Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Cotutelle internationale avec Université de Tunis - Institut Supérieur de Gestion

---

Présentée et soutenue par :
M. NAFFAKHI Najeh

le lundi 08 juillet 2013

Titre :

Un modèle de recherche d'information agrégée basée sur les réseaux bayésiens dans des documents semi-structurés.

École doctorale et discipline ou spécialité :
ED MITT : Image, Information, Hypermedia

Unité de recherche :
IRIT-UMR 5505

Directeur(s) de Thèse :

M. BOUGHANEM Mohand et Mme FAIZ Rim

Jury :
Pr. HACID Mohand-Said (Président), Pr. SAVOY Jacques (Rapporteur), Pr. GARGOURI Faïez (Rapporteur), Pr. BOUGHANEM Mohand (Directeur) et Pr. FAIZ Rim (Co-directrice)

Résumé

XML est considéré comme un métalangage permettant de décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir. L'accès à ce type de document soulève de nouvelles problématiques liées à la co-existence de l'information structurelle et de l'information de contenu. L'objectif des systèmes de Recherche d'Information Structurée (RIS) n'est plus de renvoyer le document répondant à la requête, mais plutôt l'unité documentaire (élément XML, portion du document) répondant au mieux à la requête. Ainsi, au lieu de récupérer une liste d'éléments qui sont susceptibles de répondre à la requête, notre objectif est d'agréger dans un même résultat des éléments pertinents, non-redondants et complémentaires.

Les travaux décrits dans cette thèse s'intéressent à l'agrégation des unités documentaires à partir des documents semi-structurés de type XML. Nous proposons de nouvelles approches d'agrégation et d'élagage en utilisant différentes sources d'évidence contenu et structure. Nous proposons un modèle basé sur les réseaux bayésiens. Les relations de dépendances entre requête-termes d'indexation et termes d'indexation-éléments sont quantifiées par des mesures de probabilité. Dans ce modèle, la requête de l'utilisateur déclenche un processus de propagation pour sélectionner les éléments pertinents. Dans notre modèle, nous cherchons à renvoyer à l'utilisateur un agrégat au lieu d'une liste d'éléments. En fait, l'agrégat formulé à partir d'un document est considéré comme étant un ensemble d'éléments ou une unité d'information (portion d'un document) qui répond le mieux à la requête de l'utilisateur. Cet agrégat doit répondre à trois aspects à savoir la pertinence, la non-redondance et la complémentarité pour qu'il soit qualifié comme une réponse à cette requête. L'utilité des agrégats retournés est qu'ils donnent à l'utilisateur un aperçu sur le contenu informationnel de cette requête dans la collection de documents.

Une autre source d'évidence que nous avons aussi utilisée est l'information structurelle. À l'aide des techniques d'élagage utilisées dans une première hypothèse, nous appliquons la relation de la non-inclusion entre les éléments d'un même agrégat afin d'éliminer les éléments qui véhiculent la même information. Une deuxième hypothèse basée sur la source d'évidence : l'information de contenu, est appliquée en utilisant la mesure de similarité "cosine" afin d'éliminer les éléments similaires entre les agrégats renvoyés.

D'une manière générale, nous essayons de renvoyer à l'utilisateur un nombre limité des ensembles d'éléments XML, qui satisfont à la fois aux trois aspects à savoir la pertinence, la non-redondance et la complémentarité.

Afin de valider notre modèle, nous l'avons évalué dans le cadre de la campagne d'évaluation INEX 2009 (utilisant plus que 2 666 000 documents XML de l'encyclopédie en ligne Wikipédia). Les expérimentations montrent l'intérêt de cette approche en mettant en évidence l'impact de l'agrégation de tels éléments.

Mots-clés : Recherche d'information agrégée, réseaux bayésiens, éléments XML, pertinence, redondance, complémentarité.

Abstract

XML is considered as a meta-language for writing any data domain through its extensibility. It will allow to structure, place the vocabulary and syntax of the data it will contain. Access to such documents raises new issues related to the coexistence of structural information and information content. The goal of Structured Information Retrieval systems is no longer to return the document answering the query, but the documentary unit (XML element, document's portion) that best suit the application. Thus, instead of retrieving a list of XML elements that are likely to respond to the query, our goal is to aggregate into a result space a set of XML elements that are relevant, non-redundant and complementary.

The work described in this thesis are concerned with the aggregation of XML elements. We propose new approaches to aggregating and pruning using different sources of evidence (content and structure). We propose a model based on Bayesian networks. The dependency relationships between query-terms and terms-elements are quantified by probability measures. In this model, the user's query triggers a propagation process to find XML elements. In our model, we search to return to the user an aggregate instead of a list of XML elements. In fact, the aggregate made from a document is considered an information unit (or a portion of this document) that best meets the user's query. This aggregate must meet three aspects namely relevance, non-redundancy and complementarity in order to answer the query. The value returned aggregates is that they give the user an overview of the information need in the collection.

Another source of evidence we used is the structural information. Using the pruning techniques used in a first hypothesis, we apply the relation of the non-inclusion between elements of the same aggregate to eliminate elements that convey the same information. A second hypothesis based on the source of evidence : information content, is applied using a cosine similarity measure to eliminate similar elements between the returned aggregates.

In general, we try to send to the user a limited number of sets of XML elements, which satisfy both the three aspects namely relevance, non-redundancy and complementarity.

In summary, we search to reduce the result space so that the user provides the slightest effort to find the needed information. We have validated our ap-

proach of aggregated search using INEX 2009 collection. Experiments show the usefulness of this approach by highlighting the impact of the aggregation of such elements.

Keywords : Aggregated search, Bayesian networks theory, XML documents, relevance, redundancy, complementarity.

Remerciements

Cette thèse est le fruit de quatre années d'efforts incessants, mais aussi d'échanges bénéfiques et de collaborations fructueuses entre l'IRIT et LARODEC. Ce travail n'aurait pas pu aboutir sans le concours précieux et généreux de personnes qui partagent la même passion pour la recherche scientifique. C'est avec un énorme plaisir que je remercie aujourd'hui toutes les personnes qui m'ont soutenu.

Tout d'abord, j'adresse mes plus vifs remerciements à Monsieur le Professeur Claude Chrisment qui m'accueillie au sein de son équipe SIG.

Je tiens à exprimer ma profonde gratitude à Monsieur Mohand Boughanem, Professeur à l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), pour m'avoir dirigé tout au long cette thèse. Je le remercie pour m'avoir soutenu et appuyé tout au long de ma thèse. Sa gentillesse, sa patience, son humour, sa disponibilité, ses précieux conseils, son exigence, ses commentaires et ses très nombreuses compétences ont été capitales durant ces années de recherche et m'ont profondément enrichi.

Je tiens à remercier vivement ma co-directrice de thèse, Madame Rim Faiz, Professeur à l'université de Carthage, IHEC - Tunis, pour avoir encadré et dirigé mes recherches. Je la remercie pour son souci constant de l'avancement de ma thèse et son suivi continu de mon travail, ses précieux conseils de tout ordre, sa disponibilité et sa confiance. Son expérience et ses grandes compétences ont permis l'accomplissement de ce travail. Par sa bonne humeur et sa collaboration, elle m'a toujours encouragé et aidé à surmonter les difficultés. Qu'elle trouve ici les marques de ma reconnaissance et de mon respect.

Je remercie très sincèrement Monsieur Jacques Savoy, Professeur à l'Université Neuchâtel, II - Suisse et Monsieur Faïez Gargouri, Professeur à l'Université de Sfax, ISIM - Tunisie, pour avoir accepté d'être rapporteurs de ce mémoire, et pour l'honneur qu'ils me font en participant au jury. Merci également à Monsieur Mohand-Said Hacid, Professeur à l'Université Claude Bernard Lyon 1, d'avoir accepté de juger ce travail et de faire partie du jury. Je les remercie pour leur évaluation scientifique et leur travail de synthèse.

Mes remerciements vont de même à tous les membres de l'équipe SIG à l'IRIT pour leur aide et leur gentillesse. Plus particulièrement, je tiens à

exprimer ma reconnaissance à Madame Karen Pinel-Sauvagnat, Maître de conférences à l'UPS et Madame Mouna Torjmen, Maître assistante à l'université de Sfax, ENIS. Je les remercie pour leurs aides, leurs disponibilités et leurs générosités pour faire avancer mes expérimentations. Je remercie mes amis de l'équipe qui ont contribué à la finalisation de quelques tâches d'évaluation dans ce mémoire. Je remercie Arezki Hammache, Cyril Laitang, Faten Atigui, Firas Damak, Ines Krichen, Lamjed Ben Jabeur, Madalina Mitran et M'Hamed Mataoui pour leur collaboration et leur disponibilité. Je remercie également toutes les personnes qui ont participé de façon volontaire aux expérimentations menées dans cette thèse. Je n'oublie pas non plus les docteurs qui ont été des anciens thésards : Anass El Haddadi, Arlind Kopliku, Dana Al Kukhun, Duy Dinh, Hamdi Chaker, Houssem Jerbi, Ihab Mallak, Mariam Daoud, Malik Muhammad Saad Missen et Ourdia Bouidghaghen et qui m'ont encouragé, leurs conseils m'ont toujours servi.

Merci aussi à tous les amis que j'ai connu à Toulouse et avec lesquels j'ai vécu des moments inoubliables.

Mes pensées se tournent enfin vers ma famille. Il n'existe pas de mot assez grand et fort pour remercier mes parents, mes sœurs et frères qui n'ont jamais cessé de croire en moi pendant toutes mes années d'études et qui m'ont toujours encouragé à aller de l'avant.

Le mot de la fin sera à celle à qui je dédie ce travail. Ma fiancée Abir qui m'a encouragé à y aller de l'avant, çà y est ! C'est fini ! On en parle plus ! C'est la première fois que je sens le goût du succès accompagné par un bonheur complet.

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Avec l'usage croissant des smartphones, envoi de messages sur les réseaux sociaux comme Facebook, Twitter, ... chaque individu génère, sans le savoir, une multitude d'informations précieuses. En 2010, les quantités d'informations (données, musiques, vidéos, documents, etc.) créées sont estimées à 1,2 zetta-octets [1]. La croissance de ces quantités d'informations va se poursuivre au rythme effréné de 45% par an jusqu'en 2020, prévoit le cabinet d'études IDC (cf. figure 1). Agrégées, comparées à des relevés historiques et mélangées aux données produites, ces informations constituent un réservoir considérable de connaissances utiles.

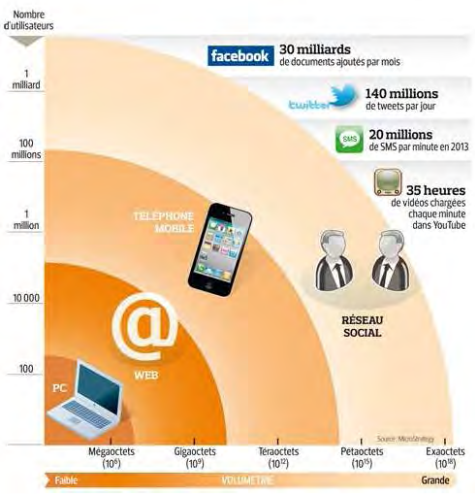

FIGURE 1 – Des volumes de données plus importants et plus complexes à traiter

Mais pour que l'abondance de l'information ne tue pas l'information, ces données doivent être gérées à l'aide de systèmes automatisés. Notre travail se situe dans le contexte de ces outils automatisés et plus précisément dans le domaine de la RI (Recherche d'Information).

1. Un zetta-octet est 10 à la puissance 21, soit 10 suivi de 20 zéros

Contexte du travail

La RI est une branche en informatique qui s'intéresse à l'acquisition, l'organisation, le stockage et la recherche des informations. Elle regroupe l'ensemble de procédures et techniques permettant de sélectionner à partir d'une collection de documents, les informations (documents ou portions de documents) pertinentes répondant à des besoins utilisateurs, exprimés à travers des requêtes.

La RI remonte à peu de temps après l'arrivée des premiers ordinateurs, et constitue l'une de plus anciennes applications de l'informatique à l'accès aux documents électroniques. À cette époque, en 1955, la plus remarquable réalisation est le WRU [2] Searching Selector, de James W. Perry *et al.* [167]. C'est une machine qui pouvait déjà résoudre jusqu'à 10 requêtes booléennes en un seul passage sur une bande magnétique. Les premiers systèmes de recherche d'information (SRI) utilisés par des libraires sont fondés sur des modèles de recherche booléens. Le développement du système SMART par Salton [197] à la fin des années 1960, et qui utilise le modèle vectoriel, a conduit à des développements novateurs.

Près de soixante ans plus tard, et une vingtaine d'années après la révolution d'Internet et ses milliards de pages accessibles sur la Toile, la RI est plus que jamais d'actualité. En effet, la banalisation de l'informatique grand public et l'accès quasi universel à Internet ont induit une énorme demande des utilisateurs vers une meilleure accessibilité aux seules données qui les intéressent : langue naturelle parlée ou écrite, images, musique, animations [55]. Cette explosion de ressources d'information et leur hétérogénéité a ramené à de nouveaux problèmes à la RI :

- Évolution des documents : collection gigantesque, dynamique et changeante, surabondance de l'information, documents structurées ou semi-structurés, documents multimédias, données réparties, multilinguisme, etc.
- Évolution des besoins : une seule requête puise désormais dans différentes sources simultanément : web, images, cartes, actualités, blogs, livres. L'ambigüité des requêtes des utilisateurs, la diversité de leurs besoins en information et de leurs situations de recherche, etc.

Ces problèmes ont remis en cause les modèles classiques de RI. En effet, les méthodes classiques d'indexation et de recherche en RI, davantage destinées aux données textuelles, ne sont pas directement applicables à ces nouveaux documents, en particulier les documents semi-structurés de type XML. En effet, la RI dans les documents semi-structurés se caractérisent par la forme

2. WRU : Western Reserve University, Cleveland (US). Voir en particulier la référence Web http ://www.libsci.sc.edu/Bob/ISP/cwru.htm

des requêtes, elles peuvent être sous forme de mots-clés et/ou de contraintes structurelles et/ou de contenu multimédia et la forme de l'unité d'information renvoyée en réponse à ces requêtes. Ces unités sont des parties du document répondant d'une manière exhaustive et spécifique à la requête.

Ces unités sont souvent renvoyées sous forme d'une liste ordonnée : chaque unité est censée répondre totalement à la requête. Or un élément peut en effet, répondre souvent partiellement à une requête. Une réponse idéale serait par exemple l'agrégation d'un élément X avec un élément Y unis d'ailleurs d'un même document que de documents différents. Nos travaux se situent précisément à la conjonction de la RIS (Recherche d'Information Structurée) et la RI agrégée. L'objectif des systèmes de RIS n'est plus de renvoyer le document entier répondant à la requête, mais plutôt l'unité d'information (ou élément XML) répondant le mieux à la requête. Pour répondre à ce challenge, plusieurs modèles de recherche ont été proposés dans la littérature (cf. chapitre 2, section 2.6). Quant à la RI agrégée, son objectif cherche à assembler des éléments provenant de sources différentes : images, vidéos (dont YouTube), livres numérisés (Google Livres), cartes (Google Maps), actualités (Google News), etc.

Nous nous intéressons dans nos travaux à l'application du paradigme de la RI agrégée en RIS afin de satisfaire l'utilisateur en lui renvoyant les meilleurs ensembles d'unités d'informations répondant à son besoin.

Problématique

La plupart des approches en RIS [202, 160, 127, 128, 177] considère que les unités d'information retournées sont sous forme d'une liste d'éléments disjoints. Ces éléments peuvent être pertinents, non pertinents ou partiellement pertinents. Le défi à relever est alors d'arriver à sélectionner automatiquement les éléments répondant à la fois de manière exhaustive et spécifique [168] à la requête de l'utilisateur.

Nous nous intéressons au problème d'agrégation d'éléments XML. Nous pensons qu'il existe des requêtes pour lesquelles, il est nécessaire d'agréger des éléments d'un même document pour former la réponse la plus complète en terme de pertinence. L'idée derrière la sélection d'un ensemble d'éléments au lieu d'un élément tout seul vient du fait que nous croyons qu'un élément pourrait être partiellement pertinents pour une requête, alors que si nous regroupons ces éléments ensembles, nous pourrons alors produire une meilleure réponse à l'utilisateur.

Les travaux décrits dans cette thèse s'intéressent à la sélection de l'agrégat

(ensemble d'éléments) qui répond le mieux à une requête composée de simple mots-clés (requêtes de type CO (Content Only)).

La question de l'agrégation des éléments XML a reçu peu d'attention dans la littérature. La première tentative proposée permettant de répondre à cette problématique est celle proposée par Bessai et Alimazighi [29].

L'émergence de la RI agrégée a permis non seulement de réviser l'accès à l'information mais aussi de remettre en cause le paradigme d'évaluation classique des systèmes de RIS. Plusieurs questions se posent dans ce contexte, elles portent en général sur la manière de :

- agréger les éléments potentiellement pertinents ;
- élaguer ceux qui sont redondants ;
- regrouper ceux qui se complètent ;
- évaluer le résultat d'une recherche ;
- prendre en compte l'information structurelle.

Dans le cadre de cette thèse, nous souhaitons mieux explorer l'impact de l'agrégation de telles unités en RIS, en étudiant notamment l'intérêt d'utiliser des ensembles d'éléments à la place d'une simple liste et en évaluant nos propositions sur des collections de documents de type XML.

Contribution

Afin de répondre aux questions listées précédemment, nous avons proposé un mécanisme complet d'agrégation d'éléments XML partant de la sélection jusqu'au renvoi d'un ensemble d'éléments répondant à une requête de type CO.

Notre approche se situe à la jonction de la recherche d'éléments les plus pertinents à partir de documents XML et leur agrégation dans un même résultat. Notre objectif est d'assembler automatiquement des éléments pertinents, non-redondants et complémentaires qui répondent ensemble le mieux au besoin de l'utilisateur formulé à travers une liste des mots-clés. Le modèle que nous proposons trouve ses fondements théoriques dans les RB (Réseaux Bayésiens). La structure réseau fournit une manière naturelle de représenter les liens entre les éléments du corpus de documents XML et leurs contenus. Quant à la théorie des probabilités, elle permet d'estimer de manière qualitative et quantitative les différents liens sous-jacents. Elle permet notamment d'exprimer le fait qu'un terme est probablement pertinent vis-à-vis d'un élément et de mesurer à quel point une réponse à la requête contient un ensemble d'éléments pertinents, non-redondants et complémentaires.

Plus précisément, au niveau de la pertinence d'éléments dans un résultat de recherche, nous estimons que la pertinence d'un agrégat en fonction d'un terme dépend non seulement de sa pertinence dans chaque élément de l'agrégat en question mais aussi de sa pertinence dans la collection afin d'éviter le problème des fréquences nulles des quelques termes.

Au niveau de l'élimination d'éléments redondants, nous avons, tout d'abord, proposé une contrainte de structure qui nous permet d'enlever les éléments qui se chevauchent. Cette contrainte d'inclusion a pour objectif de ramener dans un agrégat, les éléments qui n'ont pas une relation de parenté (ou ancêtre-descendant). Nous avons ensuite proposé une deuxième contrainte de contenu qui nous permet d'avoir dans un agrégat uniquement les éléments dissimilaires. Cette contrainte de similarité a pour objectif de renvoyer dans un agrégat les éléments qui ne sont pas semblables. Pour cela, nous avons proposé un algorithme pour fixer le seuil similarité entre les éléments redondants.

Nous avons également proposé au niveau de la complémentarité entre les éléments d'un agrégat une fonction de propagation qui favorise les éléments les plus loin de nœud racine. En effet, les éléments loin du nœud racine d'un document paraissent plus porteurs d'informations complémentaires que ceux situés plus haut dans le document. L'objectif ici est de favoriser les éléments qui se complètent mutuellement pour avoir une réponse plus complète.

Enfin, toutes nos propositions ont été évaluées sur des collections standards issues de la campagne d'évaluation INEX[3] 2009. Nous proposons également d'appliquer notre approche approche en deux modes :

- dans le premier mode, l'utilisateur n'intervient pas dans le jugement des éléments pertinents. Ce mode est utilisé pour évaluer les résultats enregistrés dans le cadre de la campagne INEX 2009 selon la stratégie *Focused* ;
- dans le deuxième mode, l'utilisateur intervient dans le jugement de la pertinence d'agrégats. Ce mode est basé sur des contextes réels d'évaluation de la redondance et la complémentarité entre les éléments du top-1 agrégat, et l'utilité de la RI agrégée contre la RIS.

Les résultats montrent l'intérêt de l'approche proposée. La combinaison des deux sources d'évidence, la structure et le contenu, permet également d'améliorer les performances de manière significative.

3. INEX : INitiative for the Evaluation of XML Retrieval. Voir http ://inex.is.informatik.uniduisburg

Organisation de la thèse

Ce mémoire de thèse est constitué de la présente introduction générale, des deux parties principales et d'une conclusion générale. La première partie présente le contexte général dans lequel se situe notre travail, à savoir la recherche d'information structurée et plus précisément la recherche agrégée dans des documents semi-structurés ; la seconde partie détaille notre contribution dans le domaine. La conclusion générale présente les principales conclusions ainsi que les perspectives de nos travaux.

L'objectif de la première partie est de porter la lumière sur le domaine de la recherche d'information structurée, puis son application pour embrasser la RI agrégée. La première partie regroupe trois chapitres.

Le chapitre 1, "**La Recherche d'Information classique**", présente les notions et concepts de base de la RI. Nous présentons brièvement les fondements de la RI classique. Ensuite, nous décrivons les principaux modèles de RI. Enfin, nous présentons les protocoles d'évaluation d'un SRI.

Le chapitre 2, "**La Recherche d'Information Structurée**", traite les enjeux de la RIS. Nous discutons la différence entre les approches orientées base de données et approches orientées recherche d'information. Nous présentons les différentes approches d'indexation et d'interrogation développées dans ce cadre. Nous décrivons ensuite les différents modèles de recherche proposés dans la littérature. Enfin, nous abordons les protocoles d'évaluation des systèmes de RIS.

Le chapitre 3,"**Vers la Recherche d'Information agrégée dans des documents semi-structurés**", présente les différentes approches en RI agrégée ainsi que les cadres d'évaluation associés. Nous présentons les limites des paradigmes recherche booléenne et recherche ordonnée. Nous décrivons ensuite les motivations vers la RI agrégée ainsi que ses différents domaines d'applications et les problèmes soulevés. Nous décrivons un état de l'art de la RI structurée et la RI agrégée. Enfin, nous présentons des modèles d'évaluation en RI agrégée, notamment l'évaluation des documents XML.

La deuxième partie détaille notre contribution dans le domaine de la RI agrégée dans des documents XML. Elle comprend deux chapitres.

Le chapitre 4, "**Un Modèle de Recherche d'Information agrégée basé sur les Réseaux Bayésiens**", présente notre approche d'agrégation des éléments XML ainsi qu'une évaluation expérimentale de cette approche. Nous présentons le cadre théorique sur lequel repose notre modèle, à savoir les RB. Nous détaillons ensuite le modèle que nous proposons. Enfin, nous illustrons le

modèle proposé à l'aide d'un exemple.

Le chapitre 5, "**Expérimentations**", présente les résultats des expérimentations que nous avons évalué. Ce chapitre présente une première évaluation expérimentale comparative entre notre résultat et les dix meilleurs résultats enregistrés par les participants à la collection de test INEX 2009 selon la stratégie de recherche *Focused*. Ce chapitre présente également une deuxième évaluation expérimentale comparative entre la RI agrégée dans des documents XML et la RI structurée.

En conclusion, nous dressons le bilan de nos travaux réalisés dans le cadre de la RI agrégée dans des documents XML. Nous introduisons ensuite les perspectives liées à nos travaux réalisés ainsi que les cadres d'évaluation appropriés.

Première partie

Recherche d'Information agrégée dans les documents semi-structurés : Aperçu sur les modèles et les cadres d'évaluation

Chapitre 1

La Recherche d'Information classique

1.1 Introduction

La RI (Recherche d'Information) est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information. Ce besoin en information est souvent formulé en langage naturel par une requête décrite par un ensemble de mots-clés. L'objectif de tout Système de RI (SRI) est alors de retrouver dans une collection de documents ceux qui sont susceptibles d'être pertinents à une requête. Un SRI peut être défini alors comme l'ensemble des programmes et des opérations permettant la gestion, la représentation, l'interrogation, la recherche, le stockage et la sélection des informations répondants à une requête [196]. L'interrogation de la collection de documents à l'aide d'une requête exige un appariement entre cette dernière et les documents. Ces documents sont souvent considérés comme des documents textuels (plats).

Ce chapitre a pour objectif de présenter les concepts de base de la RI classique. La section 1.2 présente tout d'abord les fondements de la RI classique. La section 1.3 décrit trois modèles connu en RI, à savoir le modèle booléen, le modèle vectoriel et le modèle probabiliste. La section 1.4 donne un aperçu sur les collections de test ainsi que les principales mesures d'évaluation utilisées. La dernière section 1.5 conclut le chapitre.

1.2 Processus de RI classique

Un SRI (Système de Recherche d'Information) permet de sélectionner à partir d'une collection de documents, des informations pertinentes répondant à des besoins utilisateurs, exprimés sous forme de requêtes. Dans la suite de cette section, nous abordons les concepts de base de la RI ainsi que la description du processus général d'un SRI.

1.2.1 Notions de base

Plusieurs notions clés s'articulent autour de la définition d'un SRI :

- **Document :** on appelle document toute unité d'information qui peut constituer une réponse à un besoin en information d'un utilisateur. Un document peut être un texte, une portion de texte, une image, une bande vidéo, etc.
  L'ensemble de documents exploitables et accessibles s'appelle collection de documents (ou fonds documentaire, corpus).
- **Requête :** c'est une formulation du besoin d'information d'un utilisateur. Elle peut être vue comme une description sommaire des documents ciblés par la recherche. Divers types de langage d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots-clés, mais elle peut être exprimée en langage naturel, booléen ou graphique.
- **Pertinence :** une définition simple de cette notion fondamentale est donnée dans [38] : "*La pertinence est le degré de correspondance entre un document et une requête, ou encore une mesure d'informativité du document à la requête*". On trouve également d'autres définitions de la pertinence dans [194] telle que : "*La pertinence est un degré de relation entre le document et la requête*".
  La pertinence est indispensable pour l'évaluation des SRI. Cependant, de nombreuses études menées [26, 34] autour de la notion de pertinence, montrent que la pertinence n'est pas une relation isolée entre le document et la requête et qu'elle est définie par un ensemble de critères et de préférences qui varient selon les utilisateurs. Ces critères sont des facteurs qui déterminent la pertinence accordée à l'information retrouvée par l'utilisateur dans un contexte de recherche précis. Les facteurs qui affectent les jugements de pertinence font l'objet de recherche depuis déjà des décennies [66, 34, 26]. Nous citons les critères définis par [26] et regroupés dans sept catégories : (1) le contenu informationnel des documents ; (2) le niveau d'expertise et de connaissances de l'utilisateur ; (3) les croyances et préférences de l'utilisateur ; (4) autres informations liées à l'environnement ; (5) les sources des documents ; (6) les documents comme des entités physiques ; et (7) la situation de l'utilisateur.

Compte tenu de ces facteurs, il existe plusieurs types de "*pertinence*" possibles entre un document et un besoin, nous en citons les quatre les plus importantes [211] :

1. *pertinence algorithmique (ou système)* : c'est une mesure algorithmique basée sur le calcul de la pertinence de l'information par rapport à la requête en utilisant des caractéristiques des requêtes, d'une part, et des documents, d'autre part. Le but de tout SRI est de rapprocher la pertinence algorithmique calculée par le système aux jugements de pertinence donnés par des utilisateurs. C'est le seul type de pertinence qui est indépendant du contexte.
2. *pertinence thématique* : cette pertinence est définie par le degré de couverture de l'information retrouvée au thème évoqué par le sujet de la requête. C'est la mesure de pertinence utilisée par les assesseurs dans les campagnes d'évaluation TREC [1][225].
3. *pertinence coginitive* : c'est la pertinence liée au thème de la requête, selon la perception ou les connaissances de l'utilisateur sur ce même thème ; cette pertinence est caractérisée par une dynamique qui permet d'améliorer la connaissance de l'utilisateur via l'information renvoyée au cours de sa recherche.
4. *pertinence situationnelle (ou contextuelle)* : cette pertinence est définie par l'utilité de l'information jugée relativement au contexte ou à la situation de l'utilisateur. C'est une pertinence dynamique.

Il est à noter qu'un SRI idéal doit supporter un modèle de recherche d'information qui rapproche la pertinence algorithmique calculée par le système aux jugements de pertinence donnés par des utilisateurs.

1.2.2 Mise en œuvre d'un SRI

La mise en œuvre d'un SRI fait appel à plusieurs étapes représentées par ce que l'on nomme communément, le processus en U illustré par la figure 1.1. Ce processus consiste en deux principales phases : l'indexation et l'appariement.

- **Indexation :** cette phase consiste à extraire et représenter le contenu des documents à l'aide d'un ensemble de termes significatifs, auxquels sont associés des poids pour différencier leur degré de représentativité, sous forme d'index. Cette structure d'index permet de retrouver rapidement les documents contenant les termes (mots-clés) de la requête.
- **Appariement :** cette phase consiste à mesurer la pertinence de chaque document vis-à-vis de la requête utilisateur selon une mesure de correspondance du modèle de RI, et à renvoyer à l'utilisateur une liste ordonnée des résultats.

1. TREC : Text REtrieval Conference. Voir http ://trec.nist.gov/

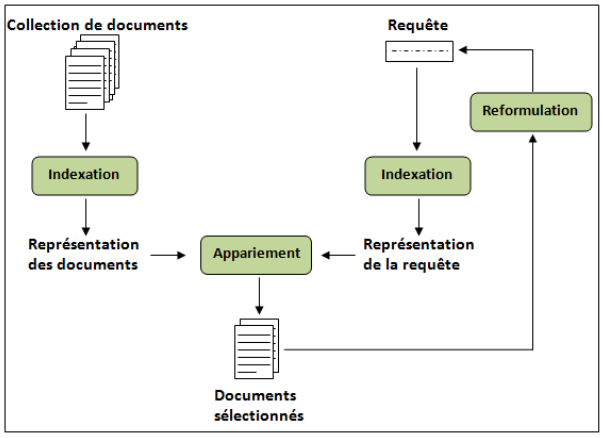

FIGURE 1.1 – Processus en U de la RI

1.2.3 Indexation

L'indexation couvre un ensemble de techniques visant à représenter le contenu des documents (ou requêtes) par une liste de termes significatifs, que l'on nomme : substituts ou descripteurs. Ces descripteurs forment le langage d'indexation. Dès lors, l'indexation consiste à détecter les termes les plus représentatifs du contenu du document.
En RI, différents modes d'indexation existent : l'indexation manuelle, automatique ou semi-automatique.

- *Indexation manuelle :* chaque document est analysé par un spécialiste du domaine (ou documentaliste) qui choisit les termes qu'il juge pertinents dans la description du contenu sémantique du document. Ce type d'indexation est subjective, d'une part, car elle dépend des connaissances de l'opérateur et d'autre part, inapplicable pour une collection volumineuse.
- *Indexation automatique :* cette indexation repose sur des algorithmes associant automatiquement des descripteurs à des parties de document. Elle peut se faire selon une méthode linguistique ou statistique.
- *Indexation semi-automatique :* c'est une combinaison des deux méthodes précédentes : un premier processus automatique permet d'extraire les termes du document. Cependant, le choix final des descripteurs est laissé au documentaliste, qui utilise un vocabulaire contrôlé sous forme de thésaurus [2] ou de base terminologique.

D'une façon générale, un processus d'indexation automatique comprend un ensemble de traitements automatiques sur les documents : extraction de mots simples, élimination de mots vides, normalisation et pondération des mots.

2. Un thésaurus est une liste organisée de descripteurs (mots-clés) obéissant à des règles terminologiques propres et reliés entre eux par des relations sémantiques.

1. **Extraction de mots simples :**
   Cette étape consiste à extraire du document un ensemble de termes ou de mots simples par une analyse lexicale permettant d'identifier les termes en reconnaissant les espaces de séparation des mots, des caractères spéciaux, des chiffres, les ponctuations, etc.
2. **Élimination de mots vides :**
   La liste de mots simples extraite précédemment peut contenir de mots non significatifs, appelés "mots vides", tels que : les pronoms personnels, les prépositions ou même des mots athématiques qui peuvent se retrouver dans n'importe quel document (par exemple des mots comme contenir, appartenir, etc). L'élimination de ces mots peut se faire en utilisant une liste dressée de mots vides (également appelée anti-dictionnaire ou *stoplist*), ou en écartant les mots dépassant un certain nombre d'occurrences dans la collection. Bien que ce traitement présente l'avantage de diminuer le nombre de termes d'indexation, il peut cependant induire des effets de silence. Par exemple, en éliminant le mot "a" de "vitamine a".
3. **Normalisation (lemmatisation ou radicalisation) :**
   Cette étape consiste à réduire les mots à leur forme canonique, à leur racine : toutes les formes d'un verbe, par exemple, sont regroupées à l'infinitif, tous les mots au pluriel sont ramenés au singulier, etc. On distingue quatre principales méthodes de normalisation :
   – par analyse grammaticale en utilisant un dictionnaire (ex : Tree-tagger [3]) ;
   – par utilisation de règles de transformation de type condition action surtout pour l'anglais (ex : l'algorithme de Porter [179]) ;
   – par troncature des suffixes à X caractères (ex : la troncature à 7 caractères) ;
   – par la méthode des n-grammes utilisée pour le chinois et très intéressante pour la radicalisation.

   Il reste cependant à mentionner que ces traitements peuvent induire certains inconvénients tels que la production de normalisation agressive, par exemple, les mots university/universe, organization/organ, policy/police sont normalisés par l'algorithme de Porter, ou l'oubli de quelques normalisations intéressantes, par exemple : matrices/matrix, Europe/European, machine/machinery ne sont pas normalisés. Il existe des techniques d'analyse de corpus pour réduire ces effets négatifs [233, 43].
4. **Pondération des termes :**
   Cette étape est généralement basée sur des formules de pondération qui affecte à chaque terme un degré d'importance (une valeur de discrimination) dans le document où il apparaît. Il existe un grand nombre de formules de pondération qui exploitent deux facteurs : fréquence de terme (tf) et fréquence inverse de document (idf) [193], définies dans ce qui suit :

3. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

– **Fréquence de terme (tf) :**
La fréquence du terme (term frequency) est le nombre d'occurrences de ce terme dans le document considéré. L'idée sous-jacente est que plus un terme est fréquent dans ce document, plus il est important dans la description de celui-ci. Soient le document d_j et le terme t_i, la fréquence tf_{ij} du terme dans le document est souvent utilisée directement ou exprimée selon l'une des déclinaisons suivantes [146] :

$$tf_{ij} = 1 + log(\#td_{ij}), tf_{ij} = \frac{\#td_{ij}}{\sum_k \#td_{kj}} \tag{1.1}$$

où $\#td_{ij}$ est le nombre d'occurrences du terme t_i dans d_j. Le dénominateur est le nombre d'occurrences de tous les termes dans le document d_j. La dernière déclinaison permet de normaliser la fréquence du terme pour éviter les biais liés à la longueur du document.

– **Fréquence inverse de document (idf) :**
La fréquence inverse de document (*inverse document frequency*) est une mesure de l'importance du terme dans l'ensemble du corpus. Elle consiste à calculer le logarithme de l'inverse de la proportion de documents du corpus qui contiennent le terme. Cette mesure est exprimée selon l'une des déclinaisons suivantes [146] :

$$idf_i = log\frac{|N|}{n}, idf_i = log\frac{|N-n|}{n} \tag{1.2}$$

où n est la proportion des documents contenant le terme et N le nombre total de documents dans collection.

