Recherche d'Information Agrégée dans des documents XML basée sur les Réseaux Bayésiens

Najeh Naffakhi[*,**], Mohand Boughanem[*], Rim Faiz[***]

[*]IRIT, Université Paul Sabatier
118, route de Narbonne, 31062 Toulouse Cedex 9
naffakhi@irit.fr
bougha@irit.fr
http://www.irit.fr
[**]LARODEC, Université de Tunis, ISG
41, rue de la liberté, cité Bouchoucha, 2000 Le Bardo, Tunis
najeh.naffakhi@isg.rnu.tn
http://www.isg.rnu.tn/larodec
[***]LARODEC, Université de Carthage, IHEC
2016 Carthage Présidence, Tunis
rim.faiz@ihec.rnu.tn
http://www.isg.rnu.tn/larodec

Résumé. Dans cet article, nous nous intéressons à la recherche agrégée dans des documents XML. Pour cela, 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 trouver des éléments. Ainsi, au lieu de récupérer une liste des éléments qui sont susceptibles de répondre à la requête, notre objectif est d'agréger dans un agrégat des éléments pertinents, non-redondants et complémentaires. Nous avons évalué notre approche dans le cadre de la compagne d'évaluation INEX 2009 et avons présenté quelques résultats expérimentaux mettant en évidence l'impact de l'agrégation de tels éléments.

1 Introduction

Typiquement, un système de Recherche d'Information (RI) retourne une liste ordonnée des documents en réponse à une requête. Ensuite, l'utilisateur doit les examiner linéairement. Un problème principal dans la RI structurée est comment sélectionner l'unité d'information qui répond le mieux à une requête composée de mots clés (CO : Content Only) (Kamps et al., 2003) (Fuhr et al., 2004). La plupart des approches en RI structurée (Sigurbjornsson et al., 2003), (Ogilvie et Callan, 2003), (Lalmas, 1997), (Lalmas et Vannoorenberghe, 2004), (Piwowarski et al., 2002) considèrent que les unités retournées sont sous forme d'une liste d'éléments disjoints. Nous supposons que cette unité pertinente n'est pas nécessairement des éléments adjascents ou un document, elle pourrait aussi être une agrégation d'éléments de ce document. Afin

de remédier à ce problème, la RI agrégée est l'une des techniques qui retourne des résultats de domaines variés (web, image, vidéo, news, etc.) et les présente ensemble dans une interface unique. Exemple des moteurs qui utilisent les techniques de la RI agrégée, nous trouvons Google's Universal Search [1], Yahoo ! alpha [2], Ask's X [3] et Microsoft's Live [4]etc. Les utilisateurs ont accès ensuite à différents types de résultats dans une seule interface. Ceci peut être bénéfique pour certaines requêtes, de type par exemple "voyage à Londres", ce qui peut retourner des cartes, des blogs, météo, etc.

Soit par exemple, un document XML qui a la structure suivante : *(Article(Title)(Chapter1 (Section1)(Section2))(Chapter2(...)))*. Si l'unité d'information pertinente est composée de *Title* et *Section1*, la majorité des systèmes de RI structurée retournent le document en entier. Dans notre modèle, nous considérons que l'unité d'information retournée est l'agrégat (ensemble d'éléments) formé des deux éléments *Title* et *Section1*.

Cet article est structuré de la manière suivante. La section 2 dresse un état de l'art des différents travaux relatifs à la RI dans des documents XML. 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. Nous conclurons en faisant le point sur notre travail et mentionnerons quelques perspectives.

2 Etat de l'art

L'objectif de la RI agrégée est de rassembler des informations à partir diverses sources pour construire des réponses, y compris les informations pertinentes à la requête.

