Un Modèle Bayésien pour l'Agrégation des documents XML

Najeh NAFFAKHI*' – Mohand BOUGHANEM* – Rim FAIZ****

**IRIT, Equipe SIG-RFI*
118, route de Narbonne, 31062 Toulouse Cedex 9
{najeh.naffakhi, mohand.boughanem}@irit.fr
***LARODEC, IHEC, Carthage Présidence, 2016 Tunis*
rim.faiz@ihec.rnu.tn

RÉSUMÉ. *Dans cet article, nous nous intéressons à la recherche agrégée dans des documents structurés XML. Pour cela, nous proposons un modèle de recherche d'information structurée 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 trouver des éléments. Ainsi, au lieu de récupérer une liste d'éléments potentiellement (ou partiellement) pertinents vis-à-vis la requête, notre objectif est de rassembler dans un agrégat des éléments pertinents, non-redondants et complémentaires susceptibles de mieux répondre à la requête. Nous avons évalué notre approche sur une collection de documents XML issus d'INEX 2005 et avons présenté quelques résultats expérimentaux mettant en évidence l'impact de l'agrégation de tels éléments.*

ABSTRACT. *In this paper, we are interested in aggregated search in structured XML documents. We present a structured information retrieval model based on the Bayesian networks theory. Relations query-terms and terms-elements are modeled through probability. In this model, the user's query starts a process of propagation to recover the elements. Thus, instead of retrieving a list of elements that are likely to answer partially the user's query, our objective is to build a virtual elements that contain relevant, non-redundant and complementary elements, that are likely to answer better the query that elements taken separatelly. We evaluated our approach using INEX 2005 collection and presented some empirical results for evaluating the impact of the aggregation approach.*

MOTS-CLÉS : *recherche agrégée, réseaux bayésiens, redondance, complementairité.*

KEYWORDS: *aggregated search, Bayesian networks theory, redundancy, complementarity.*

1. Introduction

Un Système de Recherche d'Information (SRI) retourne, en réponse à la requête d'un utilisateur, une liste de documents ordonnée selon leur score de pertinence vis-à-vis la requête. Cette présentation des résultats sous forme d'une liste ordonnée de résultats, employée par la majorité des SRI, impose à l'utilisateur de la parcourir linéairement en examinant 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. A partir de quel moment est-on certain d'avoir collecté assez d'information. La recherche agrégée vient en partie pour répondre à ce type d'attente. Son objectif est d'assembler, combiner des informations issues de sources diverses, afin de construire des réponses comportant toute l'information pertinente pour la requête (Murdock *et al.,* 2008).

Les moteurs de recherche sont capables de retrouver différents types d'information à différentes granularités. L'information recherchée sur le Web peut être composée de plusieurs éléments de documents (tableaux HTML, listes, documents XML, feuilles de style, etc.) appartenant à différentes sources et peut contenir aussi bien du texte que des images et des vidéos. La RI agrégée tente d'identifier les éléments pertinents, de les organiser et de les présenter à l'utilisateur afin de simplifier sa recherche (Sushmita *et al.,* 2008). La RI agrégée peut également offrir une vision plus riche de l'information existante. Aujourd'hui, les travaux de recherche sur les documents Web tentent d'identifier les relations entre les éléments qui peuvent être utiles pour effectuer cette agrégation (appelée aussi intégration des données à grande échelle), par exemple, le SRI *OCTOPUS* de (Cafarella *et al.,* 2009) détermine les relations entre les tableaux HTML et Kopliku (Kopliku *et al.,* 2010) utilise les listes de documents Web afin de recupérer les informations pertinentes. Yahoo ! Pipes[1] est un outil qui permet de créer visuellement des mashups[2]. C'est un outil agrégateur et manipulateur intéractif de données.

Nous nous intéressons dans cet article à l'agrégation dans la Recherche d'Information Structurée (RIS). Contrairement à la RI classique qui traite et renvoie le document en réponse à une requête, la RI structurée traite les documents XML selon une granularité plus fine, correspondant aux éléments du document qui sont donc renvoyés en réponse à une requête (Torjmen *et al.,* 2009) (Piwowarski *et al.,* 2005). D'autres SRI commencent à présenter les résultats d'une requête sous forme de résumés (Liu *et al.,* 2009) (Huang *et al.,* 2008) (Polyzotis, 2006).

