prefix : <http://example.org/ontology2#> .
2 @prefix lido: <http://www-lipn.univ-paris13.fr/~mimouni/owl/2013/12/docOntology#> .
3 @prefix metalex: <http://www.metalex.eu/metalex/2008-05-02#> .
4 @prefix dct: <http://purl.org/dc/terms/> .
5
6
7 :codification_artL362-1_21-09-00
8   a lido:Codification ;
9   rdfs:label "codification art l362-1 21-09-00"^^xsd:string ;
10  lido:referenceSource :Ordonnance_2000-914 ;
11  lido:referenceTarget :loi_91-2 ;
12  metalex:date :artL362_workCreationDate_21-09-00 ;
13  metalex:matter :loi_91-2 ;
14  metalex:result :article_L362-1 .
15
16 :Ordonnance_2000-914
17   a lido:Ordinanance ;
18   rdfs:label "Ordonnance 2000-914"^^xsd:string ;
19   dct:title "Ordonnance n°2000-914 du 18 septembre 2000 (publiée au JORF le 21/09/2000)
20 relative à la partie législative du code de l'environnement"^^xsd:string ;
21   lido:datePublication "2000-09-21"^^xsd:date ;
22   lido:dateSignature "2000-09-18"^^xsd:date .
23
24 :loi_91-2
25   a lido:Law ;
26   rdfs:label "loi 91-2"^^xsd:string ;
27   dct:identifiant "n°91-2"^^xsd:string ;
28   dct:subject "Circulation des véhicules terrestres dans les espaces naturels"^^xsd:string ;
29   dct:title "Loi n°91-2 du 3 janvier 1991 relative à la circulation des véhicules terrestres
30 dans les espaces naturels et portant modification du code des communes"^^xsd:string ;
31   lido:dateSignature "1991-01-03"^^xsd:date ;
32   lido:hasConcept :circulation_vehicule_terrestre , :espace_naturel .
33
34 :article_L362-1
35   a lido:CodifiedArticle ;
36   rdfs:label "article L362-1"^^xsd:string ;
37   metalex:realizedBy :article_L362-1_V21-09-00 .
38
39 :article_L362-1_V21-09-00
40   a lido:DocumentaryUnitExpression ;
41   rdfs:label "article L362-1 V21-09-00"^^xsd:string ;
42   lido:dateInForce "2000-09-21"^^xsd:date ;
43   lido:expressionType "\"codified article\""^^xsd:string ;
44   metalex:realizes :article_L362-1 .

```

FIGURE 7.28 – Codification de l'article L362 – 1 du code de l'environnement par l'Ordonnance n°2000 – 914.

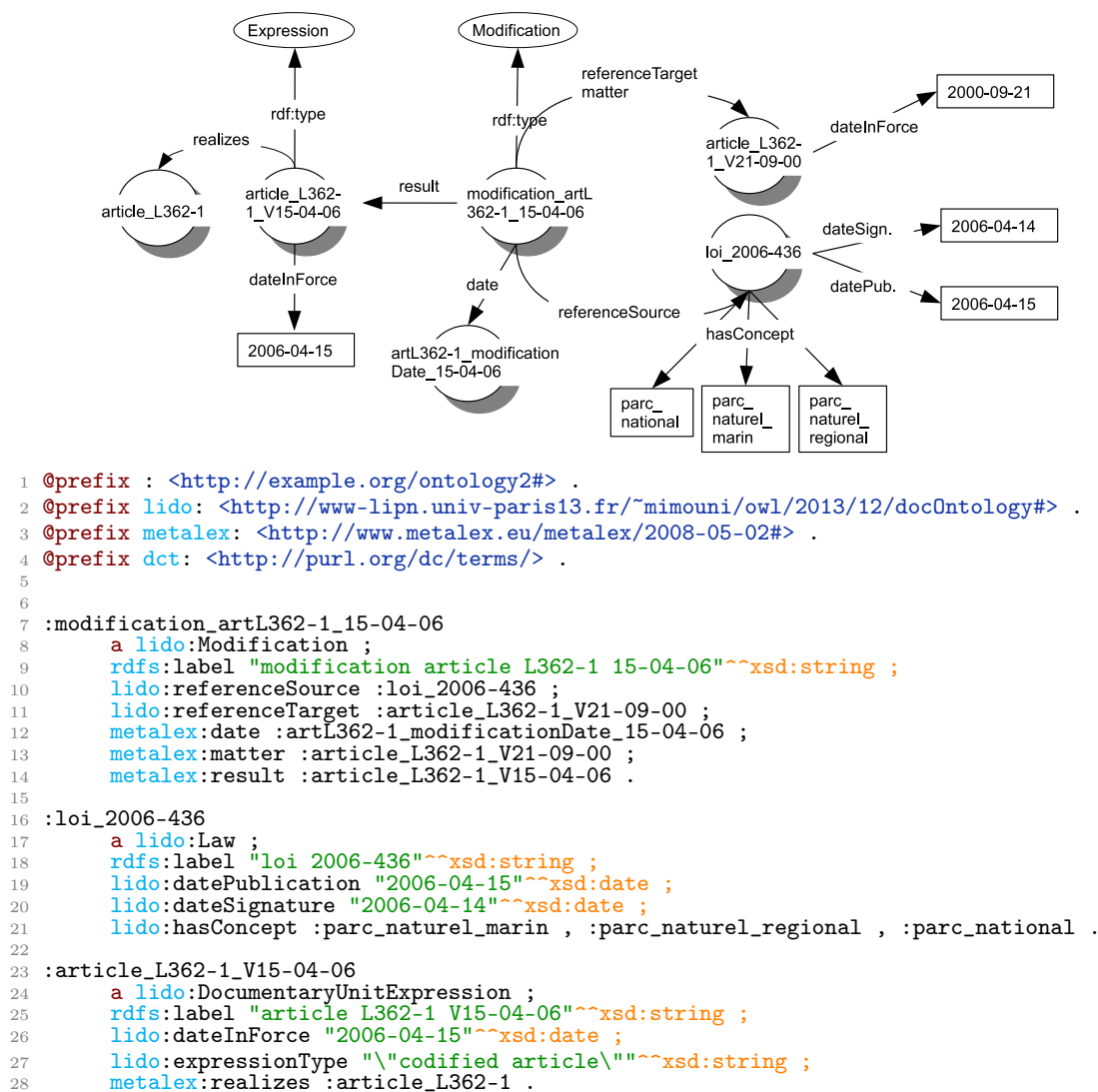


FIGURE 7.29 – Modification de l'article L362 – 1 du code de l'environnement par la Loi n°2006 – 436.

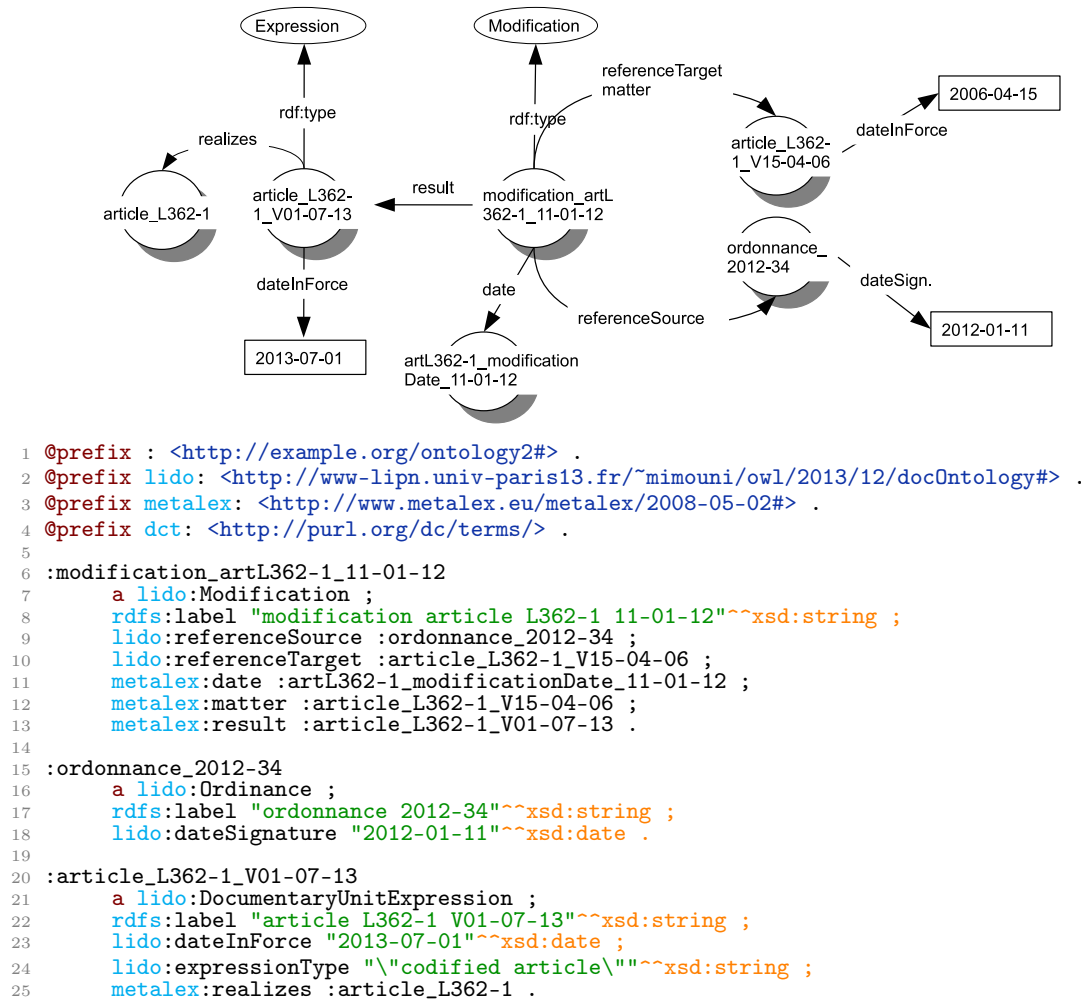


FIGURE 7.30 – Modification de l'article L362 – 1 du code de l'environnement par l'Ordonnance n°2012 – 34.

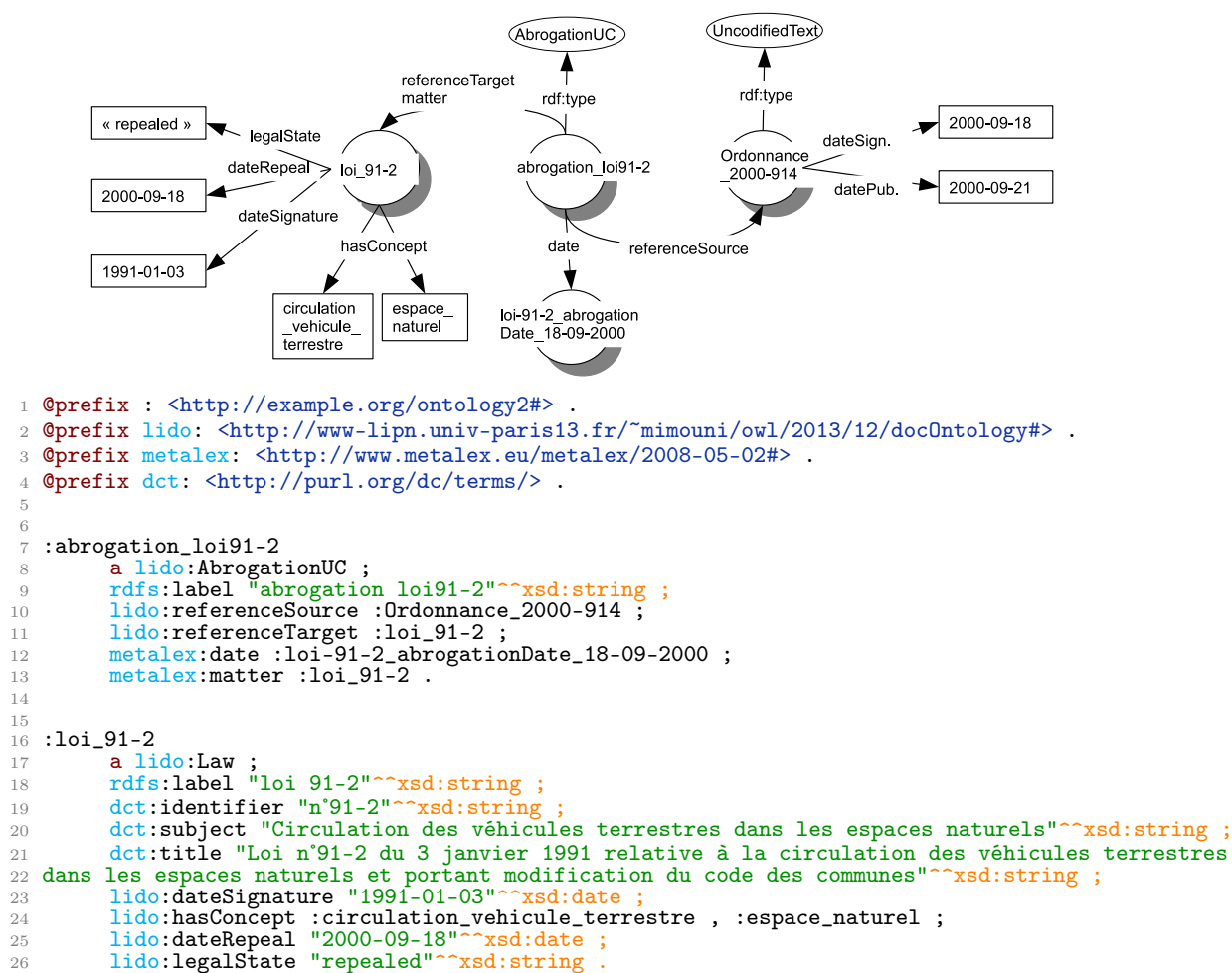


FIGURE 7.31 – Abrogation de la Loi n°2006 – 436 par l’Ordonnance n°2000 – 914.

```

1 PREFIX example: <http://example.org/ontology2>
2 SELECT ?article ?date
3 WHERE {
4   ?article rdf:type lido:DocumentaryUnitExpression .
5   ?article metalex:realizes example:article_L362-1 .
6   ?article lido:dateInForce ?date .
7   FILTER (?date <= "2007-09-26"^^xsd:date) .
8 }
9 ORDER BY DESC(?date) LIMIT 1

```

La réponse à cette requête est donnée par la version de l'article L362-1 entré en vigueur le

15/04/2006 :

article	date
article_L362-1_V15-04-06	2006-04-15

Requête 2 : *Quel texte a abrogé la Loi n°91-2 du 3 janvier 1991 ?*

Cette requête fait appel aux :

- classes Abrogation (événement d'abrogation) et AbrogationDate,
- propriétés referenceSource et referenceTarget de l'évènement.

```

1 SELECT ?texte ?date
2 WHERE {
3   ?abrogevent lido:referenceSource ?texte .
4   ?abrogevent rdf:type lido:Abrogation .
5   ?abrogevent lido:referenceTarget example:loi_91-2 .
6   ?abrogevent metalex:date ?eventDate .
7   ?eventDate rdf:type lido:AbrogationDate .
8   ?eventDate metalex:xsdDate ?date .
9 }

```

La réponse à cette requête est donnée par l'Ordonnance n°2000 – 914 et l'abrogation a eu

lieu le 18/09/2000 :

texte	date
ordonnance_2000-914	2000-09-18

Requête 3 : *Quel texte a modifié la version de l'article 362-1 du code de l'environnement en vigueur au 26/09/2007 ?*

Cette requête fait appel aux :

- classes Modification (événement de modification), ModificationDate et DocumentaryUnitExpression (version d'un article),
- les propriétés referenceSource et referenceTarget de l'évènement, realizes (l'expression réalise l'œuvre "article L362-1") et dateInForce de la version modifiée.

