République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'Informatique

Mémoire

de fin d'études

En vue de l'obtention du diplôme Master 2 en informatique

Spécialité : Conduite de projet informatique

Thème

Proposition d'une approche basée sur les réseaux bayésiens pour la recherche d'information dans les documents XML

Proposé et dirigé par :

M^me^ : Fellag.S

Réalisé par :

M^elle^ : Djebbar Sonia
M^me^: Lamrous Fadhila

Promotion : 2012-2013

Remerciements

Louange à notre créateur qui nous a incité à acquérir le savoir et nous a donné la volonté et le courage pour y arriver. C'est à lui que nous adressons toute notre gratitude en premier lieu.

Nous exprimons notre profonde gratitude à notre chère promotrice, Madame ***F. Fellag*** *pour avoir dirigé ce travail, pour la documentation qu'elle nous a fourni, pour ses critiques, ses conseils et ses orientations durant toute la période de notre travail.*

Nous remercions aussi profondément les membres de jury pour nous avoir fait l'honneur de juger notre travail.

Enfin, nous tenons à remercier nos familles respectives ainsi que toute personne ayant contribué de près ou de loin à l'aboutissement de ce travail…

SONIA & FADILA

Dédicaces

Je dédie ce modeste travail à :

- *A la mémoire de mon grand père ;*
- *Mes très chers parents;*
- *A mon frère et ma sœur (Amar, Sandra);*
- *A toutes ma famille mes oncles et tantes ;*
- *A tes mes amies : souad,cherifa,karima,djamila ..etc*
- *A tous (tes) mes collègues et amis(es) du travail service paie UMMTO (Mme Chabour ,sabrina ,fatiha ,souhila, hamid aissa ,nassim..ets);*
- *A toutes les personnes qui pensent à moi de près ou de loin;*
- *A mon binôme Fadhila et à toute sa famille;*
- *A tous les étudiants de la promotion 2012/2013.*

SONIA



Dédicaces

Je dédie ce modeste travail à :

- *A la mémoire de mon père et mon frére Hakim, que dieu leur offre le paradis eternel.*
- *A Ma très chère et tendre mère, Merci pour ton amour, ta sagesse, ta passience, ta présence et pour ton soutien continu. Bonne santé et longue vie, Amine.*
- *A ma moitié, mon mari sans qui je n'aurai pas pu contribuer à entamer ou même finaliser ce modeste travail.*
- *A mes princesses, mes filles à qui je souhaite santé, bonheur et réussite dans tous leurs projets d'études.*
- *A mes beaux parents à qui je souhaite longue vie pleine de santé.*
- *A mes mes frères et sœurs pour leur soutien et particulièrement ma jumelle pour son aide.*
- *A mes adorables neveux et nièces.*
- *A tous les beaux frères et belles sœurs.*

- *A ma binome Sonia ainsi qu'à sa famille.*

- *A mes amies et collègues de travail*

FADHILA

Table des figures

Table des figures

Sommaire :

Sommaire :

Chapitre II : Recherche d'Information Structurée (RIS)

Sommaire :

Chapitre III : les réseaux bayésien et la RI

Sommaire :

Chapitre IV : les réseaux bayésien dans la RIS

INTRODUCTION GENERALE

Le développement du document électronique et du Web ont vu émerger puis s'imposer des formats de données structurés, tels que le SGML (Standard Generalized Markup Language) et le XML (eXtensible Markup Language), permettant de représenter conjointement l'information textuelle et l'information de structure d'un document. La connaissance de la structure des documents est une ressource additionnelle qui devrait être exploitée pendant la recherche d'information afin de mieux exprimer un besoin d'information.

Dans le contexte de la RI, la question majeure soulevée par ce type de document concerne la manière de manipuler efficacement la structure et le contenu du document 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).

Le défi à relever est alors d'arriver à identifier automatiquement les parties du document XML, répondant de manière exacte à la requête de l'utilisateur. Plusieurs modèles ont été proposés dans ce sens dont les modèles probabilistes. Nous nous sommes intéressées plus particulièrement aux modèles bayésiens. L'objet de notre travail est de faire une synthèse des différents travaux effectués dans ce domaine puis d'apporter une contribution en proposant une approche de recherche d'informations dans les documents XML en utilisant les réseaux bayésiens.

Pour ce faire, nous avons organisé notre mémoire en quatre chapitres, le premier aborde les concepts de base de la recherche d'information tout en présentant les techniques d'indexation existantes, les différents modèles de RI, le processus de reformulation de la requête, ainsi, que les méthodes d'évaluation des performances d'un SRI

Le deuxième chapitre présente un état de l'art sur la Recherche d'Information Structurée tout en définissant les notions générales du Language XML, ainsi les différentes problématiques soulevées par la recherche d'information structuré et les différentes solutions proposée dans la littérature, enfin nous présenterons les différents

modèles de recherche dans les documents structuré, ainsi que les différentes mesures utilisé pour l'évaluation des SRIS.

Le troisième chapitre 3 est consacré à la définition de quelques concepts de base des réseaux bayésiens et présente les différents travaux de recherche d'information basés sur les réseaux bayésiens dans la RI. En guise de conclusion, nous Y avons effectué une synthèse

Le dernier chapitre présente et synthétise les différents modèles de RIS basés sur les réseaux bayésiens, et présente notre contribution à savoir proposition d'une approche basé sur les réseaux bayésien dans les documents XML.

I- Recherche d'Information (RI) :

I-1-Introduction :

La recherche d'information (RI) est une branche de l'informatique qui s'intéresse à la collecte, au stockage, à l'organisation, la recherche et la sélection d'information répondant aux besoins des utilisateurs exprimés sous forme de requête. Ce besoin est interprété par un système de recherche d'information, qui permet de retrouver des documents susceptibles d'être pertinents pour la requête suivant un processus de sélection, à partir d'une ou plusieurs collections de documents.
Même si les premiers systèmes automatisés de RI ont vu le jour dans les années 60, c'est sans conteste l'avènement de l'Internet qui les a complètement développés et réactualisés.

I-2-Définition de la recherche d'information RI :

"La recherche d'information (RI) est un ensemble de méthodes et procédures ayant pour objet la recherche, la collecte et l'exploitation d'informations en réponse à une question sur un sujet précis".
La RI selon [Hernandez, 06] est une activité dont la finalité est de localiser et de délivrer des granules documentaires à un utilisateur en fonction de son besoin en informations.

Ces définitions partagent l'idée que la RI a pour objet d'extraire d'un document ou d'un ensemble de documents les informations pertinentes qui reflètent un besoin d'information et cela grâce au système de recherche d'information.

I-3-Définition d'un SRI (Système de Recherche d'information):

Un **SRI** est un système qui permet de retrouver les **documents pertinents** à une **requête** utilisateur, à partir d'une base de documents volumineuse [JYN, 04].
Dans cette définition, on distingue trois notions clés: **documents, base de documents, requête, pertinence :**

I-3-1Document:

Un document peut être un texte, un morceau de texte, une page web, une image, une bande vidéo, etc. On appelle ainsi document, toute unité qui peut constituer une réponse à une requête utilisateur. Dans la RI traditionnelle l'unité d'information utilisée et recherchée lors du processus de recherche est le document.

I-3-2 Base de documents (Collection de documents ou corpus) :

Un corpus de documents est un ensemble de granules documentaires qui peuvent être des documents entiers ou bien des parties de documents.
Le contenu de la base documentaire diffère d'une base documentaire à une autre selon le domaine d'application considéré.

On distingue principalement deux types de bases documentaires **: les référothéques** et **les bibliothèques :**

a- Les référothéques:

Une référothéque est constituée d'un ensemble d'enregistrements faisant référence au document dans lequel se retrouve l'information intégrale. Une unité d'information est composée d'un résumé du texte intégral (*abstract*) et de données factuelles complémentant la description du document.

b- Les bibliothèques:

Ce sont des bases documentaires composées des textes intégraux de documents (*full texts*).

I-3-3Requête:

Une requête exprime le besoin en information d'un utilisateur. Ce besoin est l'expression mentale de ce que l'utilisateur recherche. Elle peut être exprimée selon différents langages :

- **Langage booléen:**
  La requête est construite en utilisant les opérateurs de la logique booléenne. C'est le cas des systèmes **LEXIS**, **STAIRS**, **PASCAL**, **BASIS PLUS**. Ce type d'interrogation est assez strict imposant une syntaxe difficilement accessible à un large public.
- **Langage naturel ou quasi-naturel:**

  Il est admis dans tous les cas de communication homme-machine, que le dialogue est d'autant plus aisé que le langage d'expression utilisateur est libre, non astreint au langage issu de l'environnement du système. L'interrogation en langage libre ou langage naturel, accentue l'engouement vers l'utilisation des SRI. Il apparaît cependant des difficultés de traitement des requêtes nécessitant la mise en œuvre de mécanismes d'indexation élaborés pour la traduction des requêtes en mots clés, sans perte de signification.
- **Langage graphique:**

  Dans ce cas, une interface d'aide à la formulation de requête est proposée à l'utilisateur. En effet, une vue d'ensemble de la base d'information et en particulier une vue des termes représentants le contenu sémantique des documents est donnée pour l'utilisateur pour lui faciliter la formulation de sa requête.

I 3-4-Notion de pertinence:

La notion de pertinence est très complexe. De façon générale, dans le document pertinent, l'utilisateur doit pouvoir trouver les informations dont il a besoin. C'est sur cette notion de pertinence que le système doit juger si un document doit être donné à l'utilisateur comme réponse.

Les travaux de recherche mettent en exergue deux types de pertinences :

- **La pertinence utilisateur** :
  Elle représente le point de vue de l'utilisateur. C'est la façon dont il évalue les documents retournés par le SRI en fonction de son besoin d'information (on parle de ses jugements de pertinence).

- **la pertinence système** :
  Elle est souvent traduite par un score calculé à partir des méthodes utilisées pour évaluer l'adéquation des documents et la requête.
  Contrairement à la pertinence système, la pertinence utilisateur est subjective car elle peut évoluer avec le temps et diffère d'un utilisateur à un autre.

Tout l'enjeu de la RI réside dans la mise en œuvre de mécanismes visant à rapprocher la pertinence système de la pertinence utilisateurs

I-4-Le processus en U :

Le processus de Recherche d'Information a pour but la mise en relation des informations disponibles d'une part, et les besoins de l'utilisateur d'autre part.
Ce processus, couramment appelé Processus en U de Recherche d'Information, est illustré par la figure suivante :

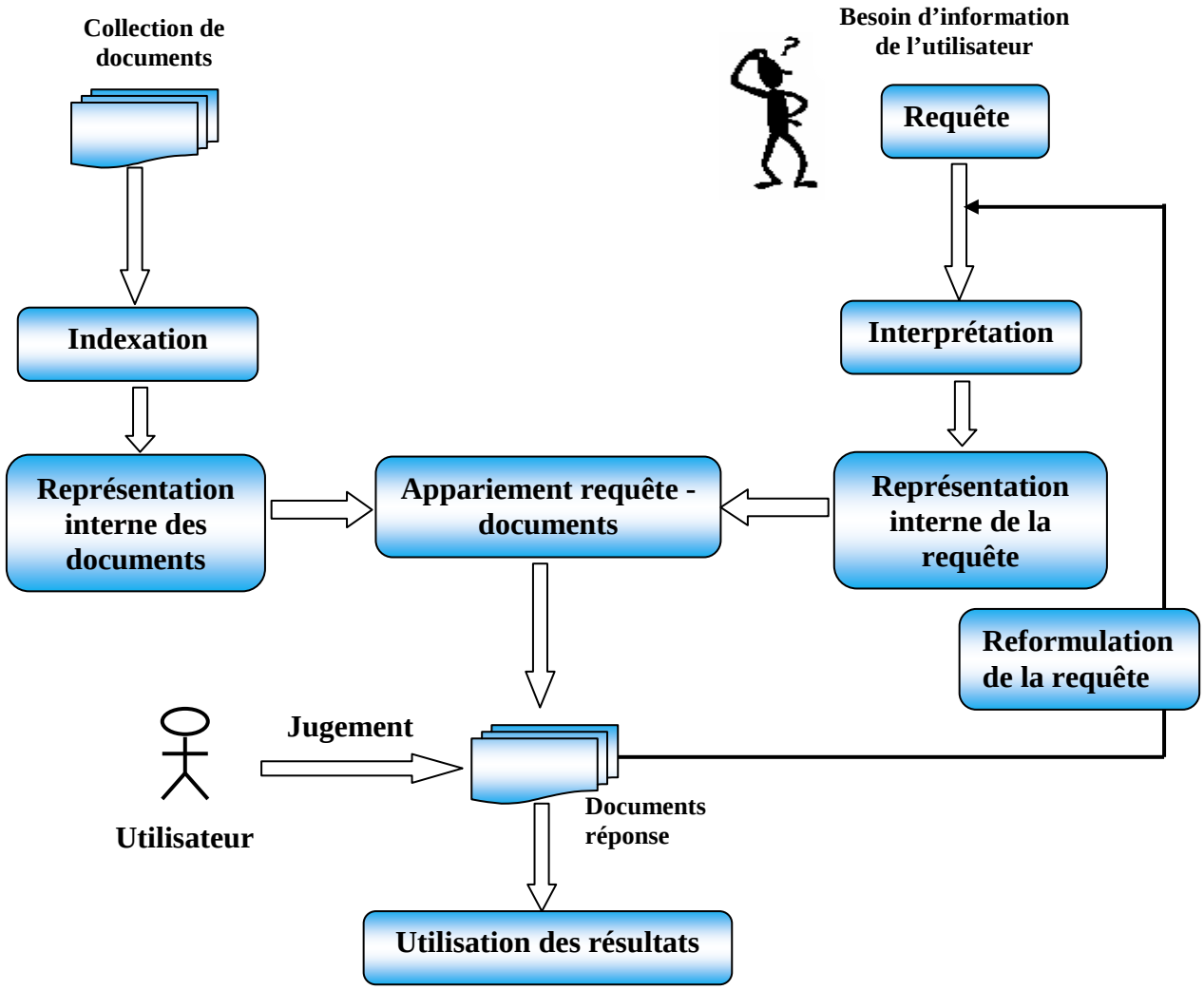


Figure I.1: Processus général de la RI

Le processus de RI a pour objectif de sélectionner les documents les plus proches du besoin en information de l'utilisateur décrit par une requête. Ces requêtes seront transformées lors du processus d'interprétation en une représentation compatible avec celle des documents issue du processus d'indexation. Ces deux représentations doivent être mises en correspondance pour pouvoir sélectionner les documents qui correspondent au mieux à la requête (pertinents).