Nous nous intéressons particulièrement au problème de l'agrégation des éléments XML répondant à des requêtes composées uniquement de mots clés. Il s'agit ainsi d'affronter des problèmatiques liées à la pertinence des éléments XML, la diversité du contenu retourné (texte, image, etc.), la couverture des différents aspects (sujets) de la requête formulée, la non-redondance (les éléments retournés ne véhiculent pas

1. http ://pipes.yahoo.com/pipes/.
2. C'est une application qui combine du contenu ou du service provenant de plusieurs applications plus ou moins hétérogènes.

la même information) ainsi qu'à leur granilarité. L'agrégation de ces éléments représente un *agrégat*. Cet agrégat cherche à produire un contenu agrégé reprenant les informations les plus pertinentes, non-redondantes et complémentaires qu'un utilisateur pourra trouver dans un corpus de documents XML en rapport avec sa requête. Notre objectif est de permettre à un utilisateur de localiser les informations les plus pertinentes.

Cet article est structuré de la manière suivante. La section 2 présente quelques travaux proches. La section 3 décrit le modèle que nous proposons. La section 4 présente quelques résultats expérimentaux évaluant l'impact de l'agrégation des éléments XML. La section 5 conclut et présente les principales perspectives associées à notre approche.

2. Travaux proches

Il est bien connu que, dans le contexte de la recherche Web, les utilisateurs accèdent généralement à un très petit nombre de documents (Jansen *et al.*, 2006). Une étude sur les utilisateurs du Web dans (Spink *et al.*, 2002), a montré, que le pourcentage d'utilisateurs qui consultent moins de documents (pages Web) par requête augmente avec le temps. Par exemple, de 1997 à 2001, le pourcentage d'utilisateurs examinant un document par requête est passé de 28,6% à 50,5%. Ce pourcentage s'est encore accru à plus de 70% après 2001 (Sushmita *et al.*, 2008). Cela donne à penser que pour une liste des documents est principalement confiné aux documents contenus dans le premier, le deuxième et parfois (au plus) le troisième rang. L'étude rapportée dans (Jansen *et al.*, 2005) a montré que sur 10 documents affichés, 60% des utilisateurs ont examiné moins de 5 documents et près de 30% ont lu un seul document.

La recherche agrégée permet d'apporter des solutions à ce problème. En effet, son objectif est d'intégrer d'autres types de documents (pour l'instant on peut trouver des documents web, des images, des vidéos, des cartes, des actualités, des livres) dans la page de résultats. Exemple des moteurs de RI qui commencent à faire l'agrégation, nous trouvons Google Universal Search[3], Yahoo ! alpha[4], etc. Les utilisateurs ont accès ensuite à différents types de documents. Ceci peut être bénéfique pour certaines requêtes, de type par exemple "voyage à Londres", peut retourner des cartes, des blogs, météo, etc. Toutefois, la recherche agrégée peut être utilisée en conjonction avec la technique Digest pages proposée par (Sushmita *et al.*, 2008) afin d'améliorer la page de résultats. Il s'agit de construire un document fictif à partir du regroupement des documents retournés par un moteur de recherche sous forme des clusters. Ce document fictif est considéré comme la réponse à la requête où chaque cluster correspond à des résumés de documents web retournés.

Une autre technique qui peut être utilisée afin d'améliorer la page de résultats est le clustering. Toutefois, il ne suffit pas simplement de retourner des clusters. Il est impor-

3. http ://www.google.com/intl/en/press/pressrel/universalsearch-20070516.html
4. http ://au.alpha.yahoo.com/

tant de fournir aux utilisateurs une sorte d'aperçu du contenu des documents formant un cluster (Sushmita *et al.*, 2008). Une approche commune pour fournir une telle vue d'ensemble est le résumé multidocuments, des exemples de systèmes basés sur cette technique, nous trouvons : NewInEssence (Radev *et al.*, 2005), QCS (Dunlavy *et al.*, 2007), etc.