La fonction de pondération de la forme $tf - idf$ consiste à multiplier les deux mesures tf et idf comme suit :

$$tf * idf = log(1 + tf) * log\frac{|N|}{n} \tag{1.3}$$

1.2.4 Appariement

La phase d'appariement du système implique un processus d'interaction de l'utilisateur avec le SRI illustré dans la figure 1.1. Cette interaction implique le scénario suivant : l'utilisateur exprime son besoin en information sous la forme d'une requête. Le système interprète la requête et crée son index qui sera compatible avec le modèle d'index des documents. Ensuite le système évalue la pertinence des documents par rapport à cette requête en utilisant une fonction de correspondance. Cette fonction exploite l'index généré dans la phase d'indexation dans le but de calculer un score de similarité (en anglais *Relevance Status Value*), notée $RSV(q, d)$, entre la requête indexée q et les descripteurs

du document d. Différents modèles de RI ont été proposés dans la littérature et tentent de formaliser la pertinence en partant des modèles naïfs basés sur l'appariement exact vers des modèles plus élaborés basés sur l'appariement rapproché [46].
Le résultat est une liste de documents triée par ordre de valeur de correspondance décroissante, et présenté à l'utilisateur. Celui ci apporte son jugement sur les documents renvoyés par le système selon des critères liés à son besoin en information et au contexte dans lequel la recherche est effectuée. Dans la suite, nous présentons les principaux modèles développés en RI.

1.3 Aperçu des principaux modèles de RI

Un modèle de RI se définit par une formalisation du processus de RI et une modélisation de la mesure de pertinence. Selon Baeza-Yates et Ribeiro-Neto [23], un modèle de RI est défini formellement par par un quadruplet (D, Q, F, $R(q_i, d_j)$), où :
- D est l'ensemble de documents ;
- Q est l'ensemble de requêtes ;
- F est le schéma du modèle théorique de représentation des documents et requêtes ;
- $R(q_i, d_j)$ est la fonction de pertinence du document d_j à la requête q_i.

Nous présentons dans la suite les principaux modèles de RI : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

1.3.1 Modèle booléen

Le modèle booléen [190] est le premier modèle de RI, et est basé sur la théorie des ensembles. Dans ce modèle, un document est représenté par une liste de termes d'indexation. Ces termes sont reliés par des connecteurs logiques ET, OU et NON. Un exemple de représentation d'un document est comme suit : $d_j = t_1 \wedge t_2 \wedge t_3 ... \wedge t_n$.

Une requête est une expression booléenne dont les termes sont reliés par des opérateurs logiques (OR, AND, AND NOT) permettant d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. Un exemple de représentation d'une requête est comme suit : $q_i = (t_1 \wedge t_2) \vee (t_3 \wedge t_4)$.

La fonction de correspondance est basée sur l'hypothèse de présence/absence

des termes de la requête dans le document et vérifie si l'index de chaque document d_j implique l'expression logique de la requête q_i. Le résultat de cette fonction est donc binaire. Cette fonction est décrite comme suit : $RSV(q_i, d_j) = \{1, 0\}$. Cette décision binaire sur laquelle est basée la sélection d'un document ne permet pas d'ordonner les documents renvoyés à l'utilisateur selon un degré de pertinence parce que les termes ne sont pas pondérés.

1.3.2 Modèle vectoriel

Initialement proposé par Salton et implémenté dans le système SMART [191], dans ce modèle la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel. Le modèle vectoriel préconise la représentation des requêtes utilisateurs et des documents sous forme de vecteurs, dans l'espace engendré (à n dimensions) par tous les termes d'indexation [191]. Les dimensions sont constituées par les termes du vocabulaire d'indexation. Chaque document est représenté par le vecteur $\vec{d_j}$=$(w_{1,j}, w_{2,j}, w_{3,j}, ..., w_{n,j})$. Chaque requête est également représentée par un vecteur $\vec{q_i} = (w_{1,i}, w_{2,i}, w_{3,i}, ..., w_{n,i})$. Avec $w_{k,j}$ (resp. $w_{k,i}$) est le poids du terme t_k dans le document d_j (resp. dans la requête q_i). La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs :

$$RSV(q_i, d_j) = cos(\vec{q_i}, \vec{d_j}) \tag{1.4}$$

Plus deux vecteurs sont similaires, plus l'angle formé est petit, et plus le cosinus de cet angle est grand. À l'inverse du modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui permet de retrouver des documents qui ne satisfont la requête qu'approximativement. Les résultats peuvent donc être ordonnés par ordre de pertinence décroissante.
Le modèle vectoriel suppose l'indépendance entre termes. En effet, la représentation vectorielle considère chaque terme séparément alors qu'on peut avoir des termes qui sont en relation sémantique entre eux.

1.3.3 Modèle probabiliste

Le modèle probabiliste a été développé dans les années 70, et sa fonction de pertinence se base sur le calcul de la probabilité de pertinence d'un document vis-à-vis d'une requête [183, 147]. Le principe de base consiste à retrouver des documents qui ont en même temps une forte probabilité d'être pertinents, et

une faible probabilité d'être non pertinents. Étant donné une requête utilisateur q_i et un document d_j, il s'agit de calculer la probabilité de pertinence du document pour cette requête. Deux événements se présentent : R, d_j est pertinent pour q_i et $\overline{R}$, d_j n'est pas pertinent pour q_i.
Le score d'appariement entre le document D et la requête Q, noté $RSV(Q, D)$, revient à calculer le rapport entre la probabilité de pertinence d'un document et sa probabilité de non pertinence. Ce score est donné par :

$$RSV(q_i, d_j) = \frac{P(R|d_j)}{P(\overline{R}|d_j)} \tag{1.5}$$

En utilisant la règle de Bayes après simplification, cela vient à ordonner les documents selon :

$$RSV(q_i, d_j) = \frac{P(d_j|R)}{P(d_j|\overline{R})} \tag{1.6}$$

Pour estimer les probabilités $P(d_j|R)$ et $P(d_j|\overline{R})$, un document sera décomposé en un ensemble d'événements $d_j(t_1, t_2, ..., t_N)$. Chaque événement dénotera la présence ou l'absence d'un terme t_i dans un document d_j. En supposant l'indépendance des termes des documents, la formule précédente devient :

$$RSV(q_i, d_j) = \sum_{i=1}^{N} log \frac{P(w_{ij}|R)}{P(w_{ij}|\overline{R})} \tag{1.7}$$

où w_{ij} indique la présence ou l'absence terme t_i dans le document d_j. Après transformation, l'équation 1.7 s'écrit :

$$RSV(q_i, d_j) = \sum_{i=1, t_i \in q}^{N} log \frac{P(w_{ij} = 1|R)P(w_{ij} = 0|\overline{R})}{P(w_{ij} = 1|\overline{R})P(w_{ij} = 0|R)} \tag{1.8}$$

Un des inconvénients de ce modèle réside dans la représentation du document. En effet, ce modèle ne prend pas en compte les fréquences des termes dans le document. Pour pallier cet inconvénient, Robertson *et al.* [185, 227] a proposé le modèle 2-Poisson basé notamment sur la notion de termes élites qui intègre différents aspects relatifs à la fréquence locale des termes, leur rareté et la longueur des documents [183]. Ceci a donné lieu à la formule BM25 :

$$w_{ij} = log(\frac{N - df + 0.5}{df + 0.5}) \times \frac{(k_1 + 1) * tf}{k_1 * ((1 - b) + b * \frac{dl}{avgdl}) + tf} \tag{1.9}$$

Avec :

- dl est la longueur du document d_j ;
- $avgdl$ est la longueur moyenne des documents dans la collection ;
- k_1 et b sont des paramètres qui dépendent de la collection ainsi que du type de la requête.

Les expérimentations ont montré que les paramètres $k_1 = 1,2$ et $b = 0,75$ ont donné les meilleurs résultats, en termes de performances, sur les collections TREC considérées.
Les modèles probabilistes comprennent également le modèle bayésien ou d'inférence [220] et le modèle de langue [178, 126].

1.4 Évaluation des performances des systèmes de RI

La validation expérimentale des SRI consiste à mesurer ses performances par comparaison de ses résultats retournées à l'aide des métriques standards à l'aide des collections de test contrôlées.

Le premier paradigme qui constitue le cadre de référence dans lequel s'inscrivent les expérimentations et la validation des SRI, se base sur une approche de type laboratoire *(laboratory-based model)*, appelé paradigme de *Cranfield*, initié par Cleverdon [60] dans le cadre du projet *Cranfield Project II*. Dans cette approche, on parle d'évaluation qualitative, car l'idée de base est de comparer, pour une requête donnée, les documents retrouvés par le système dans la collection de test, aux réponses idéales établies pour cette requête dans la collection de test, réponses qui ont été identifiées manuellement par des documentalistes (experts du domaine). Il s'agit donc bien de comparer une notion de pertinence système à une notion de pertinence utilisateur.
Cette approche est souvent adoptée dans les campagnes d'évaluation des SRI tells que TREC, INEX, CLEF [4], etc.

1.4.1 Collections de test

Généralement, chaque collection de test est composée : d'une collection de documents, aussi appelée corpus de documents, d'une liste de requêtes et des jugements de pertinence des documents par rapport à ces requêtes.

- **Collection de documents :** c'est un corpus de documents sur lesquels les SRI posent des requêtes et récupèrent les documents pertinents.
  Le choix d'une collection dépend de la tâche de recherche que l'on veut évaluer, pour garantir une représentativité par rapport à la tâche. De même que la spécification du volume des collections de documents utilisées dans l'évaluation est relativement dépendante de la tâche de recherche impliquée dans le SRI à évaluer, pour garantir une diversité des

4. CLEF : Cross Language Evaluation Forum. Voir http ://clef.iei.pi.cnr.it/

sujets et du vocabulaire. Les premiers corpus de test développés au début des années 1960 renferment quelques milliers de documents. Les corpus de test plus récents (par exemple, ceux d'INEX et de TREC) contiennent en général des millions de documents. Le travail concernant la sélection des documents des corpus est d'ailleurs très déterminant et fait l'objet de nombreuses recherches [86].

- **Requêtes :** ce sont souvent présentées sous forme de "topics" qui expriment un besoin d'information de l'utilisateur. Pour exploiter au mieux les caractéristiques de la collection de documents et avoir une évaluation assez objective, il est important de créer un ensemble de requêtes qui correspondent aux thèmes abordés dans les documents. Les requêtes doivent d'abord être extraites de log et ensuite, si ce n'est pas possible de les créer artificiellement par les assesseurs.
- **Jugements de pertinence :** pour la construction d'un corpus de test, les jugements de pertinence constituent la tâche la plus ardue. Les jugements de pertinence indiquent pour chaque document du corpus s'il est pertinent, et parfois même à quel degré il l'est, pour chaque requête. Pour établir ces listes de documents pour toutes les requêtes, les utilisateurs doivent examiner chaque document de la base de document, et juger s'il est pertinent par rapport à une requête donnée. Dans les programmes d'évaluation tels que TREC, les collections de documents contiennent plus d'un million de documents, ce qui rend impossible le jugement exhaustif de pertinence. Ainsi, dans le cas de grandes collections, les jugements de pertinence sont construits selon la technique de *pooling*, effectuée à partir des 100 premiers documents retrouvés par les systèmes participants.

Les campagnes d'évaluation ont apporté plusieurs évolutions importantes. La première évolution réside dans la taille des collections, qui se veut la plus réaliste possible par rapport aux contextes réels de la RI ; on vise ainsi des collections de plusieurs centaines de milliers à plusieurs millions de documents, construites de manière collaborative par les participants aux campagnes. La seconde évolution est l'organisation de programmes d'expérimentation : les collections sont établies en vue d'expérimentations particulières (par exemple la RI multilingue, le Web, Question-Réponse, etc.). La dernière concerne dans l'aspect compétitif des expérimentations à INEX : les participants testent leur système au cours des mêmes campagnes, et les résultats comparatifs sont présentés dans des conférences spécifiques. Ainsi se perpétue, et même se renforce, la tradition d'expérimentation de la RI [55].

1.4.2 Protocole d'évaluation

Le protocole d'évaluation dans le modèle d'évaluation orienté-laboratoire définit une méthodologie rigoureuse et efficace pour comparer plusieurs SRI,

stratégies de recherche, ou algorithmes sur une même base, en spécifiant trois composants non indépendants qui sont : le nombre de topics utilisés, les mesures d'évaluation utilisées et la différence de performance requise pour considérer qu'une stratégie de recherche est meilleure qu'une autre [44].
L'évaluation de l'efficacité de chaque stratégie de recherche consiste à évaluer la liste des résultats obtenus pour chaque requête de test. Cette évaluation est à la base de la correspondance entre la pertinence algorithmique calculée par le système et la pertinence donnée par les assesseurs. L'efficacité globale d'une stratégie de recherche est calculée comme étant la moyenne des précisions calculées selon une mesure donnée sur l'ensemble des topics dans la collection de test.

Les protocoles d'évaluation se basent sur des mesures que nous présentons les principales dans le section suivante.

1.4.3 Mesures d'évaluation

Rappel et précision : Le rappel mesure la capacité d'un SRI à retrouver *tous* les documents pertinents à une requête et la précision mesure sa capacité à ne retrouver *que* ces documents pertinents.
Généralement les SRI retournent les documents classés par ordre décroissant de leur pertinence. Plusieurs travaux se sont penchés sur cette notion de pertinence [119, 40], affirmant la subjectivité, la gradualité de cette notion. L'efficacité d'un système mesure sa capacité à satisfaire l'utilisateur en terme de pertinence des documents restitués vis-à-vis d'une requête. Le tableau de contingence 1.1 permet de mesurer cette pertinence en fonction des documents restitués et non restitués.

	Pertinent	**Non pertinent**	
Restitués	$A \cap B$	$\bar{A} \cap B$	B
Non restitués	$A \cap \bar{B}$	$\bar{A} \cap \bar{B}$	$\bar{B}$
	A	$\bar{A}$	N

TABLE 1.1 – Tableau de contingence de la pertinence

Avec :
- A est l'ensemble des documents pertinents pour une requête Q ;
- B est l'ensemble des documents restitués par le système ;
- N est le nombre de documents de la collection ;
- $|.|$ désigne la cardinalité.

Selon le tableau de contingence 1.1, nous pouvons définir les mesures de **rappel (recall)** et de **précision (precision)** comme suit :

$$rappel = \frac{|A \cap B|}{A} \tag{1.10}$$

$$précision = \frac{|A \cap B|}{B} \tag{1.11}$$

Une façon d'évaluer un SRI est de tracer une courbe de précision-rappel. Ainsi, si le résultat de recherche dépend d'un certain paramètre, par exemple le rang d'un document restitué, alors pour chaque point de rappel, les valeurs de précision peuvent être calculées. Un SRI est parfait si et seulement si les documents retrouvés sont tous pertinents, avec une précision et un rappel de 100%. En pratique, ces deux taux varient en sens inverse, la précision diminue au fur et à mesure que le rappel augmente. La figure 1.2 illustre la forme générale de la courbe rappel-précision d'un SRI.

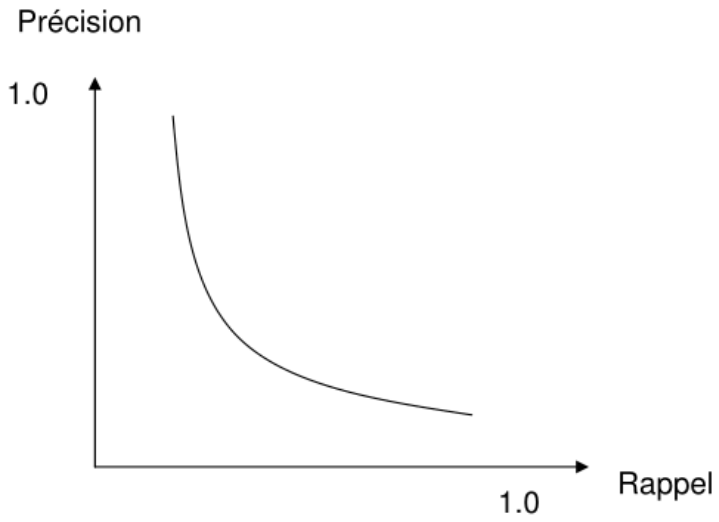

FIGURE 1.2 – Forme générale de la courbe rappel-précision d'un SRI

Comparaison entre SRI : pour comparer deux systèmes de RI, il faut les tester avec la même collection de test (ou plusieurs collection de test). Un système dont la courbe de rappel/précision est au-dessus de celle d'un autre est considéré comme un meilleur système.
D'autres mesures ont été proposées telles que :

- *Precision@X (P@X) :* cette précision mesure la proportion des documents pertinents retrouvés parmi les premiers X documents retournés par le système. Elle permet en particulier de s'intéresser à la haute précision, lorsque peu de documents sont restitués.

- *R-precision (RPrec) :* précision après que R documents ont été retrouvés, où R est le nombre de documents pertinents pour la requête considérée. Cette mesure a été introduite dans TREC2 pour limiter l'influence du nombre de documents pertinents : ce nombre varie en fonction des requêtes.
- *Précision moyenne interpolée (MAiP)* : cette précision est calculée à différents niveaux de rappel (0%, 10%, 20%, ...,100%). Pour chaque niveau de rappel, les valeurs calculées sont moyennées sur tout l'ensemble des requêtes. La *MAiP* est calculée comme suit :

$$MAiP = \frac{\sum_{q \in Q} AiP_q}{|Q|} \tag{1.12}$$

Avec :
- AiP_q est la précision interpolée moyenne d'une requête q ;
- Q est l'ensemble des requêtes ;
- $|Q|$ est le nombre de requêtes.

Dans [154], S. Mizzaro a fait une étude complète des différentes mesures d'évaluation utilisées en RI. Ceci a permis de dégager d'autres mesures de performance relativement importantes telles que :

- *F-mesure (ou F-score)* : la moyenne harmonique F-mesure qui consiste à combiner le rappel et la précision en un nombre compris entre 0 et 1 [182]. Cette moyenne harmonique a des valeurs élevées uniquement lorsque les taux de rappel et de précision sont élevés.

$$F = \frac{2 * précision * rappel}{précision + rappel} \tag{1.13}$$

Dans le cas de collections volumineuses, la construction de jugements de pertinence complets est difficile ou même impossible puisque elle est très coûteuse en terme de temps. Dans la mesure MAP, les documents non jugés sont considérés comme des documents non pertinents. Afin de pallier cet inconvénient, Buckley et Voorhees ont proposé la mesure *BPREF* [44] (Binary PREFerence-based measure).

- *BPREF* : cette mesure se focalise sur les documents réellement jugés et elle prend en compte les documents pertinents et les documents non pertinents afin de réduire l'effet du jugement de pertinence qui n'est réalisé que sur certains documents. La mesure BPREF est donnée par la formule suivante :

$$BPREF = \frac{1}{R} \sum_{r} 1 - \frac{|n|}{R} \tag{1.14}$$

Avec :
- R le nombre de documents pertinents pour la requête ;
- r est un document pertinent ;
- n est le nombre de documents non pertinents classés avant le document pertinent r.

– *Mean Reciprocal Rank (MRR)* : c'est une autre mesure, proposée par Voorhees [224], qui permet d'évaluer le nombre de documents qu'il faut considérer avant de retrouver le premier document pertinent. Elle est égale à la moyenne calculée sur l'ensemble des requêtes, du rang du premier document pertinent.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{1.15}$$

MRR est nulle pour une requête si aucun document pertinent n'est retourné par le système. Cependant, MRR donne un score élevé pour un système qui retourne des documents pertinents en haut de la liste présentée à l'utilisateur. Cette mesure est couramment utilisée dans les systèmes *Question-Réponse* où l'utilisateur s'intéresse à recevoir la bonne réponse en premier rang.

1.5 Conclusion

Nous avons présenté dans cette première partie le processus de la RI dans le cadre de la RI classique, les concepts de base ainsi que le fonctionnement global de tout SRI. Nous avons aussi décrit les modèles les plus connus de la RI ainsi que les techniques développées pour l'évaluation de tout SRI.

Ce type de SRI fonctionne sur des documents textuels plats. L'avènement des documents structurés, de type XML par exemple, a apporté une nouvelle problématique liée en particulier à la manière d'exploiter non seulement le contenu textuel de ces documents mais aussi l'information liée à la structure. Ceci amène l'utilisateur à affiner sa requête en intégrant des contraintes sur la structure de l'information recherchée. Nous présentons dans le chapitre suivant (2) les modèles traitant conjointement le contenu et la structure des documents structurés.

Chapitre 2

La Recherche d'Information Structurée

2.1 Introduction

Le développement du document électronique et du Web ont conduit à l'émergence des formats de données structurées, tels que SGML [1], HTML [2] et XML [3], permettant de représenter les documents sous une forme plus riche que le simple contenu [226]. À l'aide de ces formats, l'information textuelle et l'information structurelle sont représentées conjointement dans un document. Des modèles de RI intégrant cette relation formelle entre structure et contenu sémantique d'un document ont été développés.

En particulier, les documents semi-structurés ont donné naissance à une nouvelle thématique de la RI : la RIS (Recherche d'Information Structurée). Bien qu'elle présente de nouvelles problématiques spécifiques, la RIS s'appui fortement sur des approches déjà développées en RI. Dans le contexte de la RI dans les documents semi-structurés, appelée également RIS, la question majeure soulevée par ce type de document concerne la manière de prendre en compte efficacement de l'information du contenu et de structure pour mieux répondre aux besoins de l'utilisateur. Ces besoins peuvent être formulés par le biais de requêtes formées que de mots clé ou par des requêtes comportant des mots-clés et des contraintes structurelles (des balises).

Les systèmes d'accès aux documents structurés sont confrontés à des nouveaux problèmes dans toutes les étapes du processus de recherche à savoir :

1. SGML : Standard Generalized Markup Language
2. HTML : HyperText Markup Language
3. XML : eXtensible Markup Language

- **Indexation :** Faut-il adapter l'indexation classique afin de prendre en considération la structure des documents ? Comment indexer le contenu par rapport au structure ? Comment pondérer les termes en tenant compte de la structure ?
- **Appariement :** Quelle unité d'information faut-il sélectionner ? En effet, les techniques classiques de RI (plein texte) considèrent souvent le document entier comme un granule d'information indivisible, or dans le cas des documents XML tout élément (sous-arbre d'un document XML) peut être une réponse potentielle à la requête de l'utilisateur. Le défi à relever est alors d'arriver à identifier automatiquement l'unité d'information, en l'occurrence les parties du document XML, répondant à la fois de manière exhaustive et spécifique [168] à la requête de l'utilisateur. Ceci a conduit à l'élaboration de langages de requêtes spécifiques et à de nouveaux modèles de recherche.

Ce chapitre traite les enjeux de la RIS. Nous abordons dans la section 2.2 les différents problèmes soulevés par la RI. Dans la section 2.3, nous discutons la différence entre les approches orientées base de données et approches orientées recherche d'information. Nous présentons respectivement dans les sections 2.4 et 2.5 les différentes approches d'indexation et d'interrogation développées dans ce cadre. Nous décrivons ensuite les différents modèles de recherche proposés dans la littérature dans la section 2.6. Ces modèles de recherche visent à répondre à des requêtes basées sur le contenu seul ou à des requêtes basées sur le contenu et la structure. Dans la section 2.7, nous mettons l'accent sur les techniques d'évaluation des systèmes de RIS où nous abordons la campagne d'évaluation INEX ainsi que les différentes mesures dédiées à l'évaluation des approches et des systèmes dans le cadre de la RIS. La section 2.8 conclut le chapitre.

2.2 Enjeux de la RIS

Avant d'aborder les approches de la RIS, nous présentons brièvement les enjeux de la RIS en termes unité d'information retournée et son expression de besoin.

2.2.1 Granularité de l'information recherchée

En RI classique, les SRI renvoient des documents entiers comme réponse à une requête utilisateur. Cette granularité "document" ne satisfait pas toujours l'utilisateur vu que ce granule peut contenir du bruit, ou bien l'information

pertinente peut être dispersée sur tout le document. Il serait plus intéressant de ne retourner que la partie du document qui semble pertinente vis-à-vis de la requête. Ces hypothèses ont été largement étudiées dans la recherche de passages en RI classique (*passage retrieval*) [192].

Les documents semi-structurés contiennent outre le contenu textuel, de l'information structurelle permettant ainsi de traiter l'information avec une granularité plus fine. Le but de la RIS est alors d'identifier de manière automatique les unités de documents les plus pertinentes. Ceci a nous amène à affiner le concept de *granule* renvoyé à l'utilisateur. Une granule est une unité d'information auto-explicatif, c'est-à-dire l'information contenue ne dépend pas d'une autre unité d'information pour être comprise [97]. Généralement, l'objectif d'un SRI, dans ce contexte, est de renvoyer des unités d'information auto-explicatives à l'utilisateur, et non des points d'entrée dans les documents.

Dans le contexte de la RIS dans des documents XML, l'unité d'information correspond à un nœud de l'arbre du document (ou un sous-arbre) appelé aussi *élément*[4]. La pertinence d'un élément, réponse à une requête, peut être évalué selon deux dimensions : *exhaustivité* et *spécificité* [88]. *On dit qu'une unité d'information est exhaustive à une requête si elle contient toutes les informations requises par la requête et qu'elle est spécifique si tout son contenu concerne la requête* [81]. De ce fait, un système de RIS devrait retrouver l'unité d'information la plus exhaustive et la plus spécifique répondant à une requête.

2.2.2 Expression du besoin en information

De part leur structure, l'utilisateur interroge les collections de documents XML selon deux types de requêtes :

- **Requêtes de type CO (Content Only) :** ces requêtes sont composées de simples mots-clés, et le SRI détermine la granularité de l'information à renvoyer.
- **Requêtes de type CAS (Content And Structure) :** ces requêtes portent sur la structure et le contenu des unités d'information, dans lesquelles l'utilisateur spécifie des besoins précis sur certains éléments de structure. Dans ce type de requêtes, l'utilisateur peut utiliser des conditions de structure pour indiquer le type des éléments qu'il désire voir renvoyer.

Afin de pouvoir effectuer une recherche d'information qui tient compte de la structure logique des documents, des nouvelles techniques d'indexation et d'appariement ont été proposées. Ces techniques sont décrites dans les prochaines sections.

4. Nous utilisons dans la suite de ce rapport le terme élément pour décrire un sous-arbre d'un document XML.

2.3 Les approches de la RIS

Les approches proposées pour traiter spécifiquement la RIS peuvent être classées en deux principales catégories : (i) l'***approche orientée données*** (data-centric) utilise des techniques développées par la communauté des Bases de Données (BD), (ii) l'***approche orientée documents*** (document-centric) est prise en charge par la communauté RI. Le tableau 2.1 illustre les principes de chaque communauté pour le traitement des documents semi-structurés.

	RI	**BD**
Besoin en information	Vague	Précis
Résultat	Approché	Exact
Requête	CO ou CAS	SQL
Modèle	Modèles de RI (probabiliste,...)	Théorie des ensembles

TABLE 2.1 – RI vs. BD

2.3.1 Approches orientées documents

Les approches orientées documents considèrent les documents XML comme une collection de documents textes comportant des éléments et des relations entre ces éléments. Les éléments sont utilisés comme moyen pour mieux identifier la pertinence d'une unité de document vis-à-vis d'une autre unité. La majorité des travaux ont, en fait, adapté les modèles de RI reconnus pour traiter les documents XML [127, 95, 200, 177, 12, 160, 168].

2.3.2 Approches orientées données

Les approches orientées BD s'intéressent davantage à la structure du document. Plusieurs langages ont été définis [45], Lorel [11], XML-QL [135], XQL [54], XML-GL [52].

Ces approches permettent de traiter efficacement la structure des documents XML étant donné que les mots-clés sont examinés de façon binaire (présent/absent). Cependant, elles sont limitées pour le traitement de la partie textuelle des documents. Dans [195], Salton *et al.* ont démontré qu'en RI textuelle la prise en compte des poids des mots-clés dans un document est primordiale, voire nécessaire. Ceci permet de mesurer un degré de pertinence d'un document (ou d'une unité d'information) vis-à-vis d'une requête et donc

de renvoyer à l'utilisateur une liste triée de résultats, comme le proposent les approches de RI.

Nos travaux portent sur la RI et par conséquent, les problématiques examinées dans la suite de ce chapitre sont abordées sous la perspective des approches orientées documents.

2.4 Indexation de documents semi-structurés

Les SRI ont très longtemps utilisé des représentations de données très simples pour opérer des requêtes sur les textes, ou classer ceux-ci en différentes catégories. Si les SRI ont très longtemps utilisé des représentations de données vectorielles pour opérer des requêtes sur les textes, à partir du début des années 1990, ces représentations ont commencé à prendre en compte la structure des documents pour mener des travaux sur deux axes : la "recherche de passages" et la "recherche de sous-structures". Les premiers se limitent généralement à découper un document en sous-documents, et à ré-appliquer à ces unités d'informations les modèles habituels (souvent donc vectoriels) de la RI. La prise en compte "simultanée" du document et de ses sections pour opérer des recherches plus fines n'est introduite qu'à partir de 1994 par Wilkinson [229].

En RIS, l'objectif de l'indexation n'est plus seulement de stocker l'information textuelle mais aussi l'information structurelle et de pouvoir présenter les relations entre les deux types d'information. De ce fait, un schéma d'indexation de documents XML devrait principalement permettre la reconstruction du document XML décomposé dans les structures de stockage et la recherche par mot clé et par expressions de chemin sur la structure XML.

L'indexation de documents XML peut être rangée selon le type de l'information en question (textuelle ou structurelle). Cette catégorisation permet de mieux comprendre les différents enjeux soulevés par chaque type d'information.

2.4.1 Indexation de l'information textuelle

L'indexation de l'information textuelle consiste à extraire et pondérer les termes représentatifs. En RIS, et notamment avec les documents XML, la seule différence par rapport à la RI classique est comment lier les informations textuelles (ou termes) aux informations structurelle ? C'est ce qu'on appelle la "*portée des termes d'indexation*".

2.4.1.1 Portée des termes d'indexation

Afin de relier les termes à l'information structurelle, dans la littérature, deux solutions ont été proposées : une qui agrège le contenu des nœuds (c'est l'approche d'indexation des *sous-arbres imbriqués*) et l'autre qui indexe tous les contenus des nœuds séparément (c'est l'approche d'indexation des *unités disjointes*).

- **Sous-arbres imbriqués :** ces approches considèrent que le contenu de chaque nœud de l'index est une unité atomique [12, 202, 110]. Les termes des nœuds feuilles sont donc propagés dans l'arbre des documents. Comme les documents XML possèdent une structure hiérarchique, les nœuds de l'index sont imbriqués les uns dans les autres et par conséquent, l'index contient des informations redondantes. Dans [151], Mass *et al.* ont considéré que seuls quelques types de nœud sont informatifs (dans la collection d'INEX 2005, ils ont par exemple sélectionné : article, paragraphe, section, sous-section). Un sous-index est ensuite construit pour chaque type de nœud. L'index est l'ensemble des sous-index associés.
- **Unités disjointes :** dans ces approches, le document XML est décomposé en unités disjointes, de telle façon que le texte de chaque nœud de l'index est l'union d'une ou plusieurs parties disjointes [159, 79, 89, 118, 187].

Une fois les unités d'indexation spécifiées, il reste à pondérer les termes. Cette tâche est une adaptation des fonctions de pondération déjà proposées en RI classique.

2.4.1.2 Pondération des termes d'indexation

Dans la RI classique, la pondération des termes est basée sur les notions de *tf* et *idf* [193]. Dans la RIS, le poids d'un terme dans un élément dépend non seulement de son importance dans cet élément ou dans la collection mais aussi de son importance dans le contenu du nœud même, dans le contenu de ses descendants, dans le contenu de ses voisins directs et dans le contenu des nœuds auxquels il est relié [141, 118]. Ce dernier poids est défini par la mesure *ief* (Inverse Element Frequency). Dans la littérature, plusieurs travaux ont utilisé *ief*, par exemple [230, 90, 200, 149, 171]. Des adaptations des formules de pondération utilisées en RI classique à la RIS sont proposées dans [216]. Une adaptation de la formule *tf.idf* permettant de calculer la force discriminatoire d'un terme t pour une balise b relative à un document d, est également présentée dans [236]. La nouvelle formule adaptée est définie par *tf.itdf* (Term Frequency-Inverse Tag and Document Frequency).

Pinel-Sauvagnat et Boughanem [171] ont utilisé d'autres paramètres pour l'évaluation de l'importance de termes tels que la longueur de l'élément et la

longueur moyenne des éléments de la collection.

2.4.2 Indexation de l'information structurelle

Différentes approches ont été proposées pour indexer l'information structurelle selon des granularités variées [144]. Dans le processus d'indexation, toute l'information structurelle n'est pas forcément utilisée. Dans la littérature, on trouve trois approches pour l'indexation de l'information structurelle : Indexation basée sur les champs, Indexation basée sur des chemins et Indexation basée sur des arbres.

2.4.2.1 Indexation basée sur des champs

Cette technique permet d'associer à chaque terme le nom du champ dans lequel il apparaît. Avec ce type d'indexation, on filtre, au moment de la recherche, les champs contenant le texte en question [93]. Le tableau 2.2 illustre le résultat d'indexation du document illustré par la figure 2.1.

termes	fréquence	champs
recherche	1	(titre, 1)
information	3	(titre, 1), (sec1, 1), (sec2, 1)
indexation	3	(titre, 1), (sec1, 1), (sec2, 1)
textuelle	1	(sec1, 1)
structurelle	1	(sec1, 2)

TABLE 2.2 – Indexation basée sur les champs

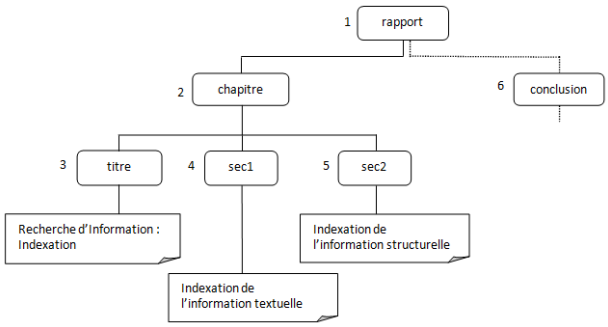

FIGURE 2.1 – Exemple d'indexation de l'information structurelle

2.4.2.2 Indexation basée sur des chemins

Cette technique a pour but de retrouver rapidement des documents ayant des valeurs connues pour certains éléments ou attributs [113, 101]. Elle facilite aussi la navigation dans les documents de manière à résoudre efficacement des expressions XPATH et d'utiliser des index pleins textes sur les contenus. Cette technique souffre cependant souvent de la difficulté de retrouver les relations ancêtres-descendants entre les différents éléments des documents. Le tableau 2.3 illustre ce type d'indexation correspondant au document de la figure 2.1. En 2009, une nouvelle approche d'indexation basée sur les chemins a été proposée par BenAouicha *et al.* [18]. Les auteurs ont proposé d'exprimer les relations entre un élément et ses descendants à l'aide d'arcs virtuels au niveau de la structure d'un document XML. Par exemple, dans la figure 2.1, le lien entre les deux éléments *rapport et sec1* est un arc virtuel, etc.

termes	**fréquence**	**chemins**
recherche	1	(/rapport/chapitre/titre)
information	3	(/rapport/chapitre/titre), (/rapport/chapitre/sec1), ...
indexation	3	(/rapport/chapitre/titre), (/rapport/chapitre/sec2), ...
textuelle	1	(/rapport/chapitre/sec1)
structurelle	1	(/rapport/chapitre/sec2)

TABLE 2.3 – Indexation basée sur les chemins

2.4.2.3 Indexation basée sur des arbres

Dans cette technique, chaque élément (nœud) du graphe représentant le document XML est identifié par un identifiant unique (*UID*) [133]. Les termes sont associés à cet identifiant afin de pouvoir localiser leurs emplacement dans les éléments et de retrouver les relations hiérarchiques entres les éléments [201]. L'UID peut également être un chemin d'accès (XPath absolu, avec les numéros des éléments) de l'élément [231]. Le tableau 2.4 illustre ce type d'indexation pour le document XML de la figure 2.1. Parmi les travaux utilisant cette technique d'indexation, nous citons [113, 101]. D'autres techniques d'indexation structurelle basée sur les arbres sont proposées dans la littérature telles que l'approche EDGE et BINARY [77], l'architecture BUS [104], etc.