```

1 SELECT ?texte
2 WHERE {
3   ?modifevent lido:referenceSource ?texte .
4   ?modifevent rdf:type lido:Modification .
5   ?modifevent lido:referenceTarget ?article.
6   {SELECT ?article
7     WHERE {
8       ?article rdf:type lido:DocumentaryUnitExpression .
9       ?article metalex:realizes example:article_L362-1 .
10      ?article lido:dateInForce ?date .
11      FILTER (?date <= "2007-09-26"^^xsd:date) .
12     }
13   ORDER BY DESC(?date) LIMIT 1 }
14 }

```

La réponse à cette requête est donnée par l'Ordonnance n°2012–34 :

texte
ordonnance_2012-34

Exemple collection Bruit Sur cette collection, les références entre les documents sont représentées par un seul type de lien « fait-référence » entre les arrêtés (source de la référence) et

les décrets (cible de la référence). La collection ne fournit pas davantage d'information sur la nature de la référence et sur ses propriétés, à savoir l'agent responsable, la date, etc. La relation de référence peut être donc identifiée à une citation dans la deuxième ontologie (classe `Citation`) entre le document source (définir la propriété `citationSource`) et le document cible (définir la propriété `citationTarget`). De plus, la collection ne précise pas de quelles versions de documents il s'agit (objets de la classe `Expression`) et à quelles dates. La création d'objets oeuvres (classe `Work`) et expressions (classe `Expression`) ne peut donc être faite. Ainsi, l'instantiation de l'exemple de la collection Bruit sur la deuxième ontologie oblige à réduire les propriétés du modèle et ne permet pas de mettre en avant le potentiel de cette modélisation par rapport au premier modèle.

7.6 Conclusion

Dans ce chapitre, nous proposons une solution basée sur les technologies sémantiques pour résoudre le problème de la gestion de contenu auquel les collectivités locales françaises et l'administration sont confrontées.

Les deux modèles ontologiques que nous avons présentés permettent de modéliser une collection documentaire avec l'ensemble de ses caractéristiques sous la forme d'un graphe RDF puis de l'interroger de manière sémantique, structurelle, temporelle et relationnelle à l'aide de SPARQL. L'interrogation est immédiate, nous pouvons répondre à toutes les requêtes du chapitre analyse des besoins et à d'autres types de requêtes, mais nous n'avons pas la navigation offerte par la structure relationnelle de l'approche conceptuelle. En terme de faisabilité, les deux ontologies sont possibles à mettre en œuvre, choisir l'une ou l'autre dépend des choix de l'application (types de requêtes, structure de la collection, etc.).

Chapitre 8

Experimentation

Sommaire

8.1	Introduction	181
8.2	Corpus OIT	182
8.2.1	Description du corpus	182
8.2.2	Requêtes OIT et réponses pertinentes	183
8.2.3	Approche conceptuelle : AFC/ARC	183
8.2.4	Approche sémantique : première ontologie	187
8.2.5	Discussion	191
8.3	Corpus LÉGILOCAL	191
8.3.1	Description du corpus	191
8.3.2	Requêtes LÉGILOCAL et réponses pertinentes	193
8.3.3	Exécution sur la première ontologie documentaire	193
8.3.4	Exécution sur la deuxième ontologie documentaire	198
8.3.5	Discussion	201

8.1 Introduction

Dans ce chapitre, nous décrivons les expérimentations que nous avons conduites pour valider les approches de modélisation et de recherche relationnelle proposées. Le but de ces expérimentations est de tester l'intérêt et la faisabilité de l'ajout d'une couche sémantique d'intertextualité. Les besoins d'interrogation intertextuelle étant émergents, il n'existe pas de collections de documents déjà annotées comme décrit dans le chapitre 5, ni de benchmark que nous pouvons utiliser directement. Nous avons cependant pu trouver une collection qui correspond partiellement à notre besoin : elle est constituée d'un ensemble de documents de deux types collectés sur le site de l'Organisation Internationale de Travail. Dans le projet Légilocal nous n'avons pu travailler que sur une petite collection. En effet, le projet a permis de définir des spécifications détaillées sur un échantillon réel de données et de mettre en place le dispositif de collecte et d'annotation de données. La construction effective de la collection est donc sur le point de débiter chez notre partenaire Victoires Éditions. Elle n'était pas exploitable pour notre travail de thèse. Pour valider notre travail, nous avons construit une collection de petite taille composée d'un sous-ensemble de documents collectés dans Légilocal. La collection a été annotée manuellement afin d'extraire la structure des documents, leurs liens intertextuels et leurs contenus sémantiques.

Nous avons testé les approches proposées en parallèle sur les deux corpus introduits dans le chapitre 5 :

- Le corpus de l’OIT contenant un nombre plus grand de documents et un type de lien : l’expérimentation est mise en œuvre avec l’approche conceptuelle et avec la première ontologie de l’approche sémantique. La collection est peu détaillée, elle est de ce fait compatible avec les approches citées.
- Le corpus Légilocal plus riche sémantiquement (types de documents, types de liens) : l’expérimentation est mise en œuvre avec les deux ontologies de l’approche sémantique. La modélisation de la deuxième ontologie étant plus riche, le but est de la comparer avec la première modélisation par rapport à la représentation des relations et la gestion des versions ainsi que l’interrogation de la collection portant sur ces propriétés.

Nous nous sommes concentrée sur les points suivants pour tester la faisabilité des approches :

1. la modélisation de la collection,
2. la formulation des requêtes,
3. l’interrogation (stratégie de recherche),
4. la navigation (si possible).

La section 8.2 décrit le corpus de données OIT, les requêtes formulées sur ce corpus, le traitement de ces requêtes par les approches correspondantes et se termine par une synthèse. La section 8.3 suit la même logique sur le corpus LÉGILOCAL.

8.2 Corpus OIT

8.2.1 Description du corpus

Ce corpus est constitué d’un ensemble d’environ 400 documents concernant le droit international du travail établis par l’Organisation Internationale du Travail¹¹¹ entre 1919 et 2007. Il y a deux types de documents : les conventions (188 documents) et les recommandations (199 documents). Les documents sont identifiés par leurs numéros (C1, C2,..., R1, R2, etc.). Nous avons utilisé une taxonomie de termes reliés au domaine du travail, également accessible sur le site, pour décrire le contenu des documents. En tout, 256 descripteurs ont servi pour annoter le corpus (par exemple : accident du travail, contrat, établissement agricole, bateau de pêche, heures supplémentaires, etc.).

Ces documents contiennent des références vers d’autres documents du corpus, ou vers des articles de la constitution de l’Organisation Internationale du Travail. Parmi les références internes au corpus, il existe à la fois des références entre conventions, entre recommandations, ou de recommandation à convention et inversement. Nous avons distingué parmi ces références différents types de relations entre documents comme par exemple la relation d’implémentation entre une convention et une recommandation, ou la relation de modification entre 2 conventions ou entre 2 recommandations. Dans la suite, nous avons utilisé le lien d’implémentation, qui part des conventions vers les recommandations, et qui est le type de lien le plus fréquent (les autres types sont rares).

Ainsi chaque document possède un ensemble de descripteurs sémantiques de contenu et peut, dans le cas des conventions, avoir une ou plusieurs relations d’implémentation vers des recommandations.

111. <http://www.ilo.org/dyn/normlex/en/f?p=NORMLEXPUB:1:0>. Corpus construit par Thibault Mondary.

8.2.2 Requêtes OIT et réponses pertinentes

Reprenons l'ensemble des requêtes formulées sur le corpus OIT présentées dans le chapitre 5 (section 5.4). Le tableau 8.1 décrit les requêtes (de 1-1 à 1-7) et donne pour chacune la réponse ou l'ensemble de réponses attendues. Nous considérons également les requêtes (de 2-1 à 2-7) proposées pour compléter les types du premier ensemble pour lesquelles nous ne disposons pas de réponses pertinentes (ce sont des requêtes génériques) mais que nous proposons de traiter dans la suite.

Ces requêtes ont été soumises à une juriste qui a identifié les documents à retourner pour chacune. Il existe cependant un biais dans cette expérimentation puisque la juriste a créé l'ensemble des requêtes et leurs réponses en ayant une connaissance parfaite du corpus : les descripteurs des documents (les descripteurs annotant déjà les documents sont utilisés dans les requêtes), le nombre de réponses (requêtes formulées au singulier ou au pluriel selon qu'il existe une ou plusieurs réponses). Les réponses à ces requêtes sont de ce fait toutes retrouvées par nos approches. Ce qui change c'est la stratégie de recherche adoptée pour les retrouver : les requêtes ne sont pas toutes traitées de la même façon.

TABLE 8.1 – Requêtes OIT avec réponses pertinentes.

	Requête	Réponse	
		Convention	Recommandation
OIT1-1	Quelle convention implémente la Recommandation 113 sur la consultation aux échelons industriel et national ?	C144	
OIT1-2	Quelle convention implémente la recommandation qui parle des accidents de travail des marins ?	C164	R142
OIT1-3	Quelles recommandations sont implémentées par la convention qui parle de l'exposition à l'amiante ?	C162	R147, R156, R164, R171
OIT1-4	Quelles sont les recommandations implémentées par les conventions qui parlent de la pollution de l'air ?	C148, C162	R112, R114, R118, R120, R144, R147, R156, R164, R171
OIT1-5	Quelles sont les recommandations implémentées par des conventions qui parlent de la convention collective et de la négociation collective ?	C147, C154	R137, R158, R107, R108
OIT1-6	Quelles conventions implémentent les recommandations qui parlent de bruit et vibrations ?	C120, C148	R118, R120
OIT1-7	Quelle recommandation qui parle du benzène, est implémentée par la convention 139 sur le cancer professionnel ?		R144

8.2.3 Approche conceptuelle : AFC/ARC

Modélisation de la collection Dans cette expérimentation, nous avons travaillé sur un corpus contenant 20 conventions et 30 recommandations annotées avec l'ensemble des attributs, pour deux raisons :

- c'est un corpus clos : le jugement avec des réponses exhaustives a été réalisé sur ce sous-ensemble de documents ;
- le temps de calcul des treillis est raisonnable.

À partir de ces données nous avons construit les contextes formels et relationnels qui modélisent la collection documentaire. La famille de contextes relationnels construite est composée

de :

- un contexte formel de conventions : 20 objets et 256 attributs (descripteurs sémantiques de contenu) ;
- un contexte formel de recommandations : 30 objets et 256 attributs (descripteurs sémantiques de contenu) ;
- un contexte relationnel définissant la relation d’implémentation (convention \times recommandation).

Nous avons mis en place un prototype de test pour la validation de l’approche : nous avons implémenté l’algorithme de recherche et de navigation décrit dans le chapitre 6 (section 6.7) en Java et nous avons créé un module de visualisation de résultats en nous appuyant sur l’API Prefuse¹¹². Nous avons utilisé l’outil Galicia¹¹³ pour la construction des treillis avant et après insertion des requêtes. L’algorithme prend en entrée la famille de treillis construite (exportés au format xml) et fournit en résultat un ensemble de concepts qui sont donnés en entrée au module de visualisation, lequel construit et affiche les graphes réponses. Ce prototype peut être vu comme le point de départ pour le développement d’un outil de recherche relationnelle par treillis de concepts (avec interfaces de saisie de requêtes et d’affichage de résultats) mis à disposition des utilisateurs sur des collections de documents liés.

Nous avons construit la famille de treillis relationnels à partir des contextes décrits ci-dessus (collection réduite de l’OIT). Rappelons que dans cette modélisation, le treillis des conventions (domaine) est enrichi par la relation `implement` vers le treillis des recommandations (co-domaine). Le tableau 8.2 décrit la FTR en nombre de concepts, nombre d’arcs, nombre de niveaux et la compare avec la FTR construite sur toute la collection OIT (en considérant les 188 conventions et les 199 recommandations avec 244 attributs).

TABLE 8.2 – Propriétés de la collection OIT : Nb. objets, Nb. attributs, Nb. concepts dans le treillis, Nb. arcs, Nb. niveaux (hauteur) du treillis.

	#Obj	#Att	#Conc	#Arc	#Niv
Conv.	188	244	5.947	21.797	16
Rec.	198	244	28.341	126.888	18
Conv. réduit	20	244	134	333	9
Rec. réduit	31	244	494	1433	13

La figure 8.1 illustre le treillis des conventions de la collection réduite de l’OIT avant enrichissement relationnel¹¹⁴. Vu leurs tailles, les treillis enrichis ne sont pas facilement visualisables pour exploration. Nous proposons la navigation par calcul de voisinage et par généralisation ou spécialisation et le module de visualisation de graphes résultats comme alternatives qui exploitent le potentiel de cette structure relationnelle. Ces possibilités sont discutées dans la suite.

Traitement des requêtes : formulation, interrogation, navigation Une fois le modèle de la collection construit, nous procédons au traitement des requêtes décrites dans le tableau 8.1. Ces requêtes sont toutes relationnelles. Au-delà de l’algorithme d’interrogation, nous avons adopté une stratégie complète de recherche (voir section 6.5.1) pour le traitement de ces requêtes. Elle consiste en plusieurs étapes :

112. <http://prefuse.org/>

113. <http://www.iro.umontreal.ca/galicia/>

114. Visualisé par Galicia.

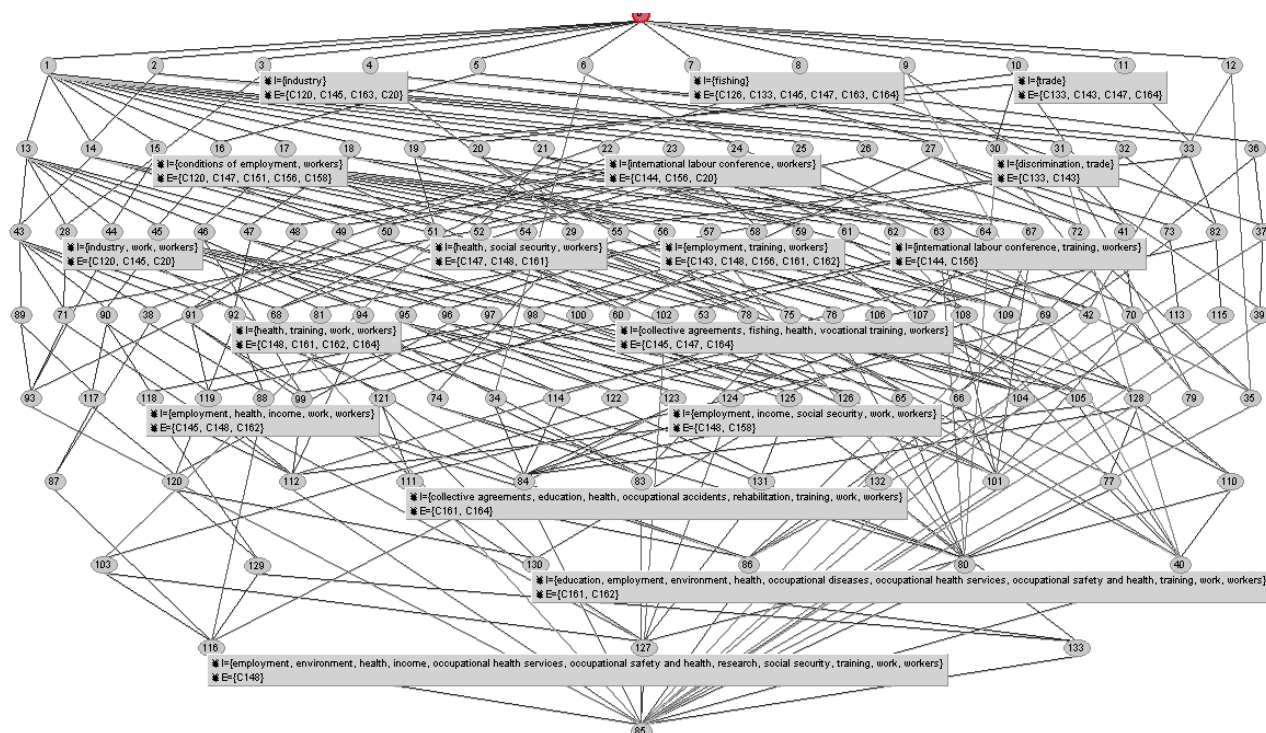


FIGURE 8.1 – Treillis des conventions avant enrichissement relationnel.

1. Décomposition/formulation de la requête : à partir de la requête en langage naturel, construire les concepts à insérer dans les treillis (extraire les descripteurs sémantiques et le type de relation). Pour les besoins de test, nous avons développé une interface par formulaire que nous avons testée sur l'exemple de la collection (arrêtés, décrets).
2. Interrogation et navigation (si possible) : construire les treillis enrichis et localiser les résultats pertinents (algorithme d'interrogation et de navigation).
3. Affichage des résultats : visualisation des graphes réponses (module de visualisation).

Le présupposé d'unicité dans les requêtes n'est pas pris en compte, nous cherchons à chaque fois toutes les réponses (listes, couples, graphes) possibles. Nous ne faisons pas la différence entre requêtes avec ou sans cible dans l'affichage des résultats : nous affichons dans le résultat tous les objets qui sont mis en relations. Les traitements des requêtes diffèrent selon le nombre d'objets virtuels à ajouter. Le cas le plus général est l'ajout de deux objets virtuels (requêtes OIT1-2 et OIT1-6), un seul objet virtuel est ajouté lorsque qu'il existe un objet identifié dans la requête (requêtes OIT1-1 et OIT1-7). Dans ces deux cas, localiser les objets virtuels permet de trouver la réponse. Lorsqu'il n'existe pas de contraintes sur les identifiants, attributs ou relations de la requête, le traitement s'effectue comme pour une requête simple puis la partie relationnelle de la réponse est lue directement sur les contextes relationnels. C'est le cas lorsque la cible est une recommandation qui n'est pas identifiée et qui n'est pas décrite par des attributs (requêtes OIT1-3, OIT1-4 et OIT1-5).

Les points suivants décrivent étape par étape le cas général de la stratégie de recherche adoptée. Ils concrétisent l'algorithme 2 de recherche relationnelle du chapitre 6.7. La première partie concerne la préparation des requêtes et la deuxième partie concerne la recherche de réponses pertinentes. L'algorithme retourne une liste de couples d'objets qui sont utilisés par le module

de visualisation pour construire les graphes réponses.

Étapes de la stratégie de recherche : Cas général

Entrée	- Famille de contextes relationnels : Conventions, Recommandations, Relation Implement. - Une requête relationnelle : " $(Conv, \{att_c\}), impl, (Rec, \{att_r\})$ ", telle que : $Conv \in \{Conv_i, QueryConv\}$: objet convention identifié ou virtuel $Rec \in \{Rec_i, QueryRec\}$: objet recommandation identifié ou virtuel $\{att_x\}$: ensemble d'attributs, qui peut être vide
Sortie	Liste de couples d'objets pertinents : $\{C\}$ (conventions) , $\{R\}$ (recommandations)

Préparation : formulation des requêtes et enrichissement des contextes formels

1. Ajout de $(QueryConv, \{att_c\})$ au contexte Conventions
2. Ajout de $(QueryRec, \{att_r\})$ au contexte Recommandations
3. Ajout de la relation $(Conv \times Rec)$ dans le contexte relationnel

Construction de la famille des treillis relationnels

Construire les treillis : $LatConv$ (conventions) et $LatRec$ (recommandation) (procédure MULTIFCA de Galicia)

Recherche d'objets pertinents

- | | |
|-------|--|
| Cas 1 | Deux objets virtuels ou un objet virtuel et un objet identifié
<ol style="list-style-type: none"> 1. \mathcal{C}_C = identifier les concepts $Conv$ dans $LatConv$ (les concepts les plus spécifiques des objets dans la même extension que $QueryConv$ ou l'objet identifié) 2. \mathcal{C}_R = identifier les concepts Rec dans $LatRec$ (les concepts les plus spécifiques des objets dans la même extension que $QueryRec$ ou l'objet identifié) 3. Résultat : $\{C\} = \cup Ext(\mathcal{C}_C)$, $\{R\} = \cup Ext(\mathcal{C}_R)$ |
| Cas 2 | Un seul objet virtuel
<ol style="list-style-type: none"> 1. \mathcal{C}_C = identifier les concepts $Conv$ dans $LatConv$ (les concepts les plus spécifiques des objets dans la même extension que $QueryConv$ ou l'objet identifié) 2. Lire sur le contexte relationnel, pour chaque objet dans \mathcal{C}_C, les objets Rec_i qu'il implémente 3. Résultat : $\{R\} = \cup \{Rec_i\}$ |

Nous détaillons dans ce qui suit le déroulement des étapes de cette stratégie sur les requêtes OIT1-1 et OIT1-2 (correspondant à chacun de ces cas) et les possibilités de navigation offertes :

OIT1-2 : Cette requête reflète le cas le plus général d'interrogation (ajout de deux objets virtuels). La requête est formulée comme suit : " $(QueryConv), impl, (QueryRec, \{accidents\ de\ travail, marin\})$ ". L'algorithme d'interrogation retourne l'objet R142 (dans l'extension du concept contenant QueryRec) et l'objet C164 (dans l'extension du concept contenant QueryConv). Le graphe réponse de cette requête, créé par le module de visualisation, est donné par la figure 8.2. Sur ce graphe, la partie centrale correspond à la réponse exacte (R142) qui possède les attributs de la requête. La partie droite contient, en plus de la réponse exacte, plusieurs réponses approchées (R107, R138, R164, R171) qui ne possèdent qu'une partie des attributs et qui sont obtenues par navigation dans le treillis (parcours de généralisation). La partie gauche représente la convention qui implémente ces recommandations avec l'ensemble de ces attributs.

Par le module de visualisation, nous avons cherché les attributs (descripteurs sémantiques)

de l'objet C164 afin de les afficher avec le graphe résultat. Ceci permet à un utilisateur d'étendre les résultats en recherchant les conventions qui parlent de sujets similaires à C164 (qui sont annotés avec une partie de ses d'attributs seulement). La fonction de recherche par l'exemple, décrite dans le chapitre 6, est destinée à ce type d'usage. Nous pouvons à ce stade (sans l'interface de navigation) proposer directement à l'utilisateur des objets similaires en effectuant une recherche simple (requête simple avec les attributs de C164) sur le treillis initial des conventions.

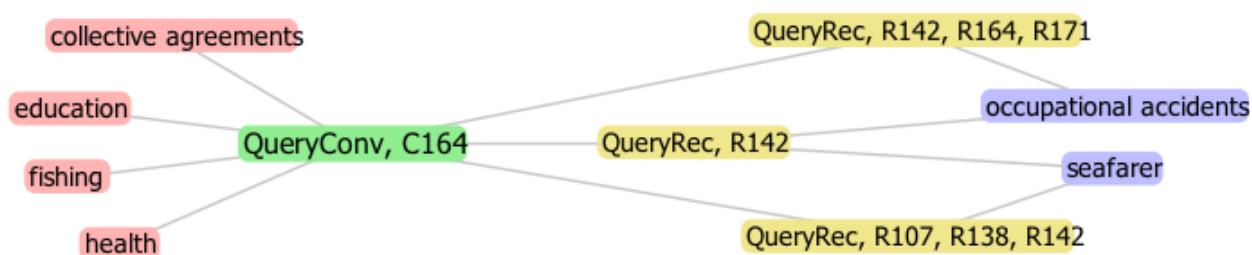


FIGURE 8.2 – Graphes réponses exactes et approchées de la requête OIT1-2.

OIT1-1 : Cette requête contient un objet identifié (R113) donc un seul objet virtuel est créé (QueryConv). La requête décrit la recommandation avec un identifiant et aussi un ensemble d'attributs. L'identifiant est utilisé pour formuler la requête pour l'interrogation : "(QueryConv), impl, (R113)". L'algorithme d'interrogation retourne l'objet C144. Le graphe réponse de cette requête est donné par la figure 8.3. Relâcher la requête pour n'utiliser que les attributs permet de retourner des réponses approchées : "(QueryConv), impl, (R113Query de travail, marin)".

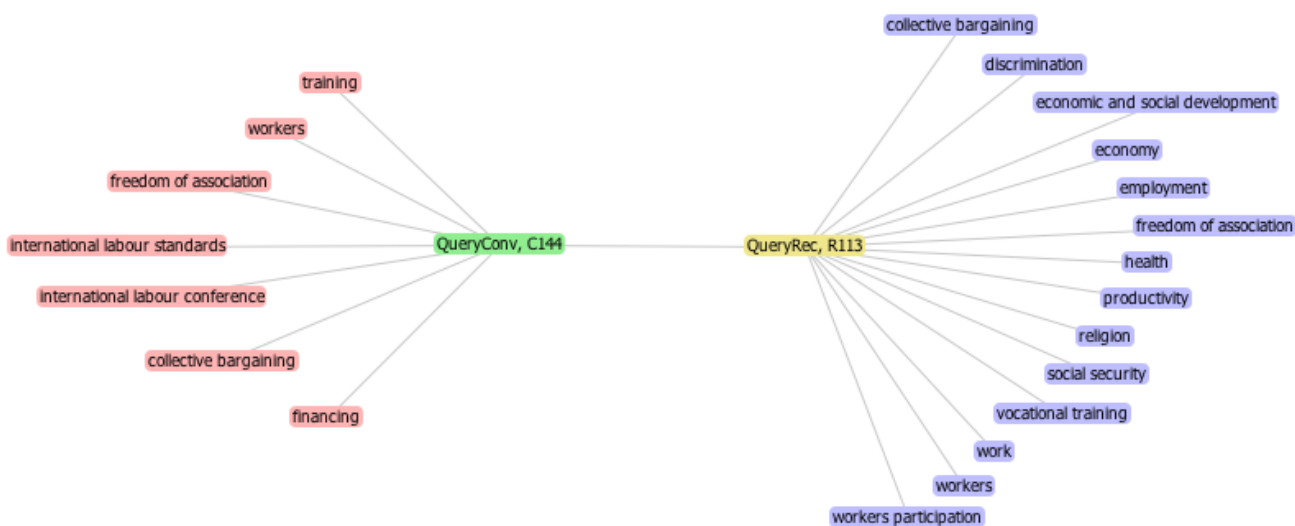


FIGURE 8.3 – Graphe réponse de la requête OIT1-1.

8.2.4 Approche sémantique : première ontologie

Dans un premier temps l'ontologie a été peuplée avec les documents du corpus (individus de la classe Convention et de la classe Recommandation) ayant la relation hasConcept avec les

descripteurs sémantiques de contenu du domaine du travail (individus de la classe `WorkConcept` de type `skos:Concept`) et reliés entre eux par la relation `implement` ayant comme source des objets de la classe `Convention` et comme cible des objets de la classe `Recommandation`. Cette opération est faite automatiquement en transformant les documents (leurs identifiants), leurs descripteurs et leurs relations en triplets en se basant sur les classes de l'ontologie.

Nous avons interrogé la base de connaissances construite avec les requêtes OIT1-1 à OIT1-7 pour lesquelles nous avons obtenu à chaque fois les réponses pertinentes, ainsi qu'avec les requêtes de OIT2-1 à OIT2-7. L'interrogation est faite avec SPARQL qui est plus expressif que le langage défini dans le chapitre 5 et nous permet d'exprimer des requêtes plus complexes. La traduction en SPARQL des requêtes est faite à la main mais en suivant quelques règles que nous explicitons sur quelques requêtes types.

OIT1-1 : *Quelle convention implémente la Recommandation 113 sur la consultation aux échelons industriel et national ?*

Cette requête cherche les objets `Convention` en relation `implement` avec l'objet identifié `Recommandation R113`. Le présupposé de l'unicité dans la requête en langage naturel n'est pas pris en compte dans la traduction SPARQL, s'il existe plus d'une réponse, elles sont toutes retournées. Comme dans l'approche conceptuelle, les descripteurs de contenu `échelon industriel` et `échelon national` ne sont pas utilisés pour la recherche puisqu'il suffit d'utiliser l'identifiant de l'objet. Si la requête ne retourne pas de résultats, ils peuvent servir à lancer une nouvelle requête qui cherche de manière plus générale les recommandations qui parlent de ce sujet. La requête OIT1-7 se traduit de la même manière, l'objet identifié est la convention `C139`.