Afin d'améliorer les résultats de la recherche, le système peut être doté d'un mécanisme de modification qui permet la reformulation de la requête. Cette amélioration est appelée bouclage de pertinence (ou « relevance feedback »).

I-4-1 L'indexation :

a- Définition:

De façon générale, « Indexer un document» « c'est lui attribuer une marque distinctive renseignant sur son contenu et permettant de le retrouver » [MAN, 02].

L'indexation est une étape très importante dans le processus de RI. Elle consiste à déterminer et à extraire les termes représentatifs du contenu d'un document afin d'en produire un ensemble de mots clés, appelés aussi ***descripteurs***. Ces derniers doivent être choisis de manière à pouvoir séparer lors d'une recherche, les documents pertinents à une requête des documents non pertinents.

b-Types d'indexation :

b-1 Indexation manuelle :

Dans l'indexation manuelle, chaque document est analysé par un spécialiste du domaine ou par un documentaliste.

- **Avantages :**

- Meilleure correspondance entre les documents et les termes choisis pour les représenter (dits termes d'indexation).
- Meilleure précision dans les documents que le système de RI retourne en réponses aux requêtes des utilisateurs.

- **Inconvénients :**

- L'effort intellectuel qu'elle exige (en temps et en nombres de personnes).
- Approche subjective car le choix des termes d'indexation dépend de l'indexeur et de ses connaissances du domaine.
- Méthode inapplicable aux corpus de textes volumineux.

b-2 Indexation semi-automatique (supervisée):

Ici un premier processus automatique permet d'extraire les termes du document. Cependant le choix final est laissé au spécialiste du domaine ou au documentaliste pour établir les relations entre les mots clés et choisir les termes significatifs.

b-3 Indexation automatique :

Elle est sans doute celle qui a été le plus étudiée en recherche d'information. Dans cette approche, chaque document est analysé à l'aide d'un processus entièrement automatisé. Elle permet de traiter les textes plus rapidement que l'indexation manuelle et elle est particulièrement adaptée aux corpus volumineux.

Dans le paragraphe suivant, nous nous proposons de décrire en détail, le processus d'indexation automatique des documents textuels.

b-3-1 Processus d'indexation automatique:

Le processus d'indexation automatique s'appui sur quatre étapes qui sont :

- L'extraction des termes d'index.
- La réduction des termes d'index.
- La pondération des mots d'index.
- Création des indexes.

a- L'extraction des termes d'index :

Le but de cette étape est d'identifier les termes représentatifs du contenu du document. Elle repose sur différents niveaux d'analyse linguistique du texte du document à indexer, dont:

- Analyse morphologique.
- Analyse lexicale.
- Analyse syntaxique.
- Analyse sémantique.

a-1 Analyse morphologique :

En analyse morphologique, l'ensemble des termes appartenant à un document sont repérés et les mots vides (articles, prépositions….) éliminés. Notons que l'ensemble des mots vides de la langue (des documents) est préalablement recensé dans un anti-dictionnaire (ou *stop-list)*.

a-2 Analyse lexicale :

En analyse lexicale les termes d'index sont normalisés en éliminant leurs variantes morphologiques (nombre, dérivations, ...) afin de réduire leur taille sans perte de signification. En effet, un mot donné peut avoir des formes différentes dans un texte dont le sens est le même ou très similaire, comme c'est le cas notamment pour les mots conjugués. En RI, il n'est pas utile de considérer la différence de forme entre ces mots. La normalisation des mots permet d'éliminer leurs différences non significatives et de réduire leur taille sans perte de signification. Ainsi, dans l'index ne sont conservées que les formes normalisées, ce qui offre un gain d'espace important. Deux principales approches de normalisation peuvent être utilisées qui sont :

- **La lemmatisation (ou racinisation) :**
  Il s'agit de ramener le mot à son radical. La recherche du radical est réalisée par élimination de suffixes par itération (en plusieurs passages) en utilisant un lexique contenant tous les suffixes possibles.
- **La troncature :**
  Elle consiste à tronquer le mot à partir d'un rang précis, afin d'obtenir son radical. Cette technique permet ainsi de réduire les variantes morphologiques des mots issus de la même racine.

a-3 Analyse syntaxique :

Lors de cette analyse, les groupes de mots (ou groupes composés) sont repérés comme unités descriptives d'un document afin d'augmenter la précision de réponse car un groupe de mots, est souvent plus précis que les mots qui le composent pris séparément. Par exemple, Le terme composé "accident de travail" est plus précis que "accident" et "travail" pris isolément.

a-4 Analyse sémantique :

Fondée sur le sens des mots (ou concepts) afin de résoudre le problème de polysémie _mots ou expressions ayant plusieurs sens_ (**exp :** mémoire = mémoire humaine, mémoire d'ordinateur, mémoire de fin d'étude), l'analyse sémantique se base souvent sur l'utilisation des réseaux sémantiques (cartographie représentant les différents sens d'un mot).

b- Réduction du langage d'indexation:

L'ensemble des termes descripteurs (ou termes d'indexation) issus de l'indexation constitue le langage d'indexation.

La réduction du langage d'indexation a pour but de trouver les meilleurs termes représentant le contenu d'un document en éliminant tous les mots non importants du langage d'indexation. La conjecture de **Luhn [**Luhn, 1958**]** à la base de cette réduction, considère que :

« L'importance d'un terme est généralement mesurée par sa fréquence d'apparition dans un document, et que les termes à fréquence très élevé (très fréquent) et les termes à faible fréquence (rares) ne permettent pas de différencier les documents du même corpus ».

De ce fait, deux seuils de fréquences sont généralement définis, un seuil de fréquence maximal ***seuil max*** et un seuil de fréquence minimal ***seuil min***, tels que seuls les termes entre ces deux seuils sont considérés comme représentatifs des contenus des documents.

c- Pondération des termes :

La pondération des termes d'indexation consiste à affecter à chaque terme d'un document, un poids mesurant son degré d'importance dans le document.

La conjecture de Luhn **[**Luhn, 1958**]** est la base sur laquelle se positionnent les SRI. La mesure de la pondération (par la fréquence d'occurrence) a été étendue de sorte à tenir compte de la spécificité du terme pour un document (par rapport aux autres documents de la collection) et de donner des meilleures performances. La formule dite ***tf*idf*** est née de cette extension.

- **Pondération en tf * idf :**

[ROB, 76] a défini la fonction de pondération des termes dans un document sous la forme de ***tf * idf*** qui permet de combiner la pondération locale (***tf***) et globale (***idf)*** d'un terme :

- ***tf* (*term frequency*) :** Mesure la fréquence d'occurrence du terme dans le document.
  Le ***tf*** est souvent exprimé selon l'une des formules suivantes :

***tf* :** utilisation brute.
ou
log (1+tf).

- ***idf* (*Inverse of Document Frequency*) :** Mesure l'importance d'un terme dans toute la collection.

  Cette mesure est exprimée selon l'une des formules suivantes :

$$idf = \log\left(\frac{N}{df}\right).$$

Ou

$$idf = \log\left(\frac{N-df}{df}\right).$$

Où ***df* :** est la proportion de documents contenant le terme.
N: nombre total de documents dans la collection.

La fonction de pondération de la forme ***tf * idf*** consiste à multiplier les deux mesures ***tf*** et ***idf***. Une formule largement utilisée est la suivante :

$$tf * idf = \log(1+tf) * \log\left(\frac{N}{df}\right)$$

La combinaison des deux mesures (***tf*** et ***idf***) donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de tailles homogènes. Cette mesure a eu en revanche un succès très limité dans les corpus de tailles très variables. Le problème posé est que les termes appartenant aux documents longs apparaissent très fréquemment et l'emportent en poids sur les termes appartenant à des documents moins longs. Les documents longs auront alors plus de chance d'être sélectionnés.

Une **normalisation** de la mesure du ***tf * idf*** par rapport à la longueur des documents a été proposée par **[SIN, 95]** comme suit :

$$tf * idf = 0.5 * \frac{tf * log\left(\frac{N-df+0.5}{df+0.5}\right)}{2 * \left(0.25 + 0.75 * \frac{dl}{\Delta l}\right) + tf}$$

Où :
***dl* :** La longueur du document en nombre de termes.
***Δl* :** La longueur moyenne des documents de la collection entière.

-Formule BM25 :
BM25 (« Best Matching » ou meilleur arrangement) est une formule de classement des documents en fonction de leurs pertinences. Développée en 1976 par Robertson et Spark Jones [Rob *et al.*, 1976], elle est généralement appelée « Okapi BM25 », référençant le SRI OKAPI, basé sur le modèle probabiliste, qui l'a implémenté en premier. A cause des bons résultats retournés par ce système, elle est devenue une formule de référence.

Description :
La fonction BM25 classe les documents en fonction des termes d'une requête fournie à l'interface d'interrogation d'un SRI, en calculant le poids associé à un terme dans un document et une approximation a été proposée pour cela dans le modèle Okapi :

$$Okapi\ BM25 = \frac{tf_i * (K_1 + 1)}{K_1 * \left((1-b) + b * \frac{dl}{avdl}\right) + tf_i} * w^1$$

Avec : *b* = constante ϵ [0,1] contrôle l'effet de longueur du document avec :
b = 1 : document long à termes répétitifs et
b = 0 : document long à termes distincts
dl est la longueur du document considéré et *avdl* = Δl longueur moyenne des documents de la collection.

w^1 est la formule du modèle probabiliste de base :

$$w_i = log \frac{p(t_i \in D/R) * p(t_i \notin D/NR)}{p(t_i \in D/NR) * p(t_i \notin D/R)}$$

D'après [Yuanhua Lv & ChengXiang Zhai, 2004], la fréquence des termes (TF) est le composant clé de la BM25, dont le poids d'un terme est contrôlé par le paramètre *K1* généralement constant.

La formule BM25 adaptée sélectionne le document *D* par similarité à la requête *Q* :

$$BM25 = \sum_{t \in q \cap d} \left(\frac{tf}{tf + k_1 \,.\, n_b} \,.\, log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \,.\, qtf\right)$$

Où :
tf : fréquence d'apparition du terme,
N : nombre totale de documents dans la collection,

dft : nombre de documents contenant le terme *t*,
qtf : fréquence d'apparition du terme *t* dans la requête,
*k*1 : paramètre d'influence de la fréquence des termes qui est ajusté à 1.2 par défaut,
nb : facteur de normalisation est calculé comme suit :

$$n_b = (1 - b) + b \cdot \frac{tl}{tl_{avg}}$$

Où :
tl : nombre de termes dans le document (longueur de document),
tlavg : nombre moyen de termes dans un document,
b = 0.75 ; l'augmentation de la valeur de *b* augmente la pénalisation des plus longs documents, avec une valeur de 1 étant la limite supérieure.

II-3-4-Création et organisation des indexes :

Les termes pondérés issus du processus d'indexation seront mémorisés dans une structure de stockage appelée index qui permet de sélectionner pour n'importe quel terme tous les documents où il apparait. Plusieurs solutions de stockage ont été proposées parmi lesquelles : les fichiers séquentiels indexés**,** les fichiers inverses (inverted files), les fichiers de signatures (signature files) et les tableaux de suffixes (suffix arrays). Nous présentons les plus utilisés :

- **Les fichiers séquentiels indexés :**

  Il s'agit d'un ensemble de trois fichiers reliés entre eux par des pointeurs:

  - Le premier contient les mots issus du processus d'indexation.
  - Le second contient une liste ordonnée de mots avec les références aux documents qui les contiennent.
  - Le troisième contient les positions des mots clés dans la phrase, le paragraphe, la section, le chapitre etc.