D'autres SRI commencent à présenter les résultats d'une requête sur des documents XML sous forme des résumés. Par exemple, eXtract (Huang *et al.*, 2008) est un SRI qui génère des résultats sous forme des fragments XML. 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. XCLUSTERs (Polyzotis, 2006) est un nouveau modèle de représentation des résumés XML. XCLUSTERs regroupe des éléments XML basés sur la structure et le contenu et utilise un petit espace pour stocker les données XML tout en atteignant des résultats précis, autant que possible pour le traitement des requêtes. Les techniques sont donc mises au point pour la compression de données, par exemple, de stocker une instance de chaque nom de balise distincte avec son nombre d'occurrences, la fréquence d'un nœud imbriqué dans un autre nœud, etc. L'objectif de ces résumés est de fournir des extraits significatifs permettant aux utilisateurs de mieux évaluer la pertinence des résultats de la requête correspondante. Cependant un résumé est souvent désagréable à lire (Liu *et al.*, 2009).

Notre approche se situe à la jonction de la recherche des 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 d'un corpus de documents XML qui répondent le mieux au besoin de l'utilisateur formulé à travers une liste de mots clés. Le modèle que nous proposons trouve ses fondements théoriques dans les réseaux bayésiens. La structure réseau fournit une manière naturelle de représenter les éléments du corpus de documents XML et leurs relations. Quant à la théorie des probabilités, elle permet d'estimer les différentes valeurs sous jacentes (Naffakhi *et al.*, 2010). Ces valeurs permet notamment de mesurer à quel point une réponse à la requête contient des éléments pertinents, non-redondants et complémentaires.

Outre les points cités ci-dessus, le cadre théorique qui supporte nos propositions, en l'occurrence les réseaux bayésiens nous différencie clairement des cadres utilisés dans les approches précédentes.

3. Un modèle de recherche agrégée basé sur le réseau bayésien

Le modèle que nous proposons est représenté par un Résean Bayésien (RB) de topologie illustré par la figure 1. D'un point de vue qualitatif, le graphe permet de représenter un document XML, ses éléments et les termes d'indexation. Les liens entre les nœuds permettent de représenter les relations de dépendances entre les différents

nœuds. Ces relations sont issues de la représentation DOM[5] d'un document XML. D'un point de vue quantitatif, notre modèle manipule des probabilités pour estimer des valeurs sur les nœuds.

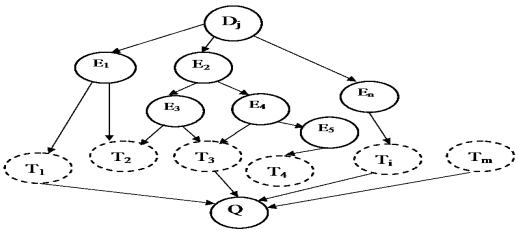

Figure 1. *Architecture générale du modèle*

3.1. *Description du modèle bayésien*

Le nœud D_j représente un document de la collection C. Chaque nœud D_j représente une variable aléatoire binaire. L'instanciation $D_j = 1$ signifie que le document est activé (choisi). Nous nous intéressons qu'au cas où le document D_j est activé, et nous le notons D_j.

Les nœuds E_1, E_2,..., E_n représentent les éléments du document D_j. Chaque nœud E_j représente une variable aléatoire prenant des valeurs binaires dans l'ensemble $dom(E_j) = \{1, 0\}$. L'instanciation $E_j = 1$ 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. Chaque nœud terme T_i représente une variable aléatoire binaire prenant des valeurs dans l'ensemble $dom(T_i) = \{1, 0\}$ où l'instanciation $T_i = 1$ signifie que le terme T_i est présent dans l'objet (nœud père auquel il est relié c.-à-d. le nœud balise E_j contient ce terme T_i ou requête Q) et donc représentatif de l'objet. Il faut noter qu'un terme est relié aussi bien au nœud qui le comporte qu'à tous les ascendants de ce dernier.

Nous ne considérons que le cas $Q = 1$ et nous le notons Q.

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_j et assigner des valeurs aux relations de dépendances entre nœud requête-nœuds termes, nœuds termes-nœuds éléments et nœuds éléments-nœud racine. Ces valeurs dépendent du sens que nous donnons à ces relations.

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_j. Chaque terme T_i,

5. On dit souvent : le *DOM*, de l'anglais *Document Object Model*.

$T_i \in T$ 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. En fait, nous considérons le sous-réseau composé des nœuds éléments et de leurs termes d'indexation.