```

1      SELECT ?conv
2      WHERE {
3          ?conv rdf:type ilo:Convention .
4          ?conv ilo:implement ilo:R113 .
5      }
```

OIT1-2 : *Quelle convention implémente la recommandation qui parle des accidents de travail des marins ?*

Cette requête cherche les objets `Convention` en relation `implement` avec des objets `Recommandation` décrits par les concepts `accidents de travail` et `marins`. Nous avons eu comme réponse exacte la recommandation `R142` qui possède les deux attributs. S'il n'y a pas de réponse à cette requête, une réponse approchée peut être intéressante à retourner à l'utilisateur. L'approche sémantique ne retourne pas de telles réponses approchées contrairement à l'approche conceptuelle qui propose des alternatives à la réponse exacte (en se basant sur la structure du treillis) soit pour enrichir l'ensemble des résultats soit pour éviter de retourner un ensemble vide (dans le cas où il n'existe pas de réponse exacte). La requête OIT1-6 se traduit avec la même structure.

```

1      SELECT ?conv ?recom
2      WHERE {
3          ?recom ilo:hasConcept ilo:occupationalaccidents , ilo:seafarer .
4          ?conv ilo:implement ?recom .
5      }
```

OIT1-5 : *Quelles sont les recommandations implémentées par des conventions qui parlent de la convention collective et de la négociation collective ?*

Cette requête (OIT1-3 et OIT1-4 ont la même structure) cherche les objets `Recommandation` en relation `implement` avec les objets `Convention` décrits par les concepts `convention`

collective et négociation collective. La réponse, décrite dans le tableau ci-dessous, est donnée par les graphes dont les noeuds sont (R107,C147), (R108,C147), (R137,C147), (R158,C154) et les arcs représentent la relation implémente-par. Sa structure est très proche de la requête précédente, seule la cible change de convention à recommandation. Lorsque la cible est une recommandation, nous ne sommes pas obligée de créer la relation inverse implémente-par, changer l'ordre des variables (devant la clause SELECT de la requête) fait l'affaire sans toucher aux schémas de graphes (dans la clause WHERE). Avec l'approche conceptuelle, ces requêtes ont nécessité un traitement différent des autres puisque, en l'absence de contraintes sur les recommandations (ni d'attributs ni de relations dont ils sont le domaine), elles étaient d'abord traitées comme des requêtes simples sur les conventions ensuite complétées à partir des contextes relationnels.

```

1      SELECT ?recom ?conv
2      WHERE {
3          ?conv ilo:hasConcept ilo:collectiveagreements, ilo:collectivebargaining .
4          ?conv ilo:implement ?recom .
5      }

```

Recommandation	Convention
R107, R108, R137	C147
R158	C154

Considérons maintenant les requêtes OIT2-1 à OIT2-7 décrites dans le chapitre 5. Nous avons formulé ces requêtes pour compléter les types du premier ensemble définis par l'expert du domaine. Ces requêtes sont génériques et ne disposent pas d'un ensemble de réponses pertinentes défini *a priori* comme dans le cas du premier ensemble. Dans la suite, nous proposons de traiter ces requêtes en décrivant leur traduction en SPARQL et en donnant un sous-ensemble des réponses retournées (seulement quelques requêtes représentatives seront décrites).

OIT2-1 : *Quelles sont les recommandations qui sont implémentées ?*

Cette requête cherche tous les objets **Recommandation** qui sont le co-domaine de la relation **implement**. Plusieurs réponses sont retournées (39), un extrait est donné dans le tableau ci-dessous.

```

1      SELECT ?recom
2      WHERE {
3          ?conv ilo:implement ?recom .
4      }

```

Recommandation
R083, R100, R105

OIT2-3 : *Quels sont les couples de conventions et de recommandations (en relation d'implémentation) qui parlent de sujets différents ?*

Dans la requête, l'expression « sujets différents » peut être comprise de deux façons : aucun descripteur de contenu en commun ou au moins un descripteur différent. La première traduction n'a pas de solution : toutes les recommandations possèdent au moins un descripteur en commun avec les conventions qui les implémentent. La deuxième traduction retourne 39 réponses dont un extrait est décrit dans le tableau ci-dessous.

```

1      SELECT ?recom ?conv
2      WHERE {
3          ?conv ilo:implement ?recom .
4          ?conv ilo:hasConcept ?concept .
5          MINUS {
6              SELECT ?recom WHERE {
7                  ?recom ilo:hasConcept ?concept .

```



```

8      }
9      }
10
11     SELECT DISTINCT ?recom ?conv
12     WHERE {
13         ?conv ilo:implement ?recom .
14         ?recom ilo:hasConcept ?concept1 .
15         ?conv ilo:hasConcept ?concept2 .
16         FILTER (?concept1 != ?concept2 ).
17     }

```

Recommandation	Convention
R083	C122
R105	C164

OIT2-5 : *Quelles sont les conventions qui implémentent la même recommandation et la recommandation qu'elles implémentent ?*

La requête cherche, pour un objet **Recommandation**, les objets **Convention** qui l'implémentent. Le mot clé **ORDER BY** permet de regrouper les résultats par recommandation. En tout, 8 réponses sont retournées, un exemple est donné dans le tableau suivant.

```

1     SELECT ?conv ?recom
2     WHERE {
3         ?conv ilo:implement ?recom .
4     }
5     ORDER BY (?recom)

```

Recommandation	Convention
R112	(C148,C161)
R111	(C122,C156)

OIT2-6 : *Quelles sont les recommandations qui sont implémentées de deux manières différentes (c'est-à-dire par au moins deux conventions différentes) ?* Cette requête retrouve l'ensemble des recommandations retournées dans la requête précédente (8 objets), un exemple de réponse est donné dans le tableau suivant.

```

1     SELECT DISTINCT ?recom
2     WHERE {
3         ?conv1 ilo:implement ?recom .
4         ?conv2 ilo:implement ?recom .
5         FILTER ( ?conv1 != ?conv2 )
6     }

```

Recommandation
R120, R144, R147

OIT2-7 : *Existe-t-il des conventions qui implémentent deux recommandations différentes ?* Dans cette requête, l'utilisation de « Existe-t-il » indique que la réponse attendue est booléenne (vrai ou faux). Nous utilisons **ASK** à la place de **SELECT** qui permet de vérifier l'existence de tels triplets dans la base. La réponse retournée pour cette requête est **TRUE**.