Exemple :

Cas d'un moteur de recherche référençant les mots par numéro de document, numéro de paragraphe, numéro de phrase et la position du mot dans la phrase.

Soit le document dont le numéro est **1** qui contient le texte suivant :

	1	2	3
Texte	Un index est une structure de stockage		

Soit le document dont le numéro est **2** qui contient le texte suivant :

	1	2	3	4
Texte	Les termes pondérés sont mémorisé dans un index			

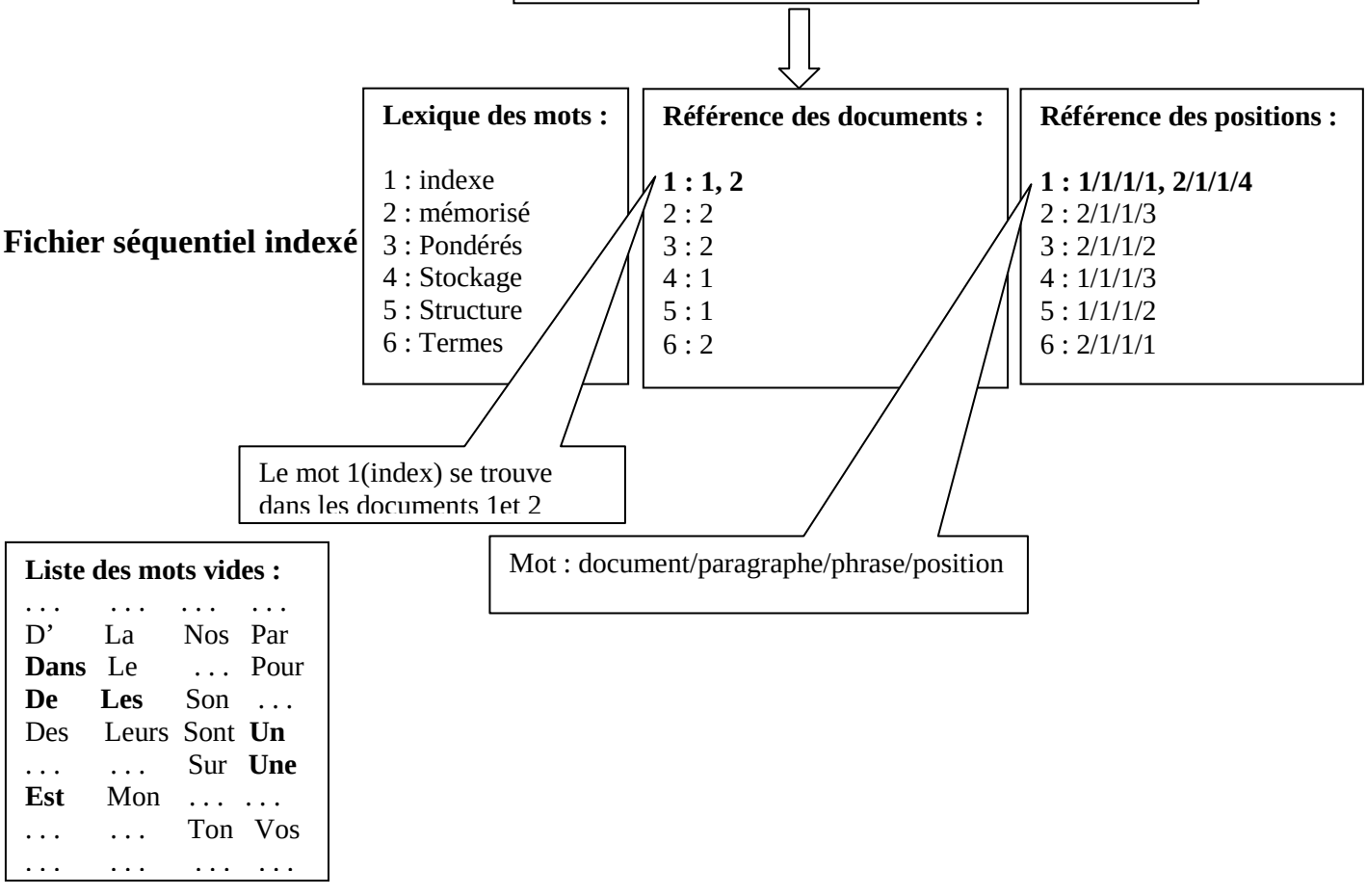


Figure I.2 : Représentation en un fichier séquentiel indexé

Les fichiers inverses :

Le fichier de base dans lequel sont stockées les données est appelé fichier maître ;il donne pour chaque document, les mots et leur position. Néanmoins, si la base documentaire est volumineuse, Cette solution s'avérera très lente.

Pour y remédier, Des fichiers inverses sont créés autour du fichier maitre. Ces fichiers sont le résultat de l'inversion de celui ci ; on donne pour chaque mot les documents qui le contiennent et sa fréquence dans chacun des documents.

Le tableau 1.2 illustre un exemple de fichier inverse : [Ben Aouicha2009]

Document	Contenu
d_1	La(1) recherche(4) d(14) '(15) information(16) gère(28) des(33) textes(37).
d_2	Un(1) système(4) de(12) recherche(15) d(25) '(26) information(27) doit(39) restituer(44) l(54) '(55) information(56) pertinente(68) à(79) l(81) '(82) utilisateur(83).
d_3	Une(1) information(5) est(17) pertinente(21) si(32) elle(35) satisfait(40) l(50) '(51) utilisateur(52).

► Tableau 1.1: Exemple de collection (fichier maître)

terme	d_1	d_2	d_3
recherche	4	15	3
information	16	27, 56	5
gère	28		
textes	37		
système		4	
restituer		44	
pertinente		68	21
utilisateur		83	52
satisfait			40

► Tableau 1.2: Exemple de fichier inverse

Figure I.3 : Représentation en un fichier inverse

I-4-2 Appariement document-requête :

Le processus d'appariement document-requête vise à apparier les documents et la requête de l'utilisateur en comparant leurs descripteurs respectifs. Pour cela, elle s'appuie sur un formalisme précis défini par un modèle de RI. Les documents présentés en résultat à l'utilisateur, et considérés comme les plus pertinents, sont ceux dont les termes d'indexation sont les plus proches de ceux de la requête.

Il existe deux méthodes d'appariement: [Zemerli, 2004]

– **Appariement exact :**

Le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis. Les documents retournés ne sont pas triés.

– **Appariement approché :**

Le résultat est une liste de documents sensés être pertinents pour la requête. Les documents retournés sont triés selon leur score de pertinence vis-à-vis de la requête.

Dans la section suivante, nous décrivons les modèles les plus souvent utilisés en RI.

I-5 Modèles de RI :

Un modèle de RI a pour rôle de fournir une formalisation au processus de recherche d'information et un cadre théorique pour la modélisation de la mesure de pertinence. De nombreux modèles ont été proposés en RI, ils sont généralement regroupés autour des trois familles comme représenté dans la figure ci dessous :

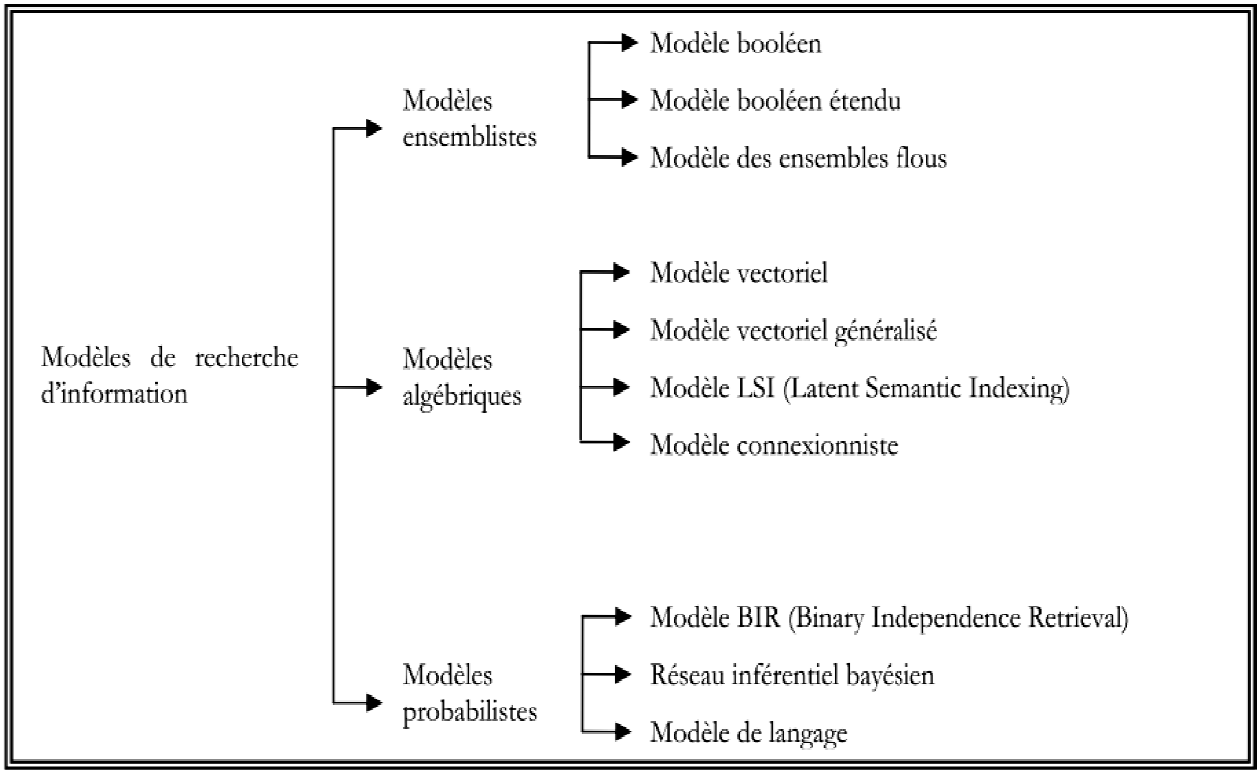


Figure I.4:Taxonomie des modèles de recherche d'information [Mataoui 2007]

I-5-1-Modèles ensemblistes :

Ils sont basés sur la théorie des ensembles. Ce sont les plus simples et les premiers à avoir été mis en œuvre. On y distingue :

I-5-1-1-Modèle booléen :

Dans ce modèle, un document est représenté comme la conjonction de l'ensemble des termes qui le composent.

Une requête est quant à elle considérée comme une expression logique dont les termes sont reliés par les opérateurs de conjonction (^), de disjonction (**v**) ou de négation (¬).

La pertinence entre le document ***Dj*** et la requête ***Q*** (**notée *RSV (Dj, Q))*** se calcule alors de la manière suivante :

- Si la requête contient un seul terme, soit $Q_k = t$ (avec ***t*** : un terme) :

$$RSV(Dj, Q) = 1 \text{ si } t \in Dj, 0 \text{ sinon}$$

- Si la requête contient deux termes reliés par l'opérateur ^, soit $Q = t1 \wedge t2$:

$$RSV(Dj, Q) = 1 \text{ si } RSV(Dj, t1) = 1 \text{ et } RSV(Dj, t2) = 1, 0 \text{ sinon}$$

- Si la requête contient deux termes reliés par l'opérateur **v**, soit $Q = t1 \vee t2$:

$$RSV(Dj, Q) = 1 \text{ si } RSV(Dj, t1) = 1 \text{ ou } RSV(Dj, t2) = 1, 0 \text{ sinon}$$

- Si la requête est composée de la négation d'un seul terme, soit $Q = \neg t$:

$$RSV(Dj, Q) = 1 \text{ si } t \notin Dj, 0 \text{ sinon}$$

➢ **Avantages :**

- Simple à appréhender.
- Efficace si l'utilisateur maîtrise parfaitement le langage de requêtes.

➢ **Inconvénients :**

- Modélisation assez «pauvre» de la notion de pertinence. Cette dernière repose en effet sur un critère exclusivement binaire : un document est soit pertinent, soit non pertinent. Ce modèle ne prend pas non plus en considération la pondération des termes : un mot a un poids égal à **1** s'il appartient au document, **0** sinon.
- Les résultats retournés à l'utilisateur ne peuvent être classés : les documents retournés ont tous la même importance.
- Les documents qui ne contiennent pas tous les termes de la requête sont automatiquement considérés comme non pertinents. Ainsi par exemple, une requête composée des termes **t1**, **t2** et **t3** ne pourra pas être appariée avec un document composé uniquement des termes **t1** et **t2**.

1.3.4.1. Le modèle booléen étendu (ou modèle P_Norm) :

Le modèle booléen a été introduit en 1983 par Salton et al. [Salton et al., 1983]. Ce modèle étend le modèle booléen de base afin de supporter l'appariement approché, ceci en assignant des poids aux termes de la requête et des documents et en mesurant un score de pertinence. Le modèle booléen étendu interprète les opérateurs de l'équation de la requête comme des distances entre requêtes et documents.