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 $T(Q)$ (resp. $T(E)$) l'ensemble des termes de la requête Q (resp. des éléments de documents). 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.

Nous considérons qu'une configuration θ_i est une instanciation possible des variables éléments. Dans une configuration, nous représentons que les varaibles instanciés à 1 . Un exemple d'une configuration $\theta_i = \{E_1 = 1, E_3 = 1, E_5 = 1\}$ ou peut aussi noté $\theta_i = \{E_1, E_3, E_5\}$. Une configuration donnée est considérée comme une réponse possible à la requête.

3.2. *Evaluation d'une 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 d'évaluation consiste à propager l'information injectée par le neœud requête vers le nœud document ; puis nous calculons pour chaque configuration potentielle sa valeur de pertinence, de non-redondance et de complémentarité. A l'issue du processus de propagation, chaque configuration aura un score issu des ces trois valeurs. La configuration retenue est celle qui présente le meilleur score. Cette configuration représentative d'un document sera rassembler avec d'autres configurations issues d'autres documents pour former un agrégat réponse à la requête. Nous considérons que le nœud R est le nœud racine de l'agrégat, réponse à une requête donnée Q. En fait, nous considérons que tous les documents sont rattachés à R. Donc, la généralisation du processus d'évaluation de notre modèle au niveau multi-documents permet de restituer un agrégat réponse à la requête de l'utilisateur. Nous décrivons dans ce qui suit, les différentes étapes pour propager une requête au niveau d'un document afin de déterminer la meilleur configuration.

Instancier le système par la réception de la requête Q. Il existe une instanciation de l'ensemble des parents de la requête, les nœuds termes, qui représentent la requête dans sa forme la plus stricte (exactement telle que formulée par l'utilisateur). Soit $T(Q)$ cette instanciation.

Le processus de propagation évalue les valeurs de probabilité entre tous les éléments d'une configuration θ_i. Avec notre modèle, la probabilité jointe d'observer une requête Q et sa réponse dans un document D_j est donnée par

$$P(Q, D_j{=}1) = P(Q|T(Q)) \times P(T(Q)|\theta_i) \times P(\theta_i|D_j{=}1) \quad [1]$$

Afin d'évaluer les différents facteurs de probabilité dans la formule [1], nous procédons de la manière suivante.

En considérant le premier facteur, $P(Q|T(Q))$, est la probabilité de la requête étant donnée ces termes, dépend de l'interprétation de la requête. En fait, plusieurs interprétations sont possibles. Les termes de la requête peuvent être connectés par une conjonction, une disjonction, ou par une somme probabiliste, ou encore par une somme probabiliste pondérée. Ces deux dernières agrégations des termes de la requête ont été déjà proposées dans les travaux (Turtle *et al.,* 1990) (Boughanem *et al.,* 2009). L'idée majeure de cette agrégation est de mesurer la conformité d'une configuration possible des termes de la requête en l'occurrence à celle trouvée dans une configuration possible qui indexent les éléments d'un document. Dans notre cas, nous nous intéressons uniquement au cas $Q = 1$. 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 que l'ensemble des termes d'indexation des éléments sont eux aussi de la requête Q et ces termes sont indépendants alors nous pouvons transformer le premier facteur en :

$$P(Q|T(Q))=P(Q|T_1,...,T_m) = \prod_{T_k \in T(Q)} P(Q|T_k) \quad [2]$$

Pour une requête donnée, le processus d'évaluation restitue une configuration dont ses éléments sont indexés par au moins un termes de la requête. Donc un terme de la requête est instancié à 1 s'il est représentatif d'un élément donnée. Ainsi, $P(Q|T_k)= 1$ pour tout terme $\forall T_k \in (T(Q) \wedge T(E))$ et 0 sinon.