```

1     ASK
2     {
3         ?conv ilo:implement ?recom1 .
4         ?conv ilo:implement ?recom2 .
5         FILTER ( ?recom1 != ?recom2 )
6     }

```

Pour le premier ensemble de requêtes (réelles, exprimées par des experts) la traduction était plus évidente que pour le deuxième (créé à des fins de test). De plus, nous avons noté pour ce premier ensemble que certaines structures de requêtes sont récurrentes. Nous proposons, comme perspective, de définir des patrons qui peuvent être utilisés pour automatiser le processus de traduction dans un système d'accès juridique.

8.2.5 Discussion

Les deux approches permettent de retrouver toutes les réponses pertinentes aux requêtes. Même si la première approche n'est pas facile à mettre en œuvre (nous n'avons pas une stratégie de recherche unique pour tous les types de requêtes), elle offre des possibilités intéressantes de navigation.

En plus des réponses exactes, l'approche conceptuelle permet de retourner des réponses approchées en explorant les treillis par généralisation ou par spécialisation ou de retourner les contextes des documents retrouvés par calcul de voisinage. Ceci n'est pas possible avec l'approche sémantique sans la formulation d'une ou plusieurs nouvelles requêtes correspondant à différentes contraintes (sur les attributs ou sur les relations). Ceci suppose que l'utilisateur possède une bonne connaissance de la base et a un coût supplémentaire en temps de calcul.

Nous considérons qu'une technique de recherche qui combine ces deux approches aura de meilleures performances (en qualité de résultats, en temps de calcul ou passage à l'échelle). Nous proposons, comme perspective, une technique de recherche qui enchaîne l'approche sémantique et l'approche conceptuelle. Nous proposons d'organiser les résultats retournés, dans un premier temps par l'approche sémantique, dans une structure conceptuelle que nous pouvons utiliser à des fins de navigation ou de visualisation (des contextes formels et relationnels sont construits à partir de l'ensemble des résultats). Par exemple, une telle technique est utile dans le cas où beaucoup de réponses sont retournées. Les organiser dans une structure de treillis facilite leur analyse et aide à repérer les éventuelles interactions qui peuvent exister entre eux.

8.3 Corpus LÉGILOCAL

8.3.1 Description du corpus

La collection sur laquelle nous avons travaillé dans le cadre du projet Légilocal contient 20 documents de 4 types différents et 29 articles, les documents peuvent être composés de plusieurs articles et possèdent plusieurs types de relations entre eux. Les documents sont collectés à partir de plusieurs sources : il s'agit de décisions publiées par des collectivités locales, de décisions de jurisprudence et de textes législatifs (lois, décrets, etc) issus de portails juridiques, principalement Legifrance¹¹⁵. Les documents et leurs relations sont décrits dans le tableau 8.3.

Les actes locaux représentent des actes des communes du comité du public et des actes cités dans les décisions de jurisprudence, dont on n'a pas le texte complet, mais seulement des extraits inclus dans le texte de la décision. Les décisions de jurisprudence sont récupérées sur Legifrance ou bien citées dans les autres décisions (et pas accessibles sur Legifrance) car elles correspondent à des étapes précédentes de la procédure. Même si on ne dispose pas de ces documents, il nous paraît pertinent de les représenter en tant qu'instances dans l'ontologie. Leur contenu est partiellement décrit dans les décisions qui les citent.

Vu que les documents ne sont pas annotés avec un standard juridique pour extraire leurs structures et leur contenu sémantique et identifier les liens de références et de citation qui existent entre eux, pour réaliser ces expérimentations, l'instantiation des ontologies avec ce corpus est faite à la main. L'effort a été réduit par la définition au moment de la conception des deux ontologies de propriétés inverses, de sous-types de propriétés et de restrictions de type `subClassOf` et `equivalentClass` et ensuite par l'exécution d'un moteur d'inférence qui permet de générer de nouveaux triplets et de les ajouter à la base.

115. Corpus et requêtes construits par Sylvie Salotti après discussions avec Eve Paul (juriste).

TABLE 8.3 – Description de la collection LÉGILOCAL : les documents, leurs types et leurs relations.

Actes Locaux	<p>Arrêté 97-17 de Champigné. Arrêté N°2007-031 de Villecresnes . Arrêté N°2011-22 de Villecresnes. Arrêté N°2012-17 de Villecresnes. Arrêté N°2012-48 de Villecresnes. Arrêté du 4 juillet 1997 du maire d’Ance (annulé par la Cour Administrative d’Appel de Bordeaux le 28/05/02). Arrêté du 24 mai 1994 du maire de Magny-le-Feule (confirmé par le Conseil d’État le 29/12/97).</p>
Législation - Codes	<p>Code de l’environnement : Articles L.362-1 à L362-8 et Articles R. 362-1 à R 362-7. Code général des collectivités territoriales : Article 2122-28, Article 2211-1, Articles 2212-1 à 2212-5, Articles 2213-1 à 2213-6-1. Code de la route. Code de la voirie routière. Code des communes : Article 131-1 (ancien texte abrogé remplacé par l’article 2212-1 du Code général des collectivités territoriales, cité par la décision du Conseil d’État du 29/12/97).</p>
Législation - Textes non codifiés	<p>Loi n° 91-2 du 3 janvier 1991 relative à la circulation des véhicules terrestres dans les espaces naturels et portant modification du Code des communes. Décret n° 92-258 du 20 mars 1992 portant modification du Code de la route et application de la loi n° 91-2 du 3 janvier 1991. Circulaire OLIN du 6 septembre 2005. Ordonnance n° 2000-914 du 18 septembre 2000 relative à la partie législative du Code de l’environnement.</p>
Jurisprudence	<p>Décision N° 99BX00597 de la Cour Administrative d’Appel de Bordeaux du 28/05/2002. Décision N°173042 du Conseil d’État en date du 29/12/1997. Jugement du tribunal administratif de Pau du 19 janvier 1999. Jugement du tribunal administratif de Caen du 5 juillet 1995.</p>

8.3.2 Requêtes LÉGILOCAL et réponses pertinentes

Reprenons l'ensemble des requêtes formulées sur le corpus LÉGILOCAL décrites dans le chapitre 5 (section 5.4). Ces requêtes étant génériques, nous les avons projetées sur la collection décrite dans la section précédente et nous avons sélectionné un sous-ensemble permettant de tester les différents aspects de l'approche sémantique. Le tableau 8.4 décrit les nouvelles requêtes avec pour chaque requête la réponse ou l'ensemble de réponses attendues.

TABLE 8.4 – Requêtes LÉGILOCAL avec réponses pertinentes.

Requête	Réponse
L1 Quelles sont les décisions de jurisprudence qui appliquent l'article L. 2213-4 du Code Général des Collectivités Territoriales ?	L'arrêt du 28/05/2002 de la Cour Administrative d'Appel de Bordeaux.
L2 Quels sont les textes d'application de la loi 91-2 du 3 janvier 1991 ?	Le décret 92-258 du 20 mars 1992.
L3 Quelle est la décision qui fait l'objet de l'arrêt N° 99BX00597 de la Cour Administrative d'Appel de Bordeaux du 28/05/2002 ?	Le jugement du Tribunal Administratif de Pau du 19/01/1999.
L4 Je cherche des arrêtés municipaux concernant la réglementation de la circulation sur les chemins ruraux qui ont été confirmés par une décision de justice.	L'arrêt de Magny-le-Feule du 24/05/1994 confirmé par la décision du Conseil d'État du 29/12/1997.
L5 Quels sont les textes législatifs sur lesquels s'appuient les décisions de jurisprudence qui ont annulé des arrêtés municipaux parlant de chemins ruraux ?	L'article L. 2213-4 du code général des collectivités territoriales.
L6 Je voudrais des arrêtés municipaux qui parlent de réglementation de la circulation sur les chemins ruraux ou les chemins forestiers avec tous les textes visés.	Ensemble de graphes <arrêtés, visasLegislation, textes législatifs visés>.
L7 Je voudrais savoir quel texte a codifié l'article L362-1 du code de l'environnement.	Ordonnance n° 2000-914 du 18 septembre 2000 relative à la partie législative du code de l'environnement.
L8 Je voudrais la dernière version (ou la version en vigueur) de l'article L362-1 du code de l'environnement.	Article L. 362-1 du code de l'environnement en vigueur au 1 ^{er} juillet 2013.
L9 Je voudrais savoir si les textes visés par l'arrêt 97-17 de Champigné ont été modifiés, et si oui, quelles sont les nouvelles versions de ces textes ainsi que les textes source de cette modification.	Article L2213-1 du Code Général des Collectivités Territoriales version du 29-01-14, Article L2213-4 du Code Général des Collectivités Territoriales version du 01-01-97.

8.3.3 Exécution sur la première ontologie documentaire

Le graphe de la figure 8.4 montre un extrait du graphe RDF de la collection modélisée avec la première ontologie. Toutes les requêtes formulées sur cette collection ont une cible unaire. Les réponses à ces requêtes sont des listes de documents. Ce type de réponse peut être suffisant dans

TABLE 8.5 – Vocabulaire utilisé pour la formation de la collection LÉGILOCAL et des requêtes associées

Types	Descriptifs
<code>CourtDecision</code>	décision de jurisprudence
<code>CourtOrder</code>	arrêt
<code>CodifiedArticle</code>	article de code
<code>LocalDecree</code>	arrêté municipal
<code>Legislation</code>	texte législatif
Relations	Descriptifs
<code>appliesLegislation</code>	une décision applique un texte législatif
<code>isSubjectOfDecision</code>	une décision qui fait l'objet d'un arrêt
<code>confirmed-by</code>	un arrêté municipal confirmé par une décision
<code>cancel</code>	une décision annule un arrêté municipal
<code>legVisasBy</code>	un texte législatif visé par un arrêté municipal
<code>isCodifiedBy</code>	un texte législatif qui codifie un article
<code>dateInForce</code>	une date d'entrée en vigueur d'un texte
Descripteurs	Equivalents terminologiques
<code>ReglementationCirculation</code>	« réglementation de la circulation »
<code>CheminRural</code>	« chemin rural »
<code>CheminForestier</code>	« chemin forestier »
<code>InterdictionCirculer</code>	« interdiction de circuler »

le cas où la requête contient un objet identifié. Dans le cas où tous les objets dans la requête ne sont pas identifiés, une réponse intéressante consiste à retourner les triplets quiinstancient le graphe de la requête. Pour faire ainsi, nous avons traduit ces requêtes avec au moins deux cibles et le résultat retourné est sous forme de graphe.

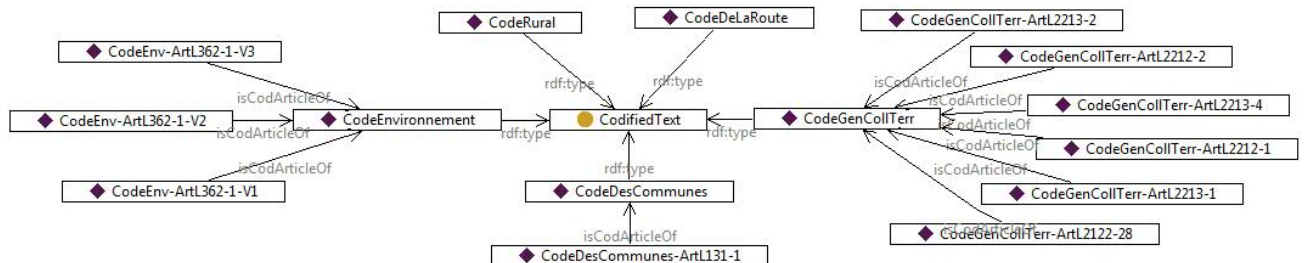


FIGURE 8.4 – Graphe RDF sur la première ontologie : instances de la classe `CodifiedText`.

L1 : *Quelles sont les décisions de jurisprudence qui appliquent l'article L. 2213-4 du Code Général des Collectivités Territoriales ?*

L'exécution de cette requête directement sur la base initiale créée au moment de l'instantiation ne retourne pas de résultats. Ceci est dû au fait que la requête porte sur les décisions de jurisprudence (objets de type `CourtDecision`) qui appliquent l'article L.2213-4 alors qu'au moment de l'instantiation, le seul document créé en relation d'application avec cet article est l'arrêt du 28/05/2002 de la Cour Administrative d'Appel de Bordeaux (de type `CourtOrder`). Pour traiter cette requête, il faut vérifier l'exécution d'une règle d'inférence d'héritage afin de reconnaître les objets de classe « Arrêt » (`CourtOrder`), sous-classe de la classe « Décision de jurisprudence » (`CourtDecision`), comme étant aussi des objets de cette dernière.

```

1      SELECT ?decision
2      WHERE {
3          ?decision rdf:type :CourtDecision .
4          ?decision :appliesLegislation :CodeGenCollTerr-ArtL2213-4
5      }

```

La requête possède une réponse donnée par :

decision
ArretCAA-Bordeaux-28-05-02 : l'arrêt du 28/05/2002 de la Cour Administrative d'Appel de Bordeaux

L2 : *Quels sont les textes d'application de la loi 91-2 du 3 janvier 1991 ?*

Requête avec une cible (dont on ne précise pas le type) et un objet identifié.

```

1      SELECT ?text
2      WHERE {
3          ?text :appliesLegislation :Loi91-2du3janvier1991 .
4      }

```

La réponse à cette requête est :

text
Decret92-258du20mars1992 : le décret 92-258 du 20 mars 1992

L3 : *Quelle est la décision qui fait l'objet de l'arrêt N° 99BX00597 de la Cour Administrative d'Appel de Bordeaux du 28/05/2002 ?*

Requête avec une cible et un objet identifié. Le type de la cible est donné : décision (de jurisprudence).

```

1      SELECT ?decision
2      WHERE {
3          ?decision rdf:type :CourtDecision .
4          ?decision :isSubjectOfDecision :ArretCAA-Bordeaux-28-05-02
5      }

```

La réponse à cette requête est :

decision
JugementTA-Pau-19-01-1999 : le jugement du Tribunal Administratif de Pau du 19/01/1999

L4 : *Je cherche des arrêtés municipaux concernant la réglementation de la circulation sur les chemins ruraux qui ont été confirmés par une décision de justice.*

Cette requête ne contient pas d'objet identifié, elle contient deux objets inconnus : elle est traduite avec deux cibles de types arrêté municipal et décision de jurisprudence.

```

1      SELECT ?decree ?decision
2      WHERE {
3          ?decree rdf:type :LocalDecree .
4          ?decree :hasConcept :ReglementationCirculation , :CheminRural .
5          ?decree :confirmed_by ?decision .
6          ?decision rdf:type :CourtDecision .
7      }

```

La réponse à cette requête est formée par les deux graphes décrits dans le tableau suivant : l'arrêté de Magny le Feule confirmé par deux décisions :

decree		decision
ArreteMagnyLeFeule94	confirmé par	DecisionCE-29-12-1997 JugementTA-Caen-5-07-1995

L5 : *Quels sont les textes législatifs sur lesquels s'appuient les décisions de jurisprudence qui ont annulé des arrêtés municipaux parlant d'interdiction de circuler ?*

Cette requête ne contient pas d'objet identifié, elle contient trois objets inconnus : elle est traduite avec trois cibles de types arrêté municipal, décision de jurisprudence et texte législatif.

```

1      SELECT ?text ?decision ?decree
2      WHERE {
3          ?decree rdf:type :LocalDecree .
4          ?decree :hasConcept :InterdictionCirculer .
5          ?decision rdf:type :CourtDecision .
6          ?decision :cancel ?decree .
7          ?text rdf:type :Legislation .
8          ?decision :appliesLegislation ?text .
9      }

```

La réponse à cette requête est donnée par un graphe composé de trois noeuds :

text		decision		decree
CodeGenCollTerr-ArtL2213-4	appliqué par	ArretCAA-Bordeaux-28-05-02	annule	Arrete-Ance97

L6 : *Je voudrais des arrêtés municipaux qui parlent de réglementation de la circulation sur les chemins ruraux ou les chemins forestiers avec tous les textes visés.*

```

1      SELECT ?decree ?text
2      WHERE {
3          ?decree rdf:type :LocalDecree .
4          {?decree :hasConcept :ReglementationCirculation , :CheminForestier .}
5          UNION {?decree :hasConcept :ReglementationCirculation , :CheminRural}
6          ?text :legVisasBy ?decree .
7      }

```

La réponse à cette requête est formée par les dix graphes suivants :

decree		text
ArreteChampigne97-17	visas Legislation	CodeGenCollTerr-ArtL2213-4 CodeGenCollTerr-ArtL2213-1 Loi91-2du3janvier1991 Decret92-258du20mars1992
ArreteMagnyLeFeule94	visas Legislation	CodeRural-ArtL161-5 CodeDesCommunes-ArtL131-1
ArreteVillegresnes2011-22	visas Legislation	CodeGenCollTerr-ArtL2212-2 CodeGenCollTerr-ArtL2212-1
ArreteVillegresnes2012-17	visas Legislation	CodeGenCollTerr-ArtL2122-28 CodeGenCollTerr-ArtL2213-2

L7 : *Je voudrais savoir quel texte a codifié l'article L362-1 du code de l'environnement.*

La relation `codified` permet de relier un texte source de la codification et le texte original qui doit être codifié. Or dans cette requête, nous ne disposons pas d'information sur ce dernier mais plutôt sur le nouveau texte issu de la codification (article L362-1 du code de l'environnement). Le lien qui peut exister entre le texte source de la codification et le texte résultat, est la relation `isCodifiedBy`. Nous avons utilisé cette relation pour traduire la requête qui a fourni une réponse décrite dans le tableau ci-dessous. Dans le cas où cette relation n'est pas créée dans la base, il n'est pas possible de répondre à cette requête avec la première modélisation. En revanche, avec la deuxième ontologie, ceci est possible grâce à la modélisation des relations comme des entités reliant tous les documents intervenant à une opération de codification (nous n'avons pas besoin de coder toutes les relations pour pouvoir retrouver les documents liés).

```

1      SELECT ?text
2      WHERE {
3          ?article rdf:type :CodifiedArticle .
4          ?article dct:title "Code de l'environnement - Article L. 362-1" .
5          ?article :isCodifiedBy ?text .
6      }

```

La réponse à cette requête est donnée par :

text
Ordonnance2000-914du18septembre2000

L8 : *Je voudrais la dernière version (ou la version en vigueur) de l'article L362-1 du code de l'environnement.*

La requête recherche la dernière version de l'article L.362-1 du code de l'environnement. Cet article possède plusieurs versions, elles ont toutes le même titre : nous utilisons cette information pour chercher les différentes versions. Nous cherchons pour chaque version sa date d'entrée en vigueur et le résultat est donné par celle qui a la date la plus récente (`ORDER By` pour ordonner les versions par date, puis `LIMIT 1` pour ne prendre que la plus récente).


```

1      SELECT ?article
2      WHERE {
3          ?article rdf:type :Article .
4          ?article dct:title "Code de l'environnement - Article L. 362-1" .
5          ?article :dateInForce ?date .
6      } ORDER BY DESC(?date) LIMIT 1

```

La réponse à cette requête est donnée par la version du 1^{er} juillet 2013

text
article_L362-1_V3

8.3.4 Exécution sur la deuxième ontologie documentaire

Nous avons créé les instances correspondant aux données du corpus LÉGILOCAL sur la deuxième ontologie et nous avons exécuté le même ensemble de requêtes décrites dans la section précédente. Toutes les requêtes que nous avons pu exécuter sur la première ontologie retournent le même ensemble de résultats sur la deuxième, seule leur traduction en SPARQL est donnée dans la suite. Les requêtes plus complexes, qui portent essentiellement sur l'historique des documents (versions), sont décrites avec leurs résultats.

L1 : *Quelles sont les décisions de jurisprudence qui appliquent l'article L. 2213-4 du Code Général des Collectivités Territoriales ?*

La relation `applique` est modélisée comme sous classe de `Citation`. Les propriétés `citationSource` et `citationTarget` permettent de relier le domaine et le co-domaine de la relation.

```

1      SELECT ?decision
2      WHERE {
3          ?decision rdf:type lido:CourtDecision .
4          ?application rdf:type lido:Application .
5          ?application lido:citationSource ?decision .
6          ?application lido:citationTarget :CGCT-ArtL2213-4
7      }

```

L2 : *Quels sont les textes d'application de la loi 91-2 du 3 janvier 1991 ?*

```

1      SELECT ?text
2      WHERE {
3          ?application rdf:type lido:Application .
4          ?application lido:citationSource ?text .
5          ?application lido:citationTarget :loi_91-2 .
6      }

```

L3 : *Quelle est la décision qui fait l'objet de l'arrêt N° 99BX00597 de la Cour Administrative d'Appel de Bordeaux du 28/05/2002 ?*

Pour identifier les documents qui sont mis en relation, nous cherchons l'opération documentaire `Decision` qui a comme source le document identifié (arrêt N° 99BX00597) et comme cible la décision en question.

```

1      SELECT ?decision
2      WHERE {
3          ?decision rdf:type lido:CourtDecision .
4          ?decisionref rdf:type :Decision .
5          ?decisionref lido:referenceSource :ArretCAA-Bordeaux-28-05-2002 .
6          ?decisionref lido:referenceTarget ?decision .
7      }

```

L4 : *Je cherche des arrêtés municipaux concernant la réglementation de la circulation sur les chemins ruraux qui ont été confirmés par une décision de justice.*

```

1      SELECT ?decree ?decision
2      WHERE {
3      ?decree rdf:type lido:LocalDecree .
4      ?decree lido:hasConcept :reglementation_circulation, :cheminRural .
5      ?decision rdf:type lido:CourtDecision .
6      ?confirmationref rdf:type :Confirmation .
7      ?confirmationref lido:referenceSource ?decision .
8      ?confirmationref lido:referenceTarget ?decree .
9      }

```

L5 : *Quels sont les textes législatifs sur lesquels s'appuient les décisions de jurisprudence qui ont annulé des arrêtés municipaux parlant d'interdiction de circuler ?*

La relation « s'appuie sur » correspond à la relation applique modélisée par la classe `Application` sous classe de `Citation` (relation binaire). La relation « annule » est modélisée par une opération documentaire `Annulation` sous-classe de `Decision` (relation ternaire).

```

1      SELECT ?text ?decision
2      WHERE {
3      ?decision rdf:type lido:CaseLaw .
4      ?decree rdf:type lido:LocalDecree .
5      ?application rdf:type lido:Application .
6      ?annulation rdf:type :Annulation .
7      ?application lido:citationSource ?decision .
8      ?application lido:citationTarget ?text .
9      ?annulation lido:referenceSource ?decision .
10     ?annulation lido:referenceTarget ?decree .
11     ?decree lido:hasConcept :interdiction_de_circuler .
12     }

```

L6 : *Je voudrais des arrêtés municipaux qui parlent de réglementation de la circulation sur les chemins ruraux ou les chemins forestiers avec tous les textes visés.*

La relation « visa » est modélisée avec la classe `VisaCitation` sous-classe de `Citation` (relation binaire).