Considérons un ensemble de termes $t_1,\ldots,t_N$, et soit *dij* le poids du terme *ti* dans le

document $Dj = (d_{1j},\ldots,d_{Nj})$, avec $1 \le i \le N$ et $0 \le dij \le 1$. La similarité entre le document *Dj* et une requête *Q* décrite sous une forme conjonctive ou disjonctive est donnée comme suit :

$$\textit{Opérateur OR :} \quad RSV(D_j, Q) = \left(\frac{\sum_{i=1}^{N} q_i^p \, d_{ij}^p}{\sum_{i=1}^{N} q_i^p} \right)^{1/p}$$

$$\textit{Opérateur AND :} \quad RSV(D_j, Q) = 1 - \left(\frac{\sum_{i=1}^{N} q_i^p \left(1 - d_{ij}^p\right)}{\sum_{i=1}^{N} q_i^p} \right)^{1/p}$$

Où P une constante $0 \le P \le \infty$, et q_{ik} le poids du terme *ti* dans la requête *Qk*.

Remarque : lorsque p=1, les deux formules étant égales. En effet :

$$\textit{Opérateur AND :} \quad RSV(D_j, Q) = 1 - \left(\frac{\sum_{i=1}^{N} q_i^1 (1 - d_{ij}^1)}{\sum_{i=1}^{N} q_i^p} \right)$$

$$1 - \left(\frac{\sum_{i=1}^{N} q_i}{\sum_{i=1}^{N} q_i^1} - \frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^1} \right) = \frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^1}$$

il n'y a aucune distinction entre les deux connecteurs ET et OU. Par conséquent, la similarité entre les requêtes et les documents peut être calculée par le produit scalaire entre leurs termes pondérés.

La littérature rapporte, qu'aucune méthode formelle n'est proposée pour la détermination de la valeur du paramètre P [Ponte, 1998].

I-5-2- Modèles algébriques :

Ces modèles proposent une représentation vectorielle des documents et requêtes. À partir de ces représentations, la mise en correspondance d'un document et d'une requête revient à appliquer un calcul algébrique de similarité entre vecteurs. On y distingue : le modèle **vectoriel** et les **réseaux de neurones**.

I-5-2-1- Modèle vectoriel :

Dans ce modèle, un document (ou une requête) est représenté par un vecteur de termes pondérés dans un espace à **N** dimensions, où **N :** représente le nombre des termes d'indexation de la collection. .

Un vecteur document ***Dj*** et un vecteur requête Q_k sont donc définis de la manière suivante :

$Dj = (d_{1j}, d_{2j}, d_{3j}, \ldots, d_{Nj})$

$Q = (q_1, q_2, q_3, \ldots, q_N)$

Avec d_{ij} **:** Poids du terme $\mathbf{t_i}$ dans le document ***Dj***.
q_i **:** Poids du terme $\mathbf{t_i}$ dans la requête ***Q***.

Le mécanisme de mise en correspondance évalue la similarité entre les vecteurs documents et le vecteur requête. Les documents considérés comme les plus pertinents sont ceux dont le vecteur est le plus proche de celui de la requête, suivant une mesure de similarité définie au préalable.

Les principales mesures de similarité utilisées sont :

- **Le produit scalaire :** $Sim\ (Q, D_j) = \sum_{i=1}^{N} q_i * d_{ij}$

- **Mesure de Jaccard :** $Sim\ (Q, D_j) = \frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^2 + \sum_{i=1}^{N} d_{ij}^2 - \sum_{i=1}^{N} q_i * d_{ij}}$

- **La mesure cosinus :** $Sim\ (Q, D_j) = \frac{\sum_{i=1}^{N} q_i * d_{ij}}{\left(\sum_{i=1}^{N} q_i^2\right)^{1/2} * \left(\sum_{i=1}^{N} d_{ij}^2\right)^{1/2}}$

➢ **Avantages :**

- La pondération améliore les résultats de la recherche.
- Représentation uniforme des documents et requêtes.
- Les mesures de similarité utilisées permettent d'ajouter à la notion de pertinence un degré d'« approximation ». Un document peut ainsi être considéré comme pertinent même s'il ne contient pas tous les termes de la requête.
- Le classement ordonné des résultats par ordre décroissant de pertinence.

➢ **Inconvénients :**

- Ne prend pas en compte l'ordre des mots. Par exemple, pour les deux requêtes : **la voile du bateau** et **le bateau à voile**, ce modèle offre la même représentation interne (hors pondération) : **Q = (*bateau, voile*)**.

- Ce modèle suppose l'indépendance entre les termes d'indexation. Or ceci n'est pas toujours le cas : en effet, d'une part un concept est souvent représenté par un groupe de mots, d'autre part les mots d'une langue entretiennent les uns avec les autres des relations de natures diverses (synonymie et de polysémie, etc).

I-5-2-2- Modèle connexionniste (basé sur les réseaux de neurones) :

L'idée de base est que la RI est un processus associatif (elle va d'une simple comparaison des requêtes et des documents à des techniques associatives basées sur des associations de documents pour l'expansion de la réponse (sélection de nouveaux documents)) qui peut être représenté par les mécanismes de propagation d'activation des réseaux de neurones.

Un réseau de neurones en RI est généralement composé de plusieurs couches:

- Une couche d'entrée qui désigne **la requête** (chaque neurone correspond à un de ses termes).

- Une couche qui représente **l'ensemble des termes de la collection** (chaque neurone équivaut à un terme d'indexation).

- **Une couche documents** où un nœud représente un document de la collection.

À partir de cette représentation, le mécanisme d'appariement de la requête et des documents est relativement simple. Il se compose des deux phases d'activation suivantes :

⇨ **Première phase d'activation :**

- L'activation initiale des neurones représentant la requête se propage vers les nœuds « termes » de la collection.
- Les nœuds « termes » de la collection, à leur tour, envoient des signaux aux nœuds documents, à travers les différentes connexions du réseau.
- Les documents les plus pertinents sont ceux qui ont reçu le plus de signaux.

⇨ **Deuxième phase d'activation :** Un traitement itératif de propagation du signal peut venir compléter la première phase d'activation :

- Les nœuds des documents considérés comme pertinents génèrent alors de nouveaux signaux en direction des nœuds « termes » de la collection (qui correspondent à des mots considérés comme très représentatifs du contenu de ces documents).
- Les nœuds « termes » de la collection envoient à leur tour de nouveaux signaux dirigés vers les nœuds documents.

Cette seconde phase, qui correspond à une forme de relevance feedback, permet de retrouver des documents dont les termes ne sont pas nécessairement contenus dans la requête.

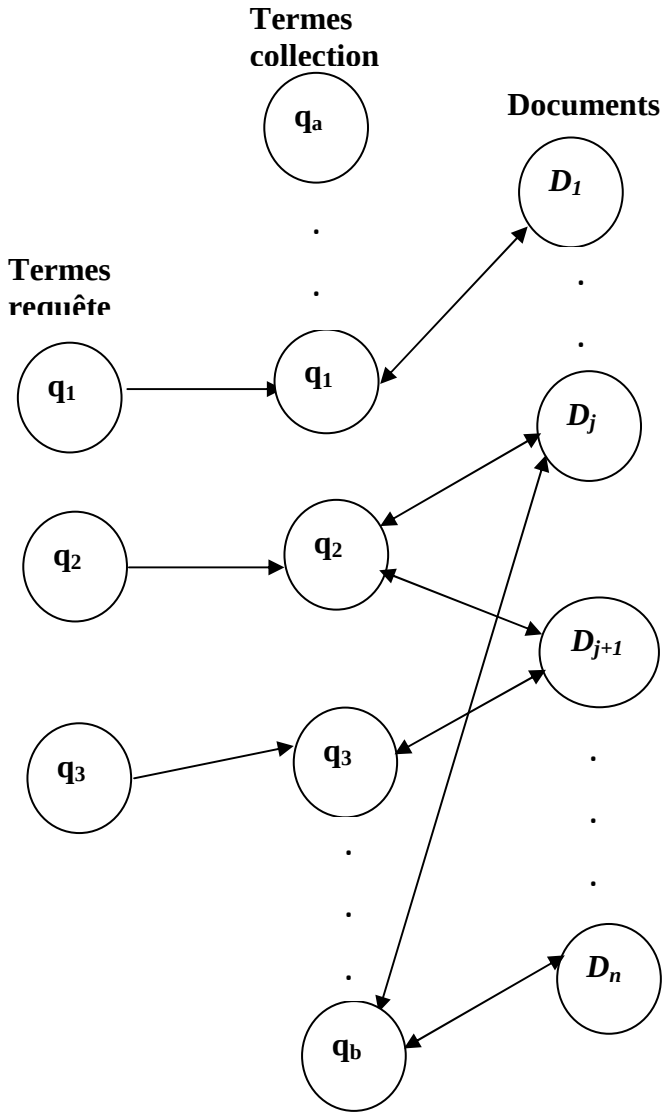


Figure I.5 Modèle de réseau de neurones pour la RI [MOR, 2006]

- **Avantage:**

- Permet de prendre en compte la dépendance entre les éléments d'informations.

- **Inconvénients :**

- Les résultats obtenus sont parfois difficilement interprétables par l'utilisateur, celui-ci ne comprenant pas toujours pourquoi tel document lui a été retourné (fonctionnement en « boîte noire »).

I-5-3-Modèles probabilistes :

Les modèles probabilistes se basent sur la théorie des probabilités. Pour ces modèles, la pertinence d'un document vis-à-vis d'une requête est vue comme une probabilité de pertinence document/requête.

Il s'agit plus précisément ici, de répondre à la question suivante :
*Étant donné un document **d** et une requête **q**, quelle est la probabilité que **d** soit pertinent pour **q** ?*

I-5-3-1- Modèle probabiliste de base :

Il a pour objectif de présenter le résultat de la recherche à l'utilisateur dans un ordre basé sur le rapport de la probabilité de pertinence d'un document pour une requête sur sa probabilité de non pertinence pour cette requête.

Soient donc:
P(R|D) **:** Probabilité qu'un document ***D*** soit pertinent ***(R)***.
P(NR|D) **:** Probabilité qu'un document ***D*** soit non pertinent ***(NR)***.

Le score de correspondance entre un document **(d)** et une requête **(q),**
noté ***RSV (D, Q)***, est donné par :

$$RSV\ (D, Q) = \frac{P(R|D)}{P(NR|D)}$$

Selon le théorème de Bayes, les deux probabilités : ***P(R|D)*** **et** ***P(NR|D)*** sont calculées de la manière suivante :

$$P(R|D) = \frac{P(D|R)P(R)}{P(D)} \text{ et } P(NR|D) = \frac{P(D|NR)P(NR)}{P(D)}$$

Où :
P(D|R) **(**resp. ***P(D|NR))*** **:** Représente la probabilité que le document ***D*** fasse partie de l'ensemble des documents pertinents ***R*** (resp. des documents non pertinents ***NR***).

P(R) **(**resp. ***P(NR))*** **:** Est la probabilité qu'un document choisi au hasard soit pertinent (resp. non pertinent).

***P(D)* :** Correspond à la probabilité qu'un document soit choisi.

Après simplification, le calcul du score de correspondance entre un document et une requête peut être noté :

$$RSV\ (D, Q) \approx \frac{P(D|R)}{P(D|NR)}$$

La probabilité qu'un document soit pertinent s'appuie sur la probabilité de pertinence de ses termes :

Pour chaque terme t_i de la requête, on calcule la probabilité qu'un document qui contienne t_i soit pertinent $(P(t_{i=1}|R))$ ou non pertinent $(P(t_{i=1}|NR))$.

- **Avantage:**

- Les documents jugés pertinents (sélectionnés) seront restitués dans l'ordre de leur pertinence.

- **Inconvénients :**

- Les calculs des probabilités sont complexes.
- Pas de prise en compte des dépendances entre les termes.

I-5-3-2- Modèles de langue :

Le principe des approches utilisant un modèle de langue est différent des approches classiques en RI. En effet, plutôt que d'évaluer le degré de similarité des documents et requêtes, le modèle de langue considère que la pertinence d'un document pour une requête est en rapport avec la probabilité que la requête puisse être générée par le document **[PON & CRO, 98].**

Formellement, soit ***Md***, le modèle de langue du document **d**; la pertinence de ***D*** vis-à-vis d'une requête ***Q*** revient à estimer ***P (Q/Md)***, c'est-à-dire, la probabilité que la requête ***Q*** soit générée par ***Md***. Etant donné une requête ***Q***, cette pertinence est mesurée par :

$$RSV\ (D,Q) = P(Q|M_d) = \prod_{i=1}^{n} P\ (t_i|D)$$

Avec : ***n* :** est le nombre de termes dans la requête.

t_i **:** Un terme de la requête **et** $1 \leq i \leq n$.

Les documents retournés à l'utilisateur sont alors classés par ordre décroissant de la probabilité $P\ (Q|M_d)$.

I-5-3-3- Réseau bayésien : (à détaillé dans le chapitre III)
Les réseaux bayésien on été utilisé en RI depuis les années 90.
On distingue deux principaux types de réseaux, les réseaux d'inférence introduit par Turtle [Turtle & Croft, 1990] et les réseaux de croyance développé par [Ribeiro-Neto et al., 1996]. Ces différentes approches seront détaillées dans le chapitre3.