Afin de bénéficier au mieux de toutes les caractéristiques du document XML, de nouvelles approches ont été proposées. Elles consistent à combiner l'approche orientée données et l'approche orientée documents [87, 214, 42, 168]. Ces approches permettent également d'indexer le contenu textuel des documents et

de pondérer les termes, ce qui rend ensuite possible un calcul de pertinence des éléments.

termes	fréquence	nœuds
recherche	1	(3)
information	3	(3), (4), (5)
indexation	3	(3), (4), (5)
textuelle	1	(4)
structurelle	1	(5)

TABLE 2.4 – Indexation basée sur les arbres

2.5 Interrogation des documents XML

Comme mentionné précédemment, l'interrogation des documents XML peut se faire selon deux types de requêtes :

- **Requêtes de type CO :** ces requêtes sont composées de simples mots-clés et imposent au SRI de décider la granularité de l'information à retourner. Elles sont utilisées lorsque l'utilisateur n'a pas une idée précise de ce qu'il recherche ou n'a pas de connaissance concernant la structure des documents.
- **Requêtes de type CAS :** ces requêtes sont composées de contraintes sur le contenu et la structure. C'est le cas lorsque l'utilisateur peut spécifier des conditions de structures pour préciser son besoin et indiquer quel type d'éléments qu'il désire lui renvoyer. Ce type de requête nécessite au moins une connaissance partielle de la structure de la collection des documents XML à interroger.

De nombreux langages de requêtes ont été proposés dans la littérature. D'une manière générale, ces langages de requêtes supportent conjointement des contraintes de contenu et de structure. Nous nous proposons d'en détailler quelques uns dans ce qui suit, suivant leur ordre chronologique d'apparition.

2.5.1 XQuery

XQuery [53] est un langage de requête pour XML proposé par le W3C dont la version 1.0 finale date de janvier 2007, et dont l'élaboration a demandé près de huit années. Il se base sur XPath pour extraire et travailler sur des fragments de documents XML. Les requêtes basiques de XQuery sont identiques à celles définies par XPath. Si l'on désire faire des requêtes simples, XPath peut donc parfaitement suffire.

XQuery est intéressant dès le moment où l'on désire faire des requêtes complexes ou encore faire appel à la récursivité. XQuery supporte des fonctions orientées systèmes documentaires : en particulier, un prédicat CONTAINS est intégré pour la recherche par mots-clés.
On trouvera ci-dessous un exemple d'une requête XQuery qui retourne les prénoms et les dates de naissance de touts les employés ayant le nom Dupont :

For $E in document ("exemple.xml")//Employe
Where $E/nom = "Dupont"
return
<dupont>{
$E/prenom,
$E/date_naissance
}</dupont>

On notera enfin que le W3C a proposé un Working Draft, qui a pour but d'étendre les caractéristiques de recherche de XQuery à la recherche plein-texte. Le langage TexQuery [16] en est une application.

2.5.2 NEXI

Le langage NEXI a été défini dans [218, 219] pour répondre aux besoins de la campagne d'évaluation INEX. Les requêtes étaient en effet précédemment exprimées en XML (pour 2002) ou XPath (pour 2003), mais dans le premier cas, le langage n'était pas assez puissant, et il était trop complexe et dans le second cas 63% des requêtes exprimées par les participants (experts en RI) contenaient des erreurs de syntaxe ! NEXI a alors été conçu comme un sous-ensemble extensible d'XPath interprétable de manière vague (il s'agit d'un langage de requête orienté RI et non BD). On utilise la syntaxe pour désigner l'élément descendant et rajoute une clause "about" pour apporter plus de précision. NEXI peut également supporter des spécifications plus complexes en utilisant les parenthèses ainsi que les opérateurs booléens.
L'exemple suivant est une requête qui renvoie une section *sec* qui est un élément du document *article* et qui contient un autre élément paragraphe *p* et qui parle de "information retrieval".

//article//sec[about(.//p,information retrieval)]

2.5.3 XFIRM

Le langage de requêtes XFIRM [168] est une extension de XPath. Ce langage permet de formuler la requête de l'utilisateur selon quatre degrés de précision comme les illustrent les exemples suivants :

- **Degré de précision P1 :**

  Toulouse OU (ville ET rose)

  Ce type de requête permet à l'utilisateur d'exprimer son besoin en information en utilisant des mots-clés indépendamment de la structure de l'unité d'information renvoyée.
- **Degré de précision P2 :**

  section[la ville rose]

  Dans cet exemple l'utilisateur désire récupérer des éléments de type *section* parlant de la *ville rose*. Avec ce type de requête, nous pourrons préciser le type des éléments à renvoyer ainsi que des conditions sur le contenu ou la valeur de ses attributs.
- **Degré de précision P3 :**

  //article[France]//section[Toulouse]

  Avec ce type de requête, l'utilisateur peut définir la structure hiérarchique entre les éléments renvoyés. Dans cet exemple, l'utilisateur désire récupérer les éléments *articles* parlant de la *France* et ayant des descendants de type *section* parlant de *Toulouse*.
- **Degré de précision P4 :**

  //article[]//ec :section[Toulouse]//par[Capitole]

  Dans cet exemple, l'utilisateur souhaite obtenir un élément de type *section* parlant de *Toulouse* ayant comme ancêtre un élément de type *article* et comme descendant un élément de type *paragraphe* parlant de *Capitole*.

L'avantage du langage XFIRM est que l'utilisateur n'est pas obligé à spécifier le type de l'unité d'information qu'il désire voir retournée. De plus, ce langage permet d'exprimer des chemins indéterminés ou partiellement connus, et permet de combiner de façon booléenne des conditions sur la structure.

2.6 Modèles de RIS

Dans la littérature, les modèles de RI classiques ont été adaptés pour tenir compte de la source d'évidence, l'information structurelle, contenue dans les documents XML, et des granularités variées de l'information. Ces modèles cherchent à répondre à des requêtes de type CO ou bien à des requêtes de type

CAS.

D'une manière générale, et indépendamment des modèles de RIS, l'appariement est effectué selon deux catégories d'approches différentes [173].

- **Approches par propagation des termes :** ces approches indexent des sous-arbres imbriqués et propagent les termes des nœuds feuilles dans l'arbre du document ;
- **Approches par propagation de pertinence :** ces approches indexent des unités disjointes et calculent les scores de pertinence au niveau des feuilles des arbres XML. Ces scores sont ensuite propagés vers les nœuds internes.

Dans cette section, nous nous proposons de détailler les différentes méthodes proposées pour adapter le modèle booléen (théorie des ensembles), le modèle vectoriel (algébrique) ou encore le modèle probabiliste. Nous nous attardons ensuite sur les modèles de RIS basés sur les RB. Notons simplement à titre d'illustration que :

- les approches présentés dans le cadre du modèle vectoriel étendu, [82, 151, 90, 200, 64, 149, 64, 150, 18] utilisent une propagation des termes et dans [83, 17, 100, 212, 168], il s'agit d'une propagation de pertinence.
- les approches de [217, 131] présentés dans le cadre du modèle booléen pondéré, utilisent une propagation des termes.
- les approches présentés dans le cadre du modèle probabiliste [56, 127, 79, 89], ou du modèle inférentiel [177, 223, 67, 134, 131, 22, 70, 68, 132, 137] ou du modèle de langue [202, 230, 138, 12, 160, 110, 157], fonctionnent tous également grâce à une propagation des termes.

2.6.1 Modèle vectoriel étendu

Le modèle vectoriel étendu permet de séparer l'information structurelle de l'information de contenu [151, 149, 150]. Dans les approches issues de ce modèle, une mesure de similarité de chaque élément à la requête est calculée, et ce à l'aide de mesures de distance dans un espace vectoriel. Les éléments sont représentés par des vecteurs de termes pondérés.
Dans la littérature, nous trouvons deux catégories d'approches. La première indexe des sous-arbres imbriqués (section 2.4.1), c'est-à-dire elles propagent les termes des nœuds feuilles dans l'arbre du document. Les éléments sont renvoyés à l'utilisateur par ordre décroissant de pertinence.

Fuller *et al.* [82] ont proposé une des premières adaptations du modèle vectoriel à la RIS. La pertinence d'un nœud est calculée à part, puis combinée avec la pertinence des nœuds descendants. Le modèle peut être généralisé en permettant le traitement des requêtes orientées contenu et structure. L'idée de base est là encore d'appliquer le modèle récursivement à chaque sous-arbre de

la hiérarchie pour ensuite effectuer une agrégation des scores.

Mass *et al.* [151, 149] ont proposé un système de recherche, appelé JuruXML, qui indexe les éléments selon leur type (un index par type d'élément) et applique ensuite le modèle vectoriel pour la pondération des éléments.

Schlieder et Meuss [200] ont développé une autre extension du modèle vectoriel, et qui consiste à intégrer la structure des documents dans la mesure de similarité du modèle vectoriel. La formulation des requêtes se fait sans besoin de connaître la structure exacte des données vu que leur modèle de requête est basé sur l'inclusion d'arbres. Afin de répondre à des requêtes orientées contenu et structure, les auteurs combinent ainsi le modèle vectoriel et le "tree matching".

BenAouicha *et al.* [18] proposent le modèle XIVIR [5] qui permet la RIS par la structure et/ou le contenu en utilisant une approche par propagation des termes :

- **Recherche par le contenu :** la propagation du texte situé au niveau des nœuds feuilles vers ses ancêtres se fait selon deux approches. La première consiste à représenter le contenu de chaque nœud feuille par un ensemble de termes pondérés. Ces derniers seront propagés vers les ancêtres de ce nœud tout en diminuant leurs poids en fonction de la distance parcourue au moment de la propagation. C'est la propagation du texte en profondeur. Quant à la deuxième approche, propagation du texte par profondeur et largeur, elle sera réalisée en fonction de la distance qui sépare le nœud feuille qui contient du texte et le nœud interne qui est censé recevoir le texte. Le facteur de propagation est calculé en fonction de cette distance.
- **Recherche par la structure :** le document XML est représenté sous forme d'un arbre défini comme un ensemble de chemins entre deux nœuds $A \rightarrow B$ où A est le nœud parent du nœud B. La relation entre A et B peut être directe (parent/fils-direct) ou indirecte (parent/descendant). Afin de refléter l'importance de la relation entre les nœuds A et B, un poids est calculé pour chaque chemin. Si la relation est directe, le poids est égal à 1, sinon, le poids w est calculé comme suit :

$$w = exp(\lambda * (1 - d(A, B))) \tag{2.1}$$

où $d(A, B)$ est la distance qui sépare les deux nœuds A et B, et λ est un coefficient d'atténuation. Pour la recherche par structure, le score de structure entre une requête q et un document d, $RSVs$, est calculé comme suit :

$$RSV_s = \sum_{A_q \overset{w_q}{\rightarrow} B_q \in E_q \equiv A_d \overset{w_d}{\rightarrow} B_d \in E_d} w_q * w_d \tag{2.2}$$

5. XIVIR : XML Information retrieval based on VIRtual links

où E_q (resp. E_d) est l'ensemble de tous les chemins pondérés de la requête (resp. du document). Soient A_q l'élément A dans la requête q et A_d est l'élément A dans le document d, $A_q \equiv A_d$ signifie que A_q est l'équivalent à A_d. Par exemple, $chapter \xrightarrow{0,37} p \equiv chapter \xrightarrow{1} p[2]$ sur la figure 2.2. Selon la structure entre la requête et le document, le score est $RSVs(E_q, E_d) = 2 + 0,37 * 0,37 = 2,14$.

– **Combinaison des scores :** Le traitement séparé du contenu et de la structure de chaque élément XML engendre deux scores : un score pour le contenu et un score pour la structure. Leur combinaison en un score définitif permet de les ordonner selon leur pertinence potentielle. Dans ce contexte, deux techniques pour la combinaison des scores sont proposées : une technique basée sur une combinaison linéaire et une deuxième technique basée sur les distributions des scores.

Les résultats obtenus au niveau de la mesure stricte de la tâche *VVCAS* montrent l'efficacité de ce modèle. Cette tâche est par essence la plus complexe, elle impose l'installation de méthodes de recherche orientées structure, et de se dissocier des méthodes traditionnelles de RI et des méthodes d'interrogation par des requêtes semblables de SQL ou XQuery.

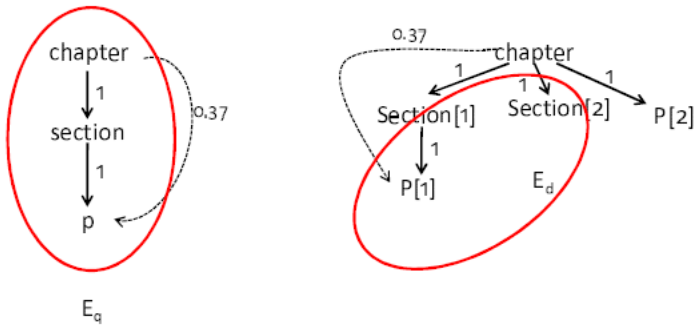

FIGURE 2.2 – Exemple de recherche par structure avec le système XIVIR [18]

On trouvera également la deuxième catégorie d'approches qui indexent des unités disjointes (section 2.4.1), c'est-à-dire elles calculent les scores de pertinence au niveau des feuilles des arbres XML et propagent ces scores ensuite vers les nœuds internes.

Dans [83], Geva a proposé un modèle simple qui a obtenu de très bons résultats pendant les campagnes d'évaluation INEX 2003 et INEX 2004. Ce modèle est basé sur un fichier inverse pour l'indexation d'un document XML. La recherche est réalisée par propagation des scores des éléments feuilles. Ce système a obtenu les meilleurs résultats dans la campagne d'évaluation INEX 2005 [84].

Dans [212], Theoblad et Weikum proposent le moteur de recherche XXL qui utilise une fonction de score basée sur tf et idf. XXL offre des fonctionnalités pour la recherche orientée pertinence de chemins, c'est à dire que la recherche est effectuée avec des conditions de chemins vagues. XXL repose sur une syntaxe SQL (select-from-where).

Dans [168, 172, 169], Pinel-Sauvagnat *et al.* proposent le système XFIRM [6] qui est basé sur un modèle de données générique permettant l'implémentation de nombreux modèles de RIS et le traitement de collections hétérogènes.
Le traitement des requêtes est effectué en deux étapes : une première qui consiste à évaluer la similarité des nœuds feuilles de l'index à la requête (on parle alors de calcul des poids des nœuds feuilles) et une seconde qui consiste à rechercher les sous-arbres pertinents. La pertinence des sous-arbres est évaluée en effectuant la propagation des poids des nœuds feuilles dans l'arbre du document. Le langage de requêtes utilisé est déjà détaillé dans la section 2.5.3.

- **Calcul du score des nœuds feuilles :** les scores des nœuds feuilles identifiés dans l'arbre du document sont calculés grâce à la fonction de similarité $RSV(q, nf)$.
  Si la requête est composée de termes et des poids associés, on a :

$$RSV(q, nf) = \sum_{i=1}^{T} w_i^q * w_i^{nf}, \; avec \; w_i^q = tf_i^q \; et \; w_i^{nf} = tf_i^{nf} * ief_i * idf_i \quad (2.3)$$

  Avec :
  - w_i^q et w_i^{nf} sont respectivement le poids du terme i dans la requête q et le nœud feuille nf ;
  - tf_i^q et tf_i^{nf} sont respectivement la fréquence du terme i dans la requête q et dans le nœud feuille nf ;
  - $idf_i = log(|D|/|d_i|)$ permet d'évaluer l'importance du terme i dans la collection de documents ;
  - $|D|$ est le nombre total de documents de la collection ;
  - $|d_i|$ est le nombre de documents contenant i ;
  - $ief_i = log(|NF||nf_i|)$ permet d'évaluer l'importance du terme i dans la collection de nœuds feuilles ;
  - $|NF|$ est le nombre total de nœuds feuilles de la collection ;
  - $|nf_i|$ est le nombre de nœuds feuilles contenant le terme i.
- **Propagation de la pertinence des nœuds feuilles :** une valeur de pertinence est ensuite calculée pour chaque nœud de l'arbre de document, en utilisant les poids des nœuds feuilles qu'il contient [172]. Les termes apparaissant près de la racine d'un sous-arbre paraissent plus porteurs d'information pour le nœud associé que ceux situés plus bas dans le sous-arbre. Il semble ainsi intuitif que plus grande est la distance entre un nœud et son ancêtre, moins il contribue à sa pertinence. Cette intuition est modélisée par l'utilisation dans la fonction de propagation du pa-

6. XFIRM : XML Flexible Information Retrieval Model

ramètre $dist(n, nf_k)$, qui représente la distance entre le nœud n et un de ses nœuds feuille nf_k dans l'arbre du document, c'est-à-dire le nombre d'arcs séparant les deux nœuds. Il paraît aussi intuitif que plus un nœud possède de nœuds feuilles pertinents, plus il est pertinent. Le paramètre $|F_n^p|$, qui est le nombre de nœuds feuilles descendants de n ayant un score non nul est alors introduit dans la fonction de propagation. Une première évaluation de la pertinence p_n d'un nœud peut être calculée selon la formule 2.4 :

$$p_n = |F_n^p| * \sum_{nf_k \in F_n} \alpha^{dist(n,nf_k)-1} * (RSV_m(q, nf_k)) \tag{2.4}$$

où F_n est l'ensemble des nœuds feuilles nf_k descendants de n, et $\alpha \in]0, 1]$ est un paramètre permettant de quantifier l'importance de la distance séparant les nœuds dans la formule de propagation.
On peut également intégrer dans la mesure du score la pertinence que l'on accorde au document entier. On parle alors de pertinence contextuelle. La valeur de pertinence d'un nœud interne est définie alors comme suit :

$$p_n = p * |F_n^p| * \sum_{nf_k \in F_n} \alpha^{dist(n,nf_k)-1} * RSV(q, nf_k) + (1 - \rho) * p_{racine} \tag{2.5}$$

Avec :
- F_n l'ensemble des nœuds feuilles nf_k descendants de n ;
- $|F_n^p|$ le nombre de nœuds feuilles descendant de n ayant un score non nul ;
- $RSV(q, nf_k)$ calculé d'après 2.3 ;
- $\rho \in [0, 1]$ est un paramètre servant de pivot et permettant d'ajuster l'importance de la pertinence du nœud racine.

Les nœuds sont ensuite renvoyés à l'utilisateur par ordre décroissant de pertinence à la requête.
Ce modèle a montré de bonnes performances au sein de la campagne d'évaluation INEX [170, 174, 171].

Enfin, on trouvera d'autres adaptations du modèle vectoriel dans [148, 17, 64, 228, 109, 100].

2.6.2 Modèle probabiliste

Les modèles probabilistes constituent un outil puissant pour les modèles de RIS vu qu'ils permettent de traiter d'une manière efficace l'incertitude intrinsèque au processus de RI. Ces modèles calculent la probabilité de pertinence des documents étant donnée une requête ou la probabilité de satisfaire

la requête étant donné le document.

2.6.2.1 Modèle inférentiel

La naissance du modèle d'inférence est le résultat de l'extension de deux idées : *(i) la proposition d'utiliser des logiques non classiques pour déterminer le degré auquel un document implique ou correspond à une requête* ; *(ii) la notion d'inférence plausible et la possibilité de combiner plusieurs sources pour inférer la probabilité de pertinence d'un document étant donnée une requête.*

Dans la RIS, les diagrammes d'inférence ont été adaptés pour exprimer les relations de causalité entre termes et structures. Plus récemment, des travaux ont essayé d'exploiter l'apport des RB pour définir des modèles de RIS. L'avantage apporté par l'utilisation des RB a été principalement de pouvoir combiner des informations provenant de différentes sources pour restituer les documents qui seraient les plus pertinents étant donnée une requête.
Parmi les travaux les plus récents, citons celui Piworwarski *et al.* [177, 175]. Les auteurs ont proposé un modèle probabiliste basé sur les RB où les dépendances de hiérarchisation sont exprimées par des probabilités conditionnelles. La probabilité de pertinence d'un élément e sachant son parent p pour une requête q est $P(e|p,q)$ est la suivante :

$$P(e = a|p = b, q) \simeq \frac{1}{1 + e^{F_{e,a,b(q)}}} \tag{2.6}$$

où $F_{e,a,b(q)}$est la pertinence de l'élément e selon le modèle Okapi.
Une requête q structurée est décomposée en un ensemble de n sous-requêtes élémentaires q_i. Chacune de ces sous-requêtes reflète une entité structurelle et un besoin d'information. Le score final est donné par la formule suivante :

$$RSV(e_i, q) = RSVq_1(e_i, q) * ... * RSVq_n(e_i, q) \tag{2.7}$$

Ce modèle est étendu, dans [223], au traitement des requêtes orientées contenu et structure.

De Campos *et al.* [67] ont également proposé un modèle de recherche basé sur les RB où le diagramme d'inférence est basé sur la probabilité conditionnelle. Deux types de diagrammes sont proposés : SID (*Simple Inference Diagram*) et CID (*Context based Inference Diagram*). Un diagramme se compose de deux parties : une partie qualitative (représentation des variables et des inférences) et une partie quantitative (probabilités des nœuds).

Plusieurs modèles ont été proposés pour l'interrogation de corpus hétérogènes. La majorité des solutions s'orientent vers la classification de documents [134,

131, 22]. La recherche se fait alors au niveau des classes de documents. Denoyer *et al.* [70] ont conçu un format intermédiaire qui permet de classifier les documents en suivant un calcul basé sur la probabilité conditionnelle.

Denoyer et Gallinari [68] ont également traité le problème de classification de documents structurés à l'aide de RB. Chaque nœud du RB comporte un libellé et des informations contextuelles. Deux sortes de variables sont envisagées :

1. Une variable structurelle s_d^i(d : document) qui dépend de ses ascendants.
2. Une variable contextuelle t_d^i qui ne dépend que de ses variables structurelles.

La probabilité de jointure d'un document d à un modèle C est calculée comme suit :

$$P(d, C) = P(c) \prod_{i=1}^{|d|} P(s_d^i / pa(s_d^i, C)) P(t_d^i / s_d^i, C) \tag{2.8}$$

Avec :
- t_d^i est une séquence de mots ;
- $pa(s)$ est le parent d'un nœud.

Ce modèle génératif permet de considérer des documents hétérogènes (texte plus image), où l'image est considérée comme un ensemble de pixels. Il est par la suite transforme en classifieur discriminant en utilisant la méthode Fisher Kernel [103].

Abiteboul *et al.* dans [10] visent à proposer un format médian dans lequel tous les documents du corpus (et éventuellement les requêtes) peuvent être transformés pour ensuite appliquer des techniques traditionnelles de traitement des requêtes structurées.

D'autres approches, comme celle proposée par Lee *et al.* dans [132] ou Lian et Cheung dans [137] visent à proposer des algorithmes de classification. Dans la première approche, les auteurs proposent un algorithme de matching entre deux documents grâce à une séquence d'opérations de transformations. Dans la deuxième approche, les auteurs proposent un algorithme pour classifier les documents en se basant sur le paramètre distance et la notion de sous-graphe qui sont codés par des chaînes de bits.

2.6.2.2 Modèle de langue

Sigurbjörnsson *et al.* [202] proposent un modèle de langue pour traiter des requêtes de type CO. Les auteurs considèrent que comme n'importe quel élément XML peut potentiellement être renvoyé à l'utilisateur, chaque élément est indexé afin d'assurer la même fonction qu'un fichier inverse en RI classique

et chaque document est indexé pour des calculs statistiques. L'arbre XML est indexé en se basant sur le post et le pré-ordre des nœuds. Par conséquent, pour chaque élément, le texte qu'il contient ainsi que le texte contenu dans ses descendants est indexé (voir approches d'indexation basées sur les sous-arbres imbriqués, section 2.4.1.1). Un modèle de langue est ensuite estimé pour chaque élément de la collection. Pour une requête donnée, les éléments sont triés par rapport à la probabilité que le modèle de langue de l'élément génère la requête. Ceci revient à estimer la probabilité $P(e, q)$, ou e est un élément et q une requête :

$$P(e, q) = P(e) * P(q|e) \tag{2.9}$$

Deux probabilités doivent donc être estimées : la probabilité a priori de l'élément $P(e)$ et la probabilité qu'il génère la requête $P(q|e)$. La première probabilité est estimée comme suit :

$$P(e) = \frac{|e|}{|C|} \tag{2.10}$$

Avec :
- $|e|$ est le nombre de mots dans l'élément e ;
- $|C|$ est le nombre de mots contenus dans tous les documents.

Pour la seconde probabilité, les auteurs considèrent que les termes de la requête sont indépendants, et utilisent une interpolation linéaire du modèle d'élément et du modèle de collection pour estimer la probabilité d'un terme de la requête. La probabilité d'une requête $t_1, t_2, ..., t_n$ est ainsi calculée de la façon suivante :

$$P(t_1, ..., t_n|e) = \prod_{i=1}^{n} (\lambda * P(t_i|e) + (1 - \lambda) * P(t_i)) \tag{2.11}$$

Avec :
- $P(t_i|e)$ est la probabilité d'observer le terme t_i dans l'élément e ;
- $P(t_i)$ est la probabilité d'observer le terme dans la collection ;
- λ est un paramètre de lissage.

Le calcul des probabilités peut être réduit à la formule de calcul des scores 2.12, pour un élément e et une requête $t_1, ..., t_n$.

$$s(e, t_1, ..., t_n) = \beta * log(\sum_t tf(t, e)) + \sum_{i=1}^{n} log(1 + \frac{\lambda * tf(t_i, e) * (\sum_t df(t))}{(1 - \lambda) * df(t_i) * \sum tf(t, e)}) \tag{2.12}$$

Avec :
- $tf(t, e)$ est la fréquence du terme t dans l'élément e ;
- $df(t)$ est le nombre d'éléments contenant t ;
- λ est le poids donné au modèle de langue de l'élément en lissant avec le modèle de la collection ;
- β est un paramètre servant à combler le fossé entre la taille de l'élément moyen et la taille de l'élément moyen pertinent.

Dans [230], l'utilisation de la fréquence inverse d'élément ief est proposée pour faciliter les pondérations par élément : un nouveau poids probabiliste pour les termes est alors formulé, utilisant ief et la fréquence du terme dans chaque élément. Les poids des termes de la requête peuvent être étendus avec des conditions sur l'appartenance du terme à un certain élément ou chemin.

On trouvera d'autres approches basées sur les modèles de langues pour la RIS dans [138, 12, 160, 110, 157].

2.6.2.3 Autres approches

Bogers *et al.* dans [32] proposent une approche basée sur le modèle de langue afin d'effectuer une recherche dans d'une collection des livres. Leur principal objectif est d'examiner l'efficacité de l'utilisation des fonctions sociales pour re-classer les résultats de recherche initiales basées sur le contenu. Ils se concentrent en particulier sur l'utilisation de techniques de filtrage collaboratif pour améliorer leurs résultats de recherche basés sur le contenu.

Dans [30], Bhaskar *et al.* décrivent un système hybride de contextualisation de tweets. Le système de RI concentré est basé sur l'architecture Nutch et le système de résumé automatique est basé sur le classement de phrases par TF-IDF et des techniques d'extraction de phrases.

Une autre approche basée sur le modèle vectoriel est proposée par Crouch *et al.* dans [65]. Cette approche réalise tout d'abord une recherche sur les documents afin d'identifier les articles pertinents à l'aide du système SMART [191]. Afin de produire d'extraits de documents correspondant à chaque article, les auteurs utilisent une approche appelée Flex pour recherche flexible [63].

L'évaluation de ces différentes approches de RIS est présentée dans la section suivante.

2.7 Évaluation des performances des systèmes de RIS

Aujourd'hui, il existe une seule campagne d'évaluation des différents systèmes de RIS. Cette campagne d'évaluation est INEX (*INitiative for the Evaluation of XML retrieval*). Elle a eu lieu depuis 2002. Elle offre un forum international pour évaluer et comparer les résultats enregistrés par les différents participants, mais aussi pour discuter les différentes problématiques qui se présentent. La

collection de test est un ensemble de documents XML, requêtes, tâches de recherche et jugements de pertinence. Le langage de requête utilisé dans INEX est NEXI [219, 218].
INEX a proposé plusieurs tâches telles que la tâche ad-hoc, la tâche multimedia, la tâche relevance feedback, la tâche hétérogène, etc.

2.7.1 Collections de test

Afin d'améliorer la qualité de l'évaluation, les collections de test proposées dans la cadre de la campagne INEX ne cessent d'évoluer. Entre 2002 et 2004, INEX a utilisée une collection composée des articles de la revue scientifique "IEEE Computer Society", balisés au format XML et d'une taille totale aux alentours 500 Mo. En 2005, la collection a été étendue pour comporter environ 17 000 articles issus de 21 revues pour une taille totale d'environ 750 Mo.

À partir de 2006, la collection IEEE a été complétée par de documents en anglais extraits de l'encyclopédie en ligne "Wikipedia", a été utilisée dans la plupart des tâches. Cette collection de 6 Go, est composée de 659 388 de documents d'une profondeur moyenne 6,72.

En 2009, une extension de la collection Wikipedia est fournie [199]. Elle est composée de 2 666 190 articles Wikipedia annotés et elle a une taille de 50,7 GB. Cette collection est utilisée dans la tâche adhoc ainsi que dans d'autres tâches. D'autres collections sont aussi fournies par la campagne d'évaluation pour évaluer d'autres tâches telles que la collection "mmwikipedia" pour une sous-tâche de la tâche *multimedia*, ou encore les collections fournies pour la tâche hétérogène. Le Guide de Planète Solitaire a été aussi utilisé et depuis 2007 une collection de livres parcourus a aussi été rendue disponible pour des tâches de recherche de livre.

2.7.2 Requêtes

Les participants à INEX ont créé deux types de requêtes (ou topics) :
- **CO :** les mots-clés de cette requête peuvent être regroupés sous forme d'expressions et précédés par les opérateurs "+" (signifiant que le terme est obligatoire) ou "-" (signifiant que le terme est exclu des éléments renvoyés à l'utilisateur).
- **CAS :** les contraintes de cette requête portent sur la structure des documents.

Pour chaque topics, différents champs permettant d'expliciter le besoin de l'utilisateur. Par exemple, le champ *title* donne une définition simplifiée de la requête, le champ *keywords* contient un ensemble de mots-clés qui ont permis l'exploration du corpus avant la reformulation définitive de la requête, et les champs description et narrative, explicités en langage naturel, indiquent les intentions de l'auteur.
À partir de 2006, ces deux types de requêtes ont été regroupés dans un seul type CO+S en rajoutant un nouveau champ *castitle* donnant la forme structurée de la requête.

2.7.3 Tâches de recherche

INEX propose plusieurs tâches d'évaluation afin d'explorer plusieurs voix de recherche dans les documents XML. Nous détaillons dans ce qui suit quelques tâches.

- **Tâche adhoc :** c'est la tâche principale de la campagne d'évaluation INEX. Elle est considérée comme une simulation de l'interrogation d'une bibliothèque, où un ensemble statique de documents XML. Plusieurs stratégies de recherche sont étudiées dans ce contexte en utilisant différents types de requêtes (CO ou CAS). Nous citons quelques-unes :
  1. La stratégie **Thorough** consiste à renvoyer à l'utilisateur les éléments fortement pertinents ;
  2. La stratégie **Focused** suppose qu'un utilisateur préfère ne pas avoir d'éléments imbriqués dans la réponse ;
  3. La stratégie **Fetch and Browse** appelée aussi All in Context, consiste à classer les résultats par article ou document. L'évaluation concerne alors d'une part les documents et d'autre part le classement des éléments dans un document donné ;
  4. La stratégie **Best in Context** permet d'évaluer les meilleurs points d'entrée dans un article donnée.
- **Tâche hétérogène :** lorsque les documents sont issus de différentes collections, ils ne possèdent pas la même DTD. Notamment avec l'apparition et l'utilisation des systèmes distribués, la tâche hétérogène s'avère un véritable challenge qui pose un certain nombre de défis :
  1. avec des requêtes de type CO, des nouvelles approches doivent être développées indépendamment des DTDs ;
  2. avec des requêtes de type CAS, s'ajoute le problème de faire correspondre des conditions structurelles appartenant à différentes DTDs.
- **Tâche recherche de livres :** il s'agit d'explorer des techniques permettant de traiter des requêtes complexes (qui va au-delà de la pertinence et qui peuvent inclure des aspects comme le genre, la nouveauté, le bien

écrit, etc.) et des sources d'information complexes (qui incluent des profils utilisateurs, des catalogues personnels et les descriptions de livres) en utilisant une collection basée sur des données provenant de Amazon et de LibraryThing.
- **Tâche contextualisation de tweets :** l'objectif est de fournir un contexte sur le sujet d'un tweet afin d'aider le lecteur à comprendre. Cette tâche consiste à répondre aux questions de la forme "Au sujet de quoi ce tweet ?" Qui peut être répondu par plusieurs phrases ou par une agrégation de textes de différents documents Wikipédia. Ainsi, l'analyse de tweet, XML/recherche par passage et le résumé automatique sont combinés afin de se rapprocher des besoins réels en information.
- **Tâche recherche d'extraits de documents :** cette tâche s'intéresse à la façon de générer des extraits d'information pour les résultats de recherche. Ces extraits doivent fournir suffisamment d'informations pour permettre à l'utilisateur de déterminer la pertinence de chaque document, sans avoir besoin de consulter le document lui-même.

2.7.4 Mesures d'évaluation

Afin de traiter les besoins supplémentaires induits par la RIS, une extension des mesures traditionnelles utilisées dans la RI classique a été proposée. Cette extension concerne plusieurs mesures d'évaluation selon les tâches et les années. Nous présentons dans cette section les mesures d'évaluation à INEX 2005 et INEX 2007.

2.7.4.1 Métriques à INEX 2005

Les mesures proposées avant INEX 2005 ne prennent pas en compte d'un problème essentiel de l'évaluation : la surpopulation de la base de rappel [116]. Cette surpopulation est due aux règles d'inférence utilisées lors de l'élaboration des jugements de pertinence [176] : si un élément est jugé pertinent, ses ancêtres doivent aussi être jugés pertinents, même si leur degré de pertinence est moindre (et ce notamment à cause de la propagation de l'exhaustivité dans l'arbre du document). Par conséquent, un taux de rappel idéal ne peut être obtenu que par les systèmes référençant tous les composants de la base de rappel, y compris les éléments imbriqués. Afin de solutionner ce problème, Kazai *et al.* établissent dans [116] la définition d'une base de rappel idéale, qui supporterait la procédure d'évaluation suivante : les éléments de la base de rappel idéale doivent être retournés par les systèmes, les éléments proches de ceux contenus dans la base de rappel idéale peuvent être vus comme des succès partiels, mais les autres systèmes ne doivent pas être pénalisés s'ils ne les renvoient pas. Les

mesures xCG sont proposées pour répondre à ces besoins. Les mesures xCG (*XML Cumulated Gain*) sont des extensions du "gain cumulatif" proposé par Järvelin et Kekäläinen dans [106].

$$xCG(i) = \sum_{j=1}^{i} xG(j) \tag{2.13}$$

où $xG(j)$ est le score obtenu pour l'élément classé à la position j par le système. La métrique xCG inclut les mesures de gain cumulé étendu normalisé ($nxCG$) données par :

$$nxCG(i) = \frac{xCG(i)}{xCI(i)} \tag{2.14}$$

où $xCI(i)$ est le gain cumulé idéal.

Les mesures de gain cumulatif ont été développées pour évaluer les systèmes selon le degré de pertinence des documents retournés. La motivation derrière XCG est d'étendre les mesures de gain cumulatif au problème des éléments imbriqués. Les premiers tests de fiabilité de la mesure sont encourageants [117].

2.7.4.2 Métriques proposées depuis INEX 2007

Depuis 2007, les mesures officielles sont basées sur l'interpolation du Rappel/Précision sur 101 niveaux [112].