```

1      SELECT ?decree ?text
2      WHERE {
3      ?decree rdf:type lido:LocalDecree .
4      {?decree lido:hasConcept :reglementation_circulation , :cheminForestier .}
5      UNION {?decree lido:hasConcept :reglementation_circulation , :cheminRural}
6      ?visa rdf:type lido:VisaCitation .
7      ?visa lido:citationSource ?decree .
8      ?visa lido:citationTarget ?text .
9      }

```

L7 : *Je voudrais savoir quel texte a codifié l'article L362-1 du code de l'environnement.*

La relation « codifie » est modélisée comme une opération documentaire avec la classe `Codification` (relation ternaire).

```

1      SELECT ?text
2      WHERE {
3      ?codification rdf:type lido:Codification .
4      ?codification lido:referenceSource ?text .
5      ?codification metalex:result :article_L362-1 .
6      }

```

L8 : *Je voudrais la dernière version (ou la version en vigueur) de l'article L362-1 du code de l'environnement.*

Dans la première modélisation, nous étions obligée de passer par le titre de l'article, une propriété commune à toutes les versions. Dans cette modélisation, la gestion des versions avec œuvre et expression facilite la recherche.

```

1      SELECT ?version
2      WHERE {
3        ?version rdf:type lido:DocumentaryUnitExpression .
4        ?version metalex:realizes :article_L362-1 .
5        ?version lido:dateInForce ?date .
6      } ORDER BY DESC(?date) LIMIT 1

```

L9 : *Je voudrais savoir si les textes visés par l'arrêté 97-17 de Champigné ont été modifiés, et si oui, quelles sont les nouvelles versions de ces textes ainsi que les textes source de cette modification.*

Dans cette requête, l'utilisateur formule une demande complexe qui combine la gestion des versions (version précédente, version suivante) et des relations qui font intervenir plus de deux documents (document source de modification, document cible et document résultat). Trouver cette information avec la première ontologie n'est pas facilement réalisable, la deuxième ontologie permet d'explicitier ces contraintes.