I-6 Reformulation de la requête :

La reformulation de la requête est un processus ayant pour objectif de modifier la requête initiale afin de mieux répondre au besoin informationnel de l'utilisateur. On distingue deux techniques de reformulation de requête :

- **La technique dite de « rétroaction » de pertinence (relevance feedback)** (qui utilise uniquement des informations issues des documents et requêtes).

- **La technique d'expansion de requêtes (**basée sur des ressources linguistiques externes).

I-6-1Technique de relevance feedback :

Cette technique consiste à prendre en considération les jugements de pertinence de l'utilisateur pour formuler une requête qui répond mieux à son besoin.
La technique fonctionne comme suit:

- L'utilisateur soumet sa requête.
- Les documents résultant de la première recherche seront retournés.
- L'utilisateur les examine et détermine les documents pertinents et non pertinents.
- Le SRI modifie la requête initiale, en donnant plus d'importance aux termes apparaissant dans les documents pertinents, et affaiblissant la force de ceux qui appartiennent dans les documents non pertinents.
- Ce processus est répété jusqu'à ce que l'utilisateur soit complètement satisfait de tous les documents trouvés.

I-6-2Technique d'expansion de requêtes :

Dans ce cas la requête est étendue automatiquement, sans intervention de l'utilisateur, par ajout de termes supplémentaires, ce qui permet de trouver des nouveaux documents pertinents qui ne contiennent pas forcément les termes de la requête initiale.

- Les termes ajoutés sont issus de différentes sources telles que :

  - **Des ressources linguistiques externes :**

L'extension de la requête peut être effectuée à partir d'un **dictionnaire, thésaurus,**...qui définit les relations entre les différents termes d'index et permet de sélectionner de nouveaux termes à ajouter à la requête initiale.

- **Des ressources internes :**

Les informations sont acquises directement à partir de la collection de documents. (**exp :** les cooccurrences des termes dans un document, ou les termes appartenant aux premiers documents retournés à l'issue d'une première recherche ...).

I-7 Evaluation des performances d'un SRI :

L'évaluation d'un SRI peut porter sur plusieurs critères :

- La pertinence.
- Le temps de réponse : durée écoulée entre l'instant où l'utilisateur interroge le système et l'instant où ce dernier restitue la réponse. Le but est bien évidemment de réduire au maximum cette durée.
- La qualité et la présentation des résultats : elle doit être conviviale, claire, simple, accessible à tous.
- La facilité d'utilisation,…

Néanmoins, le critère le plus important est celui qui mesure la capacité du système à satisfaire le besoin en information de l'utilisateur, c'est à dire la pertinence des résultats retournés. Quatre facteurs permettent d'évaluer ce critère : le rappel, la précision, le bruit et le silence.

- **Le rappel (R) :** Est une mesure du pourcentage des documents pertinents ayant été retrouvée parmi tous les documents pertinents dans la base (collection).

$$\mathbf{R} = \frac{\textbf{Nombre de documents pertinents retrouvés}}{\textbf{Nombre de documents pertinents dans la collection}}$$

- **La précision (P) :** Calcule le pourcentage de documents pertinents retrouvés parmi tous les documents retrouvés par le système. Elle est calculée comme :

$$\mathbf{P} = \frac{\textbf{Nombre de documents pertinents retrouvés}}{\textbf{Nombre total de documents retrouvés}}$$

- **Le bruit :** Il mesure le taux de documents non pertinents extraits par rapport à la totalité des documents extraits :

$$\textbf{Bruit} = \frac{\textbf{Nombre de documents non pertinents retrouvés}}{\textbf{Nombre de documents retrouvés}}$$

- **Le silence :** Il mesure le taux de documents pertinents non extraits par rapport à la totalité des documents pertinents contenus dans le corpus :

$$\textbf{Silence} = \frac{\textbf{Nombre de documents pertinents non retrouvés}}{\textbf{Nombre de documents pertinents}}$$

Remarque :

- La performance d'un SRI peut être appréciée si, d'un côté, les taux de précision et de rappel sont élevés et, de l'autre, les taux de bruit et de silence sot bas.
- Un rappel égal à **1** indique que tous les documents pertinents ont été retrouvés. Une précision égale à **1** indique que tous les documents retrouvés sont pertinents. Ces deux taux varient en sens inverse.
- Les relations entre ces quatre paramètres peuvent être formulées comme suit :

  Rappel + silence =1.

  Précision + bruit =1.

  Le calcul effectif de ces valeurs est rendu possible grâce aux collections de tests.
  Une collection de tests comprend :

1. Un ensemble de documents (ou collection de documents) à indexer, sur lesquels le système sera évalué,
2. Une liste de requêtes prédéfinies,
3. Les jugements de pertinence, manuellement établis, pour chaque requête. Il s'agit, pour chaque requête, de la liste des documents pertinents pour cette requête.

Pour évaluer un système, on l'interroge avec des requêtes et on calcule les mesures introduites ci-dessus. A partir de là, on peut juger l'efficacité et les performances du système.

Les collections de tests sont, le plus généralement, mises en place dans le cadre de campagnes d'évaluation des SRI, dont les campagnes **TREC**[1] *(Text Retrieval Conference)* **[HAR, 92]** qui constituent la référence en ce qui concerne l'évaluation des SRI.

[1] http://trec.nist.gov : TREC est un projet international initié au début des années **90** par le **NIST (**National Institute of Standards and Technology) aux Etats-Unis dans le but de proposer des moyens homogènes d'évaluation de systèmes documentaires.

Conclusion :

Dans ce chapitre, nous avons présenté un état de l'art sur la recherche d'information classique. Nous y avons abordé la définition, les techniques d'indexation utilisées en RI, les principaux modèles de recherche, ainsi, que les méthodes d'évaluation adoptées pour attester de la qualité d'un SRI.

Cependant, ces SRI traitent les documents comme une unité indivisible. L'utilisateur se trouve alors obligé de les parcourir pour trouver l'information souhaitée.

Actuellement, de nouveaux formats de structuration de l'information comme XML sont en train de s'imposer, permettant ainsi, de représenter conjointement l'information textuelle et les contraintes structurelles (balises) d'un document. La RI plein texte, évolue vers une RI structurée, prenant donc en compte la structure et le contenu des documents. Ce qui représente un nouveau défi pour la RI. C'est dans ce contexte, que se situe notre travail, qui consiste à restituer à l'utilisateur non pas un document intégral suite à une formulation de sa requête, mais un fragment de document structuré répondant ainsi, de manière ciblée et efficace à sa requête.

Nous décrivons dans le prochain chapitre, une description du langage XML et les différentes approches issues de la RI structurée.

Introduction :

La nature des collections de documents électroniques évolue. Elles intègrent de plus en plus des méta-informations[2] et notamment des informations structurelles : de simples documents texte « plat », on dispose aujourd'hui de documents structurés ou semi-structurés.
Les informations structurelles sont liées à l'utilisation de formats tels que SGML (*Standard GeneralizedMarkup Language*) [Goldfarb, 1990] ou encore XML (*eXtensibleMarkup Language*) .Ces derniers, conçus à l'origine pour faciliter l'échange et la standardisation des données, voient leur importance augmenter grâce à l'expansion d'Internet.

Pour la RI, ces types de formats soulèvent des problèmes liées à la co-existence de l'information structurelle et de l'information de contenu (ces problèmes sont soulevée dans la section II.4.2. Néanmoins, la prise en compte de l'aspect structurelle devrait permettre de mieux répondre aux besoins de l'utilisateur.

Ce chapitre a pour objectif de présenter le langage XML, soulever les différentes problématiques liées à la RI sur des collections d'informations semi-structurées et présenter les différentes solutions proposées dans la littérature à savoir de nouvelles techniques d'indexation, ainsi que de nouveaux langages d'interrogation prenant en compte la structure. Enfin, nous présenterons les différents modèles de recherche d'information proposés.

II-1Notions générales du langage XML :

II-1-1 Historique :

XML est l'acronyme de ***eXtensible Markup Language***, ou *Langage à balises extensibles*. Il est né du succès du HTML (*HyperText Markup Language*) mais aussi de la constatation de ses insuffisances. Comme ce dernier, XML est issue du langage de balise généralisé SGML (*Standard Generaized Markup Language*) et en est une simplification.
Contrairement à **HTML**, qui est à considérer comme un langage défini, figé (avec un nombre de balises limité) et ne permettant pas d'interpréter ou d'échanger les données, **XML** peut être considéré comme un métalangage permettant de définir d'autres langages tel que le XHTML, MATHML….etc.

XML a été développé par le **XML Working Group** sous la tutelle du **W3C** [3](World Wide Web Consortium) dès 1996. Depuis le 10 février 1998, les

[2]Est une information servant à définir ou décrire une autre information quelque soit son support (papier ou electronique)

[3] une association à buts non lucratifs, visant à uniformiser les langages et les technologies utilisés sur le Web., le W3C n'est pas un organisme de normalisation. Il ne définit pas des normes, mais des recommandations. Son site est accessible sur http://www.w3c.org

spécifications **XML 1.0** sont reconnues comme recommandation par le **W3C**, ultime étape du processus d'approbation de cet organisme.

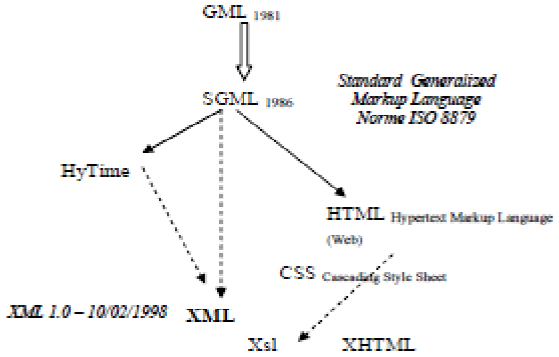


Figure II.1: Historique des langages de balisage, extrait de [C.Chrisment, 2005]

II-1-2Caractéristiques du langage XML :

XML se caractérise par les points forts suivants :
- **Lisibilité :** Le contenu d'un document **XML** est lisible dans n'importe quel éditeur de texte, et est compréhensible par une personne n'ayant pas de connaissance particulière.
- **Auto-descriptif et extensible :** On peut définir les balises de notre choix.
-**Modularité :** Réutilisation de description.
- **Une structure arborescente :** le langage permet la description arborescente des données afin d'apporter par l'intermédiaire des arbres des structures de donnée performantes et efficaces.
- **Universel et portable :** Les différents jeux de caractères de type unicode sont pris en compte.
- **Intégration :** Un document XML est utilisable par toutes applications pourvues d'un parseur texte (analyseur).

II-1-3 les composants d'un document XML :

Afin de créer un document **XML**, un certain nombre de règles de base doivent être respectées.

II-1-3-1 L'en-tête : *le prologue*

- **L'en-tête minimum** : <?xml version="1.0"?>
  version représente la version de XML utilisée.

- Cet en-tête peut être complété par l'encodage qui est utilisé pour la lecture
  Exemple : <?xml version="1.0" encoding="ISO-8859-1"?>

- Un autre élément facultatif **standalone.** Il signale si le XML en lui seul est suffisant pour être lu ou s'il nécessite l'adjonction d'un autre document externe dit DTD (voir section II.3.1.1). La valeur par défaut de **standalone** est yes :

<?xml version="1.0" standalone='yes'?>

- Si l'encoding n'est pas précisé, UTF-8 sera utilisé par défaut.