- **Précision interpolée selon quatre niveaux de rappel sélectionnés :** $iP[jR]$,$j \in [0,00; 0,01; 0,05; 0,1]$ La précision à un rang r est définie comme suit :

$$P[r] = \frac{\sum_{i=1}^{r} rsize(p_i)}{\sum_{i=1}^{r} size(p_i)} \tag{2.15}$$

  Avec :

  1. p_i est la partie du document assignée au rang i (avec $i \leq r$) dans la liste de résultats L_q des parties de documents retournées par un système de recherche pour une requête q.
  2. $rsize(p_r)$ est la taille du texte pertinent contenu dans p_r en nombre de caractères (ce texte est déterminé grâce aux jugements de pertinence qui contiennent le bon élément avec sa taille) et $size(p_r)$ est la taille totale du texte contenu dans p_r en nombre de caractères.

  Le rappel à un rang r est défini comme suit :

$$R[r] = \frac{\sum_{i=1}^{r} rsize(p_i)}{Trel(q)} \tag{2.16}$$

  où $Trel(q)$ est la quantité totale du texte pertinent pour une requête q.

La mesure de précision interpolée $iP[x]$ est la suivante :

$$iP[x] = \begin{cases} \max\limits_{1 \leq r \leq |L_q|} (P[r] \wedge R[r] \geq x) & if \quad x \leq R[|L_q|], \\ 0 & if \quad x > R[|L_q|]. \end{cases} \tag{2.17}$$

où $R[|Lq|]$ est le rappel pour tous les documents restitués. La mesure officielle utilisée pour comparer les différents systèmes est $iP[0, 01]$.

– **Moyenne des précisions moyennes interpolées selon 101 niveaux de rappel (MAiP) :** Pour n requêtes, $MAiP$ est calculée comme suit :

$$MAiP = \frac{1}{n} \sum_{t} AiP(t) \tag{2.18}$$

où Aip est la précision moyenne interpolée, elle est obtenue par la moyenne des scores de précision interpolées selon 101 niveaux standards de rappel :

$$AiP = \frac{1}{101} \sum_{x=0,00;0,01;...;1,00} iP(x) \tag{2.19}$$

Nous utilisons ces mesures dans notre première série d'expérimentations du chapitre 5, section 5.3.3.

2.8 Conclusion

Dans ce chapitre, nous avons passé en revue les méthodes, modèles et algorithmes fondamentaux utilisés en RIS. La dimension structurelle apportée au contenu textuel des documents permet de considérer l'information avec une autre granularité que le document tout entier. Le but pour les systèmes de RIS est alors de renvoyer les unités d'information (ou portions de documents) les plus spécifiques et exhaustives à la requête utilisateur.

Nous avons aussi donné un aperçu sur les nouveaux concepts d'évaluation des systèmes de RIS. Nous constatons qu'avec la structure la RI dans ses documents peut être plus spécifique et précise. Généralement, les approches actuelles renvoient des éléments indissociables, or il existe des requêtes qui nécessitent l'agrégation de résultats. Ainsi, au lieu de récupérer une liste d'éléments qui sont susceptibles de répondre à la requête, notre contribution consiste à agréger des éléments XML en utilisant des RB. L'avantage d'utiliser un modèle de RIS basé sur les RB et leur capacité à combiner des informations provenant de différentes sources pour restituer une liste d'agrégats qui seraient les plus pertinents étant donnée une requête.

Nous allons présenter dans le chapitre suivant (3) les principales motivations développées en RI agrégée comme une alternative prometteuse car elle peut assembler dans la réponse des éléments plus pertinents, non-redondants et complémentaires.

Chapitre 3

Vers la Recherche d'Information agrégée dans des documents semi-structurés

3.1 Introduction

Les modèles de RI peuvent être regroupés selon le type de modèle mathématique utilisé, à savoir : le modèle ensembliste [1], le modèle vectoriel [2] et le modèles probabiliste [3]. Ils peuvent également être regroupés selon le type de sortie à savoir une liste de documents non-ordonnés ou une liste de documents ordonnés selon un degré de pertinence. Les premiers travaux en RI étant basés sur le premier paradigme alors, actuellement, c'est le second le plus utilisé.

Il y a peu de temps, lorsqu'on soumet une requête à un moteur de recherche quel qu'il soit, ce dernier effectuait par défaut la recherche sur un serveur principal qui indexe les pages Web en HTML. Ce modèle a évolué en effet, les moteurs de recherche créent de plus en plus des moteurs annexes ou verticaux pour d'autres types de contenus, soit la plupart du temps : images, vidéos, actualités, livres, etc. Cette avancée majeur des moteurs de recherche permet donc d'ajouter des résultats complémentaires provenant d'autres sources à la

1. ces modèles trouvent leurs fondements théoriques dans la théorie des ensembles. On distingue le modèle booléen pur, le modèle booléen étendu et le modèle basé sur les ensembles flous.
2. ces modèles sont basés sur l'algébrique, plus précisément le calcul vectoriel. Ils englobent le modèle vectoriel, le modèle vectoriel généralisé, Latent Semantic Indexing et le modèle connexioniste.
3. ces modèles se basent sur les probabilités. Ils comprennent le modèle probabiliste général, le modèle de réseau de document ou d'inférence et le modèle de langue.

liste ordonnée de documents Web. La RI agrégée représente l'une des alternatives la plus prometteuse qui permet de répondre à ce type d'attente. La RI agrégée peut également offrir une vision plus riche de l'information issue des différentes sources de données.

Nous présentons dans ce chapitre un aperçu des différentes approches en RI agrégée ainsi que les cadres d'évaluation associés. La section 3.2 décrit en détail les problématiques des paradigmes recherche booléenne et recherche ordonnée. La section 3.3 décrit les motivations vers un nouveau paradigme de RI à savoir la RI agrégée. La section 3.4 décrit un état de l'art de la RI structurée et la RI agrégée. La section 3.5 décrit différents modèles d'évaluation orientés RI agrégée, notamment l'évaluation des documents XML. La dernière section 3.6 conclut le chapitre.

3.2 Limites de la recherche ordonnée

La majorité des approches de RI renvoient les résultats de recherche sous forme d'une liste de documents ordonnée selon un critère, souvent leur pertinence vis-à-vis de la requête. L'ordre des résultats permet souvent de placer des résultats pertinents en-tête de la liste. Ceci correspond au principe de classement. Dans [186], Robertson affirme qu'un SRI est optimal s'il devrait ordonner les résultats selon leur probabilité de pertinence.

Typiquement, les résultats sont ordonnés selon une fonction de classement qui combine différents facteurs générés à partir de la requête et la collection de documents. Ces facteurs sont également spécifiques au modèle RI [196, 184, 37, 178, 38].

Ce paradigme de recherche devient moins efficace lorsque les informations, que l'utilisateur souhaite avoir dans sa réponse, ne sont pas contenues dans un document unique [158]. Dans ce cas, une liste ordonnée n'est peut être pas le bon moyen de présenter les résultats car l'utilisateur doit fouiller au sein de différents documents pour collecter soi-même les informations qui satisfont son besoin d'information. Outre le fait qu'un tel parcours risque de s'avérer couteux en temps, onéreux et fastidieux ; tout le problème est de savoir *quand s'arrêter ?*

Pour certaines requêtes, les résultats de recherche ne sont pas diversifiés tant en termes de contenu que de présentation [61]. Ce paradigme de recherche donnerait une présentation uniforme à tous les résultats. Toutefois, il convient qu'il est parfois nécessaire de rechercher des images, des vidéos, des cartes ou bien encore des informations appartenant à une thématique très précise.

Par exemple, les requêtes “images of Niagara Falls”, “videos of Niagara Falls” et “Niagara Falls” auront tous retournées des extraits de pages Web à partir d’une recherche traditionnelle sur le Web. Idéalement, les deux premières requêtes doivent renvoyées respectivement des images et des vidéos, tandis que la troisième requête peut avoir des résultats divers (images, vidéos, pages web, ...). En fait, la diversification des résultats de la recherche a un intérêt croissant dans la RI selon [59, 14].

Plusieurs requêtes peuvent être ambigües en termes de besoin d’information. L’exemple référence est la requête “Jaguar”, qui peut se référer à une voiture, un animal, un système d’exploitation et ainsi de suite. Idéalement, nous devrions renvoyer une réponse par interprétation de la requête [203]. Cela peut être par plusieurs listes ordonnées ou un ensemble de résultats liés.

3.3 Vers la RI agrégée

3.3.1 Motivations

L’objectif de la RI agrégée est de rassembler des informations à partir diverses sources pour construire des réponses pertinentes à la requête. Comme nous l’avons déjà mentionné, dans le contexte de la liste ordonnée, l’utilisateur doit parcourir linéairement la liste en consultant les documents un à un jusqu’à avoir le sentiment d’avoir collecté suffisamment d’informations. Outre le fait qu’un tel parcours risque de s’avérer fastidieux, tout le problème est de savoir quand s’arrêter. *À partir de quel moment est-on certain d’avoir collecté assez d’informations ?*
Il est bien connu que dans le contexte de la recherche Web, l’utilisateur se limite principalement à des résultats au premier, deuxième et parfois (au plus) troisième rang [209]. Selon une étude rapportée dans [105], il a été montré que sur 10 documents affichés, 60% des utilisateurs ont consulté moins de 5 documents et près de 30% ont lu un seul document. De ce fait, il est important de renvoyer à l’utilisateur des résultats plus diversifiés pour fournir une bonne couverture de l’information disponible sur le Web concernant la requête [50, 180]. Autrement, les résultats retournés devraient donner un aperçu de différents intentions de l’utilisateur derrière sa requête. La question de la diversité des résultats retournés est encore plus importante pour les requêtes courtes ou ambigües. Par exemple, pour la requête “travelling to London”, il serait plus bénéfique de retourner des cartes, blogs, données météorologiques, etc.
La RI agrégée permet d’apporter des solutions à cette problématique. En effet, son objectif est d’intégrer d’autres types de documents (pages Web, images, vidéos, cartes, actualités, etc.) dans la page de résultats. Ce type d’agrégation est aujourd’hui adopté par la majorité des moteurs de recherche : Google’s

Universal Search [4], Yahoo !7 [5], Ask [6] et Microsoft's Live [7], etc. Les utilisateurs ont accès ensuite à différents types de résultats dans une seule interface. Ceci peut être favorable pour certaines requêtes, de type par exemple "jaguar". En effet, cette même requête est soumise aux deux moteurs de recherche Web Yahoo ! et ASK (consultés en novembre 2012) qui renvoient des résultats dans des pages agrégées indiquées dans les figures 3.1 et 3.2, respectivement. La page agrégée retournée par Yahoo !7 contient des informations appropriées aux différents contextes de la requête (e.g. jaguar cars, jaguar cats, etc.). Quant au moteur ASK, il affiche en plus une liste de sujets proposés associés à la requête sur le panneau latéral (e.g. jaguar Cubs, jaguar Clipart, jaguar Food Chain, etc.).

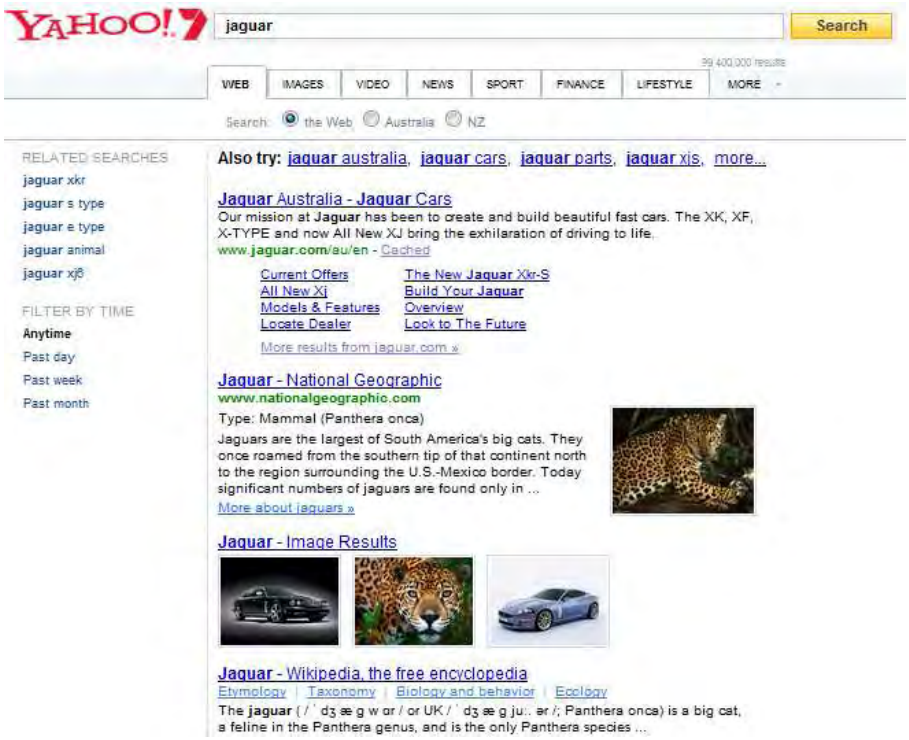

FIGURE 3.1 – Agrégation des résultats renvoyés par Yahoo !7 pour la requête "jaguar"

4. http ://www.google.com/intl/en/press/pressrel/universalsearch_20070516.html
5. http ://au.search.yahoo.com/
6. http ://www.ask.com/
7. http ://www.live.com/

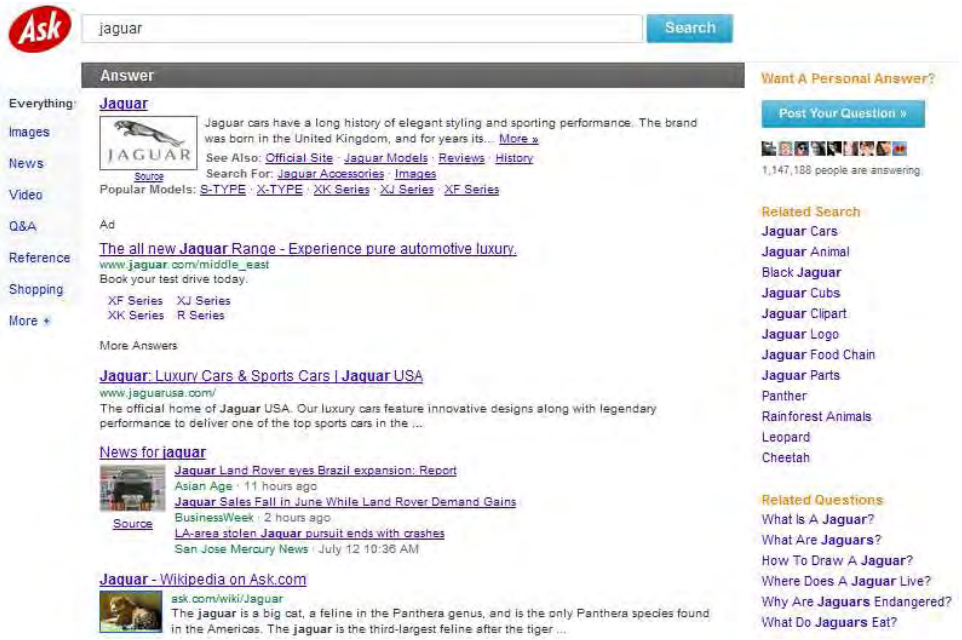

FIGURE 3.2 – Agrégation des résultats renvoyés par ASK pour la requête "jaguar"

Une autre façon d'aborder l'agrégation et aller ainsi au-delà de la notion de liste ordonnée, est de présenter ces résultats sous forme de clusters. Dans [237], Zeng *et al.* proposent une approche basée sur le regroupement (clustering). Ils considèrent que le regroupement des résultats de recherche dans des clusters permet d'avoir des documents qui se concentrent sur certains aspects de la requête. Exemple de moteur de recherche qui se base sur la technique de regroupement, on trouve clusty [8].

Une autre approche commune pour fournir une telle vue d'ensemble est le résumé multi-documents. On trouve plusieurs systèmes qui adoptent cette technique pour agréger des résultats de recherche. Par exemple, WebInEssence [73], NewsInEssence [72], NewsBlaster [152] et QCS [74].

D'autres approches combinent à la fois deux techniques à savoir le regroupement et le résumé multi-documents proposé par Sushmita *et al.* [209]. En fait, il s'agit de construire un document fictif à partir d'un regroupement des résultats par un moteur de recherche sous forme des clusters. Ce document fictif est considéré comme la réponse à la requête ou chaque chaque cluster correspond à des résumés de documents web retournés. Une amélioration considérable de l'espace résultat de l'utilisateur est constatée.

Le modèle de recherche orienté liste ordonnée devient moins efficace lorsque les informations demandées par l'utilisateur ne sont pas contenues dans un

8. http ://www.clusty.com

document unique, ou même dans une seule catégorie de ressource. On peut citer plusieurs exemples de requêtes pour lesquelles il est nécessaire de collecter et d'assembler les informations pertinentes sous forme d'une réponse ("Avatar trailer", "kamini", "Chelsea fc", etc.). Ce nouveau paradigme de RI agrégée a été défini lors de l'atelier SIGIR'2008 :
"*Aggregated search is the task of searching and assembling information from a variety of sources, placing it into a single interface*" [158].

3.3.2 Domaines d'application de la RI agrégée

La question d'agrégation de résultats a été abordée dans différents domaines. Nous illustrons dans ce qui suit les différentes instances de la RI agrégée vu sous des angles différents.

3.3.2.1 RI agrégée relationnelle

Un des cadres de RI qui demande l'agrégation des résultats est la RI agrégée relationnelle. Ce type de RI agrégée porte sur deux approches à savoir la recherche orientée entité ainsi que la recherche relationnelle.

- **Recherche orientée entité :** les entités nommées sont des concepts communs qui appartiennent à des catégories tels que les emplacements, noms de personnes, organisations, etc. Ils sont aussi appelés des instances de classes [24, 15, 122, 125].
  Kato *et al.* [115] ont montré qu'environ 71% des requêtes de recherche Web contiennent des entités nommées. Une autre étude récente [27] sur les fichiers *logs* a révélé qu'environ 73% à 87% des requêtes contiennent des entités nommées et qu'environ 18% à 39% des requêtes sont des entités nommées.
  Quand on interroge sur l'entité, on peut alors retourner un lot des informations de ce sujet. Dans la littérature, il existe des approches qui prennent une entité comme une requête et retourne un contenu connexe tel que la page d'accueil Wikipedia de l'entité [24, 25], d'images [31], de profil d'une personne dans un réseau social [235], etc.
- **Recherche relationnelle :** les approches d'extraction des entités tels que les noms de personnes, lieux, organisations, etc. permettent aussi de déterminer leurs relations tels que "John works for Motorola".
  Dans [48], les auteurs identifient les différents types de requêtes qui peuvent être satisfaites par la recherche relationnelle. Pour illustrer, nous pouvons donner quelques exemples tels que "French wines", "Capital of France", "features of iPhone" [122]. La première requête peut être répondue avec une liste d'instances (entités nommées) alors que la seconde avec un at-

tribut et le troisième avec de nombreux attributs.
La recherche relationnelle utilise des techniques d'extraction d'information [15] et de fouille des données semi-structurées [47]. Les techniques existantes peuvent découvrir des extraits d'information et leurs relations. Néanmoins, leur utilisation pour la RI reste limitée.

3.3.2.2 Recherche verticale

La recherche verticale [20, 71, 158, 206, 124, 123] traite l'agrégation des résultats de recherche provenant de différents moteurs verticaux. Un moteur vertical peut être un moteur d'images, vidéos, actualités, etc. Ce type de recherche permet aux utilisateurs d'interroger différents moteurs verticaux à partir de la même interface. Le contenu pertinent peut être clairsemé dans les différentes sources.

3.3.2.3 Autres perspectives de la RI agrégée

La RI agrégée peut être appliquer dans des domaines spécifiques. Les approches ci-après sont parfois trop spécifiques, mais il est important de les présenter parce qu'elles sont intéressantes et bien répandu dans la littérature.

- La RI agrégée est appliquée dans un service de recherche unifiée de NAVER [164], le premier moteur de recherche coréen. Ce moteur de recherche permet aux utilisateurs de rechercher dans diverses collections de documents.
- La RI agrégée est exploitée dans la recherche dans des bibliothèques numériques. Strotmann *et al.* [205] introduisent deux graphes à base de structure pour aider à naviguer dans des résultats de recherche. Le premier est un graphe sur les documents regroupés par auteur. Le second est un graphe des auteurs avec des liens basés sur l'analyse de co-citation.
- La RI agrégée est utilisée également en sciences sociales. Kaptein et Marx [161] extraient et agrègent les concepts retrouvés, leurs relations, les méthodes de recherche et l'information contextuelle. Les résultats peuvent ensuite être consultées par la méthode, la relation ou le concept de recherche. Pour chaque concept de recherche, l'utilisateur reçoit un résumé de l'information contextuelle.
- Le regroupement des actualités en fonction de la similitude et le temps a montré un effet bénéfique [189, 96]. Articles de presse à thèmes similaires et date de publication peuvent représenter l'historique d'un thème. Une telle organisation peut aider l'utilisateur à concentrer sa recherche dans un sujet et un intervalle de temps [139].
  Un contenu multimédia peut être juxtaposé à cette historique [188]. C'est

le cas pour Google News[9] (voir figure 3.3). Rohr *et al.* [188] proposent un calendrier afin de montrer l'évolution d'un thème.

FIGURE 3.3 – Résultats retournés par Google News pour la requête "chelsea", consulté en avril 2009 [121]

- La recherche géographique est devenu un axe de recherche très intéressant en RI [221, 108, 198]. L'information se rapporte à la situation géographique où les choses se passent dans un lieu géographique déterminé. Les personnes et leurs tâches sont liées à leurs positions. Cette relation devient importante lorsqu'on recherche des entités géographiques ou lorsqu'on personnalise la recherche en fonction du lieu de l'utilisateur [94, 156, 39]. Les entités géographiques peuvent être associées à d'autres types de contenu : des images [145, 120], entités liées nommés [222], actualités, etc. Ces relations peuvent devenir utiles pour d'autres RI agrégée inter-verticale ou de recherche Web.
- Enfin, on trouvera d'autres approches utilisant le paradigme de RI agrégée dans la recherche fédérée [21, 13, 102, 49, 91], les applications mash-up[10] [92, 181], les approches QR[11] [155, 232], les approches de GAT[12]

9. http ://news.google.com/
10. Les mash-up sont des outils agrégateurs et manipulateurs interactifs de données. Elles combinent d'une manière séquentielle ou parallèle des sources (contenu ou service) provenant de plusieurs applications plus ou moins hétérogènes dans des domaines spécifiques
11. QR : Question-Réponse
12. GAT : Génération Automatique de Textes

[162, 210, 163] et les discours politiques [114].

3.3.3 Problématique de la RI agrégée

Bien qu'il paraît un peu abstrait au départ, plusieurs questions se posent dans la RI agrégée. Dans ce qui suit, nous citons quelques-unes mentionnées dans [121] : **Identifier le type de réponse :** le contenu des réponses renvoyées aux requêtes peut être différent. Pour certaines requêtes, une seule unité d'information suffit comme réponse, d'autres demandent de multiples unités. Des requêtes telles que "Capital of France", "BBC home page", "height of Everest", "definition of Brontosaurus" peuvent être répondues par une seule unité d'information, tandis que des requêtes telles que "French wines by region", "ratings of Nokia E72", "Chinese restaurants at New York" et "all about Nokia E72" demandent de multiples unités. **Identifier les unités d'information les plus pertinentes :** en RI agrégée, nous pouvons récupérer des unités d'information avec des granularités différentes et de types différents. Cela permet d'avoir une réponse finale plus exhaustive. Il n'est pas anodin d'identifier les unités qui devraient être utilisées pour composer la réponse finale. Quand devrions-nous utiliser une unité d'information au lieu d'un document entier ? Quand devrions-nous utiliser le contenu multimédia (images, vidéos, etc.) ? Quand devrions-nous utiliser les moteurs de recherche spécialisés (recherche d'images, de recherche de vidéos de recherche nouvelles, etc.) ? C'est une des questions les plus difficiles dans ce domaine. **Assembler les différentes unités d'information dans un document cohérent :** la RI agrégée peut impliquer toutes les manières possibles d'assembler les résultats de recherche. Cela peut être un résumé, deux images et une définition, une table relationnelle, etc. L'un des objectifs de la RI agrégée est de choisir la meilleure agrégation selon les résultats de recherche disponibles. Quelle est la forme à laquelle le résultat final pourrait ressembler, il doit être lisible et cohérent. La principale question est de savoir comment assembler et évaluer la pertinence des résultats agrégés vis-à-vis de la requête, sachant qu'il est impossible de construire a priori toutes les combinaisons possibles des résultats.

3.4 RI agrégée dans les documents semi-structurés

3.4.1 Problématique

Comme nous l'avons mentionné dans le chapitre précédent, un problème principal de la RIS est comment sélectionner l'unité d'information qui répond le mieux à une requête de type CO [111, 80]. La plupart des approches en RIS

[202, 160, 127, 128, 177] considère que les unités retournées sont sous forme d'une liste d'éléments disjoints. Pour notre part, nous considérons que cette unité pertinente n'est pas nécessairement des éléments adjacents ou un document, elle pourrait aussi être une agrégation d'éléments de ce document.
Soit par exemple, un document XML de structure illustrée par la figure 3.4. Si nous supposons que l'unité d'information pertinente est composée d'éléments "*title*" et "*paragraph[2]*", situés au niveau de l'élément "*section[2]*". Les autres éléments ne sont pas sollicités par l'utilisateur. La majorité des systèmes de RIS retournent le document en entier comme réponse à la requête. Afin d'élaguer les éléments non-pertinents de la réponse, nous considérons que l'unité d'information retournée est l'agrégat (ensemble d'éléments) formé des deux éléments "*title*" et "*paragraph[2]*".

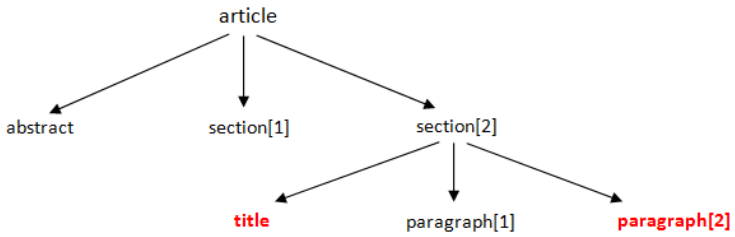

FIGURE 3.4 – Exemple d'une structure d'un document XML

L'idée derrière la sélection d'un ensemble d'éléments au lieu d'un élément tout seul vient du fait qu'un élément pourrait être partiellement pertinents pour une requête, alors qu'un ensemble d'éléments pourrait produire une meilleure réponse à l'utilisateur.

Nous présentons dans ce qui suit les premières tentatives proposées permettant de répondre à cette problématique, à savoir la RI agrégée dans des documents XML.

3.4.2 Agrégation des documents XML

La question de l'agrégation des éléments XML a reçu peu d'attention dans la littérature. En fait, le seul travail qui fait de l'agrégation dans des documents XML, au sens strict du terme, est celui proposé par Bessai et Alimazighi [29]. Pour cela, elles présentent un modèle pour la RIS, basé sur les réseaux possibilistes. Les relations document-éléments et éléments-termes sont modélisées par des mesures de possibilité et de nécessité. Dans ce modèle, la requête de l'utilisateur déclenche un processus de propagation pour retrouver des documents ou des unités d'information nécessairement ou au moins possiblement pertinents

par rapport à la requête. De plus, elles interprètent la notion de pertinence par deux dimensions :

- une dimension qui mesure à quel point il est certain qu'une "composition d'éléments d'un document" est pertinente vis-à-vis de la requête ;
- une dimension qui mesure à quel point il est possible qu'une "composition d'éléments d'un document" est possiblement pertinente pour la requête.

Pour évaluer leur approche, les expérimentations sont menées sur une sous-collection d'INEX 2005 (utilise un ensemble d'articles IEEE).

On trouve également des approches qui représentent les résultats d'une requête sous forme des résumés de documents XML. Par exemple, eXtract [99] est un système de RIS qui génère des résultats sous forme des fragments à partir des documents XML (films [13]). Un fragment XML est qualifié comme résultat s'il répond à quatre caractéristiques : autonome (compréhensif par l'utilisateur), distinct (différent des autres fragments), représentatif (des sujets de la requête) et succinct. On trouve également d'autres approches qui s'adressent au problème d'affichage des résultats de la recherche dans des documents XML [98, 142].

3.4.3 Motivations

Contrairement aux approches citées précédemment, nous proposons un modèle permettant de sélectionner automatiquement des éléments XML qui répondent le mieux à une requête de type CO à partir de chaque document ainsi que leur agrégation dans un même résultat. Afin d'assurer que les éléments assemblés ne véhiculent pas la même information et afin de diversifier les résultats retournés par notre modèle, nous avons ajouté une première hypothèse de non-redondance sur les deux sources d'évidence (le contenu et la structure). Nous proposons également une deuxième hypothèse de complémentarité ne permettant d'assembler que des éléments porteurs de l'information pertinente et additionnelle. Le défi majeur de ce travail est de sélectionner et d'assembler des éléments pertinents, non redondants et complémentaires, et s'ils sont susceptibles de mieux répondre à la requête tous ensemble qu'une liste d'éléments pris séparément.

Le modèle que nous proposons trouve ses fondements théoriques dans les réseaux bayésiens. La structure réseau fournit une manière naturelle de représenter les liens entre les éléments du corpus de documents XML et leurs contenus. Quant à la théorie des probabilités, elle permet d'estimer de manière qualitative et quantitative les différents liens sous-jacents. Elle permet notamment d'exprimer le fait qu'un terme est probablement pertinent vis-à-vis d'un élément et de

13. http ://infolab.standford.edu/pub/movies

mesurer à quel point une réponse à la requête contient des éléments pertinents, non-redondants et complémentaires.

3.5 Évaluation des systèmes de RI agrégée

L'évaluation d'un SRI consiste à mesurer ses performances et estimer sa capacité à répondre aux besoins en information des utilisateurs. La performance ou la qualité d'un SRI est mesurée en comparant les réponses du système renvoyés à l'utilisateur pour une requête donnée, aux réponses idéales que l'utilisateur espère recevoir. Dans la littérature, différents modèles d'évaluation des SRI sont proposés tels que les modèles d'évaluation orientés laboratoire, les modèles d'évaluation par utilisation des contextes réels (user studies), etc.

3.5.1 Limites des modèles d'évaluation orientés laboratoire en RI agrégée

Les premiers modèles d'évaluation des SRI sont basées sur une approche de type laboratoire (où *laboratory-based model*) initiée par *Cleverdon* [60] dans le cadre du projet *Cranfield project II*. Cette approche fournit des ressources de base pour l'évaluation d'un SRI, notamment une collection de requêtes, une collection de documents et des jugements de pertinence associés à chaque requête. Ce modèle d'évaluation orienté laboratoire est adopté dans les campagnes d'évaluation telles que TREC, INEX, etc.

L'évaluation de la RI agrégée engendre de nouvelles problématiques liées, en particulier, à la notion de document en RI agrégée et l'absence des métriques d'évaluation spécifiques.

3.5.1.1 Absence de la notion de document en RI agrégée

De manière générale, la RI agrégée peut être vue comme un moyen permettant d'assembler dans un même agrégat, du contenu pertinent provenant de plusieurs sources susceptibles de comporter une partie de l'information pertinente pour la requête.

Dans le but de comparer les agrégats résultats fournis par un système de RI agrégée et les agrégats que souhaite recevoir l'utilisateur, il faut spécifier pour chaque requête l'ensemble de réponses idéales du point de vue utilisateur. La spécification des jugements de pertinence d'agrégats associés à la requête

constituent la tâche la plus difficile dans la construction d'une collection de test. À la différence des modèles d'évaluation orientés laboratoire où les documents pertinents doivent être connus et complets pour chaque requête. En bref, la notion de document n'existe pas dans la RI agrégée.

3.5.1.2 Insuffisance des métriques quantitatives

Les métriques d'évaluation classiques tels que le rappel et la précision sont des mesures quantitatives considérées insuffisantes pour l'évaluation des systèmes de RI agrégée. En effet, l'évaluation par le biais de ces mesures se fait par rapport au nombre de documents retrouvés par le système. Ces mesures ne permettent pas d'évaluer la qualité d'un agrégat construit. Il s'agit d'évaluer, à un rang donné, un ensemble d'éléments qui peut comporter des bons et mauvais éléments : un tout pertinent ou non ! Il n'existe cependant pas des métriques spécifiques pour estimer cette qualité.

3.5.2 Modèles d'évaluation orientés RI agrégée

Jusqu'à présent, différentes méthodes d'évaluation ont été menées pour mesurer les performances des systèmes de RI agrégée. Ces méthodes sont assez hétérogènes parce qu'elles ont été conçues avec des objectifs différents. Nous pouvons les classer par rapport à leur objectif. Dans [20, 136, 140], l'objectif principal est d'évaluer la sélection des sources. Dans [206, 208, 213], l'objectif principal est de comparer les interfaces de la RI agrégée inter-verticale. Dans [19], l'objectif d'évaluer les résultats de la RI agrégée. Dans [29], l'objectif principal est de montrer l'intérêt de la RI agrégée dans des corpus de documents XML. Nous allons décrire ci-après les différentes méthodes d'évaluation.

Un protocole commun pour évaluer la sélection des sources est de demander aux participants de choisir qu'elles sont les sources pertinentes pour une requête. Liu *et al.* [140] ont effectué ce type de jugement de pertinence sur 2153 requêtes Web génériques. Dans [20], Arguello *et al.* ont évalué les résultats de recherche de 25195 requêtes en utilisant des données des utilisateurs issus des fichiers logs d'un moteur de recherche. Ce type d'évaluations est rapide, mais pas nécessairement exacte. Dans ce type jugement, on pourrait ne pas deviner le besoin d'information réelle ou négliger certaines interprétations de la requête et certaines requêtes peuvent exiger des connaissances spécifiques.

Dans [206, 208], Sushmita *et al.* comparent l'efficacité de différentes interfaces pour la RI agrégée inter-verticale. Ils montrent que les utilisateurs trouvent des résultats plus pertinents lorsque les résultats de la RI agrégée inter-

verticale sont placés ensemble avec des résultats Web. Ils montrent également que placer les résultats de la RI agrégée inter-verticale au-dessus, au-dessous ou au milieu des résultat Web peut affecter la qualité de la recherche. Dans les deux études, les participants ont montré un grand intérêt d'avoir des résultats issus des sources différentes.

Sushmita *et al.* proposent d'examiner le comportement d'utilisateurs envers les concepts proposés tels que *digest pages* (pages sommaires) et *aggregated digest pages*. Dans [35], diverses simulations des situations de tâches sont conçues à cette fin. Les résultats et les observations déduits par ces simulations peuvent informer les auteurs si les concepts proposés mèneront à une augmentation d'espaces de résultat et s'ils font que les approches sont les plus efficaces et pourquoi.

Au lieu d'évaluer les performances des systèmes via les jugements des utilisateurs, les évaluations de pertinence ont été simulées à l'aide de fichiers logs d'un moteur de recherche [207, 71, 208]. Dans [71], Diaz montre que les requêtes qui obtiennent un taux élevé dans les fichiers logs des actualités sont probablement plus intéressantes. Les fichiers logs sont également utilisés dans [208]. Sushmita *et al.* ont montré que pour certaines sources telles que la vidéo, les comportements d'utilisateurs sont déterminés à partir de fichiers logs et différents. Bien que les fichiers logs permettent une évaluation à grande échelle automatique, ils ne peuvent pas être aussi réalistes qu'une utilisation des contextes réels.

Récemment, Arguello *et al.* [19] ont proposé une méthodologie pour évaluer le classement des résultats de la RI agrégée. La pertinence des évaluations sont par paires de préférences entre des ensembles de résultats. Chaque ensemble de résultats contient des résultats issus d'une seule source. Ce travail ne se concentre pas sur la notion de pertinence de la source, mais plutôt sur l'efficacité relative au classement des résultats.

Zhou *et al.* [240] proposent de bâtir une référence d'évaluation (benchmark) pour la RI agrégée inter-verticale à travers la réutilisation des références d'évaluation existantes. Les auteurs utilisent la tâche ClueWeb dans TREC [57] et construisent artificiellement des collections verticales par classification. Puis, ils choisissent des requêtes qui couvrent de nombreuse sources. Ce travail est considéré comme une étape vers l'évaluation des performances des SRI, même si un effort plus substantiel est nécessaire dans ce sens pour rendre la distribution des requêtes, des sources et des évaluations plus réalistes.