```

1      SELECT ?text ?newversion ?oldversion ?source
2      WHERE {
3        ?visa rdf:type lido:VisaCitation .
4        ?visa lido:citationSource :ArreteChampigne97-17 .
5        ?visa lido:citationTarget ?text .
6        ?modification rdf:type lido:Modification .
7        ?modification lido:referenceTarget ?oldversion .
8        ?modification lido:referenceSource ?source .
9        ?modification metalex:result ?newversion .
10     ?text metalex:realizedBy ?oldversion .
11     }

```

La réponse est donnée par les articles L2213-1 et L2213-4 du code général des collectivités territoriales avec leurs anciennes et nouvelles versions et les textes de loi sources de la modification.

text	newversion	oldversion	source
CGCT-ArtL2213-1	CGCT-ArtL2213-1-V29-01-14	CGCT-ArtL2213-1-V24-02-96	Loi-2014-58-Art62
CGCT-ArtL2213-4	CGCT-ArtL2213-4-V01-01-97	CGCT-ArtL2213-4-V24-02-96	Loi96-1236-Art42

Le graphe de la figure 8.5 montre le graphe instance de l'opération documentaire de modification pour l'article L2213-1.

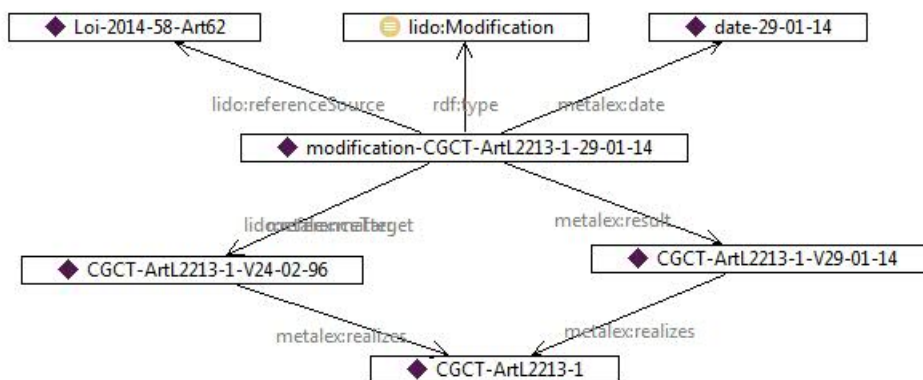


FIGURE 8.5 – Opération documentaire de modification de l'article L2213-1 : l'œuvre, les deux versions qui réalisent l'œuvre et le texte source de modification.

8.3.5 Discussion

Dans la première ontologie les relations sont représentées par des liens directs entre les objets (modélisés avec des propriétés d'objets). Cette modélisation a l'avantage d'être plus facile à instancier et à interroger mais elle est mal adaptée à la modélisation des relations complexes mettant plus de deux documents en jeu ou associant plusieurs versions à un même document. À l'inverse, la deuxième ontologie demande un plus grand effort d'instantiation mais permet de gérer les aspects liés aux chaînage de versions et des relations de référence à plus de deux documents. Elle est plus adaptée à la complexité des liens dans le domaine juridique.

Chapitre 9

Conclusion et perspectives

9.1 Conclusion

Le travail présenté dans cette thèse vise à améliorer le processus de recherche d'information sémantique dans une collection documentaire en proposant l'intégration de la dimension intertextuelle dès le départ dans le processus de recherche. L'analyse des besoins dans le domaine juridique montre, notamment à travers les requêtes des juristes, l'enjeu que représente la prise en compte de l'intertextualité dans ce domaine.

Nous avons proposé deux approches de modélisation et de recherche dans une collection de documents inter-reliés. Ces approches construisent un modèle de collection documentaire qui se base sur le contenu sémantique des documents, leur typologie ainsi que les relations intertextuelles qu'ils entretiennent. Cela permet de répondre à des requêtes relationnelles qui portent à la fois sur le contenu sémantique et sur les liens intertextuels et de retourner en réponse des graphes de documents liés. La première approche utilise l'analyse formelle de concepts et l'analyse relationnelle de concepts pour modéliser la collection de documents par des structures conceptuelles. Nous avons défini des méthodes de recherche de documents par accès direct ou par navigation pour interroger et explorer le modèle relationnel construit puis retourner des documents ou graphes de documents pertinents. La deuxième approche présente une solution plus opérationnelle basée sur les technologies du web sémantique et propose un modèle à base d'ontologies pour modéliser des collections de documents liés. Au-delà de la recherche traditionnelle, ces modèles offrent des fonctionnalités sémantiques et relationnelles de RI.

Dans la première approche, des contextes formels et des contextes relationnels sont créés respectivement en fonction des descripteurs sémantiques et des références entre les documents (les documents doivent être annotés sémantiquement et la structure des documents doit être analysée pour extraire les liens de référence). L'utilisateur formule ensuite une requête qui peut être de deux types : simple ou relationnelle. L'algorithme de recherche traite la requête et renvoie des réponses pertinentes à l'utilisateur, soit une liste de documents pertinents soit des graphes de documents. L'utilisateur a aussi la possibilité d'explorer la structure des treillis par navigation entre les catégories de documents.

Bien que cette approche ne permette pas de traiter tous les types de requêtes identifiés dans l'analyse des besoins ni de travailler sur une collection de grande taille, elle nous a permis de montrer l'intérêt d'une approche intertextuelle pour la RI. Elle a aussi l'avantage de proposer à l'utilisateur des réponses approchées en l'absence de réponses exactes.

Dans la deuxième approche, on peut modéliser plus de propriétés documentaires : la typologie des documents, les liens intertextuels et leurs différents types, la structure d'un document et

son contenu sémantique. Pour combiner toutes ces propriétés dans un seul et unique modèle utilisant les technologies du web sémantique nous avons proposé une ontologie documentaire pour les textes juridiques qui est structurée en trois modules : module document (structure), module collection (types des documents et liens intertextuels) et module sémantique (ressources sémantiques pour les concepts de domaine). La gestion avancée des versions (cycle de vie d'un document) et des opérations documentaires à l'origine des références entre les documents nous a amenée à proposer une deuxième ontologie documentaire qui prend en compte ces deux derniers aspects.

L'adoption d'un modèle de document intégré pour coder la structure des documents, leurs annotations sémantiques et la structure sémantique de la collection permet de traiter des requêtes complexes combinant des critères de recherche structuraux, intertextuels et de contenu. Le choix de la première ou de la deuxième ontologie dépend de l'application et des besoins de recherche. Nous avons pu répondre à toutes les requêtes recensées dans l'analyse des besoins et à d'autres types plus complexes que nous avons élaborés en anticipant sur les futurs besoins des utilisateurs. Par rapport à la première approche, nous avons perdu la possibilité d'avoir, sans calcul supplémentaire, des réponses approchées (avantage lié à la navigation dans la structure des treillis) mais nous avons gagné sur les détails de description des documents (structure, hiérarchie sémantique des attributs) et en échelle.

Les résultats des systèmes de RI juridique existants et les approches relationnelles proposées sont différents. Dans les systèmes de RI existants les documents retournés sont organisés dans une liste sans tenir compte des liens intertextuels qui existent habituellement entre eux. Dans les approches que nous proposons, les réponses sont présentées sous forme de graphes où les nœuds correspondent aux différents types de documents (code, loi, jurisprudence, etc.) et les arcs correspondent aux différents types de liens entre eux (modification, abrogation, etc.).

9.2 Perspectives

Dans un contexte applicatif, nous avons pu montrer par notre travail l'intérêt de traiter l'intertextualité dans un système réel d'accès juridique. Évidemment, beaucoup reste à faire pour aboutir à un système opérationnel :

- L'annotation des documents au regard d'une ressource terminologique et l'extraction de leurs structures (étape que nous avons supposée faite dans notre travail). Les documents étant de différents types, une méthode d'extraction automatique permet de prendre en compte la diversité des documents possédant des structures spécifiques selon leurs types.
- Analyse des besoins en termes d'interfaces pour étudier la manière la plus acceptable pour les utilisateurs pour entrer des requêtes relationnelles basées sur les caractéristiques de la collection (descripteurs sémantiques, types de documents, types de liens).
- Concevoir sur cette base des interfaces utilisateurs conviviales pour la création des requêtes et pour l'affichage des résultats.

À court terme, notre objectif consiste essentiellement à :

- Concevoir des interfaces simples à base de formulaires pour aider les utilisateurs à entrer des requêtes relationnelles. L'utilisation des formulaires a l'avantage de regrouper toutes les caractéristiques sur lesquelles peuvent porter les requêtes ce qui aide les utilisateurs à élargir le champ des requêtes simples habituellement posées.
- Élaborer des interfaces d'affichage de résultats qui sont retournés sous forme de graphes de documents.
- Spécifier davantage le langage de requêtes défini dans ce travail pour trouver un compro-

mis entre le plus expressif (comme SPARQL) et ce qui est effectivement utile pour les utilisateurs et appréhensible par eux.

- Combiner les deux approches proposées pour en tirer le meilleur des deux : taille de collections pour l’approche sémantique, navigation pour l’approche conceptuelle. Une approche combinée sémantique-conceptuelle consisterait :
 1. à modéliser les documents avec l’approche sémantique en créant une ontologie documentaire instanciée avec les données de la collection,
 2. à lancer la recherche sur la base construite,
 3. à extraire des données à partir des triplets récupérés pour construire les contextes formels et relationnels,
 4. à modéliser les résultats avec une famille de treillis relationnels. Les structures relationnelles construites offrent un espace de navigation dans l’ensemble des résultats retournés qui facilite leur exploitation et leur analyse.
- Affiner le modèle ontologique en étudiant plus en profondeur les spécificités des textes juridiques.
- Proposer un modèle de conception d’ontologie (*Content Ontology Design Pattern*) comme solution réutilisable pour la modélisation des références juridiques. Il s’agit de représenter une référence comme une entité pour permettre une description détaillée de la référence elle-même en termes de documents et des agents impliqués mais aussi en termes de multiplicité des types de ces références à laquelle on accorde une très grande importance dans le domaine juridique. Cette représentation permet de réduire les efforts de modélisation et d’instantiation (elle concerne un contenu récurrent dans les textes de loi) ce qui représente un avantage majeur puisque la calculabilité est un problème commun dans la représentation des connaissances juridiques.
- Explorer d’autres domaines d’application dans lesquels une recherche d’information relationnelle pourrait apporter des solutions : le domaine juridique présente un exemple extrême sur lequel nous avons jugé pertinent de tester nos approches, mais il n’est pas certain que des modèles aussi riches soient nécessaires pour d’autres domaines.

Bibliographie

- [DBL, 2009] (2009). *The 12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 8-12, 2009, Barcelona, Spain*. ACM.
- [Abasolo and Gomez, 2000] Abasolo, J. M. and Gomez, M. (2000). Melisa. an ontology-based agent for information retrieval in medicine. In *Proceedings of the First International Workshop on the Semantic Web (SemWeb2000)*, pages 73–82.
- [Abiteboul et al., 1995] Abiteboul, S., Hull, R., and Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley.
- [Agnoloni and Tiscornia, 2010] Agnoloni, T. and Tiscornia, D. (2010). Semantic web standards and ontologies for legislative drafting support. In *Proceedings of the 2nd IFIP WG 8.5 international conference on Electronic participation*, pages 184–196, Berlin, Heidelberg. Springer-Verlag.
- [Alam et al., 2013] Alam, M., Chekol, M. W., Coulet, A., Napoli, A., and Smaïl-Tabbone, M. (2013). Lattice based data access (lbda) : An approach for organizing and accessing linked open data in biology. In *Proceedings of the International Workshop on Data Mining on Linked Data, with Linked Data Mining Challenge collocated with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013), Prague, Czech Republic, September 23*, volume 1082 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Amardeilh et al., 2013] Amardeilh, F., Bourcier, D., Cherfi, H., Dubail, C., Garnier, A., Guillemin-Lanne, S., Mimouni, N., Nazarenko, A., Ève Paul, Salotti, S., Seizou, M., Szulman, S., and Zargayouna, H. (2013). The légilocal project : the local law simply shared. In *Legal Knowledge and Information Systems - JURIX 2013 : The Twenty-Sixth Annual Conference, December 11-13, 2013, University of Bologna, Italy*, pages 11–14.
- [Amardeilh et al., 2005] Amardeilh, F., Laublet, P., and Minel, J.-L. (2005). Document annotation and ontology population from linguistic extractions. In *Proceedings of the 3rd international conference on Knowledge capture (K-CAP '05)*, pages 161–168.
- [Andrews and Fox, 2007] Andrews, N. O. and Fox, E. A. (2007). Recent Developments in Document Clustering. Technical report.
- [Andrews, 2009] Andrews, S. (2009). In-close, a fast algorithm for computing formal concepts. In *the Seventeenth International Conference on Conceptual Structures*.
- [Andrews, 2011] Andrews, S. (2011). In-close2, a high performance formal concept miner. In *Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK*, Lecture Notes in Computer Science, pages 50–62. Springer.

- [Andrews and Orphanides, 2013] Andrews, S. and Orphanides, C. (2013). Discovering knowledge in data using formal concept analysis. *International Journal of Distributed Systems and Technologies (IJ DST)*, 4(2) :31–50.
- [Angles and Gutierrez, 2008] Angles, R. and Gutierrez, C. (2008). The expressive power of sparql. In *The Semantic Web - International Semantic Web Conference - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 114–129. Springer Berlin Heidelberg.
- [Arévalo et al., 2006] Arévalo, G., Falleri, J.-R., Huchard, M., and Nebut, C. (2006). Building abstractions in class models : Formal concept analysis in a model-driven approach. In *Model Driven Engineering Languages and Systems, 9th International Conference, MoDELS 2006, Genova, Italy, October 1-6*, volume 4199 of *Lecture Notes in Computer Science*, pages 513–527. Springer.
- [Ashley, 2013] Ashley, K. D., editor (2013). *Legal Knowledge and Information Systems - JURIX 2013 : The Twenty-Sixth Annual Conference, December 11-13, 2013, University of Bologna, Italy*, volume 259 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.
- [Azmeah et al., 2011a] Azmeah, Z., Driss, M., Hamoui, F., Huchard, M., Moha, N., and Tibermacine, C. (2011a). Selection of composable web services driven by user requirements. *the Application and Experience Track of ICWS 2011 - International Conference on Web Services*, pages 395–402.
- [Azmeah et al., 2011b] Azmeah, Z., Hacène-Rouane, M., Huchard, M., Napoli, A., and Valtchev, P. (2011b). Querying relational concept lattices. In *Proceedings of the 8th International Conference on Concept Lattices and their Applications (CLA'11)*, pages 377–392, Nancy, France.
- [Azouaou, 2006] Azouaou, F. (2006). *Modèles et outils d'annotation pour une mémoire personnelle de l'enseignant*. PhD thesis, Université Joseph Fourier - Grenoble I.
- [Baader, 2009] Baader, F. (2009). Description logics. In *Reasoning Web : Semantic Technologies for Information Systems, 5th International Summer School 2009*, volume 5689 of *Lecture Notes in Computer Science*, pages 1–39. Springer-Verlag.
- [Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA.
- [Baeza Yates and R., 1999] Baeza Yates, R. A. and R., N. B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman, Boston, MA, USA.
- [Barabucci et al., 2011] Barabucci, G., Palmirani, M., Vitali, F., and Cervone, L. (2011). Long-term preservation of legal resources. In *Proceedings of the Second international conference on Electronic government and the information systems perspective, EGOVIS'11*, pages 78–93, Berlin, Heidelberg. Springer-Verlag.
- [Barbut and Monjardet, 1970] Barbut, M. and Monjardet, B. (1970). *Ordre et classification : Algèbre et combinatoire, Tome II*. Hachette, Paris.
- [Baziz, 2004] Baziz, M. (2004). Towards a Semantic Representation of Documents by Ontology-Document Mapping . In Bussler, C. and Fensel, D., editors, *The Eleventh International Conference on Artificial Intelligence(AIMSA 2004) , Varna, Bulgaria, 02/09/04-04/09/04*, pages 33–43, Springer-Verlag Berlin, Heidelberg, Germany. LNCS/LNAI 3192, Springer.
- [Baziz, 2005] Baziz, M. (2005). *Indexation conceptuelle guidée par ontologie pour la recherche d'information*. PhD thesis, Institut de recherche en informatique de Toulouse, Université PaulSabatier.

-
- [Baziz et al., 2005] Baziz, M., Boughanem, M., Aussenac-Gilles, N., and Chrisment, C. (2005). Semantic cores for representing documents in ir. In *Proceedings of the 2005 ACM symposium on Applied computing, SAC '05*, pages 1011–1017, New York, NY, USA. ACM.
- [Berners-Lee, 2006] Berners-Lee, T. (2006). Linked data - design issues. <http://www.w3.org/DesignIssues/LinkedData.html>.
- [Berners-Lee, 2007] Berners-Lee, T. (2007). Giant global graph. <http://dig.csail.mit.edu/breadcrumbs/node/215>.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*.
- [Biagioli et al., 2005] Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., and Soria, C. (2005). Automatic semantics extraction in law documents. In *International Conference on AI and Law (ICAIL)*, pages 133–140.
- [Biasiotti et al., 2008] Biasiotti, M., Francesconi, E., Palmirani, M., Sartor, G., and Vitali, F. (2008). *Legal informatics and management of legislative documents*. Global Centre for ICT in Parliament.
- [Birkhoff, 1967] Birkhoff, G. (1967). Lattice theory. In *Colloquium Publications*, volume 25, pages 172–210. Amer. Math. Soc., 3. edition.
- [Bizer et al., 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3) :1–22.
- [Blomqvist and Groza, 2013] Blomqvist, E. and Groza, T., editors (2013). *Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*, volume 1035 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Boer, 2009] Boer, A. (2009). Metalex naming conventions and the semantic web. In *Proceedings of the 2009 conference on Legal Knowledge and Information Systems : JURIX 2009 : The Twenty-Second Annual Conference*, pages 31–36, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- [Boer et al., 2002] Boer, A., Hoekstra, R., and Winkels, R. (2002). *METALex : Legislation in XML*, pages 1–10. IOS Press.
- [Boer et al., 2007] Boer, A., Winkels, R., and Vitali, F. (2007). Proposed xml standards for law : Metalex and lkif. In *Proceedings of the 2007 conference on Legal Knowledge and Information Systems : JURIX 2007 : The Twentieth Annual Conference*, pages 19–28, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- [Boer et al., 2008] Boer, A., Winkels, R., and Vitali, F. (2008). Metalex xml and the legal knowledge interchange format. In *Computable Models of the Law*, volume 4884 of *Lecture Notes in Computer Science*, pages 21–41. Springer Berlin / Heidelberg.
- [Bolelli et al., 2006] Bolelli, L., Ertekin, S., and Giles, C. L. (2006). Clustering scientific literature using sparse citation graph analysis. In *Proceedings of the 10th European conference on Principle and Practice of Knowledge Discovery in Databases, PKDD'06*, pages 30–41, Berlin, Heidelberg. Springer-Verlag.
- [Bommarito and Katz, 2009] Bommarito, M. J. and Katz, D. M. (2009). Properties of the united states code citation network. *ArXiv e-prints*.
- [Bordat, 1986] Bordat, J.-P. (1986). Calcul pratique du treillis de galois d'une correspondance. *Mathématiques et Sciences Humaines*, 96 :31–47.

- [Borkar et al., 2001] Borkar, V., Deshmukh, K., and Sarawagi, S. (2001). Automatic segmentation of text into structured records. *SIGMOD Rec.*, 30(2) :175–186.
- [Boulet et al., 2009] Boulet, R., Mazzega, P., and Bourcier, D. (2009). Network analysis of the french environmental code. In *AICOL Workshops*, pages 39–53.
- [Boulet et al., 2011] Boulet, R., Mazzega, P., and Bourcier, D. (2011). A network approach to the french system of legal codes- part i : Analysis of a dense network. *Journal of Artificial Intelligence and Law*, 19 :333–355.
- [Bourcier, 2011] Bourcier, D. (2011). Sciences juridiques et complexité. un nouveau modèle d’analyse. *Droit et Cultures*, 61(1) :37–53.
- [Bourcier and Fernández-Barrera, 2012] Bourcier, D. and Fernández-Barrera, M. (2012). Recensement des ressources sémantiques réutilisables pour la modélisation du droit des collectivités locales. Livrable 2.1 - Projet Légilocal.
- [Bourcier and Mazzega, 2007a] Bourcier, D. and Mazzega, P. (2007a). Codification, law article and graphs. In Lodder, A. and (eds.), L. M., editors, *Legal Knowledge and Information Systems, JURIX*, pages 29–38. IOS Press.
- [Bourcier and Mazzega, 2007b] Bourcier, D. and Mazzega, P. (2007b). Toward measures of complexity in legal systems. In *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA*, pages 211–215. ACM.
- [Bouzidi, 2013] Bouzidi, K. R. (2013). *Aide à la création et à l’exploitation de réglementations basée sur les modèles et techniques du Web sémantique*. Phd thesis, École doctorale STIC, Université Nice Sophia Antipolis.
- [Bouzidi et al., 2011] Bouzidi, K. R., Faron-Zucker, C., Fies, B., Corby, O., and Nhan, L.-T. (2011). Modélisation de documents réglementaires dans le domaine du bâtiment. In *Actes 12e Conférence Internationale Francophone sur l’Extraction et la Gestion de Connaissance, EGC 2011*, pages 557–558, Bordeaux, France.
- [Braga et al., 1999] Braga, R., Werner, C., and Mattoso, M. (1999). Odyssey : a reuse environment based on domain models. In *Application-Specific Systems and Software Engineering and Technology, 1999. ASSET ’99. Proceedings. 1999 IEEE Symposium on*, pages 50–57.
- [Braga et al., 2000] Braga, R. M. M., Werner, C. M. L., and Mattoso, M. (2000). Using ontologies for domain information retrieval. In *Proceedings of the 11th International Workshop on Database and Expert Systems Applications, DEXA ’00*, pages 836–840, Washington, DC, USA. IEEE Computer Society.
- [Breuker and Hoekstra, 2004] Breuker, J. and Hoekstra, R. (2004). Epistemology and ontology in core ontologies : Folaw and lri-core, two core ontologies for law. In *In Proceedings of the EKAW04 Workshop on Core Ontologies in Ontology Engineering*, pages 15–27. Northamptonshire, UK.
- [Brighi and Palmirani, 2009] Brighi, R. and Palmirani, M. (2009). Legal text analysis of the modification provisions : a pattern oriented approach. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law, ICAIL ’09*, pages 238–239, New York, NY, USA. ACM.
- [Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30 :107–117.
- [Carpineto et al., 2009] Carpineto, C., Osinski, S., Romano, G., and Weiss, D. (2009). A survey of web clustering engines. *ACM Comput. Surv.*, 41(3) :17 :1–17 :38.

-
- [Carpineto et al., 2006] Carpineto, C., Pietra, A. D., Mizzaro, S., and Romano, G. (2006). Mobile clustering engine. In *European Conference on Information Retrieval (ECIR)*, pages 155–166.
- [Carpineto and Romano, 1993] Carpineto, C. and Romano, G. (1993). Galois : An order-theoretic approach to conceptual clustering. *Proceedings of 10th International Conference on Machine Learning, Amherst*, pages 33–40.
- [Carpineto and Romano, 1996] Carpineto, C. and Romano, G. (1996). A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2) :95–122.
- [Carpineto and Romano, 2000] Carpineto, C. and Romano, G. (2000). Order-theoretical ranking. *Journal of the American Society for Information Science*, 51 :587–601.