3.2.1. *Pertinence*

Le deuxième facteur de la formule [1]

$$P(T(Q)|\theta_i)=P(T_1,...,T_m|\theta_i) = \prod_{T_k \in (T(Q) \wedge T(\theta_i))} P(T_k=1|\theta_i) \quad [3]$$

Dans une configuration donnée, un terme représentatif d'un élément est un terme qui contribue à sa restitution en réponse à une requête. Le degré d'importance d'un terme dans une telle configuration est représenté par la quantité $P(T_k = 1|\theta_i)$. En fait, nous avons besoin de cette quantité pour déterminer la pertinence de cette configuration étant donnée une requête. Cette quantité est estimée par :

$$P(T_k = 1|\theta_i) = \frac{\sum_{\forall \theta_i^{j,\cdot} \in \theta_i} tf_k^{j,\cdot}}{tf_d} \quad [4]$$

avec :

– $\sum_{\forall \theta_i^{j,\cdot} \in \theta_i} tf_k^{j,\cdot}$ est la fréquence du terme T_k dans la configuration θ_i.

– tf_d est la fréquence du terme dans un document d.

3.2.2. *Redondance*

Dans chaque configuration, nous nous intéressons à agréger des éléments qui ne véhiculent pas la même inforamtion à des granularités différentes. Cette redondance est traité dans notre modèle à deux niveaux : un niveau structurelle supporté par l'hypothèse **H1** et un niveau contenu supporté par l'hypothèse **H2**.

– **Hypothèse 1 (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 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.

– **Hypothèse 2 (H2)** : cette hypothèse est considérée comme contrainte de contenu. Cette hypothèse est appliquée lorsque nous agrégeons les éléments issus de différentes configurations (inter-documents). Nous formulons cette problématique par la mesure de la redondance d'un élément θ_i^j par rapport autres éléments de l'agrégat réponse à la requête. Dans la littérature et dans le cadre d'évaluation Text REtrieval Conference (TREC), nous trouvons les approches les plus étroitement liées à la détection de nouveauté/redondance[6]. Parmi ces approches, il y a celles qui se basent sur la technique de clustering pour mesurer la redondance d'un document par sa distance à chaque cluster (Miller *et al.,* 2001) (Stokes *et al.,* 2001) (Franz *et al.,* 2001). Une autre approche mesure la redondance en se basant sur la distance entre un document et chacun des autres documents (Zhang *et al.,* 2002). Quand on demande à un ensemble des évaluateurs d'annoter un ensemble de données d'évaluation, ils ont trouvé qu'il était plus facile pour eux d'identifer un nouveau document comme étant redondant avec un autre document vu précédemment et plus difficile de l'identifier comme redondant avec un cluster de documents vu précédemment.

Afin de simplifier notre modèle, nous nous utilisons une mesure de similarité vectorielle classique pour détecter les éléments redondants dans un agrégat.

3.2.3. *Complémentarité*

Le troisième facteur de la formule [1] $P(\theta_i|D_j{=}1)$, mesure la complémentarité entre les éléments d'une configuration possible. Les éléments regroupés dans une telle configuration sont indépendants alors les hypothèses d'indépendance conditionnelle nous permettent ensuite d'écrire :

$$P(\theta_i|D_j{=}1){=}\prod_{j=1}^{|\theta_i|}P(\theta_i^j|D_j{=}1) \qquad [5]$$

6. Clarke et al. (Clarke *et al.,* 2008) proposent un cadre d'évaluation, dans TREC, pour mesurer systématiquement la nouveauté et la diversité. La mesure proposée se base sur le gain cumulé.

L'intérêt de propager une information complémentaire d'un élément θ_i^j vers la racine du document D_j dans une configuration donnée θ_i indique à quel point cet élément ajoute ce qu'il manquait en matière d'information. Les éléments loin du nœud racine du document D_j 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 alors 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_j,θ_i^j) et $dist$(D_j, $élément\,plus\,profond$(θ_i^j)), qui représentent respectivement la distance entre le nœud racine D_j et un de ses nœuds descendants θ_i^j du document (relativement à une configuration donnée θ_i) et la distance entre le nœud racine D_j et le $plus\,profond\,élément$ muni du nœud θ_i^j noté θ_i^k. La distance entre deux nœuds quelconques est déterminée par le nombre d'arcs les séparants. La mesure de probabilité de propagation d'un élément θ_i^j, supposé complémentaire dans une configuration θ_i, vers le nœud racine D_j est quantifiée comme suit :

$$P(\theta_i^j|D_j=1) = \frac{dist(D_j,\theta_i^j)}{dist(D_j,élément\,plus\,profond(\theta_i^k))} \qquad [6]$$