Bessai et Alimazighi [29] ont proposé une méthode d'évaluation afin de valider leur modèle de RI agrégée dans des documents XML. Un questionnaire a été conçu afin de récupérer les jugements des utilisateurs et permettre l'analyse des résultats. Ce questionnaire contient une description de la tâche d'évaluation, des requêtes ainsi que des questions sur le résultat obtenu par le prototype.

3.5.3 Discussion

L'évaluation des performances des systèmes de RI agrégée reste un problème ouvert. Il existe différents types d'évaluation de pertinence, différentes mesures, alors qu'il n'y a pas encore un protocole d'évaluation commun. En particulier, il n'est pas clair *quels sont les avantages de ces approches ?*, et *comment devraient-elles être évaluées ?* Nous savons que la RI agrégée inter-verticale peut fournir une orientation sur la diversité et l'exhaustivité des résultats, mais nous ne savons pas pourquoi et à quel point cette recherche peut contribuer à la RI. Les travaux de recherche doivent examiner de plus sur l'intérêt des méthodes d'évaluation orientées RI agrégée.

3.6 Conclusion

Nous avons donnée dans ce chapitre un bref aperçu sur la question de la RI agrégée. Nous avons montré quelques exemples de domaines dans lesquels la RI agrégée a un sens. Nous avons présenté le processus général suivi par ce type de recherche ainsi que les problématiques liées à chacune des étapes.

Nous avons également montré que peu de travaux de recherche ont assuré la RIS sous l'angle de l'agrégation des résultats. Nous avons également mis en évidence les problèmes liés à l'évaluation de ce type de recherche. Dans cette optique, nous développons dans la deuxième partie de ce manuscrit notre modèle de RI agrégée dans des documents XML.

Deuxième partie

Un Modèle de Recherche d'Information agrégée dans des documents XML basé sur les Réseaux Bayésiens

Chapitre 4

Un Modèle de RI Agrégée basé sur les Réseaux Bayésiens

4.1 Introduction

L'agrégation des éléments XML en RIS a été peu étudiée en littérature. En fait, comme nous l'avons signalé précédemment, la seule approche qui traite de cette problématique est celle de Bessai et Alimazighi [29]. Unes de limites de cette approche vient du fait que les agrégats peuvent contenir des éléments redondants et/ou non complémentaires. Ces propriétés ne sont pas pris en compte dans cette approche alors que la nôtre les permis. De plus le modèle proposé se base sur un cadre possibiliste alors que dans notre cas, nous nous appuyons sur un cadre probabiliste.

Dans ce chapitre, nous proposons une approche de RI agrégée des éléments XML basée sur les RB. En effet, nous proposons d'assembler automatiquement les éléments qui répondent le mieux au besoin de l'utilisateur formulé à travers une liste des mots-clés. On se limite à de requêtes de type CO. Chaque agrégat, qualifié comme réponse à la requête à partir d'un document XML, doit satisfaire aux trois propriétés suivantes : *pertinence*, *non-redondance* et *complémentarité*.

Le modèle que nous proposons trouve ses fondements théoriques dans les RB. La structure réseau fournit une manière naturelle de représenter les documents, les éléments ainsi que la requête. La La théorie des probabilités permet de mesurer les différentes valeurs sous-jacentes du modèle. Ces valeurs permettent notamment de mesurer à quel point un agrégat contient des éléments potentiellement pertinents, non-redondants et complémentaires.

Ce chapitre est organisé comme suit. La section 4.2 présente brièvement le

cadre théorique sur lequel repose notre modèle, à savoir les RB. Nous détaillons dans la section 4.3 le modèle que nous proposons. Ce modèle est basé sur un RB défini par une composante qualitative et une composante quantitative :

- la composante qualitative représente les nœuds documents, éléments, termes d'indexation et la requête et les relations de dépendance existant entre eux ;
- la composante quantitative mesure les poids des nœuds par les degrés de probabilité.

La section 4.4 illustre ce modèle par un exemple. La dernière section 4.5 conclut le chapitre.

4.2 Les Réseaux bayésiens

Les réseaux bayésiens, qui doivent leur nom aux travaux de Thomas Bayes au XVIIIe siècle sur la théorie des probabilités, sont le résultat de recherches effectuées dans les années 1980, dues à J. Pearl à UCLA et à une équipe de recherche danoise à l'Université de Aalborg. Aujourd'hui, les réseaux bayésiens se sont révélés des outils très pratiques pour la représentation de connaissances incertaines, et le raisonnement à partir d'informations incomplètes.

Définition 1 (Réseau bayésien) *Un réseau bayésien $\mathcal{B}=(\mathcal{G}, \theta)$ peut être formellement défini par :*
$\mathcal{G} = (V,E)$, un graphe acyclique orienté où V est l'ensemble des nœuds de $\mathcal{G}$, et E l'ensemble des arcs de $\mathcal{G}$.
$\theta = \{P(V_i|Pa(V_i)\}$ ensemble des probabilités de chaque nœud V_i conditionnellement à l'état de ses parents $Pa(V_i)$ dans $\mathcal{G}$.

Ainsi, un graphe est appréhendé selon un aspect qualitatif et un aspect quantitatif. L'aspect qualitatif du graphe indique les dépendances (ou indépendances) entre les variables et donne un outil visuel de représentation des connaissances, outil plus facilement appréhendable par ses utilisateurs. De plus, l'utilisation de probabilités permet de prendre en compte l'incertain, en quantifiant les dépendances entre les variables, c'est l'aspect quantitatif.

Dans [165], J. Pearl a aussi montré que les réseaux bayésiens permettaient de représenter de manière compacte la distribution de probabilité jointe sur l'ensemble des variables :

$$P(V_1, V_2, .., V_n) = \prod_{i=1}^{n} P(V_i|Pa(V_i)) \tag{4.1}$$

Cette décomposition d'une fonction globale en un produit de termes locaux dépendant uniquement du nœud considéré et de ses parents dans le graphe, est une propriété fondamentale des réseaux bayésiens. Elle permet de calculer $P(V)$ d'une manière plus rapide lorsqu'il y a des dépendances entre les variables. Elle est à la base des premiers travaux portant sur le développement d'algorithmes d'inférence, qui calculent la probabilité de n'importe quelle variable du modèle à partir de l'observation même partielle des autres variables.

4.3 Un modèle de RI agrégée basé sur les RB

4.3.1 Motivations

Les travaux qui nous proposons ont pour but de définir un modèle de RIS permettant l'agrégation des éléments XML. D'une manière générale, quel que soit le modèle proposé dans la littérature, et particulièrement ceux qui assemblent les résultats de la recherche soit par regroupement, résumé multi-documents ou agrégation, la non-redondance et la complémentarité des résultats renvoyés ne sont pas considérées.

Nous nous sommes particulièrement penchés dans nos travaux sur la résolution de trois points qui nous paraissent essentiels pour un modèle efficace et fiable en RI agrégée sur des documents XML :

- dans le premier point, nous estimons que la pertinence d'un terme dans un élément d'une configuration donnée dépend d'une part de l'ensemble d'éléments constituant la configuration et d'autre part de la collection de documents. De ce fait, l'information non disponible dans un élément à un impact sur l'importance de cet élément dans l'ensemble d'éléments récupérés. Notre modèle est basé sur les RB, les mesures de probabilités permettent de représenter l'importance d'un élément dans un document et dans la collection.
- le second point traite la redondance d'éléments véhiculant la même information. En effet, nous estimons que le fait de renvoyer des éléments qui sont similaires induit à du bruit. Nous suggérons tout d'abord d'appliquer une contrainte au niveau de la structure : les éléments d'un agrégat ne doivent pas avoir une relation d'inclusion entre eux (non-overlapping). La seconde contrainte renforce la première et sera appliquée au niveau de contenu.
- le dernier point, la complémentarité, est étroitement lié au premier point. Il découle de fait qu'on cherche à assembler dans un agrégat d'éléments qui ajoutent ce qui manquait en matière d'informations pertinentes.

Notre objectif est de permettre à un utilisateur de localiser les informations

les plus pertinentes, non-redondantes et complémentaires répondant complètement à ses besoins.

4.3.2 Architecture générale du modèle

Le modèle que nous proposons est représenté par un réseau bayésien de topologie illustrée par la figure 4.1. D'un point de vue qualitatif, le graphe permet de représenter un document XML, ses éléments, les termes d'indexation et la requête. Les arcs orientés permettent de représenter les relations de dépendances entre les différents nœuds. Ces relations sont issues de la représentation DOM [1] d'un document XML. D'un point de vue quantitatif, notre modèle estime des valeurs sur les nœuds à l'aide des mesures de probabilité.

Le nœud D représente un document de la collection C. Un document D est représenté par une variable aléatoire binaire, prenant ses valeurs dans le domaine $D = \{d, \neg d\}$. L'instanciation (ou activation) d'un nœud document, $D = d$ (resp. $\neg d$) signifie que le document est pertinent (resp. non pertinent) étant donnée une requête. Nous nous intéressons qu'au cas où le document $D = d$ est activé, et nous le notons d.

Les nœuds E_1, E_2,..., E_n représentent les éléments du document d. Chaque nœud E_j représente une variable aléatoire binaire prenant des valeurs dans l'ensemble $\{e_j, \neg e_j\}$. L'instanciation $E_j = e_j$ signifie que l'élément E_j est indexé par au moins un nœud terme.

Les nœuds T_1, T_2,..., T_m sont les nœuds termes d'indexation. Chaque nœud terme T_i représente une variable aléatoire binaire prenant des valeurs dans le domaine $dom(T_i)$=$\{t_i, \neg t_i\}$ où l'instanciation $T_i = t_i$ signifie que le terme T_i est présent dans le nœud père auquel il est relié c'est-à-dire le nœud balise e_j contient ce terme t_i. Il faut noter qu'un terme est relié aussi bien au nœud qui le comporte ainsi qu'à tous les ascendants de ce nœud.

Une requête Q, prend ses valeurs dans le domaine $dom(Q)$=$\{q, \neg q\}$. Nous sommes intéressés par l'instanciation de la requête, nous ne considérons que le cas où la requête est instanciée positivement $Q = q$, c'est-à-dire la requête introduit de l'information à travers le RB, et nous noterons Q indifféremment lorsque cela ne prête pas à confusion.

Le passage du document vers la représentation sous forme de RB se fait de manière assez simple. Il consiste à garder la structure du document d et assigner des valeurs aux différents nœuds.

1. DOM : Document Object Model

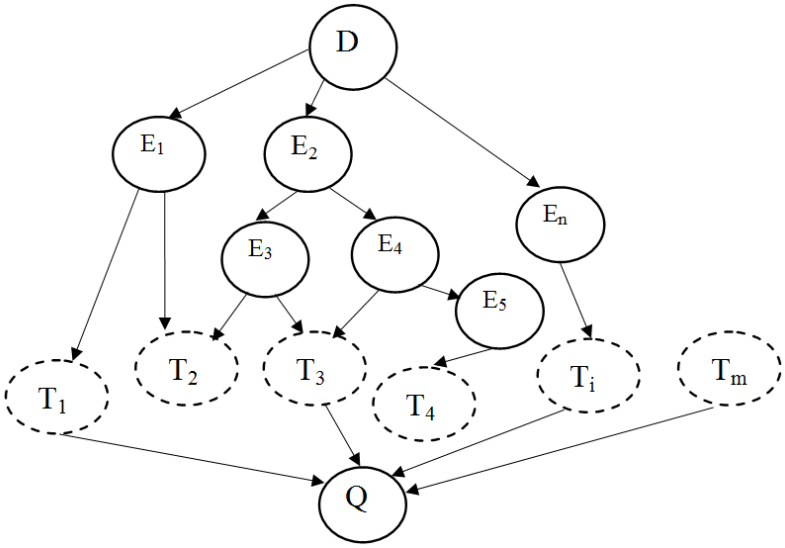

FIGURE 4.1 – Architecture simplifiée par document du modèle proposé

Considérons le sous-réseau composé du nœud document et des éléments. Les arcs sont orientés du nœud document vers les nœuds éléments exprimant les relations de dépendance entre les deux types de nœuds.

Considérons maintenant le sous-réseau composé des nœuds éléments et termes d'indexation. Les termes de ce sous-réseau n'ont une existence que parce qu'ils apparaissent dans ces nœuds éléments qui sont leurs parents. Chaque élément e_j (variable structurelle), $e_j \in E$ avec $E = \{e_1, ..., e_n\}$ dépend directement de son nœud parent dans le RB du document d. Chaque terme $t_i \in T$ avec $T = \{t_1, ..., t_m\}$, dépend uniquement des éléments où il apparaît. Il faut également noter que la représentation fait apparaître un seul document (voir figure 4.1). En fait, nous considérons que les documents sont indépendants les uns des autres, et donc nous pouvons raisonner en considérant le sous-réseau qui représente le document que nous le traitons.

Considérons à présent le sous-réseau constitué de la requête et ses termes d'indexation. La requête exprime une demande d'information à travers une liste de termes mais elle peut aussi en exclure d'autres. La requête propage l'information aux nœuds termes qui figurent dans la collection. Ces nœuds termes forment les nœuds parents de la requête. Un terme d'indexation de la requête n'apparaissant pas dans un document donné sera considéré comme un nœud terme racine, n'ayant pas de parents.

Le système est instancié par la soumission de la requête. L'instanciation de la requête propage l'information à travers le réseau en activant les nœuds termes d'indexation, parents de la requête. Cette instanciation consiste à injecter la requête à travers les arcs activés du réseau pour rechercher les documents et

les éléments pertinents par rapport à la requête. Soit θ_i cette instanciation, $\theta_i = \{E_1, E_3, E_5\}$ noté $\{e_1, e_3, e_5\}$ est un exemple d'une configuration déduite à partir de la figure 4.1. Une configuration donnée est considérée comme un résultat de la recherche. L'ensemble des instances possibles est noté θ.

Nous supposons que la requête Q est composée d'une simple liste de mots-clés : $Q = \{t_1, ..., t_m\}$. L'importance relative des termes entre eux est ignorée et nous notons $T(Q)$ l'ensemble des termes d'indexation de la requête Q, et $T(E)$ l'ensemble des termes d'indexation des éléments du document d. Les termes de la requête qui indexent les éléments de documents, $t_i \in (T(Q) \wedge T(E))$, sont évalués dans le contexte de leurs parents par $P(t_i|e_j)$, et séparés des termes de la requête absents des éléments de documents.

4.3.3 Évaluation de la requête par propagation

L'évaluation de la requête est effectuée par la propagation de l'information apportée par la requête à travers le réseau. Dans notre modèle, le processus de propagation est similaire à la propagation probabiliste bayésienne [28, 33]. Le processus d'évaluation consiste à propager l'information *injectée* par le nœud requête vers le nœud document. Les arcs reliés à la requête sont instanciés dans le but de calculer pour chaque configuration potentielle (instanciation de nœuds éléments) sa valeur de pertinence et complémentarité étant donnée cette requête. À l'issue du processus de propagation, chaque configuration aura un score global de pertinence et de complémentarité. La configuration retenue est celle qui présente le plus grand score. Cette configuration représentative d'un document forme un agrégat. Cet agrégat est le résultat de la recherche dans ce document pour une requête donnée.

Nous décrivons dans ce qui suit, les différentes étapes pour propager une requête donnée vers le nœud document.

Le modèle est instancié à la réception de la requête. Il existe une configuration possible des parents de la requête qui correspond aux nœuds termes, qui représentent la requête sous sa forme la plus stricte (exactement telle que formulée par l'utilisateur).

Le processus de propagation évalue les valeurs de probabilité entre tous les éléments d'une configuration θ_i. Dans ce modèle, la probabilité jointe d'observer une requête Q et son résultat de recherche (réponse) θ_i dans un document d est donnée par :

$$P(Q, \theta_i, d) = \sum_{\overrightarrow{T(Q)}} P(Q|T(Q)) \times P(T(Q)|\theta_i) \times P(\theta_i|d) \times P(d) \qquad (4.2)$$

$\overrightarrow{T(Q)}$ représente l'ensemble des configurations possibles des parents de Q.

La quantification totale de la pertinence et complémentarité d'une configuration d'éléments revient à quantifier chaque membre de la formule 4.2. Afin de simplifier notre modèle[2], nous nous restreignons tout d'abord au cas où $T(Q)$ ne contient que des instanciations positives des termes figurant dans la requête. Ensuite, des probabilités *a priori* sont affectées aux documents de la collection, égales à $P(d) = \frac{1}{N}$ (en fait, un seul document est instancié à la fois, excluant l'instanciation des autres documents de la collection), mais elles sont supprimées du calcul de la propagation globale parce que ce membre de la formule 4.2 est considéré comme un coefficient uniforme appliqué à tous les documents de la collection. Donc, la formule 4.2 sera simplifiée par :

$$P(Q, \theta_i, d) = P(Q|T(Q)) \times P(T(Q)|\theta_i) \times P(\theta_i|d) \tag{4.3}$$

La section 4.3.4 décrit les différentes façons que nous proposons pour estimer la valeur de probabilité du premier membre de la formule 4.3. Par la suite, nous donnons les pondérations attribuées aux termes d'indexation des éléments dans les configurations dans la section 4.3.5. Ceci correspond bien au deuxième membre de la formule 4.3. Dans la section 4.3.6, nous élaguons les configurations qui sont superflus avec la contrainte structurelle de redondance. Finalement, nous traitons le troisième membre de la formule 4.3. Il s'agit d'estimer la valeur de la complémentarité entre les éléments d'une configuration donnée dans la section 4.3.7.

4.3.4 Agrégation des termes de la requête

La probabilité de la requête étant donnée les termes d'indexation, $P(Q|T(Q))$, dépend de l'interprétation de la requête. Plusieurs interprétations sont possibles. Les termes de la requête peuvent être connectés par une conjonction, une disjonction, ou par une somme probabiliste, ou encore une somme probabiliste pondérée. Ces deux dernières agrégations ont déjà été proposées dans les travaux de Turtle [220] et Boughanem *et al.* [36].

L'idée majeure de l'agrégation des termes de la requête est de mesurer

2. L'utilisation des RB en RI a été un challenge à cause de deux principaux problèmes liés à leur utilisation : *(i)* le temps de calcul des distributions de probabilité et l'espace nécessaire à leur stockage augmentent d'une manière exponentielle avec le nombre de nœuds dans le réseau ; *(ii)* la complexité de la propagation de l'information, c'est-à-dire les inférences nécessaires à propager l'information, dans un réseau est un problème NP-complet [62] (Ceci parce que dans les réseaux généraux, il peut exister plusieurs chemins entre les paires de nœuds du graphe).

la conformité d'une configuration possible, en l'occurrence celle trouvée dans un élément donné, avec la configuration des termes de la requête. Pour ce faire, pour toute configuration, $T(Q)$ de $\overrightarrow{T(Q)}$, la probabilité conditionnelle $P(Q|T(Q))$ est spécifiée par des fonctions d'agrégation en fusionnant les fonctions de ressemblance élémentaires $P(Q|T_k = t_k)$. Chaque $P(Q|t_k)$ est le poids de la conformité entre l'instance t_k du terme T_k avec celle de la requête (dans Q). Une fonction de ressemblance élémentaire évalue donc à quel point une instance d'un terme dans une configuration donnée ressemble à l'instanciation de ce même terme dans la requête. Cette configuration est en fait la configuration telle que trouvée dans un document.

Le stockage de toutes les configurations possibles des termes de la requête est coûteux en espace et le temps de calcul croît de manière exponentielle avec le nombre de termes parents de la requête. En effet, une requête, Q de domaine binaire, composée de 20 termes de domaines binaires aussi, nécessite 2×2^{20} calculs de configurations possibles. Dans notre cas, nous nous intéressons uniquement au cas $Q = q$, que nous notons Q pour simplifier. Une organisation possible serait de pondérer chaque terme de la requête et de calculer le poids de la jointure des termes de la requête. Lorsque l'utilisateur ne fournit aucune information sur les opérateurs d'agrégation de sa requête, l'unique connaissance disponible est l'importance du terme dans la collection. Cette connaissance est disponible pour chaque terme.

Nous supposons aussi que les termes sont indépendants. En fait, les modèles basés sur les RB existants supposent l'indépendance entre les termes pour faciliter les calculs, toutefois cette supposition entrave l'exactitude de ces modèles. Mais, les conclusions des expérimentations sur différentes collections d'évaluation sont mitigées. En effet, la prise en compte des relations de dépendances entre les termes ne sont pas toujours avérées efficaces en termes de précision [41].
Le premier membre de la formule 4.3 sera transformé en :

$$\begin{aligned} P(Q|T(Q)) &= P(Q|T_1, ..., T_m) \\ &= \prod_{T_k \in T(Q)} P(Q|T_k) \end{aligned} \tag{4.4}$$

Nous donnons dans ce qui suit les différentes techniques que nous proposons pour agréger les termes de la requête. Ces techniques sont inspirées des travaux de Boughanem *et al.* [36, 40]

4.3.4.1 Agrégations booléennes des termes de la requête

Conjonction : pour une requête booléenne, ET, le processus d'évaluation restitue les éléments contenant tous les termes de la requête. Ainsi,

$$P(Q|T_k) = \begin{cases} 1 & si \quad T_k = t_k, \\ 0 & sinon. \end{cases} \tag{4.5}$$

La probabilité de la requête Q étant donnée une configuration possible, $T(Q)$, de $\overrightarrow{T(Q)}$ de tous ses parents est donnée par :

$$P(Q|T(Q)) = \begin{cases} 1 & si \quad \forall T_k \in T(Q), T_k = t_k, \\ 0 & sinon. \end{cases} \tag{4.6}$$

Dans 4.6, il faut que chaque terme T_k parent de la requête Q soit instancié dans $T(Q)$ comme dans la requête. Les éléments pertinents pour ce type de requête sont les éléments contenant simultanément tous ses termes.

Disjonction : pour une requête booléenne, OU, un élément est plus ou moins pertinent s'il contient au moins un terme d'indexation de la requête. La pertinence finale d'une configuration augmente avec le nombre de termes de la requête présents. La conjonction pure est manipulée en remplaçant $\forall$ par $\exists$ dans la requête conjonctive 4.6.

$$P(Q|T(Q)) = \begin{cases} 1 & si \quad \exists T_k \in T(Q), T_k = t_k, \\ 0 & sinon. \end{cases} \tag{4.7}$$

Cette interprétation est trop large pour discriminer entre les éléments. Dans le cas de la disjonction, le système restitue les éléments contenant au moins un terme de la requête. La configuration contenant tous les termes de la requête peut être restituée avec un score de pertinence plus faible qu'une autre configuration ne contenant qu'un terme de la requête. Dans notre approche, le calcul de la pertinence d'une configuration vis-à-vis d'une requête dépend de la valeur maximum des instances des configurations des parents de la requête. Ce maximum atteint rapidement la valeur 1, il suffit pour cela qu'au moins un terme de la requête soit instancié telle que dans la configuration. Le score de pertinence finale d'une configuration donnée dépend des poids des termes de la requête présents et absents dans l'ensemble d'éléments en question. Ainsi, soit une requête Q composée des deux termes t_1, t_2. Il n'est pas impossible que l'élément e_1 contenant le terme t_1 se retrouve avec un score de pertinence plus élevé que celui d'un élément e_2 contenant les deux termes de la requête.

Négation : la requête peut contenir la négation d'un terme, signifiant que l'utilisateur ne veut pas voir ce terme dans l'élément restitué. Lorsque l'élément contient ce terme alors la pertinence est nulle. La négation d'un terme est une

opération unaire. Ainsi :

$$P(Q|T_k) = \begin{cases} 1 & si \quad T_k = \neg t_k, \\ 0 & sinon. \end{cases} \tag{4.8}$$

Le terme parent de la requête doit être instancié à *non représentatif* lorsque la requête contient la négation du terme.

4.3.4.2 Quantification des termes de la requête

Supposons qu'une requête est satisfaite par un élément si elle contient au moins K termes communs avec l'élément. Nous considérons une fonction croissante, $f(\frac{K(T(Q)}{n})$, tel que $K(T(Q))$ est le nombre de termes de la requête instanciés dans une configuration donnée de $T(Q)$, et que la requête contient n termes. Nous posons $f(0) = 0$ et $f(1) = 1$. f est un quantificateur flou [234]. Par exemple,

$$f(\frac{i}{n}) = \begin{cases} 1 & si \quad i \geq \frac{K(T(Q))}{n}, \\ 0 & sinon. \end{cases} \tag{4.9}$$

Pour l'agrégation donnée par 4.9 il faut qu'au moins K termes de la requête soient en conformité avec $T(Q)$. D'une manière générale, f peut être une fonction non booléenne.
L'approche quantifiée pour calculer la probabilité d'une requête Q étant donnée une configuration $T(Q)$ de tous ses parents, est donnée par :

$$P(Q|T_k) = f(\frac{K(T(Q))}{n}) \tag{4.10}$$

Le tableau 4.1 présente les résultats d'une quantification sur une requête Q contenant trois termes $T1, T2, T3$. Pour cette quantification, la configuration est considérée "conforme" si au moins deux termes ont la même instanciation que dans la requête. Le choix du nombre de termes *satisfaits* de la requête reste arbitraire. Dans ce cas, cette attribution peut être une fonctionnalité du système, ou bien l'utilisateur peut spécifier dans sa requête le nombre de termes indexant l'élément à partir du quel il considère sa requête comme satisfaite. Par exemple, il peut introduire des quantificateurs du type "au moins deux termes". D'autre part, cette quantification, comme dans le cas d'une agrégation disjonctive de la requête, ne permet pas de discriminer entre les documents de la collection. En effet, seul le nombre de termes satisfaits est considéré. L'importance du terme satisfait (par exemple terme rare, terme fréquent dans la collection) n'est pas considérée.

T_1	T_2	T_3	$P(Q\|T(Q))$
t_1	t_2	t_3	1
t_1	t_2	$\neg t_3$	1
t_1	$\neg t_2$	t_3	1
t_1	$\neg t_2$	$\neg t_3$	0
$\neg t_1$	t_2	t_3	1
$\neg t_1$	t_2	$\neg t_3$	0
$\neg t_1$	$\neg t_2$	t_3	0
$\neg t_1$	$\neg t_2$	$\neg t_3$	0

TABLE 4.1 – Agrégation quantifiée des termes de la requête $P(Q|T(Q))$

La combinaison des termes de la requête peut être basée sur le "noisy-Or" [107, 36, 166]. Cet opérateur permet de quantifier les termes de la requête instanciés dans une configuration donnée comme dans la requête. Par souci de simplification de calcul, nous nous limitons à des agrégations booléennes dans notre modèle.

4.3.5 Pertinence

Nous présentons dans cette section les pondérations que nous avons proposées pour les termes d'indexation. Ces pondérations sont reliées aux relations de dépendance existantes entre un nœud terme et ses parents s'ils existent. En effet, lors du calcul de la pertinence d'une configuration de termes dans une configuration d'éléments, certains termes apparaissent dans les éléments et la requête et d'autres n'apparaissent pas dans les éléments. Dans nos travaux actuels, les termes absents dans une configuration sont considérés lors des calculs de la pertinence afin d'éviter le problème d'éléments nuls. Un terme en relation sémantique ou statistique à un terme de la requête et présent dans un élément peut apporter de l'information supplémentaire et peut constituer un élément intéressant à intégrer dans le calcul de la pertinence d'une configuration donnée.

Pour évaluer la probabilité qu'une configuration de termes d'indexation fasse partie dans une configuration d'éléments, le deuxième membre de la formule 4.3 sera transformé en :

$$\begin{aligned} P(T(Q)|\theta_i) &= P(T_1, ..., T_m|\theta_i) \\ &= \prod_{T_k \in T(Q)} P(T_k = t_k|\theta_i) \\ &= \prod_{t_k \in T(Q)} P(t_k|\theta_i) \end{aligned} \tag{4.11}$$

Dans une configuration donnée, un terme représentatif d'un élément est un terme qui contribue à sa restitution en réponse à une requête. La probabilité que le terme t_k fasse partie d'une configuration θ_i est calculée par $P(t_k|\theta_i)$. En fait, nous avons besoin de cette probabilité pour déterminer la pertinence de cette configuration de termes d'indexation dans une configuration d'éléments. Cette probabilité est estimée par : seulement les termes instanciés et qui apparaissent à la fois dans la configuration de termes $T(Q)$ et la configuration d'éléments θ_i sont considérés. Nous supposons que les termes de $T(Q)$ sont indépendants. La probabilité $P(t_k|\theta_i)$ peut être estimée en utilisant une estimation du maximum de vraisemblance sur la fréquence du terme t_i dans θ_i. Ceci correspond au premier facteur de la formule 4.12. Afin d'éviter le problème des fréquences nulles des quelques termes (quand un terme ne figure pas dans une configuration θ_i et éventuellement dans ses éléments), il faut ajouter la fréquence du terme dans la collection avec celle calculée avec le document (premier facteur de la formule 4.12). Ceci correspond au deuxième facteur de la formule 4.12. La formule 4.12 correspond en fait à une technique de lissage de type Dirichlet [238] mais appliquée à chaque élément XML.

$$\begin{aligned} P(t_k|\theta_i) &= (1-\lambda_t)\frac{tf(t_k,\theta_i)}{\sum_{\forall t\in d} tf(t,d)} + \lambda_t \frac{tf(t_k)}{\sum_{\forall t\in C} tf(t)} \\ &= (1-\lambda_t)\frac{\sum_{\forall e_j\in\theta_i} tf(t_k,e_j)}{\sum_{\forall t\in d} tf(t,d)} + \lambda_t \frac{tf(t_k)}{\sum_{\forall t\in C} tf(t)} \end{aligned} \quad (4.12)$$

Avec :

1. $tf(t_k,\theta_i)$ est la fréquence du terme t_k dans l'ensemble des éléments formant la configuration θ_i.
2. $tf(t,d)$ est la fréquence du terme t dans le document d.
3. $tf(t_k,e_j)$ est la fréquence du terme t_k dans l'élément e_j.
4. $tf(t_k)$ est la fréquence du terme t_k dans la collection de documents C.
5. $\lambda_t = \frac{\mu}{|d|+\mu}$. $\lambda_t \in [0;1]$ est un paramètre de lissage.
6. μ est une constante égale à μ=300.

4.3.6 Redondance

Définition 2 (Redondance) *Nous considérons que deux éléments sont redondants si et seulement si ils véhiculent la même information.*

Dans chaque configuration, nous nous sommes intéressés à l'agrégation d'éléments qui ne véhiculent pas la même information. La redondance est traitée dans notre modèle au niveau structurel avec une première hypothèse (**H1**) quand un agrégat est construit à partir d'un document. Une deuxième

hypothèse (**H2**) sera appliquée au niveau du contenu quand notre processus est généralisée : agrégat multi-documents [3].

- **H1** : cette hypothèse est qualifiée comme contrainte de structure ou d'inclusion permettant d'éliminer les redondances. Nous considérons que la présence d'une relation ancêtre-descendant entre deux éléments signifie que l'un est inclus dans l'autre. Autrement, nous supposons qu'un utilisateur préfère ne pas avoir des éléments imbriqués dans une configuration donnée parce que ces éléments véhiculent les mêmes informations mais à des granularité différentes. Par exemple, dans la figure 4.1, les éléments e_4 et e_5 ne doivent pas figurer dans la même configuration. De même pour l'élément e_2 et e_5. Par contre, dans une telle configuration, nous pouvons avoir à la fois les éléments e_3 et e_5 qui portent des informations différentes.
- **H2** : cette hypothèse est considérée comme une contrainte de contenu ou de détection de nouveauté/redondance. Nous supposons qu'un utilisateur préfère retrouver dans une configuration donnée des éléments non redondants à partir de plusieurs documents. Par souci de simplicité, nous supposons que la détection de nouveauté/redondance est effectuée entre les éléments d'une configuration donnée qui sont censés être pertinents. Nous formulons cette problématique par la mesure suivante $Redondance(e_i, \theta_i)$ basée sur l'hypothèse que la redondance d'un élément e_j dépend de la configuration θ_i. Dans la littérature et dans le cadre de la campagne d'évaluation TREC, nous trouvons les approches les plus étroitement liés à la détection de nouveauté/redondance de Clarke *et al.* [59] qui proposent un cadre d'évaluation dans TREC afin de mesurer systématiquement la nouveauté et la diversité. La mesure proposée se base sur le gain cumulé $nxCG$ (voir formule 2.14). Nous trouvons également d'autres approches qui se basent sur la technique de clustering pour mesurer la redondance d'un document par sa distance à chaque cluster dans [153, 204, 78]. Zhang *et al.* proposent dans [239], une autre mesure de la redondance en se basant sur la distance entre un document et chacun des autres documents. Pour simplifier notre modèle, nous utilisons la mesure de similarité *cosinus* [4] pour détecter la redondance entre les éléments de résultats de recherche. Nous supposons que la redondance d'un élément e_j dépend de θ_i, l'ensemble des éléments qualifiés comme réponse à la requête Q. Nous utilisons $Redondance(e_j, \theta_i)$ pour mesurer si e_j est redondant avec θ_i. Une façon de calculer cette redondance est de considérer e_j et θ_i représentés sous forme de vecteurs de termes.

$$Redondance(e_j, \theta_i) = cosinus(\overrightarrow{e_j}, \overrightarrow{\theta_i}) \tag{4.13}$$

Une autre façon de faire ce calcul est de mesurer la similarité entre e_j et

3. C'est un agrégat généré à partir de plusieurs documents.
4. cosine similarity, en anglais.

chacun des éléments e_p de θ_i.

$$Redondance(e_j, \theta_i) = \max_{j \neq p, \forall e_p \in \theta_i} cosinus(\overrightarrow{e_j}, \overrightarrow{e_p}) \tag{4.14}$$

Dans notre modèle, nous utilisons la formule 4.14 pour détecter les éléments redondants dans une configuration donnée θ_i.

4.3.7 Complémentarité

Définition 3 (Compléméntarité) *Nous considérons que deux éléments sont complémentaires si et seulement si l'un apporte de l'information pertinente et additionnelle à l'autre.*

Le troisième membre de la formule 4.3, $P(\theta_i|d)$, mesure la complémentarité entre les éléments d'une configuration possible. On considère que les éléments regroupés dans une telle configuration sont indépendants alors les hypothèses d'indépendance conditionnelle nous permettent ensuite d'écrire :

$$P(\theta_i|d) = \prod_{e_j \in \theta_i} P(e_j|d) \tag{4.15}$$

L'intérêt de propager une information complémentaire d'un élément e_j vers la racine du document d dans une configuration donnée θ_i indique à quel point cet élément ajoute ce qu'il manquait en matière d'information à cette configuration. On suppose que les éléments loin du nœud racine du document d paraissent plus porteurs d'informations complémentaires que ceux situés là-haut du document. Intuitivement, plus la distance entre un élément et la racine est grande, plus il contribue à la complémentarité des éléments de la configuration θ_i. Nous modélisons cette intuition par l'utilisation dans la fonction de propagation de complémentarité les deux variables $dist(d, e_j)$ et $dist(d, deepdown(e_j))$, qui représentent respectivement la distance entre le nœud racine d et un de ses nœuds descendants e_j du document (relativement à une configuration donnée θ_i), et la profondeur maximale de la branche qui passe par le nœud interne e_j noté $deepdown(e_j)$. La distance entre deux nœuds quelconques est déterminée par le nombre d'arcs qui les séparent. La mesure de probabilité de propagation d'un élément e_j, supposé complémentaire dans une configuration θ_i, vers le nœud racine d est quantifiée comme suit :

$$P(e_j|d) = \frac{dist(d, e_j)}{dist(d, deepdown(e_j))} \tag{4.16}$$

La formule 4.16 indique que plus un nœud est proche de la racine, moins il contribue à la complémentarité d'une configuration donnée. À titre d'exemple et dans la figure 4.1, les contributions des éléments E_2 et E_4 notés respectivement

e_2 et e_4 (dans ce cas, l'élément le plus profond est E_5 et sera noté par e_5), dans la complémentarité d'une configuration θ_i seront estimés comme suit :

$$P(e_2|d) = \frac{dist(d, e_2)}{dist(d, e_5)} = \frac{1}{3} \tag{4.17}$$

$$P(e_4|d) = \frac{dist(d, e_4)}{dist(d, e_5)} = \frac{2}{3} \tag{4.18}$$

Finalement, la probabilité jointe de la formule 4.3 se simplifie en :

$$P(Q, \theta_i, d) = \prod_{t_k \in T(Q)} P(Q|t_k) \times \prod_{t_k \in T(Q)} P(t_k|\theta_i) \times \prod_{e_j \in \theta_i} P(e_j|d) \tag{4.19}$$

Dans notre modèle la configuration qui sera sélectionnée est celle qui, comporte les termes de la requête, maximise la pertinence et la complémentarité de ses éléments et élimine ceux qui sont redondants. Cette configuration représentative d'un document forme un agrégat : un résultat de la recherche de la requête dans le document.