Le problème de l'agrégation d'éléments d'une collection de documents XML n'est pas abordé dans la littérature. En effet, les approches proposées sont limitées à des documents Web (Clarke et al., 2008), (Agrawal et al., 2009). Cependant, peu de systèmes de RI commencent à agréger les résultats d'une requête sur les documents XML comme résumés. Par exemple, eXtract (Huang et al., 2008) est un système de RI qui génère des résultats sous forme de fragments XML. Un fragment XML est qualifié comme résultat que s'il répond à quatre caractéristiques : Autonome (compréhensif par l'utilisateur), distinct (différent des autres fragments), représentatif (des thèmes de la requête) et succinct. XCLUSTERs (Polyzotis, 2006) est un modèle de représentation de résumés XML. Il regroupe quelques éléments XML et utilise un petit espace pour stocker les données. L'objectif est de fournir des extraits afin que les utilisateurs peuvent facilement évaluer la pertinence des résultats de la recherche.

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 des 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 liens entre les éléments du corpus de documents XML et leurs contenus. Quant à la théorie des probabilités, elle permet d'estimer le fait qu'un terme est probablement pertinent vis-à-vis d'un

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

élément et 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 différencies clairement des cadres utilisés dans les approches précédentes.

3 Modèle de RI agrégée basé sur les réseaux bayésiens

Le modèle que nous proposons est représenté par un Réseau Bayésien (RB) (Piwowarski et al., 2002) 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.

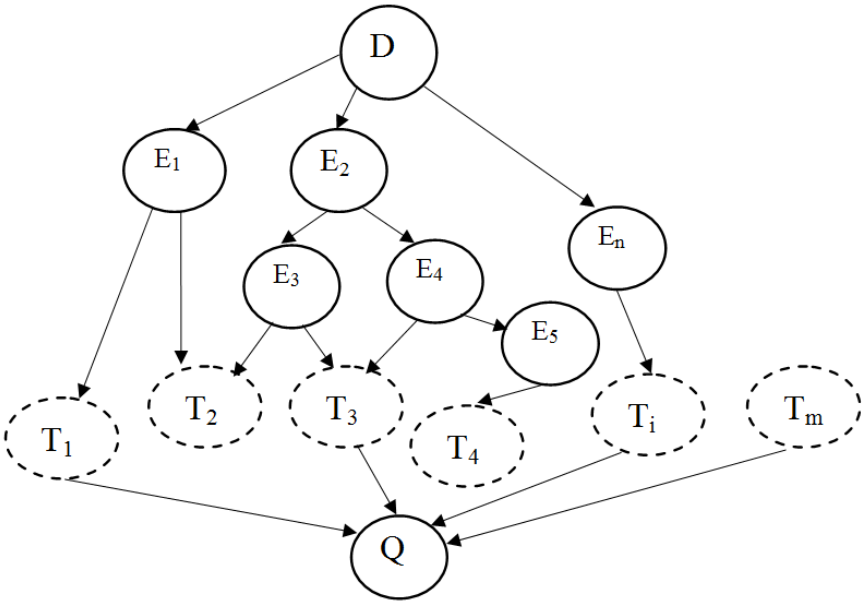

FIG. 1 – *Architecture simplifiée du modèle proposé par document.*

3.1 Architecture du modèle bayésien proposé

Le nœud D représente un document de la collection C. Chaque nœud D représente une variable aléatoire binaire $D = \{d, \neg d\}$. L'instanciation $D = d$ signifie qu'un document est choisi. Nous nous intéressons qu'au cas où le document $D = d$ est activé, et nous le notons d.

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

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

Le domaine de la requête est $dom(Q)$=$\{q, \neg q\}$. Nous ne sommes intéressés que par le cas où la requête introduit de l'information à travers le RB, c'est-à-dire lorsque le nœud est instancié positivement, 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 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. Chaque terme t_i, $t_i \in T$ 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. $\theta_i = \{e_1, e_3, e_5\}$ est un exemple d'une configuration déduit à partir de la figure 1. Une configuration donnée est considérée comme un résultat de la recherche.