La formule [6] indique que plus un nœud est proche de la racine, moins il contribue à la complémentarité d'une configuration donnée. A titre d'exemple et dans la figure 1, les contributions des éléments E_2 et E_4 notés respectivement θ_i^2 et θ_i^4 (dans ce cas, l'élément le plus profond est E_5 et sera noté par θ_i^5), dans la complémentarité d'une configuration θ_i seront estimés comme suit :

$$P(\theta_i^2|D_j=1) = \frac{dist(D_j,\theta_i^2)}{dist(D_j,\theta_i^5)} = \frac{1}{3},\ P(\theta_i^4|D_j=1) = \frac{dist(D_j,\theta_i^4)}{dist(D_j,\theta_i^5)} = \frac{2}{3} \qquad [7]$$

Finalement, la probabilité jointe de la formule [1] se simplifie en :

$$\prod_{T_k\in T(Q)} P(Q|T_k) \ \times\ \prod_{T_k\in(T(Q)\wedge T(\theta_i))} P(T_k|\theta_i) \ \times\ \prod_{j=1}^{|\theta_i|} P(\theta_i^j|D_j{=}1) \qquad [8]$$

La configuration qui sera retenue est celle qui optimise la formule 8 vérifiant

$$\underset{\theta_i^*\in\theta}{argmax}\ \Big(\prod_{T_k\in T(Q)} P(Q|T_k) \times \prod_{T_k\in(T(Q)\wedge T(\theta_i))} P(T_k|\theta_i) \times \prod_{j=1}^{|\theta_i|} P(\theta_i^j|D_j{=}1)\Big) \qquad [9]$$

La configuration qui sera sélectionnée θ_i^* sera celle qui comporte les termes de la requête et celle qui maximise la pertinence et la complémentarité de chacun de ses éléments et qui minimise leur redondance en terme de mesure de similarité. Cette configuration représentative dun document sera rassembler avec dautres configurations issues dautres documents pour former un agrégat réponse à la requête.

4. Evaluation expérimentale

Dans le but de valider notre approche, nous avons mené une expérimentation permettant d'évaluer l'impact de l'agrégation des éléments XML.

4.1. *INEX : Collection*

Nous nous appuyons pour l'évaluation des performances sur la collection de test fournie dans le cadre de la compagne d'évaluation INEX 2005 *(INitiative for the Evaluation of XML Retrieval)*. Cette collection présente une extension de la collection 2004 composée d'articles scientifiques provenant de l'IEEE Computer Society, balisés au format XML. Elle comporte 16819 articles publiés de 1995 à 2004 provenant de 21 magazines ou revues différents ayant une taille totale d'environ 1,3 gigaoctets. En moyenne, un article contient 1532 nœuds XML, où la profondeur moyenne d'un nœud est 6,9. La collection contient au total 8 millions de nœuds et 180 balises différentes.

4.2. *Stratégie d'évaluation*

En absence de cadre approprié pour l'évaluation de la valeur des agrégats, nous nous sommes limités à évaluer les éléments de ces agrégats. En effet, nous trions les agrégats selon un score calculé (cf. formule 9) puis 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 le système XFIRM (Sauvagnat, 2005). 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 le système XFIRM selon la tâche *Focused* (sans overlap). Cette tâche demande le renvoi pour chaque requête d'éléments non imbriqués. L'intérêt ici concerne les éléments les plus précis liés à un besoin d'information, sans permettre de recouvrement entre eux.

4.3. *Expérimentation*

Nous nous limitons dans cet article à des valeurs de gain cumulé à 10 et 25 éléments parce que le nombre des éléments de tous les agrégats ne dépasse pas 25. Les mesures d'évaluation utilisés durant la campagne 2005 sont basées sur les mesures nxCG et MAep. Ces mesures sont calculées en tenant compte de deux dimensions de pertinence (exhaustivité et spécificité) agrégées en une seule valeur. Deux types de fonctions de quantification sont utilisées :

– une quantification stricte pour évaluer si un SRI est capable de retrouver des éléments très spécifiques et très exhaustifs

$$f_{strict}(e,s)= \begin{cases} 1 & \text{si } e = 2 \; et \; s = 1 \\ 0 & \text{sinon} \end{cases}$$