- [Carpineto and Romano, 2004] Carpineto, C. and Romano, G. (2004). Exploiting the potential of concept lattices for information retrieval with credo. *Journal of Universal Computer Science*, 10(8) :985–1013.
- [Carpineto and Romano, 2005] Carpineto, C. and Romano, G. (2005). Using concept lattices for text retrieval and mining. In *Formal Concept Analysis*, pages 161–179.
- [Chandler, 2005] Chandler, S. J. (2005). The network structure of supreme court jurisprudence. In *Public Law and Legal Theory Series 2005-W-01 (Technical report)*. University of Houston Law Center.
- [Chein, 1969] Chein, M. (1969). Algorithme de recherche des sous-matrices premières d’une matrice. *Bull. Math. Soc. Sci. Math. R.S. Roumanie*, 13 :21–25.
- [Chekol, 2012] Chekol, M. w. (2012). *Analyse statique de requête pour le Web sémantique*. PhD thesis. Thèse de doctorat dirigée par Euzenat, Jérôme et Layaïda, Nabil Informatique Grenoble 2012.
- [Chekol and Napoli, 2013] Chekol, M. W. and Napoli, A. (2013). An FCA framework for knowledge discovery in SPARQL query answers. In [Blomqvist and Groza, 2013], pages 197–200.
- [Chevallet et al., 2007] Chevallet, J.-P., Lim, J.-H., and Le, D. T. H. (2007). Domain knowledge conceptual inter-media indexing : Application to multilingual multimedia medical reports. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM ’07*, pages 495–504, New York, NY, USA. ACM.
- [Chieze et al., 2010] Chieze, E., Farzindar, A., and Lapalme, G. (2010). An automatic system for summarization and information extraction of legal information. In *Semantic Processing of Legal Texts*, pages 216–234.
- [Cimiano et al., 2004] Cimiano, P., Handschuh, S., and Staab, S. (2004). Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web, WWW ’04*, pages 462–471, New York, NY, USA. ACM.
- [Cimiano et al., 2005] Cimiano, P., Ladwig, G., and Staab, S. (2005). Gimme’ the context : context-driven automatic semantic annotation with c-pankow. In *WWW ’05 : Proceedings of the 14th international conference on World Wide Web*, pages 332–341. ACM Press.
- [Cimiano et al., 2014] Cimiano, P., Unger, C., and McCrae, J. (2014). Ontology-based interpretation of natural language. *Synthesis Lectures on Human Language Technologies*, 7(2) :1–178.
- [Ciorascu et al., 2003] Ciorascu, C., Ciorascu, I., and Stoffel, K. (2003). Knowler - ontological support for information retrieval systems. In *Proceedings of 26th Annual International ACM SIGIR Conference, Workshop on Semantic Web*.
- [Ciravegna et al., 2002] Ciravegna, F., Dingli, A., Wilks, Y., and Petrelli, D. (2002). Adaptive information extraction for document annotation in amilcare. In *Proceedings of the 25th annual*

- international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, New York, NY, USA. ACM.
- [Clark et al., 2004] Clark, P., Thompson, J., and Porter, B. (2004). Knowledge patterns. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 191–207. Springer Berlin Heidelberg.
- [Codocedo et al., 2013] Codocedo, V., Lykourantzou, I., Astudillo, H., and Napoli, A. (2013). Using pattern structures to support information retrieval with formal concept analysis. In *FCA4AI@IJCAI, Proceedings of the International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI at IJCAI 2013), Beijing, China, August 5*, volume 1058 of *CEUR Workshop Proceedings*, pages 15–24. CEUR-WS.org.
- [Codocedo et al., 2012] Codocedo, V., Lykourantzou, I., and Napoli, A. (2012). A contribution to semantic indexing and retrieval based on FCA - an application to song datasets. In *CLA, Proceedings of The Ninth International Conference on Concept Lattices and Their Applications, Fuengirola (Málaga), Spain, October 11-14*, volume 972 of *CEUR Workshop Proceedings*, pages 257–268. CEUR-WS.org.
- [Codocedo et al., 2014] Codocedo, V., Lykourantzou, I., and Napoli, A. (2014). A semantic approach to concept lattice-based information retrieval. *Annals of Mathematics and Artificial Intelligence*, 72(1-2) :169–195.
- [Cointet and Roth, 2009] Cointet, J. and Roth, C. (2009). Socio-semantic dynamics in a blog network. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 4, pages 114–121.
- [Cole and Eklund, 2001] Cole, R. J. and Eklund, P. W. (2001). Browsing semi-structured web texts using formal concept analysis. In *Conceptual Structures : Broadening the Base, 9th International Conference on Conceptual Structures, ICCS*, pages 319–332.
- [Cole et al., 2003] Cole, R. J., Eklund, P. W., and Stumme, G. (2003). Document retrieval for e-mail search and discovery using formal concept analysis. *Applied Artificial Intelligence*, 17(3) :257–280.
- [Comparot et al., 2010] Comparot, C., Haemmerlé, O., and Hernandez, N. (2010). Expression de requêtes en graphes conceptuels à partir de mots-clés et de patrons. In *Journées Francophones d'Ingénierie des Connaissances (IC), Nîmes, 08/06/2010-11/06/2010*, pages 81–92. Cépaduès Editions.
- [Corby et al., 2004] Corby, O., Dieng-Kuntz, R., and Faron-Zucker, C. (2004). Querying the semantic web with corese search engine. In [de Mántaras and Saitta, 2004], pages 705–709.
- [Corcho et al., 2003] Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies : Where is their meeting point? *Data Knowl. Eng.*, 46(1) :41–64.
- [Crestani, 2000] Crestani, F. (2000). Exploiting the similarity of non-matching terms at retrieval time. *Journal of Information Retrieval*, 2 :25–45.
- [Croset et al., 2010] Croset, S., Grabmüller, C., Li, C., Kavaliauskas, S., and Rebolz-Schuhmann, D. (2010). The CALBC RDF triple store : retrieval over large literature content. *CoRR*, abs/1012.1650.
- [Cui et al., 2010] Cui, H., Jiang, K. Y., and Sanyal, P. P. (2010). From text to rdf triple store : an application for biodiversity literature. In *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem*, volume 47, pages 1–2. American Society for Information Science.

-
- [Dang and Viennet, 2012] Dang, T. A. and Viennet, E. (2012). Community detection based on structural and attribute similarities. In *International Conference on Digital Society (ICDS)*, pages 7–14.
- [Dao et al., 2004] Dao, M., Huchard, M., Hacene, M. R., Roume, C., and Valtchev, P. (2004). Improving generalization level in uml models iterative cross generalization in practice. In *International Conference on Computational Science (ICCS'04)*, pages 346–360.
- [d'Aquin and Motta, 2011] d'Aquin, M. and Motta, E. (2011). Extracting relevant questions to an rdf dataset using formal concept analysis. In *Proceedings of the Sixth International Conference on Knowledge Capture, K-CAP '11*, pages 121–128, New York, NY, USA. ACM.
- [Davey and Priestley, 2002] Davey, B. A. and Priestley, H. A. (2002). *Introduction to Lattices and Order (2. ed.)*. Cambridge University Press.
- [de Mántaras and Saitta, 2004] de Mántaras, R. L. and Saitta, L., editors (2004). *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*. IOS Press.
- [Demko and Bertet, 2012] Demko, C. and Bertet, K. (2012). Information retrieval by on-line navigation in the latticial space-search of a database, with limited objects access. In *FCA4AI@ECAI, Proceedings of the International Workshop "What can FCA do for Artificial Intelligence ?" (FCA4AI at ECAI 2012), Montpellier, France, August 28*, volume 939 of *CEUR Workshop Proceedings*, pages 33–40. CEUR-WS.org.
- [Desmontiles and Jacquin, 2002] Desmontiles, E. and Jacquin, C. (2002). Annotations sur le web : notes de lecture. *AS CNRS Web Sémantique*.
- [Després and Szulman, 2007] Després, S. and Szulman, S. (2007). Merging of legal micro-ontologies from european directives. *Artif. Intell. Law*, 15(2) :187–200.
- [Devignes et al., 2010] Devignes, M.-D., Franiatte, P., Messai, N., Bresso, E., Napoli, A., and Smail-Tabbone, M. (2010). Bioregistry : Automatic extraction of metadata for biological database retrieval and discovery. *International Journal of Metadata, Semantics and Ontologies (IJMSO)*, 5(3) :184–193.
- [Ding, 2011] Ding, Y. (2011). Scientific collaboration and endorsement : Network analysis of coauthorship and citation networks. *Journal of Informetrics*, 5(1) :187–203.
- [Dolques et al., 2013] Dolques, X., Ber, F. L., Huchard, M., and Nebut, C. (2013). Analyse relationnelle de concepts pour l'exploration de données relationnelles. In *Extraction et gestion des connaissances (EGC'2013), Actes, 29 janvier - 01 février 2013, Toulouse, France*, volume RNTI-E-24 of *Revue des Nouvelles Technologies de l'Information*, pages 121–132. Hermann-Éditions.
- [Domenach et al., 2012] Domenach, F., Ignatov, D. I., and Poelmans, J., editors (2012). *Formal Concept Analysis - 10th International Conference, ICFCA 2012, Leuven, Belgium, May 7-10, 2012. Proceedings*, volume 7278 of *Lecture Notes in Computer Science*. Springer.
- [Ducrou et al., 2006] Ducrou, J., Vormbrock, B., and Eklund, P. W. (2006). Fca-based browsing and searching of a collection of images. In *Conceptual Structures : Inspiration and Application, 14th International Conference on Conceptual Structures, ICCS*, pages 203–214.
- [Ducrou and Eklund, 2008] Ducrou, J. R. and Eklund, P. W. (2008). An intelligent user interface for browsing and searching mpeg-7 images using concept lattices. *International J. Foundations of Computer Science*, 19(2) :359–381.

- [Engeljehring and Scheffbeck, 2006] Engeljehring, W. and Scheffbeck, G. (2006). *The E-LAW Project in Austria. Electronic support of Law Making*. Austrian Parliament, Parliamentary Administration, Vienna. Available at : http://www.parlament.gv.at/ZUSD/PDF/2006-04-18_Publikation-Englisch.pdf.
- [Euzenat, 2001] Euzenat, J. (2001). L'annotation formelle de documents en huit (8) questions. In *Actes 6e journées sur ingénierie des connaissances (IC)*, pages 95–110, Grenoble (FR). Jean Charlet (éd).
- [Fernández et al., 2011] Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., and Motta, E. (2011). Semantically enhanced information retrieval : An ontology-based approach. *Web Semantics : Science, Services and Agents on the World Wide Web*, 9(4) :434 – 452. {JWS} special issue on Semantic Search.
- [Ferré, 2007] Ferré, S. (2007). Camelis : Organizing and browsing a personal photo collection with a logical information system. In *Proceedings of the Fifth International Conference on Concept Lattices and Their Applications, CLA 2007, Montpellier, France, October 24-26*.
- [Ferré, 2009] Ferré, S. (2009). Camelis : a logical information system to organise and browse a collection of documents. *International Journal of General Systems*, 38(4) :379–403.
- [Ferré, 2010] Ferré, S. (2010). Conceptual navigation in rdf graphs with sparql-like queries. In [Kwuida and Sertkaya, 2010], pages 193–208.
- [Fieschi et al., 2009] Fieschi, M., Staccini, P., Bouhaddou, O., Lovis, C., Jonquet, C., Shah, N., and Musen, M. A. (2009). Un service web pour l'annotation sémantique de données biomédicales avec des ontologies. In *Risques, Technologies de l'Information pour les Pratiques Médicales*, volume 17 of *Informatique et Santé*, pages 151–162. Springer Paris.
- [Formica, 2008] Formica, A. (2008). Concept similarity in formal concept analysis : An information content approach. *Knowledge Based Systems*, 21(1) :80–87.
- [Fowler et al., 2007] Fowler, J., Johnson, T., J.F., S., S., J., and P.J., W. (2007). Network analysis and the law : Measuring the legal importance of precedents at the u.s. supreme court. *Political Analysis*, 15 :324–346.
- [Fowler and Jeon, 2008] Fowler, J. H. and Jeon, S. (2008). The authority of supreme court precedent. *Social Networks*, 30 :16–30.
- [Gangemi, 2005] Gangemi, A. (2005). Ontology design patterns for semantic web content. In *Proceedings of the 4th International Conference on The Semantic Web, ISWC'05*, pages 262–276, Berlin, Heidelberg. Springer-Verlag.
- [Gangemi, 2007] Gangemi, A. (2007). Design patterns for legal ontology constructions. In Casanovas, P., Biasiotti, M. A., Francesconi, E., and Sagri, M.-T., editors, *LOAIT , Proceedings of the 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques June 4th, 2007, Stanford University, Stanford, CA, USA*, volume 321 of *CEUR Workshop Proceedings*, pages 65–85. CEUR-WS.org.
- [Gangemi et al., 2002] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening ontologies with dolce. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, EKAW '02*, pages 166–181, London, UK, UK. Springer-Verlag.
- [Gangemi et al., 2003] Gangemi, A., Sagri, M.-T., and Tiscornia, D. (2003). Metadata for content description in legal information. In *In Proc.s of LegOnt Workshop on Legal Ontologies*.

-
- [Gangemi et al., 2005] Gangemi, A., Sagri, M.-T., and Tiscornia, D. (2005). A constructive framework for legal ontologies. In Benjamins, V., Casanovas, P., Breuker, J., and Gangemi, A., editors, *Law and the Semantic Web*, volume 3369 of *Lecture Notes in Computer Science*, pages 97–124. Springer Berlin Heidelberg.
- [Ganter, 1984] Ganter, B. (1984). Two basic algorithms in concept analysis. FB4-Preprint 831, Technische Hochschule Darmstadt.
- [Ganter et al., 2005] Ganter, B., Stumme, G., and Wille, R., editors (2005). *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer.
- [Ganter and Wille, 1999a] Ganter, B. and Wille, R. (1999a). *Formal Concept Analysis*. Springer, mathematical foundations edition.
- [Ganter and Wille, 1999b] Ganter, B. and Wille, R. (1999b). *Formal Concept Analysis*. Springer, mathematical foundations edition.
- [Geist, 2009] Geist, A. (2009). *Using Citation Analysis Techniques for Computer-Assisted Legal Research in Continental Jurisdictions*. PhD thesis, Edinburgh, EH8 9YL, United Kingdom.
- [Giannopoulos et al., 2010] Giannopoulos, G., Bikakis, N., Dalamagas, T., and Sellis, T. K. (2010). Gontogle : A tool for semantic annotation and search. In *ESWC (2)*, pages 376–380.
- [Gillard, 2002] Gillard, L. (2002). Indexation de documents annotés. Technical report.
- [Godin et al., 1995a] Godin, R., Mineau, G., and Missaoui, R. (1995a). Incremental structuring of knowledge bases. In Ellis, G., Levinson, R. A., Fall, A., and Dahl, V., editors, *Proceedings of the 1st International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency (KRUSE'95), Santa Cruz (CA), USA*, pages 179–193. Department of Computer Science, University of California at Santa Cruz.
- [Godin et al., 1995b] Godin, R., Mineau, W., and Missaoui, R. (1995b). Méthodes de classification conceptuelle basées sur les treillis de galois et applications. *Revue d'intelligence artificielle*, 9 :105–137.
- [Godin et al., 1995c] Godin, R., Missaoui, R., and Alaoui, H. (1995c). Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. *Computational Intelligence*, 11 :246–267.
- [Godin et al., 1993] Godin, R., Missaoui, R., and April, A. (1993). Experimental comparison of navigation in a galois lattice with conventional information retrieval methods. *International Journal of Man-machine Studies*, 38 :747–767.
- [Governatori, 2009] Governatori, G., editor (2009). *Legal Knowledge and Information Systems - JURIX 2009 : The Twenty-Second Annual Conference on Legal Knowledge and Information Systems, Rotterdam, The Netherlands, 16-18 December 2009*, volume 205 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) :199 – 220.
- [Guenoche and Mechelen, 1993] Guenoche, A. and Mechelen, I. V. (1993). Galois approach to the induction of concepts. In Mechelen, I. V., Hampton, J., Michalski, R., and Theuns, P., editors, *Categories and Concepts. Theoretical Views and Inductive Data Analysis*, pages 287–308. Academic Press, London.

- [Guissé et al., 2012] Guissé, A., Lévy, F., and Nazarenko, A. (2012). From regulatory texts to brms : How to guide the acquisition of business rules? In Bikakis, A. and Giurca, A., editors, *Rules on the Web : Research and Applications*, volume 7438 of *Lecture Notes in Computer Science*, pages 77–91. Springer Berlin Heidelberg.
- [Guissé et al., 2009] Guissé, A., Lévy, F., Nazarenko, A., and Szulman, S. (2009). Annotation sémantique pour l’indexation de règles métiers. In L’Homme, M.-C. and Szulman, S., editors, *Conférence Internationale sur la Terminologie et l’Intelligence Artificielle (TIA 2009)*, page (electronic medium). Université Paul Sabatier - Toulouse.
- [Gultemen and van Engers, 2013] Gultemen, D. and van Engers, T. (2013). Graph-based linking and visualization for legislation documents (glvd). In *In : Network Analysis in Law Workshop, at ICAIL 2013 : XIV International Conference on AI and Law, NAIL2013@ICAIL, Rome, Italy, June 14th, 2013*.
- [Haav and Lubi, 2001] Haav, H.-M. and Lubi, T.-L. (2001). A survey of concept-based information retrieval tools on the web. In *5th East-European Conference, ADBIS 2001*, Vilnius, Lithuania.
- [Hacene et al., 2011] Hacene, M. R., Valtchev, P., and Nkambou, R. (2011). Supporting ontology design through large-scale fca-based ontology restructuring. In *Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK, July 25-29*, volume 6828 of *Lecture Notes in Computer Science*, pages 257–269. Springer.
- [Harth and Decker, 2004] Harth, A. and Decker, S. (2004). Yet another rdf store : Perfect index structures for storing semantic web data with contexts. Technical report, DERI Technical Report.
- [Heath and Bizer, 2011] Heath, T. and Bizer, C. (2011). *Linked Data : Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers.
- [Henzinger, 2000] Henzinger, M. (2000). Link analysis in web information retrieval. *IEEE DATA ENGINEERING BULLETIN*, 23 :3–8.
- [Hoekstra, 2011] Hoekstra, R. (2011). The metalex document server : legal documents as versioned linked data. In *Proceedings of the 10th International Conference on the Semantic Web, ISWC’11*, pages 128–143, Berlin, Heidelberg. Springer-Verlag.
- [Hoekstra et al., 2009] Hoekstra, R., Breuker, J., Bello, M. D., and Boer, A. (2009). Lkif core : Principled ontology development for the legal domain. In *Proceedings of the 2009 conference on Law, Ontologies and the Semantic Web : Channelling the Legal Information Flood*, pages 21–52, Amsterdam. IOS Press.
- [Hossain and Angryk, 2007] Hossain, M. S. and Angryk, R. A. (2007). Gdclust : A graph-based document clustering technique. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, ICDMW ’07*, pages 417–422, Washington, DC, USA. IEEE Computer Society.
- [Hubert et al., 2009] Hubert, G., Mothe, J., Ralalason, B., and Ramanonjisoa, B. (2009). Modèle d’indexation dynamique à base d’ontologies. In *CORIA*, pages 169–184. LSIS-USTV.
- [Huchard et al., 2007] Huchard, M., Hacene, M. R., Roume, C., and Valtchev, P. (2007). Relational concept discovery in structured datasets. *Ann. Math. Artif. Intell.*, 49 :39–76.
- [IFLA, 1998] IFLA (1998). *Functional requirements for bibliographic records : final report*, volume 19 of *new series*. UBCIM publications, München. by IFLA Study Group on the Functional Requirements for Bibliographic Records.

-
- [Jo et al., 2007] Jo, Y., Lagoze, C., and Giles, C. L. (2007). Detecting research topics via the correlation between graphs and texts. In *KDD'07 - INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*, pages 370–379. ACM.
- [Kengue et al., 2005] Kengue, J. F. D., Valtchev, P., and Djamegni, C. T. (2005). A parallel algorithm for lattice construction. In Ganter, B. and Godin, R., editors, *Formal Concept Analysis*, volume 3403 of *Lecture Notes in Computer Science*, pages 249–264. Springer Berlin Heidelberg.
- [Kirchberg et al., 2012] Kirchberg, M., Leonardi, E., Tan, Y. S., Link, S., Ko, R. K. L., and Lee, B.-S. (2012). Formal concept discovery in semantic web data. In [Domenach et al., 2012], pages 164–179.
- [Kiryakov et al., 2004a] Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., and Goranov, K. M. (2004a). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2 :49–79.
- [Kiryakov et al., 2004b] Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., and Goranov, K. M. (2004b). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2 :49–79.
- [Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46 :604–632.
- [Koester, 2006] Koester, B. (2006). Conceptual knowledge retrieval with fooca : Improving web search engine results with contexts and concept hierarchies. In *Industrial Conference on Data Mining*, pages 176–190.
- [Krajca et al., 2008] Krajca, P., Outrata, J., and Vychodil, V. (2008). V. : Parallel recursive algorithm for fca. In *Palacky University, Olomouc*, pages 71–82.
- [Kuznetsov and Obiedkov, 2001] Kuznetsov, S. and Obiedkov, S. (2001). Algorithms for the construction of concept lattices and their diagram graphs. In Raedt, L. and Siebes, A., editors, *Principles of Data Mining and Knowledge Discovery*, volume 2168 of *Lecture Notes in Computer Science*, pages 289–300. Springer Berlin Heidelberg.
- [Kuznetsov et al., 2012] Kuznetsov, S. O., Neznanov, A. A., and Poelmans, J. (2012). A system for knowledge discovery in big dynamical text collections. In *FCA4AI@ECAI, Proceedings of the International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI at ECAI 2012), Montpellier, France, August 28*, volume 939 of *CEUR Workshop Proceedings*, pages 81–87. CEUR-WS.org.
- [Kuznetsov and Obiedkov, 2002] Kuznetsov, S. O. and Obiedkov, S. A. (2002). Comparing Performance of Algorithms for Generating Concept Lattices. *Journal of Experimental & Theoretical Artificial Intelligence*, 14 :189–216.
- [Kwuida and Sertkaya, 2010] Kwuida, L. and Sertkaya, B., editors (2010). *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*, volume 5986 of *Lecture Notes in Computer Science*. Springer.
- [Lau, 2004] Lau, G. T. (2004). *A comparative analysis framework for semi-structured documents, with applications to government regulations*. PhD thesis, Stanford, CA, USA. AAI3145557.
- [Law, 2009] Law, L. C. (2009). Metalex naming conventions and the semantic web. In *Legal Knowledge and Information Systems : JURIX 2009, the Twenty-second Annual Conference*, volume 205, page 31. IOS Press.
- [Lopez et al., 2007] Lopez, V., Uren, V., Motta, E., and Pasin, M. (2007). Aqualog : An ontology-driven question answering system for organizational semantic intranets. *Web Semant.*, 5(2) :72–105.

- [Lortal et al., 2006] Lortal, G., Todirascu, A., and Lewkowicz, M. (2006). Soutenir la coopération par l’indexation semi-automatique d’annotations. In *Actes d’IC, IC 2006 : Ingénierie des connaissances 2006 (Proceedings of the 17th French Knowledge Engineering Conference)*, Nantes, France, June 26-30, 2006, pages 61–70.
- [Losada and Barreiro, 2001] Losada, D. E. and Barreiro, A. (2001). A logical model for information retrieval based on propositional logic and belief revision. *The Computer Journal*, 44 :410–424.
- [Lu et al., 2011] Lu, Q., Conrad, J. G., Al-Kofahi, K., and Keenan, W. (2011). Legal document clustering with built-in topic segmentation. In Macdonald, C., Ounis, I., and Ruthven, I., editors, *CIKM*, pages 383–392. ACM.
- [Ma et al., 2013] Ma, Y., Lévy, F., and Nazarenko, A. (2013). Annotation sémantique pour des domaines spécialisés et des ontologies riches. In *Actes de la 20ème conférence du Traitement Automatique du Langage Naturel (TALN 2013)*.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- [Mercatali et al., 2005] Mercatali, P., Romano, F., Boschi, L., and Spinicci, E. (2005). Automatic translation from textual representations of laws to formal models through uml. In Moens, M.-F. and Spyns, P., editors, *JURIX*, volume 134 of *Frontiers in Artificial Intelligence and Applications*, pages 71–80. IOS Press.