Les deux notions redondance et complémentarité seront discutées dans la section 5.4.5 du chapitre suivant (5).

4.4 Illustration du modèle proposé

Le but de cette section est de faire une exécution à la main de notre modèle. Pour illustrer notre approche, nous avons pris un exemple d'une requête de type CO : "pyramids of Egypt", cherchant des éléments (title, abstract, section, paragraph, etc.) dans des documents XML sur les pyramides d'Egypte. Un exemple de document XML (un extrait d'un document) ainsi que le RB qui lui est associé sont présentés respectivement dans les figures 4.2 et 4.3.

```
<Article>
<Title>Great Pyramid of Egypt </Title>
<Abstract>It's the oldest ... of the Seven Wonders... .
It's ... the Great Pyramid ... </Abstract>
<Section>
<TitleSection>Construction </TitleSection>
<Paragraph>At construction, the Great Pyramid was ...</Paragraph>
</Section>
...
</Article>
```

FIGURE 4.2 – Extrait d'un document XML

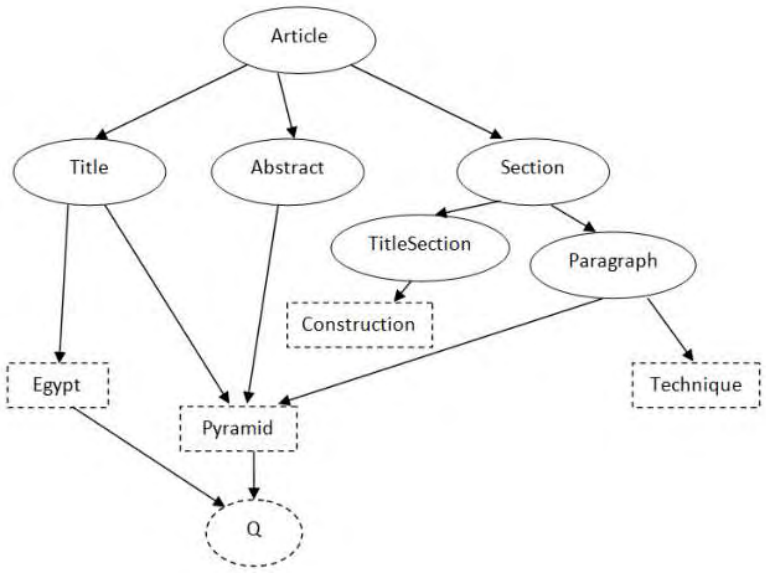

FIGURE 4.3 – Réseau bayésien relatif à la requête et au document XML

Pour cet exemple, l'ensemble des éléments relatifs au document est présenté par $E = \{e_1 = Title, e_2 = Abstract, e_3 = Section, e_4 = TitleSection, e_5 = Paragraph\}$. L'ensemble des termes d'indexation des éléments, calculé en utilisant le contenu de chaque élément ainsi que celui de ses éléments fils dans chaque configuration, est tel que $T(E) = \{t_1 = Egypt, t_2 = Pyramid, t_3 = Technique, t_4 = Construction\}$. L'ensemble des termes d'indexation de la requête est $T(Q) = \{t_1 = Egypt, t_2 = Pyramid\}$. On ne considère que quelques termes pour ne pas encombrer l'exemple. Il s'agit de répondre à la requête Q contenant une fois chacun des termes t_1 et t_2.
La réception de la requête conduit à la propagation vers le nœud document. Le processus de propagation de l'information apportée par la requête entraîne le calcul des probabilités conditionnelles de chaque configuration d'un document étant donnée la requête selon la topologie du graphe dans la figure 4.3. Pour calculer les valeurs de pertinence et complémentarité de chaque configuration possible dans un document donné, nous avons besoin de calculer la probabilité jointe $P(Q, \theta_i, d)$ donnée par la formule 4.19.
D'une manière générale, le processus d'évaluation des configurations étant donnée une requête est déclenché pour tous les documents de la collection contenant au moins un terme de la requête. L'instanciation positive d'un document D, $D = d$, entraine le développement suivant :

Agrégation booléenne des termes de la requête : le tableau 4.2 donne les valeurs de la probabilité conditionnelle de la requête Q dans le contexte de ses parents, $T(Q)$. Les valeurs sont proposées pour une agrégation booléenne de type conjonctive, ET, et disjonctive, OU pour chaque configuration possible

des parents.

T_1T_2	**ET**	**OU**
t_1t_2	1	1
$t_1\ \neg t_2$	0	1
$\neg t_1\ t_2$	1	0
$\neg t_1\ \neg t_2$	0	0

TABLE 4.2 – Probabilités conditionnelles des parents de la requête, $T(Q)$

Lorsque la requête est en conjonction de termes, il n'existe qu'une seule configuration possible qui la satisfait, à savoir t_1, t_2. Dans l'exemple que nous présentons, le seul élément du document qui n'est indexé ni par t_1 ni t_2 est e_4. Logiquement, l'ensemble θ est égale à $2^5 - 1$ configurations possibles parce que nous avons cinq éléments dans $T(E)$ et la configuration vide n'est pas considérée. Et comme l'élément e_4 n'est pas indexé par aucun élément de la requête, le nombre des configurations possibles devient alors égale à $2^4 - 1$.

Redondance : l'ensemble des configurations générées doit vérifier la première hypothèse **H1**. Cette hypothèse, qualifiée comme étant une contrainte d'inclusion, exige que deux éléments dans une même configuration possible θ_i ne se chevauchent pas (not overlapping). En appliquant **H1**, pas mal des configurations seront élaguées à partir de θ. Nous avons réellement 11 configurations possibles parmi les $2^4 - 1$. En effet, dans une configuration donnée de la figure 4.3, nous ne pouvons pas avoir les deux éléments e_3 et e_5 parce qu'ils se chevauchent.

Le tableau 4.3 donne toutes les configurations possibles θ déduites à partir de la figure 4.3 qui respecte l'hypothèse **H1**.

Pertinence des termes dans les configurations : Le tableau 4.4 donne les probabilités conditionnelles des termes instanciés positivement étant donné une une configuration possible.

Nous rappelons qu'un terme est relié aussi bien au nœud qui le comporte ainsi qu'à tous les ascendants de ce nœud. Certaines valeurs considérées dans le tableau 4.4 sont prises à titre d'exemple. Elles ne correspondent pas toujours aux résultats des formules considérées car nous ne disposons pas de tous les paramètres pour effectuer le calcul.
Un point intéressant qui peut être remarqué, c'est que quand un terme de requête ne figure pas dans une configuration donnée, cette probabilité est lissée par la fréquence des termes dans la collection comme défini par la formule 4.12. Ces valeurs ne laissent pas de place pour une telle ignorance possible.

θ_i	e_1	e_2	e_3	e_5
θ_1	1	1	1	0
θ_2	1	1	0	1
θ_3	1	1	0	0
θ_4	1	0	1	0
θ_5	1	0	0	1
θ_6	1	0	0	0
θ_7	0	1	1	0
θ_8	0	1	0	1
θ_9	0	1	0	0
θ_{10}	0	0	1	0
θ_{11}	0	0	0	1

TABLE 4.3 – Ensemble des configurations possibles

$P(t_k\|\theta_i)$	t_1	t_2
$P(t_k\|\theta_1)$	0,17	0,25
$P(t_k\|\theta_2)$	0,22	0,34
$P(t_k\|\theta_3)$	0,19	0,219
$P(t_k\|\theta_4)$	0,114	0,108
$P(t_k\|\theta_5)$	0,121	0,17
$P(t_k\|\theta_6)$	0,24	0,1001
$P(t_k\|\theta_7)$	0,075	0,121
$P(t_k\|\theta_8)$	0,091	0,143
$P(t_k\|\theta_9)$	0,094	0,0911
$P(t_k\|\theta_{10})$	0,026	0,0897
$P(t_k\|\theta_{11})$	0,049	0,081

TABLE 4.4 – Distribution de probabilité $P(t_k|\theta_i)$

Complémentarité : la tableau 4.5 présente les probabilités conditionnelles d'un élément étant donné la racine du document où il apparaît. Les valeurs déterminées dans ce tableau sont basées sur la formule 4.16 à partir de la figure 4.3.

Sélection de l'agrégat : la probabilité jointe de la formule 4.19, pour chaque configuration, est déterminée dans le tableau 4.6. Ainsi, la configuration qui sera qualifiée comme réponse à la requête dans le document D est celle qui possède le meilleur score. Nous appelons cette configuration agrégat. Dans notre exemple, θ_2 est qualifié comme agrégat.

e_i	$P(e_j \mid d = Article)$
e_1	$\frac{1}{1}$=1
e_2	$\frac{1}{1}$=1
e_3	$\frac{1}{2}$=0,5
e_5	$\frac{2}{2}$=1

TABLE 4.5 – Distribution de probabilité $P(e_j|d)$

θ_i	**Score**
θ_1	0,02125
θ_2	**0,0748**
θ_3	0,04161
θ_4	0,006156
θ_5	0,02057
θ_6	0,024024
θ_7	0,0045375
θ_8	0,013013
θ_9	0,0085634
θ_{10}	0,0011661
θ_{11}	0,003969

TABLE 4.6 – Calcul du score de chaque configuration possible

D'une manière générale, les agrégats sont alors restitués par ordre décroisant de leur probabilité de pertinence et complémentarité. Nous montrons dans le chapitre des expérimentations (Chapitre 5) des agrégats assemblant des éléments pertinents, non redondants et complémentaires et nous discutons leurs effets sur les performances du système de RI ainsi que l'utilité d'une telle agrégation dans des documents XML.

4.5 Conclusion

Nous avons décrit dans ce chapitre un nouveau modèle de RI agrégée dans des documents XML. Ce modèle traite la pertinence, la redondance et la complémentarité des éléments assemblés dans des agrégats d'une manière originale basée sur la théorie des probabilités et particulièrement les réseaux bayésiens. Les nœuds dans ce réseau représente un document XML, ses éléments, les termes d'indexation et la requête. Les arcs entre les nœuds permettent de représenter les relations de dépendances entre les différents nœuds. Ces nœuds sont quantifiés par une mesure de probabilité afin de calculer un score pour chaque configuration possible. La configuration qui possède le meilleur score et qui répond à la première contrainte d'inclusion structurelle, sera qualifiée

comme le résultat de recherche dans le document d étant donné une requête Q. Et cette configuration sera appelé *agrégat*. Nos contributions peuvent être essentiellement en trois directions :

- assembler des éléments pertinents par documents ;
- élaguer ceux qui sont redondants en appliquant l'hypothèse **H1**. Si nous souhaitons générer des agrégats multi-documents, nous appliquons dans ce cas l'hypothèse **H2** ;
- favoriser dans la formule de calcul de score d'une configuration (cf. formule 4.15) les éléments qui se complètent mutuellement pour avoir une réponse plus complète (pertinence additionnelle).

Il est indéniable que les points cités ci-dessus sont étroitement liés. Finalement, nous avons tenté de proposer des poids aux termes dans le but de calculer le degré de spécificité dans une collection des documents. Ces poids ont été utilisés dans notre approche pour mesurer l'absence des termes de la requête des éléments d'une configuration lors de calcul des valeurs de pertinence (cf. formule 4.12). D'autre part, nous avons considéré que la restitution d'un agrégat en réponse à une requête peut être considérée dans un cadre d'inférence. En effet, la restitution d'un agrégat est "causée" par la soumission d'une requête au système. Les techniques sur lesquelles se basent la plus part des modèles en littérature pour restituer des agrégats ou une liste d'éléments en réponse à un besoin informationnel ne traitent pas les deux notions : redondance et complémentarité, alors que le mien les permis. Plutôt, ils se limitent à la notion pertinence.

Le dernier chapitre est consacré à la phase de mise à l'épreuve de nos propositions sur la collection de test INEX 2009.

Chapitre 5

Expérimentations

5.1 Introduction

Les expérimentations que nous décrivons dans ce chapitre ont été effectuées sur la collection de test fournie dans la cadre de la campagne d'évaluation INEX 2009. Nous avons développé un système de recherche agrégée basée sur le modèle inférentiel que nous avons proposé.

Nous avons mené deux types d'expérimentations. La première série d'évaluation mesure les performances de notre modèle en comparant notre résultat avec les meilleurs résultats enregistrés par les participants à INEX 2009. La seconde série d'évaluation concerne du cœur de notre modèle, évaluer l'intérêt de la pertinence d'un agrégat pour répondre à une requête ainsi que les impacts de la redondance et la complémentarité sur les performances des résultats enregistrés.

Ce chapitre est organisé comme suit. La section 5.3 présente la première série d'évaluation. Dans cette section, nous décrivons rapidement la collection de test utilisée, à savoir INEX 2009, la stratégie d'évaluation utilisée ainsi qu'une évaluation comparative avec les meilleurs résultats enregistrés selon la stratégie *Focused*. La seconde série d'expérimentations est décrite dans la section 5.4, en l'absence de protocole ainsi que de collections de test appropriés, nous avons élaboré notre propre cadre. Nous avons exploité aussi la collection INEX 2009 pour ce cadre. Dans cette section, nous présentons le protocole d'évaluation ainsi que l'analyse des résultats enregistrés de différentes expérimentations dans ce cadre afin d'évaluer l'impact de la RI agrégée.

5.2 Collection de test

Pour l'évaluation des performances, nous nous appuyons sur la collection de test fournie dans le cadre de la campagne d'évaluation INEX 2009.

5.2.1 Collection de documents

À partir de 2006 et jusqu'à 2008, la collection "Wikipedia" [69] a été utilisée dans la plupart des tâches. Cette collection de 6 Go, est composée de 659 388 documents d'une profondeur (nombre de niveaux) moyenne de 6,72. Le nombre moyen de nœuds XML par document est 161,35. Cette collection est également utilisée dans la tâche multimedia, elle contient environ 246 730 images.

En 2009, une extension de la collection Wikipedia est fournie [199]. Elle comporte 2 666 190 articles Wikipedia annotés et ayant une taille totale aux alentours de 50,7 Go. Cette collection contient 101 917 424 éléments XML ayant au moins 50 caractères (y compris les espaces blancs). Cette collection est utilisée dans la tâche adhoc ainsi que dans d'autres tâches.

5.2.2 Topics

Les topics adhoc ont été créés par les participants suivant des instructions précises. Les topics contenaient une courte requête CO, une option de requête structuré CAS, un titre, une ligne décrivant la requête et le récit avec quelques détails de la requête et le contexte de travail dans lequel le besoin d'information se pose. Pour les topics sans le champ $< castitle >$, par défaut requête CAS est ajouté sur la base de la requête CO : //*[about(., "CO-requête")]. La figure 5.1 présente un exemple d'une topic adhoc. En fait, 115 topics ont été sélectionnés pour faire l'évaluation dans la campagne INEX 2009 et sont numérotées 2009001-2009115 [85].

```
<topic id="2009114" ct_no="310">
  <title>self-portrait</title>
  <castitle>//painter//figure[about(.//caption, self-portrait)]</castitle>
  <phrasetitle>"self portrait"</phrasetitle>
  <description>Find self-portraits of painters.</description>
  <narrative>
    I am studying how painters visually depict themselves in their
    work.  Relevant document components are images of works of art, in
    combination with sufficient explanation (i.e., a reference to the
    artist and the fact that the artist him/herself is depicted in the
    work of art).  Also textual descriptions of these works, if
    sufficiently detailed, can be relevant.  Document components
    discussing the portrayal of artists in general are not relevant, as
    are artists that figure in painters of other artists.
  </narrative>
</topic>
```

FIGURE 5.1 – Topic 2009114 de la campagne INEX 2009

5.3 Évaluation du modèle selon la stratégie de recherche *Focused* d'INEX

En absence de cadre approprié pour l'évaluation de la pertinence des agrégats, nous avons adapté notre agrégat pour répondre à la stratégie de recherche *Focused* définis dans la cadre d'INEX.
Nous allons décrire dans ce qui suit la stratégie de recherche *Focused*, la collection évaluée ainsi que la manière dont nous avons adapté notre résultat pour pouvoir effectuer ces évaluations.

5.3.1 Stratégie de recherche *Focused* d'INEX

Plusieurs stratégies de recherche sont proposées dans la tâche ad-hoc, parmi lesquelles on peut citer la stratégie "focused". Cette stratégie consiste à décider quels éléments doivent être retournés en se focalisant sur le besoin de l'utilisateur. Ces éléments doivent être les plus exhaustifs et spécifiques et ne doivent pas être imbriqués les uns dans les autres. Ce type de recherche suppose que l'utilisateur préfère l'élément (un seul) le plus pertinent d'un sous arbre pertinent [112].

5.3.2 Adaptation de notre résultat

Nous rappelons que dans notre approche, nous renvoyons des agrégats. Un agrégat comporte un ensemble d'éléments non redondants et complémentaires. Dans cette expérimentation, nous trions les éléments d'un agrégat selon un score de pertinence. Ainsi, nous comparons les éléments de notre agrégat avec la liste d'éléments renvoyés par les meilleurs résultats enregistrés par les participants à

INEX 2009. Pour que les résultats soient comparables, nous avons transformé nos agrégats sous forme d'une liste. Pour cela, nous parcourons les agrégats en largeur et en longueur afin de construire une liste d'éléments équivalente à celle retournée par les participants selon la stratégie de recherche *Focused*.

5.3.3 Résultats

Dans cette expérimentation, nous utilisons les mesures officielles pour l'évaluation de notre résultat à savoir la précision interpolée selon certains niveaux de rappel sélectionnés $iP[x]$ et la moyenne de ces précisions interpolées moyennées $MAiP$ selon 101 niveaux de rappel [112]. L'intérêt de ces mesures est d'évaluer la pertinence des fragments de document et pas du document entier. Pour cela, le rappel et la précision ne sont pas calculés en terme de nombre de documents mais plutôt en terme de quantité d'information exprimée grâce au nombre de caractères. Ces mesures sont déjà présenté dans le chapitre 2, section 2.7.4.2.

Le tableau 5.1 présente les meilleures résultats obtenus par les participants à INEX 2009 selon la stratégie *Focused* en utilisant uniquement des requêtes CO. La dernière ligne de ce tableau présente les résultats enregistrés par notre approche. La première colonne détermine le rang des runs. La deuxième colonne donne l'identifiant de chaque run. De la troisième à la cinquième colonne, nous donnons la précision interpolée aux points de rappel 0%, 1% et 5%. La dernière colonne donne la $MAiP$ sur les 101 niveaux de rappel (0%, 1%, ..., 100%).

D'après les résultats enregistrés, l'approche proposée est moins performantes que les approches existantes. En comparant notre résultat aux autres, nous remarquons que seulement sept résultats utilisent des requêtes de type CO. Les trois résultats suivants : le cinquième (p6-UamsFSsec2docbi100), le sixième (p5-BM25BOTrangeFOC) et le septième (p16-Spirix09R001) utilisent des requêtes de type CAS. Le premier résultat (p78-UWatFERBM25F) effectue une recherche par passage (passage retrieval). Le deuxième résultat (p68-I09LIP6-Okapi), le quatrième résultat (p60-UJM15525) et le septième résultat (p16-Spirix09-R001) récupèrent seulement des articles complets. Par élimination, il nous reste à comparer notre résultat avec ceux qui sont les plus spécifiques, à savoir le troisième (p10-MPII-COFoBM), le huitième (p48-LIG-2009-focused-1F), le neuvième (p22-emse2009-150) et le dixième (p25-ruc-term-coF), puisque notre approche récupère uniquement les éléments les plus spécifiques. Ça montre bien que notre résultat vient juste après ces quatre.

Rang	Participant	iP[0,00]	iP[0,01]	iP[0,05]	MAiP
1	p78-UWatFERBM25F	0,6797	**0,6333**	0,5006	0,1854
2	p68-I09LIP6Okapi	0,6244	**0,6141**	0,5823	0,3001
3	p10-MPII-COFoBM	0,6740	**0,6134**	0,5222	0,1973
4	p60-UJM-15525	0,6241	**0,6060**	0,5742	0,2890
5	p6-UamsFSsec2docbi100	0,6328	**0,5997**	0,5140	0,1928
6	p5-BM25BOTrangeFOC	0,6049	**0,5992**	0,5619	0,2912
7	p16-Spirix09R001	0,6081	**0,5903**	0,5342	0,2865
8	p48-LIG-2009-focused-1F	0,5861	**0,5853**	0,5431	0,2702
9	p22-emse2009-150	0,6671	**0,5844**	0,4396	0,1470
10	p25-ruc-term-coF	0,6128	**0,4973**	0,3307	0,0741
11	Notre résultat	0,5659	**0,4935**	0,3112	0,06547

TABLE 5.1 – Comparaison des résultats enregistrés dans le cas de la tâche CO de la collection INEX 2009 selon la stratégie Focused

5.4 Évaluation du modèle d'agrégation

En raison de l'absence d'un cadre approprié pour évaluer la pertinence des agrégats, nous avons adopté une stratégie d'évaluation basée sur l'utilisation des utilisateurs sollicités pour évaluer la pertinence des éléments agrégés.

Pour réaliser cette série d'expérimentations, nous avons sélectionné un ensemble de vingt requêtes CO. Ces requêtes sont numérotés 2009n avec n : 001-006, 010-015, 020, 023, 026, 028, 029, 033, 035, 036. Pour les participants, nous avons sollicité vingt-trois utilisateurs (doctorants et étudiants en M2) de notre laboratoire pour évaluer ces requêtes. La tâche d'évaluation est la suivante. Pour chaque requête soumise au système, le résultat de la recherche est une liste ordonnée des agrégats (voir formule 4.19). En moyenne, cinq agrégats par requête évalués par les utilisateurs. Chaque requête a été évaluée par quinze utilisateurs.

L'utilisateur juge chaque agrégat en fonction de trois dimensions : la pertinence (voir la section 5.4.2 pour plus de détails), la redondance (voir la section 5.4.3 pour plus de détails) et la complémentarité (voir la section 5.4.4 pour plus de détails).

5.4.1 Distribution d'éléments

Dans cette expérimentation, nous mesurons le nombre moyen d'éléments retourné par agrégat et par requête. L'objectif est d'étudier l'effet de la première hypothèse **H1** (voir section 4.3.6). En fait, les agrégats ne sont pas des éléments uniques. Nous arrivons à récupérer un agrégat par document qui est souvent formé de plusieurs éléments. Ensuite, nous construisons une série de cinq agrégats par requête. Nous faisons la somme des éléments constituant les agrégats formés et nous divisons cette somme par cinq afin de déterminer le nombre moyen d'éléments par agrégat et par requête.
La figure 5.2 présente la répartition des vingt requêtes CO sur la base des éléments retournés. En moyenne, il y avait cinq éléments par agrégat qui sont retournés. Nous constatons que pour les requêtes suivantes : Q001, Q002, Q003, Q010, Q012, Q014, Q020, Q026, Q028, Q033, Q035 et Q036, le nombre d'éléments retournés est inférieur à la moyenne globale. Ceci est du en raison de l'hypothèse **H1** qui permet d'élaguer les éléments qui se chevauchent.

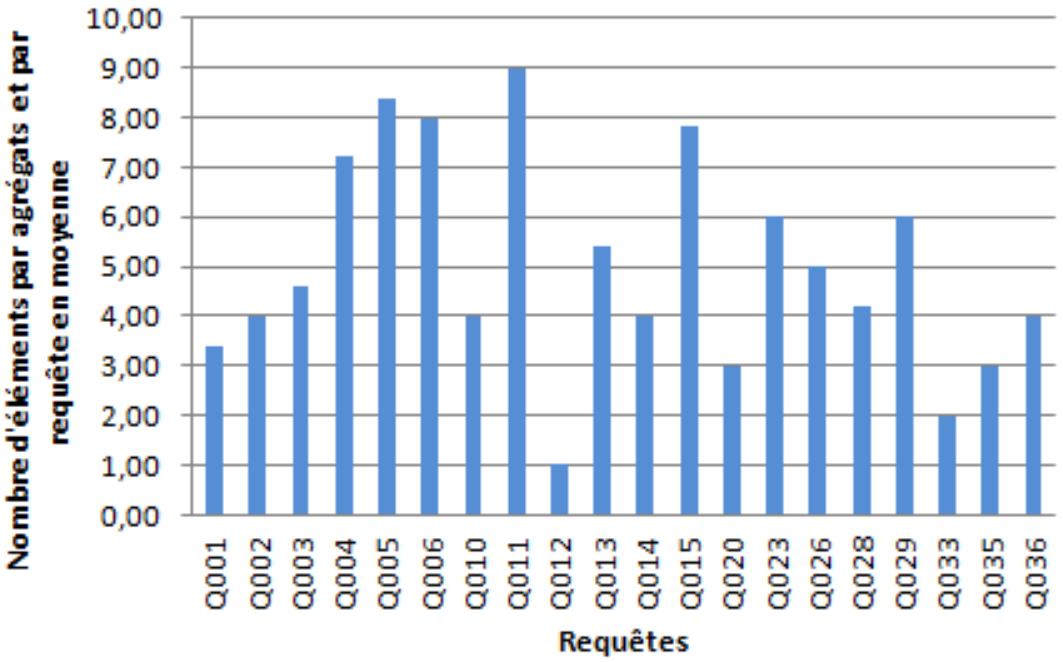

FIGURE 5.2 – Impact de l'hypothèse **H1** sur le nombre d'éléments par agrégat et par requête

5.4.2 Évaluation de la pertinence d'agrégats

Notre objectif dans cette section est d'évaluer la pertinence d'un agrégat. Pour cela, nous avons demandé aux utilisateurs de juger la pertinence d'un agrégat en fonction de trois niveaux de pertinence définis comme suit :

Définition 4 (Agrégat totalement pertinent) *Un agrégat est totalement pertinent si tous ses éléments sont pertinents.*

Définition 5 (Agrégat partiellement pertinent) *Un agrégat est partiellement pertinente s'il contient des éléments pertinents.*

Définition 6 (Agrégat non pertinent) *Un agrégat est non pertinent s'il ne contient que des éléments non pertinents.*

Dans cette première expérimentation, nous étudions la pertinence des agrégats avant d'appliquer l'hypothèse **H2** (voir section 4.3.6). Ainsi, il est possible d'avoir des éléments redondants dans un agrégat. La figure 5.3 liste le pourcentage d'agrégats pertinents, non pertinents et partiellement pertinents par requête sur l'ensemble des utilisateurs.
Les premiers résultats intéressants montrent que 87% d'agrégats sont pertinents, soit 29% totalement ou 58% partiellement pertinents. Les résultats montrent que seulement 13% des agrégats ne sont pas pertinents (la moyenne de la partie verte de la figure 5.3).

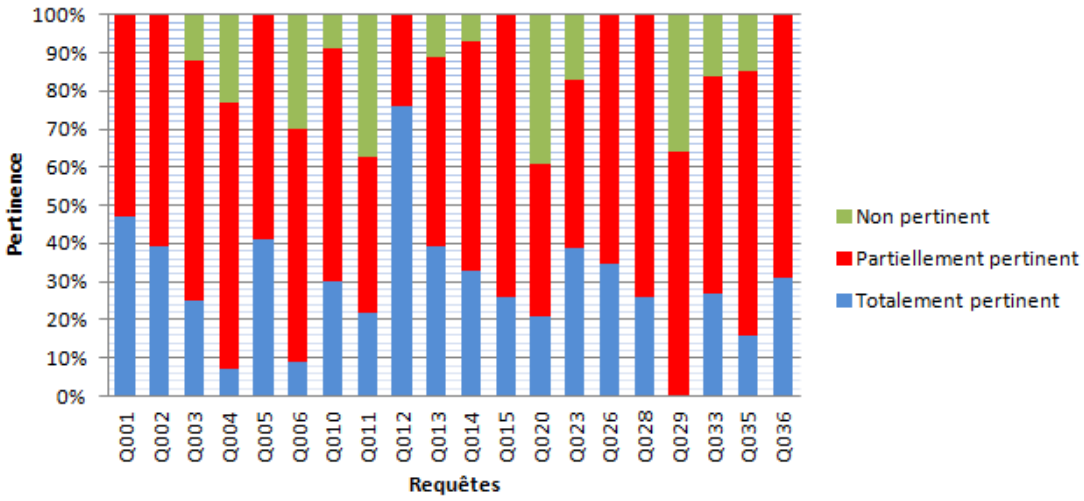

FIGURE 5.3 – Distribution de la pertinence d'agrégats par requête

Afin d'obtenir une analyse plus fine de ces résultats, nous étudions le nombre d'éléments pertinents renvoyés par agrégat et par requête. Pour cela, nous mesurons la précision dans les top-5 agrégats. Nous définissons tout d'abord la précision d'un agrégat k par :

$$P_{ag(k)} = \frac{Nombre\ d'éléments\ pertinents\ dans\ ag(k)}{Nombre\ total\ d'éléments\ dans\ ag(k)} \tag{5.1}$$

où $ag(k)$ est un agrégat au rang (k).

La précision moyenne pour une requête q, notée $AP_q@k$, est calculée par la moyenne des précisions pour les top-k agrégats comme suit :

$$AP_q@k = \frac{\sum_{i=1}^{k} P_{ag(i)}}{|k|} \tag{5.2}$$

Ainsi, la moyenne des précisions moyennes $MAP@k$ pour toutes les requêtes est calculée comme suit :

$$MAP@k = \frac{\sum_{q \in Q} AP_q@k}{|Q|} \tag{5.3}$$

Avec :

1. $AP_q@k$ est la précision moyenne pour une requête q.
2. Q est l'ensemble des requêtes.

Dans cette deuxième expérimentation, nous testons la précision par requête pour les top-k agrégats à $P_{ag(1)}$, $P_{ag(2)}$, $P_{ag(3)}$, $P_{ag(4)}$ et $P_{ag(5)}$. Les résultats sont présentés sur la figure 5.4.

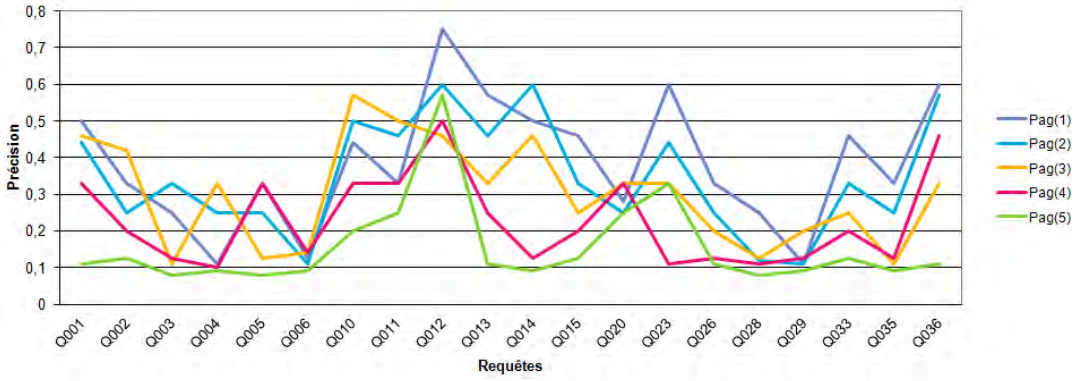

FIGURE 5.4 – Pertinence d'agrégats par requête à $P_{ag(1)}$, $P_{ag(2)}$, $P_{ag(3)}$, $P_{ag(4)}$, $P_{ag(5)}$

Pour les vingt requêtes de test et en utilisant la mesure proposée dans la formule 5.1 à $P_{ag(1)}$, huit requêtes avaient plus de 40% des éléments pertinents, onze requêtes avaient entre 10% et 40% des éléments pertinents. À $P_{ag(5)}$, parmi les vingt requêtes de test, une seule a plus de 40% des éléments pertinents, onze requêtes avaient entre 10% et 40% d'éléments pertinents, et huit requêtes ont moins de 10% des éléments pertinents par agrégat. La plus grande (resp. faible) valeur $AP@5$ est pour le $Q012$ requête (resp. $Q006$) et elle est égale à 0,576 (resp. 0,121). La $MAP@5$ pour les vingt requêtes est égale à 0,28. Ainsi, notre approche renvoie plus d'éléments pertinents dans le premier top-k agrégats, guide l'utilisateur à identifier les éléments pertinents d'un document XML et réduit également les efforts déployés par l'utilisateur afin de localiser les informations recherchées. Toutefois, dans certains cas, $Q010$ et $Q020$, le précision $P_{ag(3)}$ est supérieure à $P_{ag(1)}$.

Ces résultats sont évalués par l'utilisateur sans se demander si un agrégat contient des éléments redondants et/ou complémentaires. Ces questions sont abordées dans les expérimentations ci-après.

5.4.3 Impact de la redondance

Cette troisième expérimentation est conçue comme un test de cohérence de la redondance au niveau des résultats retournés. En effet, nous avons fourni deux degrés pour mesurer la redondance au sein d'un agrégat : *redondants* et *non-redondants*. Nous avons demandé aux utilisateurs de vérifier chaque agrégat et répondre à la question de la redondance :

Définition 7 (Redondants) *Si un utilisateur juge qu'un ou quelques éléments d'un agrégat n'apportent pas de nouvelles informations.*

Définition 8 (Non-redondants) *Si chaque élément d'un agrégat apporte une nouvelle information.*

Pour chaque requête, chaque utilisateur est invité à évaluer la redondance entre les éléments de premier agrégat (top-1 agrégat). Il convient de noter que nous ne regardons pas si les éléments sont pertinents. La figure 5.5 montre les résultats qui sont très intéressants. En effet, nous constatons que 90,85% des jugements considèrent que les agrégats renvoyés contiennent des éléments qui ne véhiculent pas la même information. Il est tout à fait logique, car à ce moment-là, nous avons travaillé avec un document unique. Même si cela se produit, la première hypothèse **H1** a déjà été appliquée afin d'éviter l'inclusion structurelle entre les éléments d'un même agrégat.

FIGURE 5.5 – Distribution des jugements de la redondance par requête

En ce qui concerne l'hypothèse **H2**, son impact est minime (même sans effet), mais la question qui se pose : *À quoi elle sert ?* Tout simplement, notre modèle est censé également fonctionner si l'agrégat est construit à partir de plusieurs documents (multi-documents). Dans ce cas, il est fort probable d'avoir d'éléments qui portent la même information et le recours à cette hypothèse sera indispensable.

5.4.4 Impact de la complémentarité

Dans cette quatrième expérimentation, nous voulons évaluer si les éléments de l'agrégat sont complémentaires afin d'avoir une vue d'ensemble sur les résultats retournés. Nous avons également cherché à mesurer l'intérêt d'un agrégat par rapport à des éléments pris individuellement. Pour cela, nous présentons chaque top-1 agrégat de toutes les requêtes à chaque utilisateur et nous lui posons la question suivante : *Est-ce que les éléments d'un agrégat se complètent ?* En d'autres termes, si chaque utilisateur trouve de l'information pertinente et additionnelle, par rapport à son besoin d'information, entre les éléments de l'agrégat.