II-1-3-2 Les éléments (balises) :

Les éléments représentent les composants les plus importants des documents **XML.** Ces éléments structurent le contenu du document sous la forme d'un arbre. De ce fait, tout document **XML** doit posséder un élément racine unique dans lequel sont imbriqués tous les autres éléments. Les éléments sont définis par des balises. Il existe deux types d'éléments:

- Pour tout élément avec contenu, une balise de début et de fin sont obligatoires.
  <nom_balise>contenu_balise</nom_balise>

- Pour un élément sans contenu on peut utiliser une écriture raccourcie
  <nom_balise/> ou bien <nom_balise></nom_balise>

Exemple de document XML :

La description d'un livre donné, pourrait se présenter ainsi(en remarquant l'élément racine **livre**) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<livre>
   <titre>XML cours et exercices</titre>
   <tome/>
   <auteur>
      <prenom>Alexandre</prenom>
      <nom>Brillant</nom>
    </auteur>
    <editeur>Eyrolles</editeur>
</livre>
```

- un élément peut soit contenir du texte, soit d'autres éléments soit un mélange des deux.

Exemple d'élément mixte :

<paragraphe>
XML permet l'extraction de l'information d'une manière extrêmement simple et puissante **<note>** Il est important de donner des noms, aux balises, relativement explicites. **</note>** pour le besoin d'une requête.
</paragraphe>

Remarques :

- ✓ Les balises, dans un document XML, ne peuvent se superposer.
- ✓ Les noms des éléments doivent être des *noms XML* :
  - le premier caractère est une lettre quelconque ou un _ (underscore ou tiret bas) ;

- les caractères suivants peuvent être des lettres, des chiffres, des tirets bas (_), des traits d'union (-) ou des points (.) ;
- il n'y a pas de limitation sur la longueur d'un nom XML.

✓ pour ce qui est du texte certains caractères sont interdits. Ainsi < et > seront remplacés par leur référence <(lower than) et >(greater than)

II-1-3-3 Attribut :

Un élément peut être porteur d'un ou plusieurs attributs sous la forme de paires (nom_attribut= 'valeur_attribut').

Exemple :

<livre id ='inf25'> Présentation du XML </livre>

✓ Les noms des attributs suivent les mêmes règles d'écriture que ceux des éléments.
✓ Les attributs ne peuvent contenir que du texte encadré par des apostrophes (' ') ou par des guillemets ('' '').
✓ Un élément ne peut posséder deux attributs de même nom

II-1-3-4 Espace de nom (namespace) :

Les namespaces (espaces de nom) sont la solution trouvée au problème des balises homonymes.

En effet, XML étant un métalangage, deux concepteurs peuvent avoir fait les mêmes choix de noms de balises sans que leurs interprétations soient identiques.

Un namespace se présente sous la forme d'un « attribut » xmlns (**XMLn**ame**s**pace) suivi d'une URI (*Uniform Resource Identifier*).

Un namespace peut se déclarer tel quel; dans ce cas, toutes les balises contenues dans la balise porteuse, et qui ne sont pas porteuses d'un autre namespace, seront considérées comme incluses dans l'espace de nom.

Exemple :

```
<?xml version="1.0"?>
<racine>
   <a xmlns="NStest1">
      <b>
         <c/>
      </b>
      <d xmlns="NStest2"/>
   </a>
   <e/>
</racine>
```

Les balises a, b, c appartiennent à l'espace de nom Nstest1. La balise **d** appartient à l'espace de nom **Nstest2.** Les balises **racine** et **e** n'appartiennent à aucun de ces deux namespaces.

II-1-3-5. Les sections CDTA:

Les sections XML CDATA contiennent du texte brut qui doit être inclus, mais non analysé, avec le XML qui le contient. Une section XML CDATA peut contenir n'importe quel texte, notamment les caractères XML réservés. Une section **CDTA** est définie par la syntaxe suivante :

<![CDATA[Texte_ici]]>

Exemple :

<Remarque><![CDATA[le prologue est encapsulé par <? et ?> et il n'existe pas d'espaces (de blancs) entre le début du document et cet élément**.]]></Remarque>**

- Ces sections peuvent se trouver à n'importe quel endroit acceptable pour des données textuelles.

II-1-3-6 Commentaires :

Ils se positionnent n'importe où après le prologue et peuvent figurer sur plusieurs lignes.

Exemple :

<!-- Date de création : 30/09/07 -->

II-1-3-7 *Les instructions de traitement:*

Les instructions de traitement (*processing instruction* ou PI) servent à donner à l'application qui utilise le document XML des actions à exécuter. Ces dernières sont totalement libres et dépendent avant tout du concepteur de l'application de traitement. On les positionne à n'importe quel endroit du document.
Une processing-instruction commence par <? et se termine par ?>

Le plus souvent elles servent à intégrer un script au document ou à lui adjoindre une feuille de style

Exemple :

<? xml-stylesheet type="text/xsl" href="classe_08.xsl"?>

Cette ligne (destinée, par défaut, à un navigateur) affecte une feuille de style (stylesheet) au document. La première partie indique le langage de feuille de style utilisé qui est ici XSL (eXtensible Stylesheet Language). La deuxième information à fournir est la localisation du fichier de style (href signifiant Hypertext REFerence).

II-2- Les documents semi-structurés et les documents structurés :

Le format XML permet de produire des documents *structurés* ou *semi-structurés.*
Contrairement aux documents plats, dans les documents entièrement structurés, nous ne parlons plus de "*texte*", mais plutôt de données. 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 de leurs éléments est généralement non significatif.
Les documents *semi-structurés* sont un pont entre les données structurées et non structurées. Ils possèdent des contenus hétérogènes et une structure flexible qui *n'est pas aussi rigide, aussi régulière ou complète que la structure requise par les systèmes de gestion de bases de données traditionnels.* La modification, l'ajout ou la suppression d'une donnée entraîne une modification de la structure de l'ensemble.

Dans notre contexte, nous nous intéressons plus particulièrement à la recherche d'information dans des documents semi-structurés, les documents structurés servant plutôt à conserver des données au sens bases de données. Par abus de langage, on parlera cependant de RI structurée.

II-2-1 Les documents bien formés et les documents valides :

Un document est appelé "document XML" s'il est *bien formé*, ce qui signifie qu'il respecte les règles syntaxiques de XML. Il est *valide* si, en plus, il respecte la grammaire du langage, contenue dans un fichier appelé DTD (voir section II-3-1-1). La norme XML n'impose pas l'utilisation d'une DTD pour un document XML (un tel document définit son propre balisage de manière informelle), mais elle impose par contre le respect exact des règles de base de la norme XML (il doit être bien formé).

II-3- la Galaxie XML :

Une galaxie de standards ou de recommandations a émergé conjointement à XML afin de définir des outils et des applications autour du langage.

II-3-1La validation :

XML permet d'utiliser un fichier afin de vérifier qu'un document XML est conforme à une syntaxe donnée. On distingue :

II-3-1-1 la Validation par une DTD :

La norme XML définit une grammaire appelée DTD (*Document Type Definition :* une définition de document type) permettant de vérifier la conformité du document XML. Une DTD peut être définie de 3 façons :

a- DTD Interne : Cela consiste à inclure la grammaire au sein même du document (après le prologue).

Syntaxe :

```
< !DOCTYPE nom_element_racine [
        contenu  DTD ] contenu  document  XML
```

Exemple très simple de document xml avec une DTD interne :

```
<?xml version="1.0"  ?>
<!DOCTYPE salutation [
<!ELEMENT salutation (#PCDATA)>
]>
<salutation>Bonjour et Bienvenus à tous </salutation>
```

La DTD indique que le document est composé d'un seul élément qui ne peut que contenir des données textuelles (#PCDATA -Parsed Character Data). La dernière ligne présente un document XML respectant cette DTD.

b-La DTD Externe : la DTD est référencée en appelant un fichier contenant la grammaire à partir d'un fichier local ou bien en y accédant par son URI.

Syntaxe :

```
< !DOCTYPE nom_élément_racine     SYSTEM   URI_du_fichier  DTD>
```

À noter également que la première ligne du document XML (le prologue) doit faire apparaître l'attribut " standalone" avec la valeur "no".

Exemple : **DTD Externe**

Soit la grammaire (DTD) stockée dans le fichier document. dtd(à gauche) et un exemple de document valide (à droite) définis comme suit :

La DTD externe	**le document XML valide**
<!ELEMENT article (titre, paragraphe+, commentaire?)> <!ATTLIST article num CDATA #REQUIRED> <!ELEMENT titre (#PCDATA)> <!ELEMENT pragraphe (#PCDATA)> <!ELEMENT commentaire (#PCDATA)	<?xml version='1.0' encoding='iso-8859-1' standalone="no"?> <**!DOCTYPE** article SYSTEM "document.dtd" > **<article num='12'>** **< titre>** Présentation de XSLT **</titre>** **<paragraphe>** Il sert à transformer des documents XML dans divers formats… **</paragraphe>** **<paragraphe>** la feuille de Transformation ….**</paragraphe>** **<commentaire/>** **</article>**

Cette DTD décrit un article composée d'un titre, d'un ou plusieurs paragraphes et un commentaire qui est optionnel. L'élément racine (article) possède un attribut (num) qui est obligatoire(Required).

c- La DTD Mixte :

Il est possible de mélanger les deux notations pour avoir une partie de la DTD dans un fichier séparé et une autre partie embarquée dans le document XML :

❖ **Les entités :**

Il s'agit de définir des raccourcis (ou des alias) qui seront utilisables dans les documents XML.

- Certaines entités sont déjà définies en XML :

Exemple : < (<),> (>),& (&)," ("), et ' (').

- Pour définir des *entités générales, on suit la syntaxe :*

*< **!ENTITY** nom_entité valeur_entité>*

La valeur de l'entité est mentionnée entre guillemets ou apostrophes.

- Cette définition sera incluse dans la DTD associée au document.

Exemple :

```
<?xml version="1.0"?>
<!DOCTYPE racine [
  <!ENTITY hel "hello world!!!">
]>
<racine>&hel;</racine>

est équivalente à :

<?xml version="1.0"?>
<racine>hello world!!!</racine>
```

- La valeur de l'entité peut contenir du texte, des balises et des entités (le code XML ainsi défini doit être bien formé).

II-3-1-2 validation par un schéma :

Les schémas XML se présentent comme une alternative aux DTDs permettant d'apporter des réponses aux limites de ces dernières (ex : elles ne gèrent pas les espaces de noms et leur système de type est pauvre : le contenu des nœuds textes -#PCDATA- ne peut pas être typé). Comme une DTD, un schéma permet de définir un ensemble de règles visant à définir un document XML, et notamment les marqueurs autorisés, leurs attributs et relations les uns par rapport aux autres. Mais contrairement à une DTD, un schéma permet de définir des types pour les données.

De plus, un Schéma XML est un document XML à part entière et peut donc être édité et manipulé à partir de n'importe quel outil d'édition ou de traitement XML.

Bien que les DTD et les schémas soient les langages de validation les plus utilisés, il existe néanmoins d'autres outils de validation qui seront décrit en annexe A.

II-3-2. Les parseurs :

Un document XML, c'est avant tout un fichier ou un flux texte. S'il doit être utilisé par un langage, celui-ci doit choisir la façon de le stocker en mémoire et de le parcourir, c'est le rôle d'un parseur.
Cet outil logiciel, appelé parfois un analyseur syntaxique, permet à une application cliente de valider (s'il est validant) un document XML et de le lire, voire de le modifier. Un parseur met à disposition de l'application cliente les données XML lues au travers d'API, les plus répandues étant SAX et DOM. Une autre catégorie, plus récente, intègre une technologie PULL, associant les avantages de SAX - définie comme une technologie PUSH - et DOM sans leurs inconvénients.

II-3-2-1. DOM :

DOM (*Document Object Model*) est une API (***A****pplication* ***P****rogramming* ***I****nterface ou interface de programmation d'application*) permettant d'accéder au contenu d'un document XML sous la forme d'une structure arborescente. Le document XML, après avoir été totalement chargé en mémoire, est accessible au travers d'un ensemble d'objets correspondant aux différents types de nœuds qui s'y trouvent, et exposant les méthodes permettant de parcourir l'arbre, de façon hiérarchique ou transversale.

L'arbre généré se compose d'une racine **Document**, de nœuds internes représentant les éléments ou les attributs, et de nœuds feuilles (texte) contenant les valeurs d'éléments ou d'attributs.

Exemple :