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 nœud requête vers le nœud document ; puis nous calculons pour chaque configuration potentielle sa valeur de pertinence et complémentarité. A l'issue du processus de propagation, chaque configuration aura une valeur. La configuration retenue est celle qui présente la plus grande valeur. Cette configuration représentative d'un document forme un agrégat. Cet agrégat est un résultat de recherche.

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 son résultat de recherche θ_i dans un document d est donnée par :

$$P(Q, \theta_i, d) = \sum_{\overrightarrow{T(Q)}} P(Q|T(Q)) \times P(T(Q)|\theta_i) \times P(\theta_i|d) \quad (1)$$

$\overrightarrow{T(Q)}$ représente toutes les configurations possibles de $T(Q)$. Afin de simplifier notre modèle, nous considérons uniquement la configuration qui contient tous les termes de la requête. La formule (1) se simplifie par :

$$P(Q, \theta_i, d) = P(Q|T(Q)) \times P(T(Q)|\theta_i) \times P(\theta_i|d) \tag{2}$$

Les différents facteurs de probabilité dans la formule (2) sont détaillés comme suit :

3.2.1 Estimation de $P(Q|T(Q))$

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 Croft, 1990), (Boughanem et al., 2009).

$$\begin{aligned} P(Q|T(Q)) &= P(Q|T_1, ..., T_m) \\ &= \prod_{T_k \in T(Q)} P(Q|T_k = t_k) \\ &= \prod_{t_k \in T(Q)} P(Q|t_k) \end{aligned} \tag{3}$$

L'idée majeure est que pour une instantiation $T(Q)$, la probabilité condiotionnelle $P(Q|T(Q))$ est estimée par une fonction d'agrégation de la propabilité maximale (maximum likelihood) $P(Q|t_k)$ avec t_k est une instance de T_k dans $T(Q)$. Chaque $P(Q|t_k)$ représente le poids de conformité de t_k d'une instantiation de T_k dans une requête Q.

Nous supposons que les termes sont indépendants. En effet, les modèles basés sur les RB existants supposent l'indépendance entre les termes pour faciliter les calculs (i. temps de construction de la structure de dépendance entre les termes, et ii. temps de calcul lors de la propagation de l'information), mais 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.

Quand l'utilisateur ne donne aucune information sur les opérateurs d'agrégation, la seule évidence que nous pourrrons utilisés est l'importance des termes dans la collection.
$P(Q|T(Q))$ peut être estimé par :

$$P(Q|t_k) = \begin{cases} 1 & if \quad \forall t_k \in T(Q), \\ 0 & otherwise. \end{cases} \tag{4}$$

La propabilité que la requête Q soit inférée par les termes T_k est égale à 1 signifie qur le terme t_k dans $T(Q)$ est instancié dans la requête. Le processus d'évaluation génére tous les éléments contenant tous les termes de la requête.

3.2.2 Pertinence

Le deuxième facteur de la formule (2) estime la probabilité qu'une configuration réponde aux termes de la requête :

$$\begin{aligned} P(T(Q)|\theta_i) &= P(T_1, ..., T_m|\theta_i) \\ &= \prod_{T_k \in T(Q)} P(T_k = t_k|\theta_i) \\ &= \prod_{t_k \in T(Q)} P(t_k|\theta_i) \end{aligned} \quad (5)$$

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 $P(t_k|\theta_i)$. En fait, nous avons besoin de cette quantité pour déterminer la pertinence de cette configuration étant donnée une requête. Cette quantité est estimée par : 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 indépendents. La probabilité $P(t_k|\theta_i)$ peut être estimée en utilisant une estimation du maximum de vraisemblance qui se calcule par la fréquence du terme t_i dans θ_i. Ceci correspond au premier facteur de la formule (6). Afin d'éviter le problème des fréquences nulles des quelques termes (Quand un terme ne figure pas dans une configuration θ_i) , il faut ajouter la fréquence du terme dans la collection avec celle calculée avec le document (premier facteur de la formule (6)). Ceci correspond au deuxième facteur de la formule (6). La formule (6) correspond en fait à une technique de lissage de type Dirichlet (Zhai, 2008) mais appliqué à chaque élément XML.