La distribution des jugements de la complémentarité entre les vingt requêtes est présenté dans la figure 5.6. Nous avons constaté que les utilisateurs considèrent que les éléments du top-1 agrégat apportent des informations pertinentes et supplémentaires pour plus de 62,42% des jugements [1]. On remarque que pour la plupart des requêtes, ces éléments peuvent être sémantiquement complémentaires. Cela prouve la capacité de notre modèle à agréger d'éléments qui se complètent

1. Nombre totale de jugements = 15 utilisateurs × 20 requêtes, soit au total 300 jugements

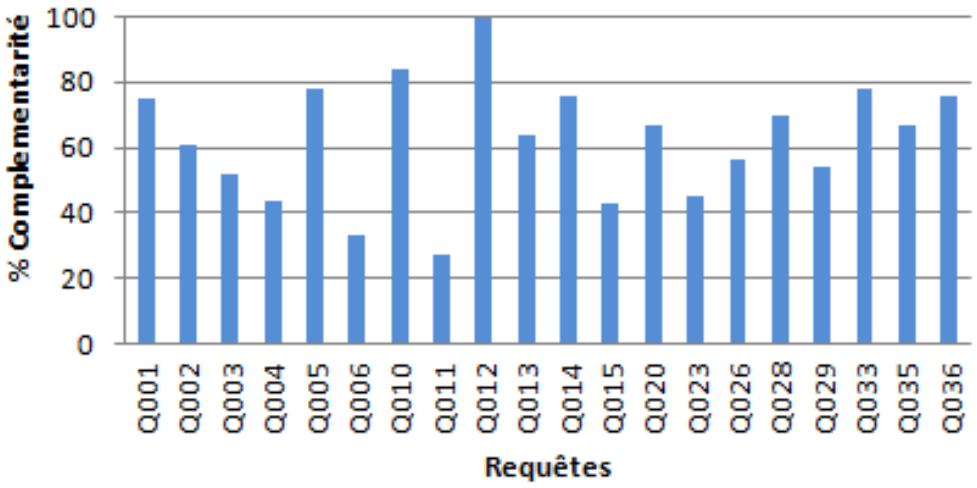

FIGURE 5.6 – Distribution des jugements de la complémentarité par requête

mutuellement c'est-à-dire chaque élément est qualifié pour fournir des informations pertinentes et supplémentaires.

5.4.5 Complémentarité vs. Redondance

Une des questions que nous aimerions discuter dans cette section porte sur la différence entre la redondance et la complémentarité, en d'autres termes si nous avons besoin de ces deux notions ou une seule d'entre elles est suffisante. Afin de mieux comprendre la différence, considérons une requête ambigüe, par exemple la requête "jaguar" (voiture vs animal), il y aura plusieurs éléments retournés qui parlent de l'usine automobile ou du parc animalier dans chaque agrégat. Dans ce cas, ces éléments seront non-redondants parce que chaque élément porte une nouvelle information par rapport au sujet de la requête. Mais, cela ne signifie pas que ces éléments sont complémentaires, car ils n'apportent aucune information supplémentaire vis-à-vis le besoin informationnel de l'utilisateur. Mais si nous avons un autre élément qui apporte l'adresse d'une usine ou d'un parc. Dans ce cas, on peut considérer que ce dernier élément est complémentaire aux éléments déjà récupérés (si nous parlons des voitures ou des animaux).

Maintenant, revenons aux deux figures 5.5 et 5.6, et vérifions le comportement des deux requêtes, à savoir, $Q012$ et $Q035$. Les agrégats de ces requêtes sont totalement non-redondants (voir figure 5.5), mais ils se comportent différemment sur le facteur de complémentarité. La figure 5.6 montrent que les éléments de l'agrégat de la requête $Q012$ sont complémentaires à 100% alors que pour l'agrégat de la requête $Q035$, 67% de ses éléments sont complémentaires. La

principale conclusion qu'on peut en tirer est que deux éléments sont complémentaires alors ils doivent d'abord être non-redondants. Et donc, la non-redondance est une condition nécessaire mais non suffisante pour la complémentarité.

5.4.6 RI agrégée vs. Liste ordonnéee

L'objectif principal de ce travail est de fournir aux utilisateurs d'agrégats au lieu d'une liste d'éléments pris séparément. La principale question que nous tentons d'évaluer dans cette cinquième expérimentation concerne l'intérêt de renvoyer des résultats agrégés par rapport à la traditionnelle liste triée d'éléments. Cette tâche n'est pas destinée à évaluer la façon de présenter les résultats (à travers une interface), mais l'utilité d'assembler les éléments dans des agrégats par rapport à une liste ordonnée. Donc, nous avons demandé aux utilisateurs de répondre à la question suivante : *Que préférez-vous la recherche agrégée ou une liste ordonnée ?*

Rappelons que pour chaque requête (parmi les vingt requêtes), nous avons quinze participants qui répondront à la question ci-dessus. Soit un total de 300 jugements. Dans 177 de jugements (soit 59%), les utilisateurs préfèrent les résultats retournés soient assemblés en agrégats qu'une simple liste ordonnée (cf. figure 5.7). Cela montre implicitement que la recherche agrégée est utile parce que souvent un seul élément ne suffit pas, alors que les éléments d'un agrégat peuvent se compléter mutuellement pour aboutir à une réponse plus complète. En résumé, la recherche agrégée fournit de meilleurs résultats que la RI structurée dans la majorité des requêtes.

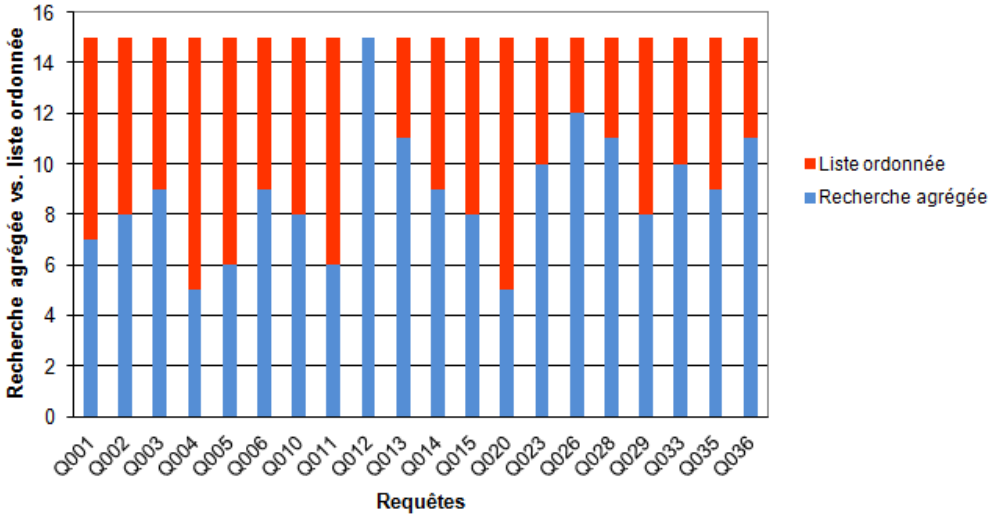

FIGURE 5.7 – Utilité de la RI Agrégée

5.4.7 Dégré d'accord entre participants et temps consacré à chaque requête

Dans cette expérimentation, nous essayons de déterminer le degré d'accord entre les jugements des utilisateurs à l'aide de coefficient de *Kappa* (K). Nous utilisons le coefficient *Kappa de Fleiss* [76] comme mesure pour évaluer la fiabilité entre un nombre fixe d'utilisateurs. Cette mesure est utilisée pour mesurer l'accord entre deux participants. Dans [129], les auteurs ont donné les intervalles suivants pour interpréter les valeurs de K. $K < 0$ (désaccord), $K \in [0,01; 0,2]$ (accord très faible), $K \in [0,21; 0,4]$ (accord faible), $K \in [0,41; 0,6]$ (accord modéré), $K \in [0,61; 0,8]$ (accord fort) et $K \in [0,81; 1]$ (accord presque parfait). Notons que la durée d'une session est le temps moyen nécessaire pour qu'un utilisateur évalue une requête pour chaque expérimentation. La durée de chaque session ainsi que le degré d'accord pour chaque expérimentation sont présentés dans le tableau 5.2.

Expérimentation	**1**	**2**	**3**	**4**	**5**
Durée (en secondes)	315	264	24	37	167
Degré d'accord	0,40	0,36	0,60	0,44	0,46

TABLE 5.2 – Durée et degré d'accord basés sur des contextes réels (user studies)

Le degré d'accord de nos expérimentations n'affecte pas la validité des résultats mentionnés dans les sections précédentes parce qu'en RI la valeur de K est toujours faible entre les utilisateurs. Ce même constat est également reconnu dans le cadre de campagnes d'évaluation tels que INEX et TREC [130].

Pour conclure, les utilisateurs sont en accord faible pour évaluer les expérimentations 1 et 2 sur la pertinence des agrégats. En outre, ces deux expérimentations sont assez longues car l'évaluation concerne les top-5 agrégats. D'autre part, les deux autres expérimentations 3 et 4 sont plutôt rapides parce que l'évaluation porte uniquement sur le top-1 agrégat de chaque requête et elles sont en accord modéré. En ce qui concerne la dernière expérimentation, le temps d'évaluation est moyen par rapport aux autres et il fait également partie de l'accord modéré.

5.4.8 Discussion

A notre connaissance, notre approche est parmi les premières qui abordent le problème de la recherche agrégée dans des documents XML. L'évaluation

expérimentale montre que la recherche agrégée peut contribuer dans la recherche d'information dans des documents XML. En effet, nous exigeons qu'un agrégat soit qualifié comme réponse à une requête s'il répond à trois caractéristiques a savoir la pertinence, non redondante et complémentarité. Pour répondre à la première caractéristique, nous essayons d'identifier les éléments les plus significatifs dans l'agrégat sélectionné à partir d'un document XML. Dans ce cas, un agrégat pertinent permet d'améliorer l'interprétation des résultats, guider l'utilisateur à identifier les éléments pertinents dans un document XML et réduire également les efforts déployés par l'utilisateur qui doit fournir pour localiser les informations souhaitées. Pour satisfaire la deuxième caractéristique, nous avons besoin de générer des agrégats sous contraintes les deux hypothèses **H1** et **H2**. Pour remplir la troisième caractéristique, nous exigeons que les éléments d'un agrégat apportent des informations pertinentes et additionnelles entre eux. Toutefois, dans quelques cas si des éléments ne sont pas complémentaires ceci ne veux pas dire que ces éléments ne sont pas sémantiquement liés à la requête de l'utilisateur.

Ce type d'agrégation est très utile car il permet une distinction très fine de différentes thématiques exprimées dans la requête de l'utilisateur lorsque son besoin en information est générique. Il vise également à donner à l'utilisateur un aperçu sur les différentes informations disponibles dans le corpus de documents et qui sont liées à son besoin. Dans le cas échéant, il peut reformuler sa requête.

5.5 Conclusion

Nous avons abordé la problématique d'évaluation des agrégats générés à partir des documents XML. Nous avons pris en considération l'évaluation des agrégats selon trois caractéristiques à savoir la pertinence, la redondance et la complémentarité.
Nous avons fourni un cadre d'évaluation spécifique à la recherche agrégée à l'aide de plusieurs séries d'expériences. D'une manière générale, ces expérimentations permettent de démontrer que :

- l'utilisateur peut trouver dans les agrégats générés plus d'informations pertinentes et réduit ainsi l'effort à fournir afin de satisfaire son besoin d'information (voir section 5.4.2) ;
- dans la plupart des agrégats renvoyés, ses éléments ne véhiculent pas la même information (voir section 5.4.3) ;
- dans plus la moitié des agrégats sélectionnés, ses éléments portent des informations pertinentes et additionnelles (voir section 5.4.4) ;
- l'intérêt de la RI agrégée par rapport à la RI structurée (voir section 5.4.6).

Conclusion générale

Synthèse

Les travaux présentés dans cette thèse s'inscrivent dans le contexte général de la RI et plus particulièrement dans le cadre de la RI agrégée dans des documents semi-structurés de type XML.
En RI Structurée (RIS), les éléments potentiellement pertinents renvoyés par un système en réponse à une requête sont présentés à l'utilisateur sous forme d'une simple liste ordonnée de résultats. Plusieurs questions se posent dans ce contexte. Les principales sont : à partir de quel moment est-on certain d'avoir collecté assez d'information ? Comment sélectionner l'unité d'information qui répond le mieux à une requête ? La plupart des systèmes de RIS retournent les résultats de recherche sous la forme d'une liste d'éléments disjoints, d'autres commencent à présenter les résultats de la recherche sous la forme de résumés multi-documents. D'autres questions plus techniques font aussi le sujet de cette thèse, elles concernent les résultats retournés : Doit-on renvoyer des résultats qui véhiculent la même information ? Dans ce cas, quelle est l'utilité d'une telle recherche ? Peut-on avoir des résultats qui se complètent ?

Notre objectif est d'apporter des réponses à ces questions. Nous avons alors proposé un modèle de RIS permettant une "meilleure" forme de construction des résultats répondant à la requête. Notre modèle trouve ses fondements théoriques dans les RB. Plus précisément, le modèle que nous proposons est basé sur un réseau pour chaque document. Dans chaque réseau, les nœuds représentent un document, ses éléments, les termes d'indexation et la requête. La topologie du réseau permet de prendre en compte naturellement les relations de dépendance entre ces nœuds.

Plus précisément, nos contributions présentées dans cette thèse ont porté sur quatre volets : l'agrégation des éléments les plus potentiellement pertinents, l'élagage d'éléments redondants à partir d'un ou plusieurs documents, la détermination d'éléments porteurs d'informations pertinentes et additionnelles et la proposition d'un cadre d'évaluation d'agrégats.

1. L'utilisation des RB en RI s'est avérée intéressante grâce notamment à

leur puissance pour inférer la pertinence des documents vis-à-vis d'une requête ainsi qu'à leur capacité de représenter de manière naturelle les différents liens existants entre les objets manipulés en RI, à savoir les documents, les éléments, les termes et la requête. L'évaluation de la pertinence d'une configuration vis-à-vis d'une requête est effectuée par un processus de propagation à travers les nœuds termes reliés à cette requête. Les termes de la requête absents dans les représentations d'agrégats via ses éléments sont donc naturellement et explicitement considérés dans le calcul des scores de pertinence contrairement aux systèmes actuels de RI. Compte tenu de l'intérêt que nous avons accordé à cette notion d'importance (ou de représentativité) d'un terme dans une configuration, nous avons proposé une estimation du maximum de vraisemblance sur la fréquence d'un terme dans une configuration permettant de mieux quantifier l'importance d'un terme dans une configuration. Afin d'éviter le problème des fréquences nulles des quelques termes (si un terme ne figure pas dans une configuration) et éventuellement dans ses éléments, il faut ajouter la fréquence du terme dans la collection avec celle calculée avec le document. En fait, nous utilisons une technique de lissage de type Dirichlet appliquée à chaque élément XML de la configuration en question ;

2. Dans notre processus de propagation, nous nous sommes intéressés à l'agrégation d'éléments qui ne véhiculent pas la même information dans une configuration donnée. Les techniques d'élagage proposées, afin d'éliminer les éléments redondants dans la même configuration, portent aussi bien sur la première source d'évidence à savoir la structure à l'aide d'une première hypothèse (**H1**) et sur la deuxième source d'évidence à savoir le contenu à l'aide d'une deuxième hypothèse (**H2**) quand notre processus de propagation est généralisé.
   - **H1** : cette hypothèse est qualifiée comme contrainte de structure permettant d'éliminer les éléments redondants. Nous considérons que la présence d'une relation ancêtre-descendant entre deux éléments signifie que l'un est inclus dans l'autre ;
   - **H2** : cette hypothèse est considérée comme une contrainte de contenu. Nous supposons qu'un utilisateur préfère retrouver dans une configuration donnée des éléments non redondants à partir de plusieurs documents. Par souci de simplicité, nous proposons d'utiliser la distance cosinus pour détecter la redondance entre les éléments renvoyés.
3. De plus, nous avons proposé d'assembler des éléments qui se complètent dans la même configuration. La complémentarité indique à quel point un élément ajoute ce qu'il manquait en matière d'information à un ensemble d'éléments. Pour modéliser cette caractéristique, nous avons également proposé une fonction de propagation qui favorise les éléments les plus loin de nœud racine. En effet, les éléments loin du nœud racine d'un document paraissent plus porteurs d'informations complémentaires que ceux situés plus haut dans le document. Intuitivement, plus la distance entre

un élément et la racine est grande, plus il contribue à la complémentarité des éléments d'une telle configuration. L'objectif de cette caractéristique est de favoriser dans les configurations les éléments qui se complètent mutuellement pour avoir une réponse plus complète :"pertinence additionnelle" ;

4. Le dernier volet de notre contribution consiste en la définition d'un cadre d'évaluation approprié pour la RI agrégée dans des documents XML. Le cadre proposé consiste à utiliser les ressources de la collection de test fournie dans le cadre de la campagne d'évaluation INEX 2009.
   Les expérimentations menées portent essentiellement sur :
   - l'évaluation de la pertinence des agrégats : les premiers résultats intéressants montrent que par parmi les agrégats renvoyés 29% sont totalement pertinents et 58% sont partiellement pertinents. Seulement 13% sont non-pertinents ;
   - l'évaluation de la redondance : nous avons trouvé que 91% des agrégats renvoyés contiennent des éléments qui ne véhiculent pas la même information ;
   - l'évaluation de la complémentarité : nous avons constaté que les utilisateurs considèrent que les éléments du top-1 agrégat apportent d'informations pertinentes et additionnelles pour plus de 63% des agrégats ;
   - l'évaluation des performances ainsi que l'utilité de la RI agrégée par rapport à la recherche d'information structurée (RIS) : Les résultats obtenus de cette comparaison montrent que notre modèle est efficace et performant pour agréger des éléments à partir d'un document. En effet, nous avons trouvé 59% de jugements, des utilisateurs qui préfèrent les agrégats qu'une simple liste ordonnée d'éléments. Ces résultats peuvent être considérés intéressants ;
   - l'évaluation de degré d'accord entre les jugements d'utilisateurs à l'aide de test statistique de Kappa de Cohen.

Il est également à noter que notre approche est applicable sur des requêtes de type CO.

Limites et perspectives

L'évaluation expérimentale de notre modèle a montré son efficacité selon plusieurs aspects, et ouvrent des perspectives à court terme portant sur l'utilisation de requêtes CAS, l'intégration d'un processus itératif à la recherche pour la reformulation de requêtes, la définition des relations de dépendances dans un cadre qualitatif et d'autres à long terme portant sur l'intégration des relations de dépendances entre des paires de termes d'indexations ou de documents, l'intégration des relations entre paires de documents dans un cadre

ordinal.

Plus particulièrement, nos perspectives a court terme portent essentiellement sur les volets suivants :

1. Étendre notre modèle pour supporter aussi des requêtes orientées contenu et structure. Nous proposons également d'étendre notre modèle pour supporter des collections hétérogènes (c'est à dire ayant des documents aux structures différentes).
2. Intégrer un processus itératif à la recherche pour la reformulation de requêtes. Pour ce faire, deux techniques existant dans les modèles basés sur les RB probabilistes pourraient être adaptées à notre approche. La première préconise l'ajout des nœuds ou d'arcs dans le réseau pour recalculer les distributions de probabilité. Cette technique permet ainsi d'ajouter des relations de dépendance entre des termes et la requête. Ces termes peuvent être issus d'agrégats jugés par l'utilisateur ou les termes des n premiers agrégats restitués initialement par le système. La seconde technique considère la requête reformulée comme une nouvelle information à introduire dans le système ;
3. Définir les relations de dépendance dans un cadre qualitatif. Les valeurs affectées à ces relations traduiraient des ordres partiels de préférence. La théorie des possibilités offre deux cadres de travail. Le cadre qualitatif ou ordinal et le cadre numérique. Nous avons proposé notre modèle dans un cadre numérique basé sur la théorie des probabilités. Nous proposons ici de traduire ce modèle dans un cadre ordinal basé sur les réseaux possibilistes. Ainsi, des préférences pourraient être définies entre les termes d'indexation pour représenter les documents et/ou la requête. Ces préférences peuvent être données par des experts, ou par des études statistiques sur le texte, etc. Ces préférences permettraient par la suite, de restituer des agrégats classés par préférence de pertinence. Il serait possible dans un tel cadre de mesurer le point auquel un agrégat a_1 est préféré à l'agrégat a_2 ou de mesurer la préférence d'un agrégat a_1 par rapport à un ensemble d'agrégats a_3,a_4.

   À long terme nous prévoyons de :

4. Intégrer des relations de dépendance entre des paires de termes d'indexation ou des paires de documents. Cette perspective peut être en relation avec la perspective précédente. Dans ce contexte, les arcs sont mesurés par des valeurs numériques traduisant des quantités et non pas des ordres partiels. Afin de quantifier ces relations, nous pourrions nous baser sur la connaissance représentée dans une ontologie. Une ontologie permet de formaliser des liens sémantiques entre des concepts unités de sens. Définie dans un cadre probabiliste, elle pourrait ajouter de l'information pertinente à considérer lors du processus de propagation déclenchée par la requête. Le réseau serait composé d'un sous réseau documents et d'un sous réseau requête. Ces sous réseaux pourraient être reliées à travers

une ontologie ;

5. Intégrer des relations entre paire d'agrégats dans un cadre numérique ou ordinal. Les relations de dépendances entre paires d'agrégats pourraient traduire des liens sémantiques ou statistiques évaluant les distributions des termes communs à des paires ou ensembles d'agrégats. Les termes ou les agrégats peuvent ainsi être regroupés dans des classes communes ;
6. Mettre en place un cadre d'évaluation standard pour la RI agrégée dans des documents XML où l'évaluation est vigoureusement contrôlée en utilisant une collection de test réelle dont les requêtes sont émises par des utilisateurs et leurs interactions sont exploitables pour fournir des jugements de pertinence sur des agrégats construits pour les vingt requêtes proposées.

Annexe A

Les documents semi-structurés

A.1 XML : concepts de base

A.1.1 Documents structurés et documents semi-structurés

La structure d'un document est l'agencement de ses différents *éléments* afin de lui donner sa cohérence, sa forme et sa rigidité. Une *balise* (ou *tag*) est une suite de caractère encadrés par "<" et ">", comme par exemple <*titre*>. Un élément est une unité syntaxique encadrant les fragments d'informations par une balise de début et une balise de fin, comme par exemple <*titre*> RI Structurée < /*titre*>. Les éléments d'un document peuvent être imbriqués comme le montre l'exemple de la figure A.1, mais ils ne doivent pas se recouvrir. Les *attributs* des éléments sont intégrés à la balise de début en utilisant la syntaxe *nomattribut*=″*valeur*″. Par exemple, <titre *sujet*=″*xml*″> RI Structurée < /titre>.

Le langage de description à balises SGML (Standard Generalized Markup Language) [87], de norme ISO [1] (International Organization for Standardization) et sa version simplifiée XML (eXtensible Markup Language) permettent de produire des documents *structurés* ou *semi-structurés*. Les documents structurés possèdent une structure régulière, ne contiennent pas d'éléments mixtes (c'est à dire d'éléments contenant du texte et d'autres éléments) et l'ordre des différents éléments qu'ils contiennent est généralement non significatif.
Les documents semi-structurés sont des documents qui possèdent une structure flexible et des contenus hétérogènes. La modification, l'ajout ou la suppression d'une donnée entraine une modification de la structure de l'ensemble [9].

1. ISO est un organisme créé en 1947 et a pour but de produire des normes internationales dans les domaines industriels et commerciaux appelées normes ISO

```xml
<?xml version="1.0"?>
<!--Exemple d'un document XML-->
<rapport université="Paul Sabatier - Toulouse 3">
<titre>Recherche Agrégée</titre>
<auteur>Najeh Naffakhi</auteur>
<date>Décembre 2012</date>
<contenu>
    <introduction> </introduction>
    <chapitre>
        <titre> La Recherche d'Information</titre>
        <section>La RI est une discipline de recherche...</section>
        <section>Le but fondamental d'un SRI est de sélectionner ...</section>
    </chapitre>
    <chapitre>
        <titre>La RI structurée et les RB</titre>
        <section>Dans le contexte de la RIS ...</section>
        <section>La naissance du modèle d'inférence ...</section>
    </chapitre>
    <conclusion> </conclusion>
</contenu>
</rapport>
```

FIGURE A.1 – Exemple d'un document XML

Dans ce contexte, nous nous intéressons à la RI dans des documents semi-structurés. Les documents structurés servent à conserver des données au sens Bases de données. Par abus de langage, on parlera de la *RIS*. Le format XML nous permet d'illustrer nos propos.

A.1.2 Les fondements de XML

XML [2] est un standard mis en place par le W3C [3] (World Wide Web Consortium) et dérivé du langage SGML. Selon [51], la définition d'un document XML est la suivante : *"Un document en XML constitue [...] un terme technique, qui ne correspond pas nécessairement à la notion classique d'un document narratif, c'est-à-dire à un ensemble de données textuelles organisées et mises en forme à l'attention d'un lecteur. Il s'applique également à toute structure de données à vocation d'échange inter-applications."*

Un document XML est hiérarchisé sous forme d'un arbre. Chaque nœud de l'arbre est un élément XML. Cette structure logique permet de faire des recherches très pointues sur les éléments d'un document XML. Ces éléments ne peuvent pas se chevaucher mais ils peuvent s'imbriquer. Le choix du nom de ses éléments et leurs attributs ainsi que leur organisation est laissé au choix

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

```
<?xml version="1.0"?>
<!--Exemple de DTD pour le document XML précédent-->
<!ELEMENT rapport (titre, auteur, date, contenu)>
<!ATTLIST rapport université CDATA #REQUIRED>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT contenu (introduction, chapitre+, conclusion)>
<!ELEMENT introduction (#PCDATA)>
<!ELEMENT chapitre (titre, section+)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT section (#PCDATA)>
<!ELEMENT conclusion (#PCDATA)>
```

FIGURE A.2 – Exemple de DTD correspondant au document XML da la figure A.1

de l'auteur. C'est pourquoi le langage XML est dit *générique*.

XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (*Document Type Definition*) [143]. C'est un sous langage restreignant décrivant la structure des documents y faisant référence grâce à une organisation prédéfinie. Ainsi un document XML doit suivre scrupuleusement les conventions de notation XML et peut éventuellement faire référence à une DTD décrivant l'imbrication des éléments possibles. Un document suivant les règles de XML est appelé *document bien formé*. Un document XML possédant une DTD et étant conforme à celle-ci est appelé *document valide*. La figure A.2 présente une DTD correspondante au document XML A.1.

XML permet donc de définir un format d'échange selon les besoins de l'utilisateur et offre des mécanismes pour vérifier la validité du document produit. Il est donc essentiel pour le receveur d'un document XML de pouvoir extraire les données du document. Cette opération est possible à l'aide d'un outil appelé analyseur (en anglais parser, parfois francisé en parseur).

Le parseur permet d'une part d'extraire les données d'un document XML (on parle d'analyse du document ou de parsing) ainsi que de vérifier éventuellement la validité du document. Il existe deux types d'analyseurs de documents XML, le parseur s'appuyant sur des flux d'évènements SAX (*Simple API for XML*) et le parseur DOM [4] qui produit un graphe d'objets.

Le DOM représente en mémoire les éléments, les attributs et le texte des éléments au sein des nœuds d'un arbre comme illustre la figure A.3. Grâce à ses

4. http ://www.w3.org/DOM

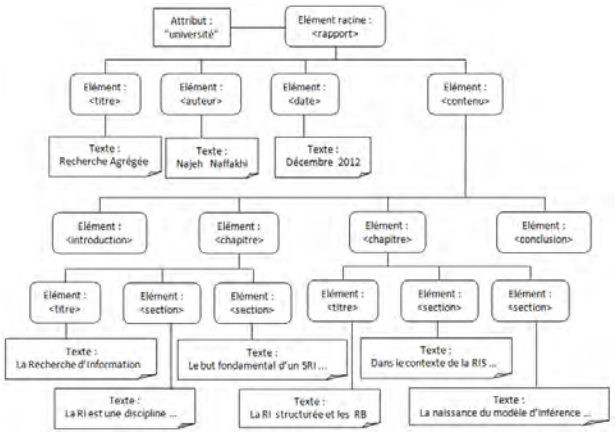

FIGURE A.3 – Exemple de DOM correspondant au document XML de la figure A.1

fonctions, le DOM permet de consulter et de modifier le contenu et la structure d'un document chargé en mémoire. Il est recommandé d'utiliser le DOM pour se repérer efficacement dans un document XML, relativement à un élément de l'arbre XML. Si le besoin en information est exprimé selon un chemin XML absolu, il devient ardu d'utiliser DOM et d'avoir recours à d'autres standards tel que *XPath*.

XPath est un langage d'expression s'appliquant à XML ; il s'agit d'un langage permettant de sélectionner des sous-arbres d'un document XML. Il possède une syntaxe simple et non ambigüe et implémente des types usuels (chaînes, nombres, booléens, variables, fonctions) [58].
XPath est une spécification conçue pour parcourir une collection de documents XML, et de sélectionner un ensemble de nœuds en exploitant notamment les relations existantes entre ces derniers. Ces nœuds devront répondre à certaines contraintes structurelles ou sémantiques (contenu) pour être sélectionnés. Les contraintes sont sous la forme d'un chemin. L'utilisateur doit décrire des expressions de chemin dans l'arbre d'un document XML pour retourner des fragments de document.

A.2 Stockage des documents XML

Le stockage des collections de documents XML peut se faire selon trois techniques : utilisation des fichiers textes, utilisation des SGBD relationnels et utilisation d'un SGBD XML natif [215].

A.2.1 Modèles de fichiers textes

Les fichiers textes constituent le moyen le plus simple de stocker les documents XML. Ils présentent l'avantage de pouvoir être lus et édités par un utilisateur. Ce format constitue de plus le moyen d'échange le plus simple des données XML sur un réseau. Pour l'interrogation, XQuery [75] permet d'interroger ces documents après une traduction préalable sous forme d'un arbre d'objets en mémoire selon le standard DOM.

A.2.2 Modèles de SGBD relationnels

Les principaux SGBD relationnels (Oracle, SQL server, etc.) ont été étendus pour les données XML. Deux méthodes de stockage existent :

- définir un nouveau type de données adapté à XML et stocker les documents XML comme des objets dans une colonne,
- réaliser une correspondance entre un document XML et un ensemble de tables en s'appuyant sur le DTD du document (destruction du document XML afin de stocker les éléments et les attributs en colonnes de tables).

Les documents stockés peuvent êtres manipulés en SQL par un jeu de fonctions prédéfinies, par exemple l'extraction des objets par une expression XPath.

A.2.3 Modèles de SGBD XML natifs

Les SGBD natifs sont développés spécifiquement pour XML. Ils stockent et manipulent directement des arbres XML au lieu de passer par une structure intermédiaire (table relationnelle). Ils possèdent des index spécialisés permettant d'accéder aux composants d'un arbre de documents XML : éléments, attributs et texte. Les langages d'interrogation pour ce type de modèles sont les langages de requête XPath et XQuery.

Bibliographie personnelle

[1] N. Naffakhi, and R. Faiz. Less is More : aggregating meaningful elements for xml keyword search. In Cépadues-Editions, *International Journal on Information - Interaction - Intelligence (I3)*, volume 12, number 1, 2012.

[2] N. Naffakhi, M. Boughanem, and R. Faiz. Recherche d'Information Agrégée dans des documents XML basée sur les Réseaux Bayésiens. In D. A. Zighed et G. Venturini, editor, *Revue des Nouvelles Technologies de l'Information (RNTI)*, volume 1, pages 369–380. Hermann, 2012.

[3] N. Naffakhi, and R. Faiz. Using Bayesian Networks Theory for Aggregated Search to XML retrieval. In *The 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS), Craiova, Romania, 13/06/2012-15/06/2012*, pages 71, ICPS, ACM digital library, 2012.

[4] N. Naffakhi, and R. Faiz. Aggregated Search in XML Documents : What to retrieve ?. In *IEEE International Conference on Information Technology and e-Services (ICITeS), Sousse, Tunisia, 24/03/2012-26/03/2012*, pages 121–126, March 24-26, 2012. IEEEXplore digital library.

[5] N. Naffakhi, M. Boughanem, and R. Faiz. Un Modèle Bayésien pour l'Agrégation des Documents XML. In *Conférence francophone en Recherche d'Information et Applications (CORIA), Avignon, France, 16/03/2011-18/03/2011*, pages 335–348, Association ARIA, Mars 2011. Université d'Avignon.

[6] N. Naffakhi, M. Boughanem, and R. Faiz. Réseau bayésien pour un modèle de Recherche d'Information agrégée dans des documents semi-structurés. In *Actes de XXVIIIème Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID), Marseille, France, 25/05/2010-28/05/2010*, pages 111–126, Association INFORSID, Mai 2010. Université de Provence.

[7] N. Naffakhi, and R. Faiz. Modèle basé sur les réseaux bayésiens pour agréger des éléments XML pertinents et non-redondants. In *Atelier de Recherche et Fouille d'Information sur le Web (RFIW) en conjonction avec la 11ème Conférence Internationale Francophone : Extraction et Gestion des Connaissances (EGC), Brest, France, 25/01/2011-28/01/2011*, pages 58–69, Hermann-Éditions, Janvier 2011. Université de Bretagne Occidentale.

[8] N. Naffakhi. Un modèle bayésien pour l'agrégation des documents semi-structurés. In *Rencontres des Jeunes Chercheurs en Recherche d'Information, en conjonction avec Colloque International Francophone sur l'Écrit et le Document et COnférence en Recherche d'Information et Applications (RJCRI :CIFED-CORIA), Sousse, Tunisie, 18/03/2010-20/03/2010*, CPU, pages 495–500, Mars 2010.

Bibliographie

[9] S. Abiteboul. Querying semi-structured data. In *6th International Conference on Data Theory (ICDT)*, volume 1186 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 1997.

[10] S. Abiteboul, I. Manolescu, B. Nguyen, and N. Prada. A test plateform for the inex heterogeneous track. In *Pre-proceedings Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, pages 177–182, 2004.

[11] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J.-L. Wiener. Query language for semi-structured data. *International Journal on Digital Libraries (IJDL)*, 1(1) :68–88, 1997.

[12] M. Abolhassani and N. Fuhr. Applying the divergence from randomness approach for content-only search in xml documents. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 409–419, 2004.

[13] P. Aditya and K. Jaya. Leveraging query association in federated search. In *Proceedings of the ACM SIGIR 2008 Workshop on Aggregated Search*, pages 31–39, 2008.

[14] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM)*, pages 5–14, 2009.

[15] E. Alfonseca, M. Pasca, and E. Robledo-Arnuncio. Acquisition of instance attributes via labeled and related instances. In *Proceedings of 33rd international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 58–65, 2010.

[16] S. AmerYahia, C. Botev, and J. Shanmugasundaram. Texquery : A full-text search extension to xquery. In *Proceedings of World Wide Web (WWW) Conference*, pages 253–265, 2004.

[17] V. Anh and A. Moffat. Compression and an ir approach to xml retrieval. In *Proceedings of the First Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, pages 253–265, 2002.

[18] M. B. Aouicha. *Une Approche Algébrique pour la Recherche d'Information Structurée*. Thèse de Doctorat de l'Université Paul Sabatier, Toulouse, France, 2009.

[19] J. Arguello, F. Diaz, J. Callan, and B. Carterette. A methodology for evaluating aggregated search results. In *Proceedings of the 33rd European conference on Advances in information retrieval (ECIR)*, pages 141–152, 2011.

[20] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *Proceedings of 32nd international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 315–322, 2009.

[21] T. Avrahami, L. Yau, L. Si, and J. Callan. The fedlemur project : Federated search in the real world. *Journal of the American Society for Information Science and Technology (JASIST)*, 57(3) :347–358, 2006.

[22] M. Azevedo, L. Amorim, and N. Ziviani. A universal model for xml information retrieval. In *Proceedings of the INEX Workshop*, pages 311–321, 2004.

[23] R. Baeza-Yates and R. Ribeiro-Neto. *Modern Information Retrieval*. New York : ACM Press ; Harlow England : Addison-Wesley, cop., 1999.

[24] K. Balog, A. Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the trec 2009 entity track. In *TREC 2009 Working Notes*. Springer-Verlag, 2009.

[25] K. Balog, A. Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the trec 2009 entity track. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*. Springer-Verlag, 2010.

[26] C. L. Barry. User-defined relevance criteria : an exploratory study. *Journal of the American Society for Information Science*, 45 :149–159, 1994.

[27] M. Bautin and S. Skiena. Concordance-based entity-oriented search. *Web Intelligence and Agent Systems (WIAS)*, 7(4) :303–319, 2009.