```
< ?xml version='1.0' encoding= ''ISO-8859-1'' ?>

<Ouvrage code=''inf125'' >
    <Titre > XML  </Titre >
    <auteur> J. Marcus</auteur>
    <Résumé>Devant le nombre croissant  de documents …</Résumé>
</Ouvrage>
```

Son arbre DOM est :

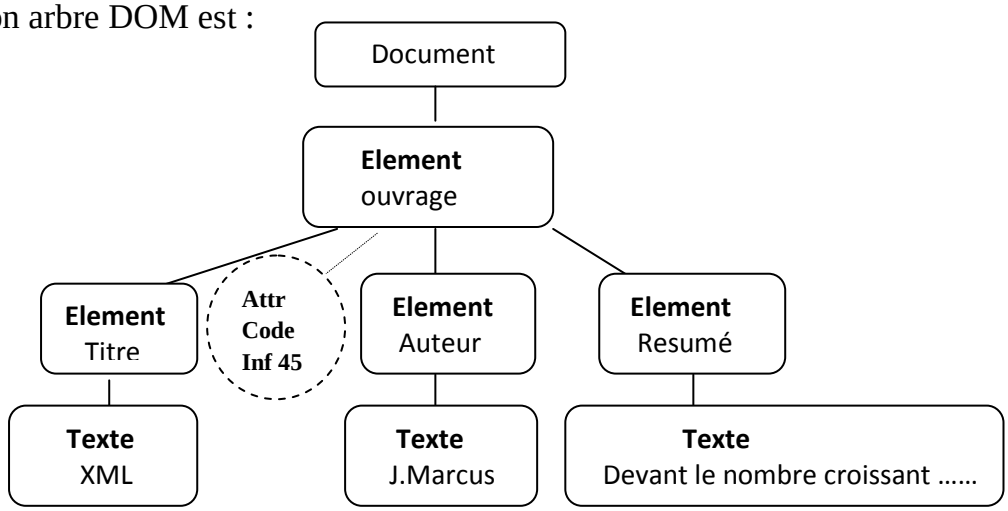


Figure II.2 l'arbre DOM

Remarques :

- DOM est une recommandation du W3C, proposée en plusieurs versions (level) proposant des fonctionnalités croissantes et dont la compatibilité est ascendante.
- Il existe des implémentations de DOM dans pratiquement tous les langages interprétés ou compilés existants pouvant lire des documents XML.

II-3-2-2 SAX (*Simple API for XML*):

Cette API fournit une interface événementielle pour parcourir un document XML : elle renvoie à l'application qui manipule le document XML des "événements" (ouverture de balise, fermeture de balise, contenu textuel...). Elle permet donc de traiter à la volée l'occurrence de telle ou telle balise. Le début d'un élément est capturé par une méthode et sa fin par une autre méthode. Ainsi chacun des événements déclenchés demande une implémentation de l'interface SAX afin de travailler avec l'information fournie par le parseur.

SAX a comme avantage, grâce à son fonctionnement, de ne lire le code que par petites portions, ce qui lui évite de le charger en mémoire intégralement, contrairement à DOM. Il est donc très intéressant pour la lecture des gros documents. Il ne sera par contre pas adapté aux cas où l'application cliente ne peut pas se contenter d'un parcours linéaire et par "petits bouts" du document XML.

II-4 les enjeux de la recherche d'information structurée :

II-4-1 Granularité du résultat d'une recherche :

Comme on l'a vu dans le premier chapitre, en RI, dite désormais, traditionnelle l'unité d'information restituée, en réponse à la requête de l'utilisateur, est le document tout entier ;
Ce qui oblige l'utilisateur à chercher l'information qu'il désire avoir à l'intérieur du document. Dans le cas des documents XML, l'unité d'information pouvant être renvoyée à l'utilisateur correspond à un noeud de l'arbre du document, c'est-à-dire à un *sous-arbre*.
L'information doit maintenant être traitée sans connaissance *a priori* de la structure des documents, et en tenant compte en plus des liens d'adjacence, de la présence de liens d'inclusions entre les balises.
La pertinence d'un nœud vis-à-vis d'une requête est évaluée selon les deux notions suivantes : l'*exhaustivité* et la *spécificité* [Lalmas, 1997].
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.
L'objectif de la recherche d'information structurée serait donc de trouver les sous-arbres de taille minimale les plus exhaustifs et spécifiques pour une la requête.

II-4-2 Les problèmes soulevés par la Recherche d'information structurée :

La dimension structurelle des documents structurés, même si c'est une ressource complémentaire qui devrait être exploitée pendant la recherche d'information, a soulevé plusieurs problématiques relatives à chaque étape du processus de recherche.

- La première problématique est *l'indexation de ces documents* qui doit prendre en considération l'information structurelle qui s'ajoute au contenu, ce qui engendre de nouvelles interrogations : Que doit-on indexer de la structure des documents ? Comment relier cette structure au contenu même du document ? Comment pondérer les termes d'indexation ?. L'objectif du processus d'indexation dans ce cas, est alors de *concilier contenu et structure* pour pouvoir *effectuer des recherches* de contenu et/ou de structure sur les documents.

- La deuxième problématique est celle de *l'interrogation des corpus de documents semi structurés.* Un système doit pouvoir permettre à un utilisateur d'exprimer ses besoins d'une manière simple et en exploitant les deux types d'information contenues dans les documents semi structurés (information textuelle et information structurelle). Pour manipuler et exploiter les documents XML, il faut chercher à simplifier les langages d'interrogation des bases de données (documents structurées) tout en proposant de nouvelles fonctionnalités.

- La dernière problématique est celle *des modèles de recherche et de tri des unités d'informations.* Les systèmes de recherche doivent pouvoir décider de la

granularité de l'information à renvoyer s'il s'agit d'une requête orientée contenu seulement. S'il s'agit d'une requête orientée contenu et structure deux cas sont envisageables :

- l'utilisateur spécifie le type d'éléments à renvoyer ;
- l'utilisateur ne spécifie pas le type d'éléments à renvoyer ou certains éléments de la requête sont imbriqués les uns dans les autres[4]. Dans ce cas, c'est au système de décider de la granularité de l'information à renvoyer.

Remarque :

La RI structurée englobe deux approches qui tentent de proposer des méthodes pour l'indexation, l'interrogation, la recherche et le tri des documents XML. Ces deux approches sont : L'approche orientée données et L'approche orientée document que nous présentons en Annexe A.

II-5 Indexation des documents XML :

L'indexation des documents semi-structurés diffère de celle utilisée pour des documents textes « plats » par le fait de la prise en compte de la dimension structurelle qui s'ajoute au contenu.

II-5-1Critère d'indexation des documents XML :

Un schéma d'indexation doit respecter les critères suivants :

- permettre la reconstruction du document XML décomposé dans les structures de stockage;
- permettre le traitement des expressions de chemin sur la structure XML;
- permettre le traitement de prédicats vagues et précis sur le contenu de documents XML;
- permettre la navigation dans des documents XML;
- permettre la recherche par mots-clés.

Remarque :

Le stockage des documents XML est abordé en Annexe A

II-5-2 les techniques d'indexation :

Même si l'indexation des informations de contenu et des informations structurelles sont étroitement liées, nous nous proposons de les décrire séparément, afin de mieux comprendre les différents enjeux soulevés par l'une et l'autre.

II-5-2-1 Indexation de l'information textuelle :

L'indexation de l'information textuelle, c'est-à-dire l'extraction et la pondération des termes, est similaire à la RI classique. Sa spécificité dans les documents XML, réside dans la description des relations entre les termes et l'information structurelle : c'est ce qu'on appelle la "portée des termes d'indexation".

[4] C'est le cas de la requête : chercher un chapitre, une section ou un paragraphe qui parle de la recherche d'information

❖ Portée des termes d'indexation

Nous tentons dans cette partie de trouver une réponse à la question : « comment rattacher les termes à l'information structurelle ? ». Deux solutions ont été proposées dans la littérature :

- l'approche d'indexation par propagation des termes qui cherche à agréger le contenu des nœuds ;
- l'approche d'indexation des unités disjointes qui indexe tous les contenus des nœuds séparément.

a- sous arbres imbriqués (propagation des termes) :

Ces approches considèrent que le texte complet de chaque nœud de l'index est un document atomique et propagent donc les termes des nœuds feuilles dans l'arbre des documents [Abolhassani et al, 2004].

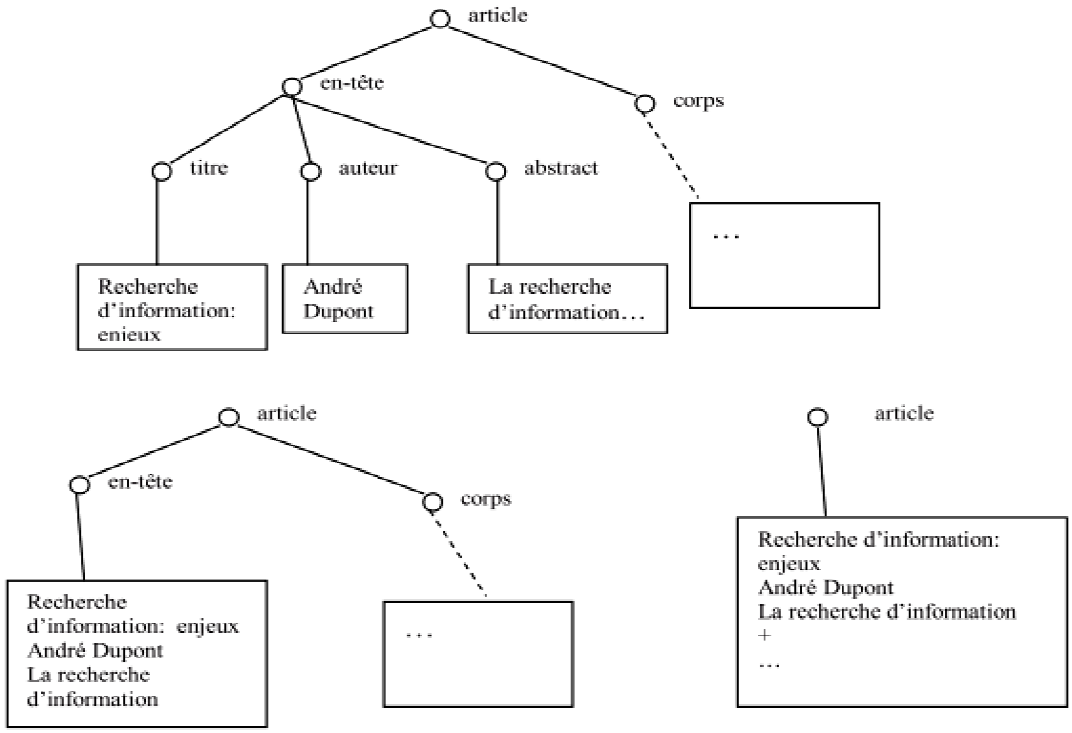


Figure II.3 exemple de propagation de termes [sauvagnat, 2005]

Les termes "André Dupont" sont reliés aux nœuds /article/entête/auteur, /article/en-tête, et /article

b-Unités disjointes :

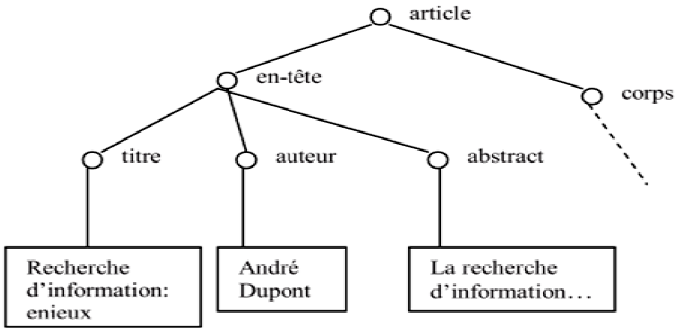
Dans ces approches le document XML est décomposé en unités disjointes, de telle façon que le texte de chaque nœud de l'index est l'union d'une ou de plus de ces parties disjointes.
Les termes des nœuds feuilles sont uniquement reliés au noeud parent qui les
Contient [Sauvagnat, 2005]. Si on reprend en exemple l'arbre de la figure II.4, les termes "recherche d'information enjeux" seront uniquement reliés au noeud /article/en-tête/titre, les termes "André Dupond" au noeud /article/en-tête/auteur et les termes "la recherche d'information" au noeud/article/en-tête/abstract.

II-5-2-2 Indexation de l'information structurelle :

Dans la littérature on distingue trois classes d'approches pour l'indexation de l'information structurelle, la première dite basée sur des champs, la seconde dite basée sur des chemins, et la dernière dite basée sur des arbres. Ces trois approches sont indépendantes de la manière d'utiliser l'information textuelle (l'approche BD ou bien l'approche RI).

- **Indexation basée sur des champs :**

Dans cette méthode d'indexation, le document est représenté comme un ensemble de champs et du contenu associé à chaque champ. Plusieurs façons permettent d'obtenir les différents champs d'un document XML :
- ils peuvent être codés en tant que métadonnées dans les fichiers XML, par exemple en utilisant RDF ;
- dans le cas d'un document d'un format quelconque transformé en XML, ils peuvent provenir du document dans son format original ;
- ils peuvent être retrouvés à l'aide de différentes techniques d'extraction ;
- ils sont simplement extraits de la DTD ou du schéma XML associé.



Termes	Champs
Recherche	(Titre), (abstract)
Information	(Titre), (abstract)
Enjeux	(Titre)
André	(auteur)
Dupont	(auteur)

Figure II.4 – Exemple d'indexation basée sur des champs [sauvagnat, 2005]