$$\begin{aligned} P(t_k|\theta_i) &= (1-\lambda_t)\frac{tf(t_k|\theta_i)}{\sum_{\forall t} tf(t,d)} + \lambda_t \frac{df(t_k)}{\sum_{\forall t} df(t)} \\ &= (1-\lambda_t)\frac{\sum_{\forall e_j \in \theta_i} tf(t_k, e_j)}{\sum_{\forall t} tf(t,d)} + \lambda_t \frac{df(t_k)}{\sum_{\forall t} df(t)} \end{aligned} \quad (6)$$

avec :

- $tf(t_k|\theta_i)$ est la fréquence de t_k dans une configuration θ_i.
- $tf(t,d)$ est la fréquence du terme t dans le document d.
- $df(t)$ est le nombre des documents ou le terme t apparaît.
- $tf(t_k, e_j)$ est la fréquence du terme t dans l'élément e_j.
- $\lambda_t = \frac{\mu}{|d|+\mu}$. $\lambda_t \in [0,1]$ est un facteur de lissage.
- μ est une constante avec μ=0.5.
- $|d|$ est la taille de d.

3.2.3 Redondance

Dans chaque configuration, nous sommes intéressés à l'agrégation d'éléments qui ne véhiculent pas pas la même information. La redondance est traitée dans notre modèle au niveau structurel : Hypothèse **H1**.

– **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.

3.2.4 Complémentarité

Le troisième facteur de la formule (2) $P(\theta_i|d)$, 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 d'écrire :

$$P(\theta_i|d) = \prod_{e_j \in \theta_i} P(e_j|d) \tag{7}$$

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. Les éléments loin du nœud racine du document d 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 $dist(d, e_j)$ and $dist(d, deepdown(e_j))$, 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 distance entre le nœud racine d et le plus profond élément muni du nœud e_j noted $e_{deepdown}$. 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 :

$$P(e_j|d) = \frac{dist(d, e_j)}{dist(d, deepdown(e_j))} \tag{8}$$

La formule (8) 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 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 :

$$P(e_2|d) = \frac{dist(d, e_2)}{dist(d, e_5)} = \frac{1}{3} \tag{9}$$

$$P(e_4|d) = \frac{dist(d, e_4)}{dist(d, e_5)} = \frac{2}{3} \tag{10}$$

Finalement, la probabilité jointe de la formule (2) se simplifie en :

$$P(Q, \theta_i, d) = \prod_{t_k \in T(Q)} P(Q|t_k) \times \prod_{t_k \in (T(Q))} P(t_k|\theta_i) \times \prod_{e_j \in \theta_i} P(e_j|d) \qquad (11)$$

En fait, nous construisons des configurations θ par document. Dans chaque document et pour chaque configuration θ_i, nous calculons un score défini par la formule 11. La configuration qui présente le score le plus élevé sera qualifiée comme agrégat, réponse à la requête Q.

4 Expérimentations et résultats

Dans le but de valider notre approche, nous décrivons dans cette section le corpus des documents INEX'2009 *(INitiative for the Evaluation of XML Retrieval)*, la catégorie de requête utilisée, la tâche d'évaluation ainsi que les résultats des expérimentations.

4.1 Corpus

Nous nous appuyons pour l'évaluation des performances sur le corpus de test fournie dans le cadre de la compagne d'évaluation INEX 2009. Cette collection est composée d'articles Wikipédia en anglais. Ce corpus comporte 2 666 190 articles Wikipédia ayant une taille totale 50,7 Go. Dans ce corpus, il y a 101 917 424 éléments XML.