[28] S. BenFerhat, D. Dubois, D. Garcia, and H. Prade. Possibilistic logic bases and possibilistic graphs. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 57–64, 1999.

[29] F. Bessai-Mechmache and Z. Alimazighi. Aggregated search in xml documents. *Journal of Emerging Technologies in Web Intelligence (JETWI)*, 4(2) :181–188, 2012.

[30] P. Bhaskar, S. Banerjee, and S. Bandyopadhyay. A hybrid tweet contextualization system using ir and summarization. In S. Geva, J. Kamps, and R. Schenkel, editors, *Proceedings of the 10th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 164–175. Lecture Notes in Computer Science, Springer Verlag, 2012.

[31] T. Bilyana, M. Kacimi, and G. Weikum. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In *Proceedings of the the third ACM international conference on Web Search and Data Mining (WSDM)*, pages 431–440, 2010.

[32] T. Bogers, K. Christensen, and B. Larsen. Rslis at inex 2011 : Social book search track. In S. Geva, J. Kamps, and R. Schenkel, editors, *Proceedings of the 10th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 45–56. Lecture Notes in Computer Science, Springer Verlag, 2012.

[33] C. Borgelt, J. Gebhardt, and R. Kruse. Possibilistic graphical models. In *Computational Intelligence in Data Mining, Courses and Lectures*, pages 51–68. Springer, 2000.

[34] P. Borlund. The concept of relevance in ir. *Journal of the American Society for Information Science*, 54(10) :913–925, 2003.

[35] P. Borlund and P. Ingwersen. The development of a method for the evaluation of interactive retrieval systems. *Journal of Documentation*, 53(3) :225–250, 1997.

[36] M. Boughanem, A. Brini, and D. Dubois. Possibilistic networks for information retrieval. *International Journal of Approximate Reasoning (IJAR)*, 7(50) :957–968, 2009.

[37] M. Boughanem, C. Chrisment, and C. Soulé-Dupuy. Query modification based on relevance back-propagation in adhoc environnement. *Information Processing Management Journal*, 35(2) :121–139, 1999.

[38] M. Boughanem and J. Savoy, editors. *Recherche d'information états des lieux et perspectives*. Hermès Science Publications, 2008.

[39] O. Bouidghaghen, L. Tamine-Lechani, and M. Boughanem. Dynamically personalizing search results for mobile users. In *Proceedings of In Flexible Query Answering (FQAS)*, pages 99–110, 2009.

[40] A. Brini and M. Boughanem. Relevance feedback : introduction of partial assessments for query expansion. In *Proceedings of the Conference of the EUropean Society for Fuzzy Logic And Technology (EUSFLAT)*, pages 67–72, 2003.

[41] A. H. Brini. *Un modèle de Recherche d'Information basé sur les réseaux possibilistes*. Thèse de Doctorat de l'Université Paul Sabatier, Toulouse, France, 2005.

[42] C. W. Bruce. *Organizing and Searchning Large Files of Document Descriptions*. Ph.D thesis, University of Cambridge, Massachusetts, USA, 1979.

[43] E. Brunet. Le lemme comme on l'aime. In *actes de la 6ème Journées Internationales d'Analyse Statistique des Données Textuelles*, pages 221–232, 2002.

[44] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, New York, NY, USA, 2000. ACM.

[45] P. Buneman, G. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization techniques for unstructured data. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 505–516. ACM, 1996.

[46] H. Bunke. Recent developments in graph matching. In *ICPR*, pages 2117–2124, 2000.

[47] J. M. Cafarella, Y. A. Halvey, and N. Khoussainova. Data integration for the relational web. In *Proceedings of the 36th international conference on Very large data bases (VLDB)*, pages 1090–1101, 2010.

[48] M. Cafarella, M. Banko, and O. Etzioni. Relational web search. Technical report, University of Washington, 2006.

[49] J. Callan. Distributed information retrieval. In *Advances in Information Retrieval*, pages 235–266. Kluwer Academic Publishers, 2000.

[50] J. Carbonell and J. Goldstein. The use of mmr, diversity-based re-ranking for reordering documents and producing summaries. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[51] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld : Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2) :1026–1038, 1999.

[52] S. Ceri, S. Comai, E. Damiani, P. Fraternali, S. Paraboschi, and L. Tanca. Xml-gl : A graphical language for querying and restructuring www data. In *Proceedings of WWW Conference*, pages 1171–1187, 1999.

[53] D. Chamberlin, J. Robie, A. Berglund, and S. Boag. Xquery 1.0 : An xml query language (second edition). Technical report, http ://www.w3.org/TR/xquery/, 2010.

[54] D. Chamberlin, J. Robie, and D. Florescu. Quilt : An xml query language for heterogeneous data sources. In *Proceedings of the 3rd Internation Workshop on World Wide Web and databases*, pages 1–25, 2000.

[55] Y. Chiaramella and P. Mulhem. De la documentation automatique à la recherche d'information en contexte. *Document numérique*, 10(1) :11–38, 2007.

[56] Y. Chiaramella, P. Mulhem, and F. Fourel. A model for multimedia information retrieval. Technical report, FERMI ESPRIT BRA 8134, University of Glasgow, 1999.

[57] C. Clark, N. Craswell, and I. Soboroff. verview of the trec 2009 web track. Technical report, 2010.

[58] J. Clark and S. DeRose. Xml path language (xpath) version 1.0. Technical report, World Wide Web Consortium, 1999.

[59] C. L. A. Clarke, M. Kolla, V. G. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. Mackinnon. Novelty and diversity in information retrieval evaluation. In *Proceezdings of SIGIR*, pages 659–666, 2008.

[60] C. Cleverdon. Readings in information retrieval. In *The cranfield tests on index language devices*, pages 47–59, 1997.

[61] P. Clough, M. Sanderson, M. Abouammoh, S. Navarro, and L. M. Paramita. Multiple approaches to analysing query diversity. In *Proceedings of SIGIR*, pages 734–735, 2009.

[62] G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks (research note). *Artif. Intell.*, 42(2-3) :393–405, 1990.

[63] C. Crouch. Dynamic element retrieval in a structured environment. *ACM Trans. Inf. Syst.*, 24(4) :437–454, 2006.

[64] C. Crouch, S. Apte, and H. Bapat. An approach to structured retrieval based on extended vector model. In *Proceedings of the INEX 2003 Workshop*, pages 89–93, 2002.

[65] C. Crouch, D. Crouch, N. Acquilla, R. Banhatta, S. Chittilla, N. Nagalla, and R. Navenvarapu. Focused elements and snippets. In S. Geva, J. Kamps, and R. Schenkel, editors, *Focused Retrieval of Content and Structure*, pages 295–299. Lecture Notes in Computer Science, Springer Verlag, 2012.

[66] A. C. Cuadra and V. R. Katter. Opening the black box of relevance. *Journal of Documentation*, 23(4) :291–303, 1967.

[67] L. M. De Campos, J. M. Fernãndez luna, and J. F. Huete. Using context information in structured document retrieval : an approch based on influence diagrams. *Information Processing and Management*, 40(5) :829–847, 2004.

[68] L. Denoyer and P. Gallinari. Bayesian network model for semi-structured document classification. *Information Processing Management*, 40(5) :807–827, 2004.

[69] L. Denoyer and P. Gallinari. The wikipedia xml corpus. In *The 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR Forum*, pages 64–69, 2006.

[70] L. Denoyer, G. Wisniewski, and P. Gallinari. Document structure matching for heterogenous corpora. In *Proceedings of the 27th ACM SIGIR 2004 workshop on XML and Information Retrieval*, pages 1–7, 2004.

[71] F. Diaz. Integration of news content into web results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM)*, pages 182–191, 2009.

[72] R. Dragomir, J. Otterbacher, A. Winkel, and S. B. Goldensohn. Newsinessence : summarizing online news topics. In *Communications of the Association of Computing Machinery (ACM)*, pages 95–98, 2005.

[73] R. Dragomir, R. Weiguo, and F. Zhu. Webinessence : a personalized web-based multi-document summarization and recommendation system. In *NAACL Workshop on Automatic Summarization*, pages 79–88, 2001.

[74] D. Dunlavy, D. O'Leary, J. M. Conroy, and J. D. Schlesinger. Qcs : A system for querying, clustering and summarizing documents. In *International Journal : Information Processing and Management (IPM)*, pages 1588–1605, 2007.

[75] M. F. Fernãndez, T. Jim, K. Morton, N. Onose, and J. Simeon. Highly distributed xquery with dxq. In *Proceedings of the 2007 ACM SIGMOD International Conference (SIGMOD)*, pages 1159–1161, 2007.

[76] J. Fleiss. Measuring nominal scale agreement among many raters 1971. *Psychological Bulletin*, pages 378–382, 1971.

[77] D. Florescu and D. Kossmann. Storing and querying xml data using an rdmbs. *IEEE Data Engineering Bulletin*, 22(3) :27–34, 1999.

[78] M. Franz, A. Ittycheriah, J. McCarley, and T. Ward. First story detection : Combining similarity and novelty based approaches. Technical report, Topic detection and tracking Workshop report, 2001.

[79] N. Fuhr and K. Grossjohann. Xirql : a query language for information retrieval in xml documents. In *Proceedings of the 24th annual international ACM SIGIR Conference*, pages 172–180, 2001.

[80] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik. Xml information retrieval : Inex 2004. In *Advances in XML Information Rretreival and evaluation*, pages 409–410, 2004.

[81] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik. Advances in xml information retrieval, third international. In *Proceedings of the INEX 2004 Workshop*. Lecture Notes in Computer Science, Springer, 2005.

[82] M. Fuller, E. Mackie, R. Sacks-Davids, and R. Wilkinson. Structural answers for a large structured document collection. In *Proceedings of the ACM SIGIR 1993*, pages 204–213, 1993.

[83] S. Geva. Gpx-gardens point xml information retrieval at inex 2004. In *Proceedings of the INEX 2004 Workshop*, pages 211–223, 2004.

[84] S. Geva. Gpx-gardens point xml information retrieval at inex 2005. In *Proceedings of the INEX 2005 Workshop*, pages 240–253, 2005.

[85] S. Geva, J. Kamps, M. Lethonen, R. Schenkel, J. Thom, and A. Trotman. Overview of the inex 2009 ad hoc track. In *Proceedings of the INEX 2009 Workshop Pre-proceedings*, pages 16–50. IR Publications, Amsterdam, 2009.

[86] L. Goeuriot. *Découverte et caractérisation des corpus comparables*. Thèse en informatique, Université de Nantes, Nantes, France, 2009.

[87] C. F. Goldfarb. *The SGML Handbook*. Oxford University Press, 1990.

[88] N. Gövert. Assessments and evaluation measures for xml document retrieval. In *Proceedings of the INEX 2002 Workshop*, 2002.

[89] N. Gövert, M. Abolhassani, N. Fuhr, and K. Grossjohan. Content oriented xml retrieval with hyrex. In *Proceedings of the INEX 2002 Workshop*, pages 26–32, 2002.

[90] T. Grabs and H. Schek. Eth zürich at inex : Flexible information retrieval from xml with powerdb-xml. In *Proceedings of the INEX 2002 Workshop*, pages 141–148, 2002.

[91] L. Gravano, H. G. Molina, and A. Tomasic. The effectiveness of gioss for the text database discovery problem. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 126–137, 1994.

[92] O. Greenshpan, T. Milo, and N. Polyzotios. Autocompletion for mashups. In *Proceedings of VLDB*, pages 538–549, 2009.

[93] A. Gutierrez, R. Motz, and D. Viera. Building databases with information extracted from web documents. In *Proceedings XX International Conference of the Chilean Computer Sciences Society*, pages 41–49, 2000.

[94] S. Hattori, T. Tezuka, and K. Tanaka. Context-aware query refinement for mobile web search. In *Proceedings of International Symposium on Applications and the Internet Workshops (SAINT-W)*, pages 15–, 2007.

[95] Y. Hayashi, J. Tomita, and G. Kikoi. Searching text-rich xml documents with relevance ranking. In *Proceedings ACM SIGIR 2000 Workshop on XML and IR*, pages 27–35, 2000.

[96] S. Hennig and M. Wurst. Incremental clustering of newsgroup articles. In *Proceedings of the 19th international conference on Advances in Applied Artificial(IEA/AIE)*, pages 332–341, 2006.

[97] L. Hlaoua. *Reformulation de requêtes par réinjection de Pertinences dans les documents semi-structurés*. Thèse de Doctorat de l'Université Paul Sabatier, Toulouse, France, 2007.

[98] V. Hristidis, Y. Papakonstantinou, and A. Balmin. Keyword proximity search on xml graphs. In *proceedings of International Conference on Data Engineering ICDE*, pages 367–378, 2003.

[99] Y. Huang, Z. Liu, and Y. Chen. Query biased snippet generation in xml search. In *proceedings of Special Interest Group on Management Of Data SIGMOD'08*, pages 315–326, 2008.

[100] G. Hubert. A voting method for xml retrieval. In *Proceedings of the 3rd International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX)*, pages 183–196, 2005.

[101] G. Huck, I. Macherius, and P. Fankhauser. Pdom : Lightweight persistency support for the document object model. In *OOPSLA'99 workshop proceedings : Business Object Design and Implementation III*, pages 106–123, 1999.

[102] G. P. Ipeirotis. *Classifying and searching hidden-web text databases*. PhD thesis, New York, NY, USA, 2004.

[103] T. S. Jaakkola, M. Diekhans, and D. Haussler. Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference Intelligent Systems for Molecular Biology (ISMB)*, pages 149–158, 1999.

[104] H. Jang, Y. Kim, and D. Shin. An effective mechanism for index update in structured documents. In *Proceedings ACM Conference on Information and Knowledge Management (CIKM)*, pages 383–390, 1999.

[105] B.-J. Jansen and A. Spink. *An Analysis of document viewing pattern of web search engine user*. Idea Publishing Group, Hershey PA, 2005.

[106] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4) :422–446, 2002.

[107] F. Jensen and D. Nielsen. Springer, Verlag, 2007.

[108] B. T. Jones and S. R. Purves. Geographical information retrieval. In *Encyclopedia of Database Systems*, pages 1227–1231, 2009.

[109] V. Kakade and P. Raghavan. Encoding xml in vector spaces. In *Proceedings of ECIR*, 2005.

[110] J. Kamps, M. de Rijkeek, and B. Sigurbj:ornsson. Length normalization in xml retrieval. In *Proceedings of the SIGIR International Conference*, pages 80–87, 2004.

[111] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson. Xml retrieval : What to retrieve ? In C. L. A. Clarke, G. Cormack, J. Callan, D. Hawking, and A. Smeaton, editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 409–410. ACM Press, New York NY, 2003.

[112] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. Inex 2007 evaluation measures. In *Proceedings of INEX 2007 Workshop*, pages 24–33, 2007.

[113] C.-C. Kanne and G. Moerkotte. Efficient storage of xml data. In *Proceedings of the 16th International Conference on Data Engineering (ICDE)*, pages 359–381, 2000.

[114] R. Kaptein and M. Marx. Focused retrieval and result aggregation with political data. *Information Retrieval*, 13(5) :412–433, 2010.

[115] P. M. Kato, H. Ohshima, S. Oyama, and K. Tanaka. Query by analogical example : relational search using web search engine indices. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 27–36, 2009.

[116] G. Kazai, M. Lalmas, and A. P. de Vries. The overlap problem in content oriented xml retrieval evaluation. In *Proceedings of SIGIR 2004 International Conference*, pages 72–79, 2004.

[117] G. Kazai, M. Lalmas, and A. P. de Vries. Reliability tests for the xcg and inex-2002 metrics. In *Pre-Proceedings of INEX 2004 Workshop*, pages 33–39, 2004.

[118] G. Kazai, M. Lalmas, and T. Roelleke. Focused structured document retrieval. In *The 9th String Processing and Information Retrieval Symposium (SPIRE)*, pages 241–247, 2002.

[119] J. Kekäläinen and K. Järvelin. Evaluating information retrieval systems under the challenges of interaction and multidimensional dynamic relevance. In *Proceedings of the CoLIS 4 Conference*, pages 253–270, 2002.

[120] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th ACM WWW*, pages 297–306, 2008.

[121] A. Kopliku. *Approaches to implement and evaluate aggregated search.* Thèse de Doctorat de l'Université Paul Sabatier, Toulouse, France, 2011.

[122] A. Kopliku, M. Boughanem, and K. Pinel-Sauvagnat. Towards a framework for attribute retrieval. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM)*, pages 515–524, 2011.

[123] A. Kopliku, F. Damak, K. Pinel-Sauvagnat, and M. Boughanem. Interest and evaluation of aggregated search. In *Proceedings of the International Conference on Web Intelligence (IEEE/WIC/ACM)*, pages 154–161, 2011.

[124] A. Kopliku, K. Pinel-Sauvagnat, and M. Boughanem. Aggregated search : Potential, issues and evaluation. Technical report, Institut de Recherche en Informatique de Toulouse, 2009.

[125] A. Kopliku, K. Pinel-Sauvagnat, and M. Boughanem. Attribute retrieval from relational web tables. In *Proceedings of the Symposium on String Processing and Information Retrieval (SPIRE)*, pages 117–128, 2011.

[126] J. Lafferty and C. Zhai. Language models, query models, and risk minimization for information retrieval. In *Research and Development in Information Retrieval, In Proceedings of the ACM SIGIR*, pages 111–119, 2001.

[127] M. Lalmas. Dempster-shafer's theory of evidence applied to structured documents : modeling uncertainty. pages 110–118, Philadelphia, USA, 1997. ACM.

[128] M. Lalmas and P. Vannoorenberghe. Indexation et recherche de documents xml par les fonctions de croyance. In *Proceedings of COnférence en Recherche d'Information et Applications (CORIA)*, pages 143–160, 2004.

[129] J. R. Landis and G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1) :159–174, 1977.

[130] B. Larsen, S. Malik, and A. Tombros. A comparison of interactive and adhoc relevance assessments. In *N. Fuhr, M. Lalmas and A. Trotman editors, INEX'07*, pages 348–358. springer, Dagstuhl Castle, Germany, 2007.

[131] R. R. Larson. Cheshire ii at inex : using a hybrid logistic regression and boolean model for xml retrieval. In *Proceedings of the INEX 2002 Workshop*, pages 18–25, 2002.

[132] K.-H. Lee, Y.-C. Choy, and S.-B. Cho. An efficient algorithm to compute differences between structured documents. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(8) :965–979, August 2004.

[133] Y. K. Lee, S.-J. Yoo, K. Yoon, and P. B. Berra. Index structures for structured documents. In *Proceedings of the first ACM international conference on Digital Libraries (DL)*, pages 91–99, 1996.

[134] M. Lehtonen. Extirp2004 : Towards heterogeneity. In *Proceedings of INEX Workshop*, pages 372–381, 2004.

[135] A. Levy, M. Fernãndez, D. Suciu, D. Florescu, and A. Deutsch. Xmlql : A query language for xml. Technical report, World Wide Web Consortium, 1998.

[136] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, 2008.

[137] W. Lian and D. Cheung. An efficient and scalable algorithm for clustering xml documents by structure. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(1) :82–96, August 2004.

[138] J. A. List, V. Mihajlovic, A. Vries, G. Ramirez, and D. Hiemstra. The tijah xml-ir system at inex 2003. In *Proceedings of INEX Workshop*, pages 102–109, 2003.

[139] K.-L. Liu, W. Meng, J. Qiu, C. Yu, V. Raghavan, Z. Wu, Y. Lu, H. He, and H. Zhao. Allinonenews : development and evaluation of a large-scale news metasearch engine. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 1017–1028, 2007.

[140] M. Liu, J. Yan, and Z. Chen. A probabilistic model based approach for blended search. In *Proceedings of the 18th international conference on World Wide Web ACM WWW*, pages 1075–1076, 2009.

[141] S. Liu, Q. Zou, and W. Chu. Configurable indexing and ranking for xml information retrieval. In *Proceedings of the 27th annual international ACM SIGIR*, pages 88–95, 2004.

[142] Z. Liu and Y. Chen. Identifying meaningful return information for xml keyword search. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD)*, pages 329–340, 2007.

[143] S. Lu, Y. Sun, M. Atay, and F. Fotouhi. On the consistency of xml dtds. *Data & Knowledge Engineering (DKE)*, 52(2) :231–247, 2005.

[144] R. Luk, H. Leong, T. Dillon, A. Shan, B. Croft, and J. Allan. A survey in indexing and searching xml documents. *Journal of the American Society for Information Science and Technology (JASIST)*, 53(3) :415–435, 2002.

[145] M. Maaman, Y. Song, A. Paepcke, and H. Garcia-Molina. Assigning textual names to sets of geographic coordinates. *Computers, Environment and Urban Systems*, 30(4) :418–435, 2006.

[146] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, NY, USA, July 2008.

[147] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *ACM Journal*, 7(3) :216–244, 1960.

[148] M. Marx, J. Kamps, and M. de Rijke. The university of amsterdam at inex 2002. In *Proceedings of the INEX Workshop*, pages 23–28, 2002.

[149] Y. Mass and M. Mandelbord. Retrieving the most relevant xml components. In *Proceedings of INEX 2003 Workshop*, pages 53–58, 2003.

[150] Y. Mass and M. Mandelbord. Component ranking and automatic query refinement for xml retrieval. In *Proceedings of the INEX 2004 Workshop*, pages 73–84, 2004.

[151] Y. Mass, M. Mandelbord, E. Amitay, Y. Maarek, and A. Soffer. Juruxml - an xml retrieval system at inex'02. In *Proceedings of the INEX Workshop*, pages 73–80, 2002.

[152] K. McKeown, R. Brazilay, J. Chen, D. Elson, D. Evans, J. Kalvans, A. Nenkova, B. Schiffman, and S. Sigelman. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285, 2002.

[153] D. Miller, T. Leek, and R. Schawartz. markov model information retrieval system. In B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proceedings of the ACM SIGIR*, pages 214–221, 2001.

[154] S. Mizzaro. Relevance, the whole (hi) story. *Journal of the American Society for Information Science and Technology (JASIST)*, 48(9) :810–832, 1997.

[155] V. Moriceau and X. Tannier. Fidji : using syntax for validating answers in multiple documents. *Information Retrieval Journal*, 13 :507–533, 2010.

[156] D. Mountain and A. Macfarlane. Geographic information retrieval in a mobile environment : evaluating the needs of mobile individuals. *Journal of Information Science*, 33(5) :515–530, 2007.

[157] P. Mulhem and J.-P. Chevallet. Modèle de langue par type de doxel pour l'indexation de documents structurés. In *Proceedings of COnférence en Recherche d'Information et Applications (CORIA)*, pages 361–372, 2010.

[158] V. Murdock and M. Lalmas. Workshop on aggregayted search. In *Proceedings of SIGIR*, pages 80–83, 2008.

[159] P. Ogilvie and J. Callan. Combining documents representations of known-item search. In *Proceedings of annual international ACM SIGIR Conference on research and development in Information retrieval*, pages 143–150, 2003.

[160] P. Ogilvie and J. Callan. Using language models for flat text queries in xml retrieval. In *Proceedings of the the Second Annual Workshop of*

the Initiative for the Evaluation of XML retrieval (INEX), pages 12–18, 2003.

[161] S. Ou and S. Khoo. Aggregating search results for social science by extracting and organizing research concepts and relations. In *SIGIR 2008 Workshop on aggregated search*, pages 1–8, 2008.

[162] C. Paris, S. Wan, and P. Thomas. Focused and aggregated search : a perspective from natural language generation. *Information Retrieval Journal*, 44(3) :434–459, 2010.

[163] C. Paris, S. Wan, R. Wilkinson, and M. Wu. Generating personal travel guides - and who wants them ? In *Proceedings of the 8th International Conference on User Modeling (UM)*, pages 251–253. Springer-Verlag, 2001.

[164] S. Park and J. H. Lee. Unified search service of naver, a major korean search engine. In *Proceedings of the ACM SIGIR 2008 Workshop on Aggregated Search*, pages 17–19, 2008.

[165] J. Pearl. Fusion, propagation, and structuring in belief networks. *Journal of Artificial Intelligence*, 29 :241–288, 1986.

[166] J. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[167] J. Perry, M. Berry, and A. Kent. *Machine literature searching*. Western Reserve University Press, Cleveland, Ohio, USA, 1956.

[168] K. Pinel-Sauvagnat. *Modèle flexible pour la Recherche d'Information dans des corpus de documents semi-structurés*. Thèse de Doctorat de l'Université Paul Sabatier, Toulouse, France, 2005.

[169] K. Pinel-Sauvagnat and M. Boughanem. Xfirm : A flexible information retrieval model for indexing and searching xml documents. In *Proceedings of ECIR*, pages 17–18, 2004.

[170] K. Pinel-Sauvagnat and M. Boughanem. A la recherche des nœuds informatifs dans des corpus des documents xml. In *Proceedings CORIA*, pages 119–134, 2005.

[171] K. Pinel-Sauvagnat and M. Boughanem. Propositions pour la pondération des termes et l'évaluation de la pertinence des élémens en recherche d'information structurée. *Journal of Information - Interaction - Intelligence (I3)*, 6(2) :77–98, 2006.

[172] K. Pinel-Sauvagnat, M. Boughanem, and C. Chrisment. Answering content and structure-based queries on xml documents using relevance propagation. *Information Systems Journal*, 31(7) :621–635, 2006.

[173] K. Pinel-Sauvagnat and C. Chrisment. Xml et recherche d'information. In M. Boughanem and J. Savoy, editors, *Recherche d'information : état des lieux et perspectives*, volume 1, chapter 4, pages 99–138. Hermès, avril 2008.

[174] K. Pinel-Sauvagnat, L. Hlaoua, and M. Boughanem. Xml retrieval : what about using contextual relevance ? In *Annual ACM Symposium on Applied Computing (SAC)*, pages 1114–1120, 2006.

[175] B. Piwowarski. *Techniques d'apprentissage pour le traitement d'information structurées : application à la recherche d'information*. Thèse de Doctorat de l'Université Paris 6, Paris, France, 2003.

[176] B. Piwowarski. Working group report : the assessment tool. In *Proceedings of INEX 2003*, pages 181–183, 2003.

[177] B. Piwowarski, G. Faure, and P. Gallinari. Bayesian networks and inex. In *Proceedings of the First Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, pages 149–154, 2002.

[178] B. J. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[179] M. Porter. An algorithm for suffix stripping. *Program*, 14 :130–137, 1980.

[180] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of SIGIR*, pages 691–692, 2006.

[181] A. Ranganathan, A. Riabov, and O. Udrea. Mashup based information retrieval for domain experts. In *Proceedings of the 18th ACM Conference on Information and knowledge Management (CIKM)*, pages 711–720, 2009.

[182] V. C. Rijsbergen. *Information Retrieval*. Butterworth & Co (Publishers)Ltd, London, 1979.

[183] S. Robertson. The probability ranking principle in information retrieval. *Journal of Documentation*, 33(4) :294–304, 1977.

[184] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, 1994.

[185] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec 3. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, pages 109–126, 1994.

[186] S. E. Robertson. Readings in information retrieval. chapter The probability ranking principle in IR, pages 281–286. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[187] T. Roelleke, M. Lalmas, G. Kazai, J. Ruthven, and S. Quicker. The accessibility dimension for structured document retrieval. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 284–302, 2002.

[188] C. Rohr and D. Tjondronegoro. Aggregated cross-media news visualization and personalization. In *Proceedings of the 1st ACM international*

conference on Multimedia Information Retrieval (MIR), pages 371–378, 2008.

[189] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman. Incremental hierarchical clustering of text documents. In *Proceedings of the 15th ACM international Conference on Information and Knowledge Management (CIKM)*, pages 357–366, 2006.

[190] G. Salton. A comparison between manual and automatic indexing methods. *Journal of American Documentation (JAD)*, 20(1) :61–71, 1971.

[191] G. Salton, editor. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[192] G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, 1993.

[193] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, 1987.

[194] G. Salton and M. McGill. *The concept of "relevance" in information science : A historical review*. R.R. Bowker, New York, 1970.

[195] G. Salton and M. McGill, editors. *Introduction to modern information retrieval*. McGraw-Hill Int. Book Co, 1983.

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

[197] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of American Documentation (JAD)*, 29(4) :351–372, 1973.

[198] M. Sanderson and J. Kohler. Analyzing geographic queries. In *Workshop on Geographic Information Retrieval*, pages 1–2, 2006.

[199] R. Schenkel, F. Suchanek, and G. Kasneci. Yawn : A semantically annotated wikipedia xml corpus. In *12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW 2007)*, volume 103, pages 277–291. Lecture Notes in Informatics, 2007.

[200] T. Schlieder and H. Meuss. Querying and ranking xml documents. *Journal of the American Society for Information Science and Technology (JASIST)*, 53(6) :489–503, 2002.

[201] D. Shin, H. Jang, and H. Jin. Bus : an effective indexing and retrieval scheme in structured documents. In *Proceedings of the third ACM international conference on Digital Libraries (DL)*, pages 235–243, 1998.

[202] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An element-based approach to xml retrieval. In *Proceedings of INEX 2003 workshop*, pages 19–26, 2003.

[203] K. Sparck-Jones, S. E. Robertson, and M. Sanderson. Ambiguous requests : implications for retrieval tests, systems and theories. In *Proceedings of SIGIR forum*, pages 8–17, 2007.

[204] N. Stokes and J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In *Proceedings of the ACM SIGIR*, pages 424–425, 2001.

[205] A. Strotmann and D. Zhao. Bibliometric maps for aggregated visual browsing in digital libraries. In *SIGIR 2008 Workshop on aggregated search*, pages 9–16, 2008.

[206] S. Sushmita, H. Joho, and M. Lalmas. A task-based evaluation of an aggregated search interface. In *Proceedings of the 16th International Symposium on String Processing and Information Retrieval (SPIRE)*, pages 322–333, 2009.

[207] S. Sushmita, H. Joho, M. Lalmas, and J. M. Lose. Understanding domain relevance in web search. In *WWW 2009 Workshop on Web Search Result Summarization and Presentation*, pages 70–74, 2009.

[208] S. Sushmita, H. Joho, M. Lalmas, and R. Villa. Factors affecting clickthrough behavior in aggregated search interfaces. In *Proceedings of the 19th ACM international Conference on Information and Knowledge Management(CIKM)*, pages 519–528, 2010.

[209] S. Sushmita, M. Lalmas, and A. Tombros. Using digest pages to increase user result space : preliminary designs. In *Proceedings of the ACM SIGIR 2008 Workshop on Aggregated Search*, pages 20–26, 2008.

[210] Z. Szlávik, A. Tombros, and M. Lalmas. Feature and query-based table of contents generation for xml documents. In *Proceedings of the 29th ECIR Conference*, pages 456–467. Spring-Verlag, 2007.

[211] L. Tamine and S. Calabretto. Recherche d'information contextuelle et web. In M. Boughanem and J. Savoy, editors, *Recherche d'information : état des lieux et perspectives*, volume 1, chapter 7, pages 201–224. Hermès, avril 2008.

[212] A. Theoblad and G. Weikum. The index-based xxl search engine for querying xml data with relevance ranking. In *Proceedings of the 8th International Conference on Extending Database Technology (EDBT)*, pages 477–495, 2002.

[213] P. Thomas, K. Noack, and C. Paris. Evaluating interfaces for government metasearch. In *Proceedings of the third symposium on Information interaction in context (IIiX)*, pages 65–74, 2010.

[214] H. Tong, J. He, M. Li, C. Zhang, and W.-Y. Ma. Graph based multimedia learning. In *Proceedings of the 13th annual ACM International Conference on Multimedia*, pages 862–871, 2005.

[215] G. Torsten. *Storage and retrieval of xml documents within a cluter of database systems*. Thèse de Doctorat, Institut fédéral de technologie, Zurich, Suisse, 2003.

[216] A. Trotman. Choosing document structure weights. *International Journal of Information Processing and Management (IPM)*, 41(2) :243–264, 2005.

[217] A. Trotman and R. A. O'Keefe. Identifing and ranking relevant document element. In *Proceedings of INEX 2003 Workshop*, pages 149–154, 2003.

[218] A. Trotman and B. Sigurbjörnsson. Narrowed extended xpath i (nexi). In *Proceedings of INEX 2004 Workshop* [81], pages 219–237.

[219] A. Trotman and B. Sigurbjörnsson. Nexi, now and next. In *Proceedings of INEX 2004*, pages 10–15, 2004.

[220] H. Turtle. *Inference networks for document retrieval*. Ph.D. Thesis, University of Massachusetts, Amherst, MA, USA, 1991.

[221] S. Vaid, B. C. Jones, H. Joho, and M. Sanderson. Spatio-textual indexing for geographical search on the web. In *Proceedings of the 9th international conference on Advances in Spatial and Temporal Databases (SSTD)*, pages 218–235, 2005.

[222] D. Vallet and H. Zaragoza. Inferring the most important types of a query : a semantic approach. In *Proceedings of the the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 857–858, 2008.

[223] J.-N. Vittaut, B. Piwowarski, and P. Gallinari. An algebra for structured queries in bayesian networks. In *Pre-proceedings of INEX 2004*, pages 58–65, 2004.

[224] E. M. Voorhees. Proceedings of the 8th text retrieval conference. In *TREC-8 Question Answering Track Report*, pages 77–82, 1999.

[225] E. M. Voorhees, N. K. Gupta, and J. Laird. The collection fusion problem. In *TREC*, 1994.

[226] H.-T. Vu, L. Denoyer, and P. Gallinari. Un modèle statistique pour la classification de documents structurés. In *Actes de 3ème conférence internationale francophone Extraction et Gestion des Connaissances, EGC 2003*, pages 233–246, 2003.

[227] S. Walker, S. E. Robertson, M. Boughanem, G. J. F. Jones, and K. Sparck Jones. Okapi at trec-6 automatic ad hoc, vlc, routing, filtering and qsdr. In *TREC*, pages 125–136, 1997.

[228] F. Weigel, K. Shulz, and H. Meuss. Ranked retrieval of structured doucments with the sterm vector space model. In *Proceedings of the INEX 2004 Workshop*, pages 126–133, 2004.

[229] R. Wilkinson. Effective retrieval of structured documents. In *the 17th ACM SIGIR 1994*, pages 311–317, 1994.

[230] J. E. Wolff, H. Florke, and A. B. Cremers. Searching and browsing collections of structural information. In *Proceedings of IEEE Advances in Digital Libraries (ADL)*, pages 141–150, 2000.

[231] A. Woodley and S. Geva. Nlpx at inex 2004. In *N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors, INEX'04*, pages 382–394. springer, 2004.

[232] M. Wu and M. Fuller. Supporting the answering process. In *Proceedings of the Second Australian Document Computing Symposium*, pages 65–73, 1997.

[233] J. Xu and B. Croft. Corpus based stemming using cooccurrence of word variants. In *ACM Transactions on Information Systems*, pages 61–81, 1998.

[234] R. Yager and H. L. Larsen. Retrieving information by fuzzification of queries. *Journal of Intelligent Information Systems*, 4(2) :106–119, 1993.

[235] G.-W. You, S.-W. Hwang, Z. Nie, and J.-R. Wen. Social search : enhancing entity search with social network matching. In *Proceedings of the 14th International Conference on Extending Database Technology (EDBT)*, pages 515–519, New York, NY, USA, 2011. ACM.

[236] H. Zargayouna. Contexte et sémantique pour une indexation de documents sémi-structurés. In *Proceedings CORIA*, pages 571–581, 2004.

[237] H. Zeng, Q. He, Z. Chen, and W. Ma. Learning to cluster web search results. In *Proceedings of the ACM SIGIR*, pages 210–217, 2004.

[238] C.-X. Zhai. Statistical language models for information retrieval a critical review. *Journal Foundations and Trends in Information Retrieval (FTIR)*, 2(3) :137–213, 2008.

[239] Y. Zhang, P. J. Callan, and P. T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the ACM SIGIR*, pages 81–88, 2002.

[240] K. Zhou, R. Cummins, and M. Lalmas. Evaluating large scale distributed vertical search. In *Proceedings of the 9th International Workshop on Large-Scale and Distributed Systems for Information Retrieval (LSD-SIR)*, pages 9–14, 2011.