– une quantification généralisée pour évaluer les éléments selon leur degré de pertinence

$$f_{gen}(e,s)= \; e * s$$

Dans le tableau 1, nous montrons les résultats obtenus en gain cumulé nxCG et MAep selon la quantification généralisée alors que dans le tableau 2 la quantification est stricte. Le nxCG[i] reflète le gain relatif de l'utilisateur accumulé jusqu'à un rang i, comparé à ce qu'il aurait dû atteindre si le système avait produit une liste triée optimale. Le MAep indique la moyenne non interpolée d'effort-précision. L'effort-précision (ep) à un niveau donné de gain-rappel (gr) mesure l'effort d'un utilisateur pour atteindre un gain relatif au gain total qu'il peut obtenir. La MAep est utilisé pour moyenner les valeurs d'effort-précision pour chaque rang auquel chaque réponse pertinente est retournée. Un premier résultat important que l'on peut tirer de cette

Overlap=on, Quant=gen			
RunId	nxCG[10]	nxCG[25]	MAep
Meilleur résultat	0.2688	0.2325	0.0737
XFIRM	0.1037	0.1044	0.0203
Notre Modèle	0.1764	0.1343	0.0371

Tableau 1. *Résultats comparatifs, tâche CO.Focused, quantification* f_{gen}

Overlap=on, Quant=strict			
RunId	nxCG[10]	nxCG[25]	MAep
Meilleur résultat	0.1401	0.1432	0.0741
XFIRM	0.0135	0.0204	0.0036
Notre Modèle	0.0182	0.0243	0.0039

Tableau 2. *Résultats comparatifs, tâche CO.Focused, quantification* f_{strict}

expérimentation est qu'on observe des améliorations significatives à partir des 10 premiers éléments (nxCG[10]) et au niveau du MAep pour les deux mesures généralisées et strictes par rapport au modèle XFIRM. Nous remarquons que nos résultats sont un peu loin par rapport aux meilleurs résultats obtenus par les participants d'INEX 2005. D'autres facteurs influencent sur les résultats comme l'indexation, la propagation du contenu doivent être encore étudiés.

5. Conclusion et Perspectives

Dans cet article, l'approche proposée fournit un cadre formel pour agréger des éléments pertinents et non-redondants. La complémentarité entre les éléments se fonde sur le contenu et la structure du document XML en utilisant les mesures de probabilités. Ces mesures sont utilisées pour quantifier les relations de dépendances entre la requête et une configuration donnée.

L'un de nos objectifs ultérieurs est d'essayer d'autres formules de pondération de termes afin d'améliorer nos résultats expérimentaux. Nous visons, en outre, à proposer une mesure d'évaluation de la redondance entre les éléments agrégés et d'en évaluer l'approche proposée sur une version plus récente d'INEX.

6. Bibliographie

Boughanem M., Brini A., Dubois D., « Possibilistic networks for information retrieval », *International Journal of Approximate Reasoning*, vol. 50, p. 957-968, 2009.

Cafarella J. M., Halevy A., Khoussainova N., « Data integration for the Relational Web », *35th International Conference on Very Large Data Bases*, 2009.

Carbonell J., Goldstein J., « The use of MMR, diversity-based reranking for reordering documents and producing summaries », *Special Interest Group on Information Retrieval SIGIR'98*, p. 335-336, 1998.

Clarke C., Kolla M., Cormack G., Vechtomova O., Ashkan A., Büttcher S., I. M., « Novelty and Diversity in Information Retrieval Evaluation », *Special Interest Group on Information Retrieval SIGIR'08*, p. 659-664, 2008.

Dunlavy D. M., O'Leary D. P., Conroy J. M., Schlesinger J. D., « QCS : A system for querying, clustering and summarizing documents », *Information Processing and Management*, p. 1588-1605, 2007.

Franz M., Ittycheriah A., McCarley J., Ward T., « First story detection : Combining similarity and novelty based approaches », *Topic Detection and Tracking Workshop Report*, 2001.

Huang Y., Liu Z., Chen Y., « Query biased snippet generation in XML search », *Special Interest Group on Management Of Data SIGMOD'08*, p. 315-326, 2008.

Jansen B. J., Spink A., « An Analysis of document viewing pattern of web search engine user », *Web Mining : Applications and Techniques*, p. 339-354, 2005.

Jansen B. J., Spink A., « How are we searching the world wide web ? : a comparison of nine search engine transaction logs », *Information Processing and Management*, p. 248-263, 2006.