4.2 Requêtes

Les requêtes (ou Topics) sont crées par les différents participants et représentatives des demandes de l'utilisateur. Pour chaque Topic, différents champs permettent d'expliciter le besoin de l'utilisateur : le champs *Title* donne la définition simplifiée de la requête, le champ *Keywords* contient un ensemble de mots clés qui permettent l'exploration du corpus, et les champs Description et Narrative, explicités en langage naturel, indiquent les intentions de l'auteur. Les topics se divisent en deux catégories principales : CO (Content Only) et CAS (Content And Structure). Dans notre travail, nous nous intéressons aux requêtes CO : ce sont des requêtes composées de simples mots clés.

4.3 Tâche d'évaluation

En absence de cadre approprié pour l'évaluation de la valeur des agrégats, nous avons adopté une stratégie d'évaluation basée sur des jugements des utilisateurs (appelé en anglais, User Studies). En effet, dans notre modèle nous cherchons à juger chaque ensemble d'éléments qui se complètent (chaque agrégat) et non plus d'éléments pris séparément (une liste). Si notre objectif était de retourner une liste des éléments triés, nous pourrons comparer nos résultats à ceux enregistrés par les participants à la compagne d'évaluation INEX2009 et nous utiliserons les mesures nxCG et MAep pour faire l'évaluation. Mais plutôt, notre objectif était d'évaluer l'agrégat entier (un tout pertinent).

C'est pour cela que nous avons pris un ensemble de 20 requêtes CO du corpus INEX'2009. Ces requêtes sont numérotées 2009n avec n : 001-006, 010-015, 020, 023, 026, 028, 029, 033, 035, 036. Pour les participants, nous avons pris 23 utilisateurs des doctorants et étudiants de Master en RI. La tâche d'évaluation est la suivante : chaque requête est soumise au système. Les résultats de la recherche sont affichés sous forme d'une liste d'agrégats. Nous nous limitons aux cinq premiers agrégats par requête. Les résultats de chaque requête sont évalués par 15 utilisateurs. Un utilisateur juge chaque agrégat selon trois dimensions : Pertinence, redondance et complémentarité.

4.4 Expérimentation 1 : Pertinence des agrégats

Dans cette expérimentation, nous voulons savoir, à travers l'analyse du critère de pertinence, si les agrégats retournés par le système sont pertinents ? La figure 2 présente le pourcentage d'agrégats pertinents, non-pertinents et partiellement pertinents par requête. Les résultats montrent que seuls quelques pourcents (13%, la partie verte de la barre) des agrégats ne sont pas pertinents. Les autres sont soit pertinente (29%, la partie bleue de la barre) ou partiellement pertinente (58%, la partie rouge de la barre). Un agrégat est partiellement pertinent s'il contient quelques elements pertinents. Par exemple, les réultats de la requête n=012 *vitiligo pigment disorder cause treatment* sont partiellement pertinents ou non pertinents. Cette requête est une requête générique nécessite une reformulation afin d'améliorer l'interprétation des résultats.

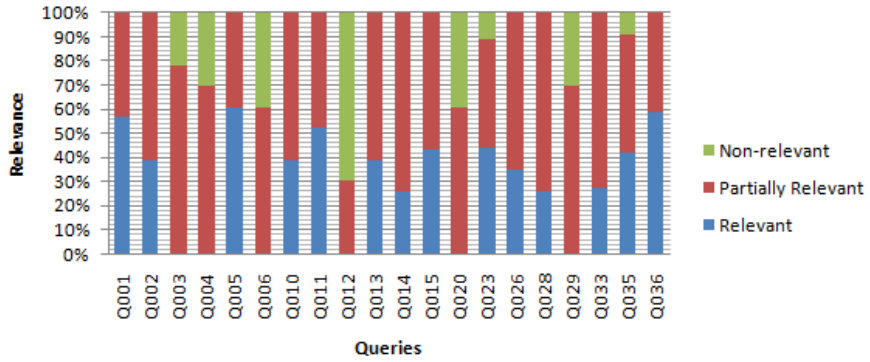

FIG. 2 – *Distrubtion des résultats de pertinence de l'agrégat par rapport à la requête.*