- [Messai et al., 2005] Messai, N., Devignes, M.-D., Napoli, A., and Smaïl-Tabbone, M. (2005). Querying a bioinformatic data sources registry with concept lattices. In *ICCS*, pages 323–336.
- [Messai et al., 2006] Messai, N., Devignes, M.-D., Napoli, A., and Smaïl-Tabbone, M. (2006). Treillis de concepts et ontologies pour interroger l’annuaire de sources de données biologiques bioregistry. *Ingénierie des Systèmes d’Information (ISI)*, 11(1) :39–60.
- [Miklos et al., 2003] Miklos, Z., Neuman, G., Zdun, U., and Sintek, M. (2003). Querying semantic web resources using triple views. In *Proceedings of the 2nd International Semantic Web Conference (ISWC03)*, Sanibel Island, Florida, USA.
- [Mimouni et al., 2013] Mimouni, N., Fernández, M., Nazarenko, A., Bourcier, D., and Salotti, S. (2013). A relational approach for information retrieval on XML legal sources. In *International Conference on Artificial Intelligence and Law, ICAIL ’13, Rome, Italy, June 10-14, 2013*, pages 212–216.
- [Mimouni et al., 2012] Mimouni, N., Nazarenko, A., and Salotti, S. (2012). Classification conceptuelle d’une collection documentaire - intertextualité et recherche d’information. In *Proceedings of the 9th French Information Retrieval Conference (CORIA’12)*, pages 123–134.
- [Mimouni and Slimani, 2006] Mimouni, N. and Slimani, Y. (2006). Indexing and Searching Video Sequences Using Concept Lattices. In *Fourth International Conference on Concept Lattices and their Applications - CLA’06*, pages 285–290, Yasmine Hammamet, Tunisia.
- [Minard et al., 2011] Minard, A.-L., Ligozat, A.-L., and Grau, B. (2011). Extraction de relations dans des comptes rendus hospitaliers. In *22es Journées Francophones d’Ingénierie des Connaissances, IC 2011*, pages 491–506, Chambéry, France.
- [Missaoui, 2013] Missaoui, R. (2013). Analyse de réseaux sociaux par l’analyse formelle de concepts. In *Extraction et gestion des connaissances (EGC’2013), Actes, 29 janvier - 01 février 2013, Toulouse, France*, volume RNTI-E-24 of *Revue des Nouvelles Technologies de l’Information*, pages 3–4. Hermann-Éditions.

-
- [Moha et al., 2008] Moha, N., Hacene, A. R., Valtchev, P., and Guéhéneuc, Y.-G. (2008). Refactorings of design defects using relational concept analysis. In *Formal Concept Analysis, 6th International Conference, ICFCA 2008, Montreal, Canada, February 25-28*, volume 4933 of *Lecture Notes in Computer Science*, pages 289–304. Springer.
- [Mokhtari, 2010a] Mokhtari, N. (2010a). *Extraction et exploitation d’annotations sémantiques contextuelles à partir de texte*. PhD thesis, Université Sophia Antipolis.
- [Mokhtari, 2010b] Mokhtari, N. (2010b). *Extraction et exploitation d’annotations sémantiques contextuelles à partir de texte*. PhD thesis, Université de Nice-Sophia Antipolis.
- [Mokhtari and Dieng-Kuntz, 2008] Mokhtari, N. and Dieng-Kuntz, R. (2008). Extraction et exploitation des annotations contextuelles. In Guillet, F. and Trouse, B., editors, *Extraction et gestion des connaissances (EGC’2008), Actes des 8èmes journées Extraction et Gestion des Connaissances, Sophia-Antipolis, France, 29 janvier au 1er février 2008, 2 Volumes*, volume RNTI-E-11 of *Revue des Nouvelles Technologies de l’Information*, pages 7–18. Cépaduès-Éditions.
- [Mommers, 2010] Mommers, L. (2010). Ontologies in the legal domain. In Poli, R. and Seibt, J., editors, *Theory and Applications of Ontology : Philosophical Perspectives*, pages 265–276. Springer Verlag.
- [Mondary et al., 2007] Mondary, T., Bouffier, A., and Nazarenko, A. (2007). Between browsing and search, a new model for navigating through large documents. In Stella Vosniadou, D. K. and Athanassios Protopapas, editors, *proceedings of EuroCogSci07, the european cognitive science conference EuroCogSci07, The European Cognitive Science Conference 2007*, pages 634–639, Delphi Greece. Lawrence Erlbaum Associates.
- [Mooers, 1958] Mooers, C. N. (1958). A mathematical theory of language symbols in retrieval. In *International Confernece on Scientific Information*, pages 61–70. Zator Company.
- [Mrabet et al., 2012] Mrabet, Y., Bennacer, N., and Pernelle, N. (2012). Enrichissement contrôlé de bases de connaissances à partir de documents semi-structurés annotés. In *23es Journées Francophones d’Ingénierie des Connaissances, IC 2012*, Paris.
- [Mrabet et al., 2010] Mrabet, Y., Bennacer, N., Pernelle, N., and Thiam, M. (2010). Une approche pour la recherche sémantique de l’information dans les documents semi-structurés hétérogènes. In Centre de Publication Universitaire 2010, editor, *CONFérence en Recherche d’Informations et Applications - CORIA 2010, 7th French Information Retrieval Conference, Sousse, Tunisia, March 18-20, 2010. Proceedings. CONFérence en Recherche d’Infomations et Applications - CORIA 2010.*, pages 195–210, Sousse Tunisia. Fondation DIGITEO, projet SHIRI.
- [Nauer and Toussaint, 2008] Nauer, E. and Toussaint, Y. (2008). Classification dynamique par treillis de concepts pour la recherche d’information sur le web. In *CORIA ’08 : CONFérence en Recherche d’Information et Applications*, pages 71–86.
- [Nešić et al., 2010] Nešić, S., Crestani, F., Jazayeri, M., and Gašević, D. (2010). Concept-based semantic annotation, indexing and retrieval of office-like document units. In *Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO ’10*, pages 134–135, Paris, France, France.
- [Newman, 2004] Newman, M. E. J. (2004). Coauthorship networks and patterns of scientific collaboration. In *Proceedings of the National Academy of Sciences*, pages 5200–5205.
- [Nguifo and Njiwoua, 2005] Nguifo, E. M. and Njiwoua, P. (2005). Treillis de concepts et classification supervisée. *Technique et Science Informatiques*, 24(4) :449–488.

- [Nicolas Bonnel, 2006] Nicolas Bonnel, M. C. (2006). Evaluation des interfaces utilisateur d'information. In *Atelier Visualisation et extraction de connaissances - EGC*, Lille, France.
- [Norris, 1978] Norris, E. M. (1978). An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2) :243–250.
- [Oberle et al., 2006] Oberle, D., Lamparter, S., Grimm, S., Vrandečić, D., Staab, S., and Gangemi, A. (2006). Towards ontologies for formalizing modularization and communication in large software systems. *Appl. Ontol.*, 1(2) :163–202.
- [Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking : Bringing order to the web. Technical report, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.
- [Palmirani and Benigni, 2002] Palmirani, M. and Benigni, F. (2002). Norma-system : A legal information system for managing time.
- [Palmirani and Brighi, 2009] Palmirani, M. and Brighi, R. (2009). Model regularity of legal language in active modifications. In *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue - International Workshops AICOL-I/IVR-XXIV Beijing, China, September 19, 2009 and AICOL-II/JURIX 2009, Rotterdam, The Netherlands, December 16, 2009 Revised Selected Papers*, pages 54–73.
- [Palmirani and Brighi, 2010] Palmirani, M. and Brighi, R. (2010). Model regularity of legal language in active modifications. In *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue*, volume 6237, pages 54–73. Lecture Notes in Computer Science.
- [Palmirani et al., 2003] Palmirani, M., Brighi, R., and Massini, M. (2003). Automated extraction of normative references in legal texts. In *Proceedings of the 9th international conference on Artificial intelligence and law, ICAIL '03*, pages 105–106, New York, NY, USA. ACM.
- [Palmirani and Cervone, 2009] Palmirani, M. and Cervone, L. (2009). Legal change management with a native xml repository. In [Governatori, 2009], pages 146–155.
- [Palmirani et al., 2009] Palmirani, M., Cervone, L., and Vitali, F. (2009). Legal metadata interchange framework to match cen metalex. In [DBL, 2009], pages 232–233.
- [Palmirani et al., 2012a] Palmirani, M., Ognibene, T., and Cervone, L. (2012a). Legal rules, text and ontologies over time. In *Proceedings of the RuleML2012@ECAI Challenge, at the 6th International Symposium on Rules, Montpellier, France, August 27th-29th, 2012*.
- [Palmirani et al., 2012b] Palmirani, M., Pagallo, U., Casanovas, P., and Sartor, G., editors (2012b). *AI Approaches to the Complexity of Legal Systems. Models and Ethical Challenges for Legal Systems, Legal Language and Legal Ontologies, Argumentation and Software Agents - International Workshop AICOL-III, Held as Part of the 25th IVR Congress, Frankfurt am Main, Germany, August 15-16, 2011. Revised Selected Papers*, volume 7639 of *Lecture Notes in Computer Science*. Springer.
- [Pejtersen, 1998] Pejtersen, A. M. (1998). Semantic information retrieval. *Commun. ACM*, 41 :90–92.
- [Pérez et al., 2009] Pérez, J., Arenas, M., and Gutierrez, C. (2009). Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3).
- [Pham et al., 2008] Pham, N.-K., Morin, A., and Gros, P. (2008). Recherche d'images par l'analyse factorielle des correspondances. In *CORIA'08 : Conférence en Recherche d'Information et Applications*, pages 23–38.

-
- [Pivovarov and Trunov, 2011] Pivovarov, G. and Trunov, S. (2011). Clustering and classification in text collections using graph modularity. *Journal of Machine Learning Research, CoRR*, abs/1105.5789.
- [Poelmans et al., 2011] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G., and Kuznetsov, S. O. (2011). Text mining scientific papers : a survey on fca-based information retrieval research. In *Advances in Data Mining. 11th Industrial Conference, ICDM 2011, New York, USA, September/August 2011, Poster and Industry Proceedings, Workshop on Data Mining in Life Sciences*, pages 82–96. IBAI Publishing.
- [Poelmans et al., 2013a] Poelmans, J., Ignatov, D. I., Kuznetsov, S. O., and Dedene, G. (2013a). Formal concept analysis in knowledge processing : A survey on applications. *Expert Syst. Appl.*, 40(16) :6538–6560.
- [Poelmans et al., 2013b] Poelmans, J., Kuznetsov, S. O., Ignatov, D. I., and Dedene, G. (2013b). Formal concept analysis in knowledge processing : A survey on models and techniques. *Expert Syst. Appl.*, 40(16) :6601–6623.
- [Pol et al., 2008] Pol, K., Patil, N., Patankar, S., and Das, C. (2008). A survey on web content mining and extraction of structured and semistructured data. In *Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology, ICETET '08*, pages 543–546, Washington, DC, USA. IEEE Computer Society.
- [Poshyvanyk and Marcus, 2007] Poshyvanyk, D. and Marcus, A. (2007). Combining formal concept analysis with information retrieval for concept location in source code. In *ICPC*, pages 37–48.
- [Pradel et al., 2012] Pradel, C., Haemmerlé, O., and Hernandez, N. (2012). Des patrons modulaires de requêtes sparql dans le système swip. In *23es Journées Francophones d'Ingénierie des Connaissances, IC 2012*, Paris, France.
- [Presutti and Gangemi, 2008] Presutti, V. and Gangemi, A. (2008). Content ontology design patterns as practical building blocks for web ontologies. In Li, Q., Spaccapietra, S., Yu, E. S. K., and Olivé, A., editors, *ER, Conceptual Modeling - ER 2008, 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20-24, 2008. Proceedings*, volume 5231 of *Lecture Notes in Computer Science*, pages 128–141. Springer.
- [Priss, 2000] Priss, U. (2000). Faceted knowledge representation. *Electronic Transactions on Artificial Intelligence*, 4(C) :21–33.
- [Rada et al., 1989] Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1) :17–30.
- [Ralalason, 2010] Ralalason, B. J. V. (2010). *Représentation multi-facette des documents pour leur accès sémantique*. Thèse de doctorat, Université de Toulouse, Toulouse, France.
- [Ramírez, 2007] Ramírez, R. C. M. (2007). Semantic information retrieval : a return on experience. *Engineering Letters*, 15(2) :234–239.
- [Reeve, 2005] Reeve, L. (2005). Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 1634–1638. ACM Press.
- [Ren and Bracewell, 2009] Ren, F. and Bracewell, D. B. (2009). Advanced information retrieval. *Electron. Notes Theor. Comput. Sci.*, 225 :303–317.
- [Renard et al., 2009] Renard, A., Calabretto, S., and Rumpler, B. (2009). Recherche d'information sémantique : Appariement sémantique flou de documents semi-structurés. *Atelier RISE - INFORSID*.

- [Reymonet et al., 2007] Reymonet, A., Thomas, J., and Aussenac-Gilles, N. (2007). Modelling ontological and terminological resources in owl dl. *Proceedings of ISWC*, 7.
- [Rocha et al., 2004] Rocha, C., Schwabe, D., and Aragao, M. P. (2004). A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 374–383, New York, NY, USA. ACM.
- [Rouane et al., 2007] Rouane, M. H., Huchard, M., Napoli, A., and Valtchev, P. (2007). A proposal for combining formal concept analysis and description logics for mining relational data. In *Proceedings of the 5th international conference on Formal concept analysis, ICFCA 2007*, LNAI, pages 51–65. Springer-Verlag.
- [Rouane et al., 2010] Rouane, M. H., Huchard, M., Napoli, A., and Valtchev, P. (2010). Using formal concept analysis for discovering knowledge patterns. In *CLA'10 : 7th International Conference on Concept Lattices and Their Applications*, CEUR, pages 223–234. University of Sevilla.
- [Rouane et al., 2013] Rouane, M. H., Huchard, M., Napoli, A., and Valtchev, P. (2013). Relational concept analysis : mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence*, 67(1) :81–108.
- [Rouane-Hacene et al., 2010] Rouane-Hacene, M., Fennouh, S., Nkambou, R., and Valtchev, P. (2010). Refactoring of ontologies : Improving the design of ontological models with concept analysis. In *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on*, volume 2, pages 167–172.
- [Ruhl, 1997] Ruhl, J. B. (1997). Thinking of environmental law as a complex adaptive system : how to clean up the environment by making a mess of environmental law. *Hous. L. Rev.*, 34 :933–1002.
- [Saada et al., 2012] Saada, H., Dolques, X., Huchard, M., Nebut, C., and Sahraoui, H. A. (2012). Learning model transformations from examples using fca : One for all or all for one? In *CLA, Proceedings of The Ninth International Conference on Concept Lattices and Their Applications, Fuengirola (Málaga), Spain, October 11-14*, volume 972 of *CEUR Workshop Proceedings*, pages 45–56. CEUR-WS.org.
- [Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11) :613–620.
- [Sartor et al., 2010] Sartor, G., Palmirani, M., Francesconi, E., and Biasiotti, M. A. (2010). *Legislative XML for the Semantic Web*. Springer-Verlag.
- [Sartor et al., 2011] Sartor, G., Palmirani, M., Francesconi, E., and Biasiotti, M. A. (2011). *Law, Governance and Technology : Legislative Xml for the Semantic Web : Principles, Models, Standards for Document Management*. Law, Governance and Technology Series, 4. Springer London, Limited.
- [Savvas and Bassiliades, 2009] Savvas, I. and Bassiliades, N. (2009). A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration. *Expert Systems with Applications*, 36(3, Part 1) :4467 – 4478.
- [Schaeffer, 2007] Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1) :27–64.
- [Shaheed, 2005] Shaheed, J. (2005). A top-level language-biased legal ontology. In *In : Workshop Proceedings, Legal Ontologies and Artificial Intelligence Techniques, International Association for Artificial Intelligence and Law, Workshop Series No 4, Wolf Legal Publishers, 2005*, pages 13–24.

-
- [Shi et al., 2011] Shi, L., Toussaint, Y., Napoli, A., and Blansch e, A. (2011). Mining for reengineering : An application to semantic wikis using formal and relational concept analysis. In *The Semantic Web : Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, Proceedings, Part II*, volume 6644 of *Lecture Notes in Computer Science*, pages 421–435. Springer.
- [Sintek and Decker, 2002] Sintek, M. and Decker, S. (2002). Triple - a query, inference, and transformation language for the semantic web. In *Proceedings of the First International Semantic Web Conference on The Semantic Web, ISWC '02*, pages 364–378, London, UK, UK. Springer-Verlag.
- [Sma il-Tabbone et al., 2005] Sma il-Tabbone, M., Osman, S., Messai, N., Napoli, A., and Devignes, M.-D. (2005). Bioregistry : A structured metadata repository for bioinformatic databases. In *Computational Life Sciences, First International Symposium, CompLife 2005, Konstanz, Germany, September 25-27, 2005, Proceedings*, pages 46–56.
- [Strok and Neznanov, 2010] Strok, F. and Neznanov, A. (2010). Comparing and analyzing the computational complexity of fca algorithms. In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, SAICSIT '10*, pages 417–420, New York, NY, USA. ACM.
- [Studer et al., 1998] Studer, R., Benjamins, V., and Fensel, D. (1998). Knowledge engineering : Principles and methods. *Data and Knowledge Engineering*, 25(1-2) :161 – 197.
- [Stumme et al., 2002] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., and Lakhal, L. (2002). Computing iceberg concept lattices with titanic. *Data Knowl. Eng.*, 42(2) :189–222.
- [Tiscornia,] Tiscornia, D. The lois project : Lexical ontologies for legal information sharing. *Library*, 2000(1) :189–204.
- [Tulloch, 1995] Tulloch, G. (1995). On the desirable degree of detail in the law. *European Journal of Law and Economics*, 2 :199–209.
- [Unger et al., 2012] Unger, C., B uhmann, L., Lehmann, J., Ngonga, A.-C. N., Gerber, D., and Cimiano, P. (2012). Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, pages 639–648. ACM.
- [Uren et al., 2006a] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., and Ciravegna, F. (2006a). Semantic annotation for knowledge management : Requirements and a survey of the state of the art. *Journal of Web Semantics*, 4.
- [Uren et al., 2006b] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., and Ciravegna, F. (2006b). Semantic annotation for knowledge management : Requirements and a survey of the state of the art. *Web Semant.*, 4(1) :14–28.
- [Vallet et al., 2005] Vallet, D., Fernandez, M., and Castells, P. (2005). An ontology-based information retrieval model. In *In ESWC*, pages 455–470. Springer.
- [Valtchev et al., 2002] Valtchev, P., Missaoui, R., and Lebrun, P. (2002). A partition-based approach towards constructing galois (concept) lattices. *Discrete Math.*, 256(3) :801–829.
- [Ven et al., 2007] Ven, S. V. D., Hoekstra, R., and Winkels, R. (2007). Metavex : Regulation drafting meets the semantic web. In *In Proc. of the Workshop on Semantic Web technology for Law (SW4Law)*.
- [Ventos and Soldano, 2005] Ventos, V. and Soldano, H. (2005). Les treillis de galois alpha. *Revue d'Intelligence Artificielle*, 19(4-5) :799–827.

- [Voorhees, 1994] Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '94*, pages 61–69.
- [Waiyamai and Lakhal, 2000] Waiyamai, K. and Lakhal, L. (2000). Knowledge discovery from very large databases using frequent concept lattices. In Lopez de Mantaras, R. and Plaza, E., editors, *Machine Learning : ECML 2000*, volume 1810 of *Lecture Notes in Computer Science*, pages 437–445. Springer Berlin Heidelberg.
- [Wang and Xu, 2000] Wang, N. and Xu, X. (2000). A method to build ontology. In *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on*, volume 2, pages 672–673 vol.2.
- [Wijaya and Bressan, 2006] Wijaya, D. T. and Bressan, S. (2006). Clustering web documents using co-citation, coupling, incoming, and outgoing hyperlinks : a comparative performance analysis of algorithms. *IJWIS*, 2(2) :69–76.
- [Wille, 1982] Wille, R. (1982). Restructuring lattice theory : an approach based on hierarchies of concepts. *Ordered sets*, pages 445–470.
- [Wille, 1984] Wille, R. (1984). Line diagrams of hierarchical concept systems. *International Classification*, 2 :77–86.
- [Winkels et al., 2003] Winkels, R., Boer, A., and Hoekstra, R. (2003). Metalex : An xml standard for legal documents. In *Proceedings of the XML Europe Conference, London (UK)*.
- [Winkels et al., 2013] Winkels, R., Boer, A., and Plantevin, I. (2013). Creating context networks in dutch legislation. In [Ashley, 2013], pages 155–164.
- [Winkels and de Ruyter, 2011] Winkels, R. and de Ruyter, J. (2011). Survival of the fittest : Network analysis of dutch supreme court cases. In [Palmirani et al., 2012b], pages 106–115.
- [Wray and Eklund, 2011] Wray, T. and Eklund, P. W. (2011). Exploring the information space of cultural collections using formal concept analysis. In *Formal Concept Analysis - 9th International Conference, ICFCA*, pages 251–266.
- [Wyner and Hoekstra, 2012] Wyner, A. and Hoekstra, R. (2012). A legal case owl ontology with an instantiation of *popov v. hayashi*. *Artificial Intelligence and Law*, 20(1) :83–107.
- [Yan et al., 2011] Yan, S., Lee, D., and Wang, A. H. (2011). Costco : Robust content and structure constrained clustering of networked documents. In *Computational Linguistics and Intelligent Text Processing - 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part II*, Lecture Notes in Computer Science, pages 289–300.
- [Yoo et al., 2007] Yoo, I., Hu, X., and Song, I.-Y. (2007). A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. *BMC Bioinformatics*, 8(S-9).
- [Zargayouna, 2004] Zargayouna, H. (2004). Contexte et sémantique pour une indexation de documents semi-structurés. In *CORIA*, pages 161–178.

Résumé

Une collection documentaire est généralement représentée comme un ensemble de documents mais cette modélisation ne permet pas de rendre compte des relations intertextuelles et du contexte d'interprétation d'un document. Le modèle documentaire classique trouve ses limites dans les domaines spécialisés où les besoins d'accès à l'information correspondent à des usages spécifiques et où les documents sont liés par de nombreux types de relations. Ce travail de thèse propose deux modèles permettant de prendre en compte cette complexité des collections documentaire dans les outils d'accès à l'information. Le premier modèle est basée sur l'analyse formelle et relationnelle de concepts, le deuxième est basée sur les technologies du web sémantique. Appliquées sur des objets documentaires ces modèles permettent de représenter et d'interroger de manière unifiée les descripteurs de contenu des documents et les relations intertextuelles qu'ils entretiennent.

Mots-clés: Réseau de documents, Intertextualité, Recherche d'information, Analyse formelle et relationnelle de concepts, Ontologies.

Abstract

A collection of documents is generally represented as a set of documents but this simple representation does not take into account cross references between documents, which often defines their context of interpretation. This standard document model is less adapted for specific professional uses in specialized domains in which documents are related by many various references and the access tools need to consider this complexity. We propose two models based on formal and relational concept analysis and on semantic web techniques. Applied on documentary objects, these two models represent and query in a unified way documents content descriptors and documents relations.

Keywords: Document network, Intertextuality, Information Retrieval, Formal and Relational Concept Analysis, Ontologies.