Jones W., Furnas G. W., « Pictures of relevance », *Journal of the American Society for Information Science*, 1987.

Kazi S., Lalmas M., « INEX 2005 Evaluation Measures », *4th International Workshop of INitiative for the Evaluation of XML Retrieval 2005*, 2005.

Kopliku A., Boughanem M., Sauvagnat K., « Querying by examples », *COnférence en Recherche d'Information et Applications CORIA*, p. 407-408, 2010.

Kraaij W., Pohlmann R., Hiemstra D., « Twenty-one at TREC-8 : using language technology for information retrieval », *Proceedings of the 8th Text Retrieval Conference (TREC)*, 1999.

Lee L., « Measures of distributional similarity », *37th Annual Meeting of the Association for Computational Linguistics ACL*, 1999.

Lee Y. K., Yoo S., Yoon K., Berra P., « Index structures for structured documents », *in DL'96 : Proc. of the first ACM international conference on Digital Libraries*, p. 91-99, 1996.

Liu Z., Sun P., Huang Y., Chen Y., « Challenges, Techniques and Directions in Building XSeek : an XML Search Engine », *Proceedings of the IEEE Data Engineering Bulletin*, p. 36-43, 2009.

Miller D., Leek T., Schwartz R., « A hidden markov model information retrieval system », *Special Interest Group on Information Retrieval SIGIR'01*, p. 214-221, 2001.

Murdock V., Lalmas M., « Workshop on aggregated search », *Special Interest Group on Information Retrieval SIGIR'08*, p. 80-83, 2008.

Naffakhi N., Boughanem M., Faiz R., « Réseau bayésien pour un modèle de recherche d'information agrégée dans des documents structurés », *INFormatique des ORganisations et Systèmes d'Information et de Décision INFORSID'2010*, p. 111-126, 2010.

Pearl J., *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

Piwowarski B., Gallinari P., « A Bayesian ramework for XML information retrieval : searching and leraning with the INEX collection », *Information Retrieval*, p. 655-681, 2005.

Polyzotis N., « XCluster Synopses for Structured XML Content », *In Proceedings of the 22nd International Conference on Data Engineering*, p. 406-507, 2006.

Radev D., Otterbacher J., Winkel A., Blair-Goldensohn S., « NewsInEssence : summarizing online news topics », *Communications of the Association of Computing Machinery*, p. 95-98, 2005.

Radlinski F., Dumais S., « Improving personalized web search using result diversification », *Special Interest Group on Information Retrieval SIGIR'06*, p. 691-692, 2006.

Sauvagnat K., Modèle flexible pour la recherhce d'information dans des corpus de documents semi-structurés, Thèse de doctorat, Université Paul Sabatier, Toulouse 3, 2005.

Spink A., Jansen B. J., Wolfram D., Saracevic T., « From E-Sex to E-Commerce : Web Search Changes », *IEEE Computer Science*, vol. 35, p. 107-109, 2002.

Stokes N., Carthy J., « Combining semantic and syntactic document classifiers to improve first story detection », *Special Interest Group on Information Retrieval SIGIR'01*, p. 224-225, 2001.

Sushmita S., Lalmas M., « Using digest pages to increase user result space : Preliminary designs », *Special Interest Group on Information Retrieval 2008 Workshop on Aggregated Search*, 2008.

Torjmen M., Pinel-Sauvagnat K., Boughanem M., « XML multimedia Retrieval : From relevant textual information to relevant multimedia fragments », *31th European Conference on Information Retrieval*, p. 150-161, 2009.

Turtle H., Croft W., « Inference networks for document retrieval », *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 1-24, 1990.

Yager R., Larsen H., « Retrieving information by fuzzification of queries », *Journal of Intelligent Information Systems*, p. 106-119, 1993.

Zhai C., Lafferty J., « A study of smoothing methods for language models applied to adhoc information retrieval », *in Proc of the 24th Annual Int ACM Special Interest Group on Information Retrieval SIGIR Conference*, p. 334-342, 2001.

Zhang Y., Callan J., Minka T., « Novelty and Redundancy Detection in Adaptive Filtering », *Special Interest Group on Information Retrieval SIGIR'02*, p. 81-88, 2002.