4.5 Expérimentation 2 : Redondance

Cette expérimentation permet d'évaluer la redondance entre les éléments d'un agrégat par requête. Plus spécifiquement, nous avons fourni deux degrés de redondance, *un peu redondant* et *non-redondant* ; les utilisateurs peuvent les appliquer en fonction de leurs attentes sur la façon dont un bon système devrait se comporter. Pour chaque requête, chaque utilisateur est invité à évaluer la redondance entre les éléments du premier agrégat retourné par le système pour chaque requête. Si un utilisateur juges qu'un élément d'un agrégat donné ne contient pas

de nouvelles informations, il le considère comme *peu redondant*. En revanche, si un nouvel élément d'un agrégat présente quelques informations nouvelles, il sera considérer comme *non-redondant*. La figure 3 montre les résultats des jugements.

Nous trouvons que des agrégats contiennent des éléments qui ne véhiculent pas les mêmes informations, par exemple les requêtes n=035 et n=012. Cette sorte de RI agrégée dans les documents XML est très utile car elle renseigne l'utilisateur sur la diversité des informations du corpus en rapport avec son besoin en information.

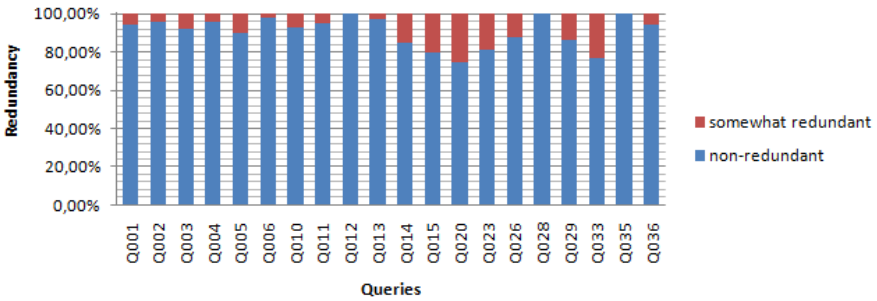

FIG. 3 – *Répartition des résultats par requête par rapport au critère de Redondance.*

4.6 Expérimentation 3 : Complémentarité

Dans cette expérimentation, nous voulons savoir si les éléments d'un agrégat sont complémentaires. Pour chaque requête, chaque utilisateur est invité à évaluer la complémentarité entre les éléments de chaque sommet un agrégat. La distribution des jugements du critère de complémentarité entre les 20 requêtes est montré dans la figure 4. Nous notons que pour la plupart des requêtes, les jugements sont positifs. Cela prouve la capacité de notre modèle d'agréger d'éléments qui contiennent des informations pertinentes et supplémentaires. Pour les requêtes n={004, 005, 012, 028, 033, 035}, les éléments des agrégats ne sont pas complémentaires par rapport à la requête. En effet, pour le Q004 requête *mean average precision reciprocal rank references precision recall proceedings journal*, nous trouvons des thèmes différents sont exprimés comme la précision, la réciprocité, le rappel qui sont très génériques. C'est pourquoi les éléments d'un agrégat ne peuvent pas être sémantiquement similaire à l'égard des besoins de l'utilisateur et donc pas complémentaires. De même pour les autres requêtes.

5 Conclusions

Ce papier présente une nouvelle approche pour la RI agrégée dans des documents XML et qui repose sur les réseaux bayésien. Notre modèle montre comment assembler dans un agrégat des éléments XML pertinents, non-redondants et complémentaires afin d'améliorer les résultats de recherche. En effet, les agrégats orientent l'utilisateur plus rapidement pour

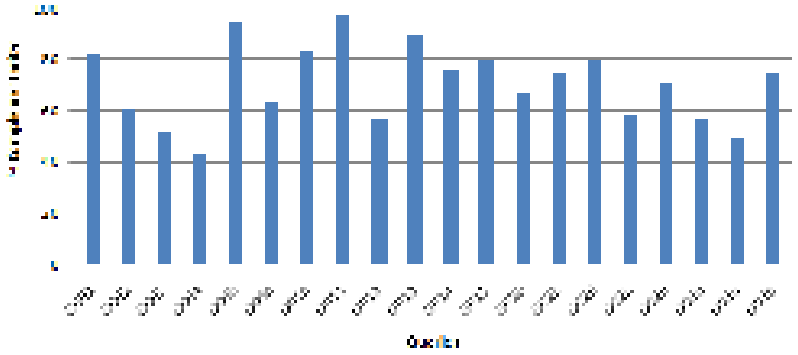

FIG. 4 – *Distribution des jugements par requête par rapport au critère de Complementarité.*

sélectionner les éléments pertinents, non redondants et complémentaires et donnent un aperçu sur les informations disponibles dans le corpus par rapport à la requête.

Ainsi, il semble très intéressant de penser à dévélopper des outils d'évaluation qui évaluent les systèmes de RI agrégée afin de comparer leurs performances.

Références

Agrawal, R., S. Gollapudi, et A. Halverson (2009). Diversifying search results. In *ACM Int. Conference on WSDM*, pp. 5–14.

Boughanem, M., A. Brini, et D. Dubois (2009). Possibilistic networks for information retrieval. In *IJAR*, Volume 50, pp. 957–968.

Clarke, C., M. Kolla, G. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, et M. I. (2008). Novelty and diversity in information retrieval evaluation. In *SIGIR'08*, pp. 659–664.

Fuhr, N., M. Lalmas, S. Malik, et Z. Szlavik (2004). Xml information retrieval: Inex 2004. In *Advances in XML Information Rretreival and evaluation*, pp. 409–410.

Huang, Y., Z. Liu, et Y. Chen (2008). Query biased snippet generation in xml search. In *SIGMOD'08*, pp. 315–326.

Kamps, J., M. Marx, M. De Rijke, et B. Sigurbjörnsson (2003). Xml retrieval: What to retrieve? In *Conference on Research and Development in Information Retrieval*, pp. 409–410.

Lalmas, M. (1997). Dempster-shaferŠs theory of evidence applied to structured documents: Modelling uncertainty. In *Proceedings of the 20th Annual International ACM SIGIR*, pp. 110–118.

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

Ogilvie, P. et J. Callan (2003). Using language models for flat text queries in xml retrieval. In *Proceedings of INEX 2003 Workshop*, pp. 12–18.

Piwowarski, B., G.-E. Faure, et P. Gallinari (2002). Bayesian networks and inex. In *In Proceedings of the First Annual Workshop of the Initiative for the Evaluation of XML retreival*, pp. 149–153.

Polyzotis, N. (2006). Xcluster synopses for structured xml content. In *ICDE*, pp. 63.

Sigurbjornsson, B., J. Kamps, et M. de Rijke (2003). An element-based approach to xml retrieval. In *Proceedings of INEX 2003 Workshop*, pp. 80–83.

Turtle, H. et W. Croft (1990). Inference networks for document retrieval. In *Proc. of the ACM-SIGIR Conference*, pp. 1–24.

Zhai, C. (2008). Statistical language models for information retreival a critical review. In *the Found. Trends Information Retreival*, pp. 137–213.

Summary

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.

Query-terms and terms-elements relations are modeled through probability. In this model, the user's query starts a propagation process to recover the XML elements. Thus, instead of retrieving a whole document or a list of disjoint elements that are likely to answer partially the query, we attempt to built a virtual document that aggregates a set of elements, that are relevant, non-redundant and complementary. We evaluated our approach using the INEX 2009 collection and presented some empirical results for evaluating the impact of the aggregation approach.