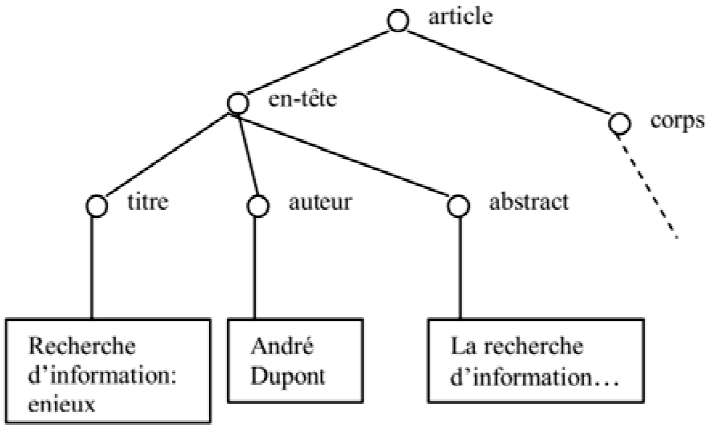
Cette technique est fortement limitée car elle ne permet pas la navigation dans l'arbre document ni même de retrouver le chemin menant à un noeud. C'est ce que proposent les approches d'indexation des chemins.

- **l'indexation par les segments :**

L'indexation par les segments divise les documents structurés en régions.
Les modèles qui y sont associés sont très efficaces, mais antérieurs à l'apparition du format XML, et ils en supportent mal toutes les possibilités.

- **Indexation basée sur des chemins**

Ce type de techniques facilite la navigation dans les documents en permettant la résolution des expressions XPath, il permet aussi de retrouver des documents ayant des valeurs connues pour certains éléments ou attributs, et utilise des index pleins textes sur les contenus.
Ces techniques souffrent de la difficulté de retrouver les relations ancêtre-descendant entre les différents nœuds des documents. La figure suivante (figure II.5) représente une illustration de ce type d'indexation.



Termes	Chemins
Recherche	(/article /en-tête/titre), (/article/en-tête/abstract)
Information	(/article /en-tête/titre), (/article/en-tête/abstract)
Enjeux	(/article /en-tête/titre)
André	(/article /en-tête/auteur)
Dupont	(/article /en-tête/auteur)

Chemins	Documents
/article /en-tête/titre	Article.xml
/article /en-tête/auteur	Article.xml
/article/en-tête/abstract	Article.xml
……	……..

Figure II.5 : Exemple d'indexation basée sur des chemins [sauvagnat, 2005]

❖ Indexation basée sur des arbres :

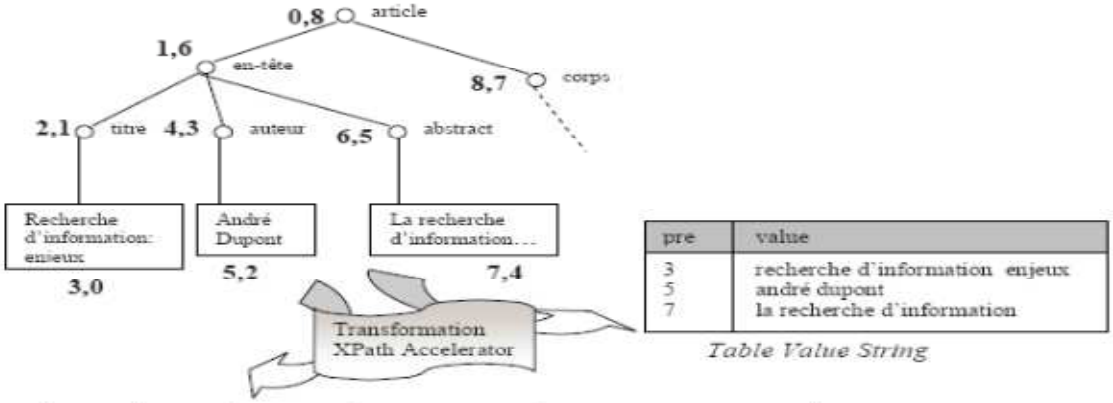
Ce type d'indexation permet de résoudre la difficulté de la technique basée sur des chemins à savoir : retrouver les relations ancêtre-descendant. Dans cette technique, chaque noeud (élément) du graphe représentant le document XML est identifié par un identifiant unique. Les termes sont donc associés à cet identifiant, ce qui permet de localiser de façon précise l'endroit où ces termes sont apparus et de retrouver les relations hiérarchiques entre les éléments.

Les nœuds d'un arbre sont numérotes dans l'index de façon à pouvoir reconstruire la structure arborescente des documents. Parmi les techniques d'indexation, nous citons :

A- Codage Préordre / postordre :

Un identificateur de noeud est une paire ***(pré,post)*** où ***pré*** est la valeur du préordre et ***post*** la valeur postordre du noeud en question. Évidemment cette approche supporte uniquement des bases de données sous forme d'arbre.
Cette approche a été adaptée dans plusieurs systèmes de recherche, parmi lesquels citons les systèmes : *EDGE*, *BINARY* [Florescu, 1999] *et XPATHACCELERATOR* [Grust, 2002] ….

La figure II.6 illustre la meilleure technique d'indexation utilisée qui est Xpath accelerator. En stockant de plus, la dimension du prédécesseur du noeud parent, un champ indiquant la présence d'attributs et le nom de balise de chaque noeud, la navigation devient efficace et il est possible de répondre à des expressions XPath qui n'ont pas pour origine la racine du document.



pre	value
3	recherche d'information enjeux
5	andré dupont
7	la recherche d'information

Table Value String

pre	post	par	att	tag
0	8	NULL	0	article
1	6	0	0	en-tête
2	1	1	0	titre
3	0	2	0	NULL
4	3	1	0	auteur
5	2	4	0	NULL
6	5	1	0	abstract
7	4	6	0	NULL
8	7	0	0	corps

XPath Accelerator table

Figure II.6- Exemple d'indexation basée sur la technique Xpath accelerator [Sauvagnat ,2005]

B- *Codage d'intervalle (*Interval Encoding*):*

Le codage d'intervalle [Li et al, 2001] est basé sur le parcours préordre du graphe de la base de données. Au lieu de la valeur de postordre, chaque noeud est annoté avec la compensation de la plus grande valeur de préordre parmi ses descendants, comme indiqué dans la Figure II.7 (a). De là pour n'importe quel noeud n_1 avec l'ID ***(pre1, size1), size1*** dénote la taille de l'intervalle des IDs de nœuds enjambés par le sous-arbre dont la racine est n_1.Pour éviter l'incrémentation des valeurs de préordre de tous les nœuds des documents après insertion (dans le parcours préordre), [Li et al, 2001] propose de choisir la taille de n'importe quel intervalle plus grande que le nombre réel de nœuds dans le sous-arbre correspondant, une technique appelée ***préordre étendu***. Ainsi l'index supporte quelques mises à jour progressives avant qu'il ne doive être reconstruit à nouveau.

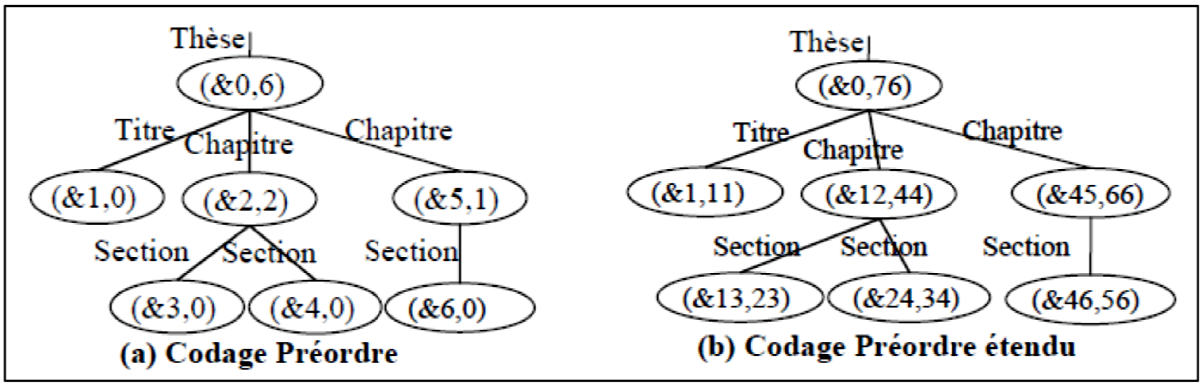


Figure II.7: Interval Encoding [Dahak, 2006]

La Figure II.7.b décrit la même base de données que dans (a), mais avec des intervalles étendus par une valeur de 10. Semblable au codage préordre/postordre, le stockage du niveau dans l'IDs de nœuds permet au codage d'intervalle de choisir des descendants à une distance spécifique d'un noeud ancêtre.

C- Nœuds virtuels (Virtual nodes) :

Les nœuds des documents sont numérotés comme si l'arbre de la base de données était un arbre complet d'arité uniforme ***a***, en réservant un ID bien qu'il n'apparaisse pas physiquement dans la base de données. La Figure II.8 illustre une base de données avec ***a*** = 3.

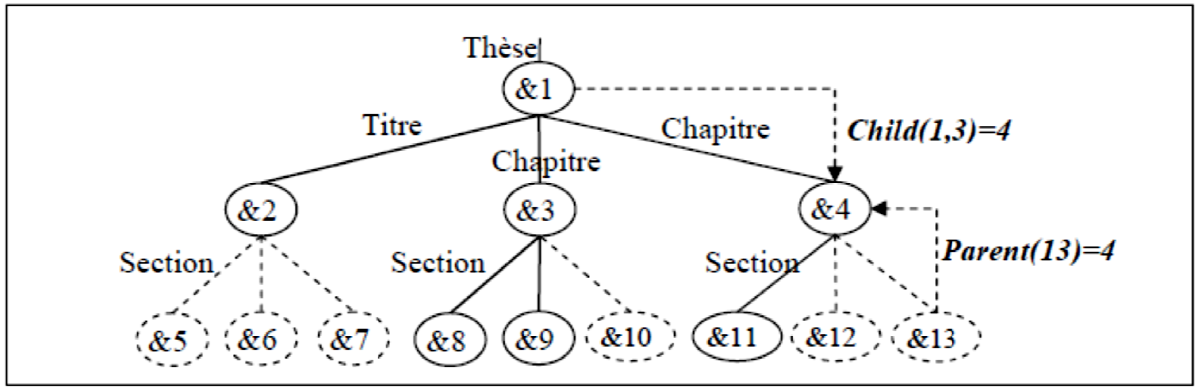


Figure II.8: Codage avec les nœuds virtuels [Dahak, 2006]

Le codage du Noeud Virtuel supporte des mises à jour progressives pour un certain nombre d'insertions de nœuds avant qu'une nouvelle indexation complète ne soit exigée. De plus, les IDs du parent et de l'enfant de n'importe quel noeud peuvent être calculés de son propre identificateur dans un temps constant. Les deux fonctions pour calculer l'ID du parent et pour calculer l'ID du kieme enfant sont définies comme suit

$$parent(i) = \left[\frac{(i-2)}{a} + 1\right] \qquad child(i,k) = a(i-1) + k + 1$$

Formule II.3 : Virtual nodes - Calcul des ID des nœuds

II-5-3 Pondération des termes d'indexation d'un document XML:

Les approches d'indexation structurées extraient les termes d'indexation selon des processus similaires à ceux utilisés en recherche d'information traditionnelle. La pondération de ces termes doit cependant être vue sous un nouvel angle. Alors qu'en recherche d'information traditionnelle, le poids d'un terme cherche à rendre compte de son importance de manière locale au sein du document et de manière globale au sein de la collection, s'ajoute en recherche d'information structurée l'importance du terme au niveau de l'élément qui le contient. Les occurrences des termes ne suivent plus forcément une loi de Zipf [Zip, 1949].Le nombre de répétitions des termes peut être (très) réduit dans les documents semi-structurés et l'utilisation d'***idf*** n'est pas forcément appropriée. L'utilisation d'***ief*** (***Inverse Element Frequency***) a été proposée par de nombreux auteurs
[Wolff et al, 2000]. On trouvera des exemples d'adaptation des formules de pondération traditionnellement utilisées en recherche d'information à la recherche d'information structurée dans [Trotman, 2004].

$$ief(g,t) = \log\left(\frac{|E_g|}{|E_g(t)|}\right)$$

Formule 1 : formule de ief

Où ***t*** est un terme, ***g*** est un groupe spécifique (noeud), |***Eg***| est le nombre total d'éléments dans le groupe ***g*** assigné au noeud de la requête avec un attribut contenant le terme ***t*** et |***Eg(t)***| est le nombre d'éléments dans le group ***g*** contenant ***t***.

Dans [Zargayouna, 2004], le calcul du poids des termes est influencé par le contexte (l'unité d'indexation) dans lequel ils apparaissent. Ce calcul de poids s'inspire de la méthode ***tf-idf*** qu'on applique aux balises. Ainsi, l'auteur définit le ***tf-itdf*** (***Term Frequency - Inverse Tag and Document Frequency***), qui permet de calculer la force discriminatoire d'un terme ***t*** pour une balise ***b*** relative à un document ***d***.

$$TF-ITDF(t,b,d) = TF(t,b,d) \bullet ITF(t,d) \bullet IDF(t,b)$$
$$IDF(t,b) = \log\left(|B|_d \,/\, TagF(t,b)\right)$$
$$ITF(t,d) = \log\left(|D|_b \,/\, DF(t,b)\right)$$

Formule 2 : TF-ITDF

Où ***T*** est l'ensemble de tous les termes qui figurent dans le corpus, ***B*** l'ensemble de tous les modèles de balises, ***D*** l'ensemble de tous les documents du corpus. $|D|_b$ est le nombre total de documents où le modèle de balise ***b*** est présent dans leur structure. $|B|_d$ est le nombre total de balises dans le document ***d***. ***DF(t,b)*** : (***Document Frequency***) est le nombre de documents qui contiennent la balise ***b*** et dans laquelle le mot ***t*** apparaît au moins une fois. ***TagF(t,b)***: (***Tag Frequency***) est le nombre de balises dans le document ***d*** et dans lesquelles le mot ***t*** apparaît au moins une fois.

- Formule 3 : Formule BM25 :

La formule a été adaptée à la recherche dans les documents XML dans
[Géry et al, 2008] pour le calcul du poids d'un terme ti dans un élément ej ainsi :

$$W_{ji} = \frac{tf_{ij} * (K_1 + 1)}{K_1 * \left((1-b) + (b * ndl)\right) + tf_{ji}} * log \frac{N - df_i + 0.5}{df_i + 0.5}$$

$$ndl = \frac{|e_j|}{\Delta|e_j|}$$

Avec :
- |*ej*| est la taille de l'élément *ej* en nombre de termes, Δ|*ej* | est la taille
– *tfii* : la fréquence de *ti* dans *ej* ;
– *N* : le nombre d'éléments dans la collection ;
– *dfi* : le nombre d'éléments qui contiennent le terme *ti* ;
– *ndl* : le ratio entre la taille de *ej* et la taille moyenne des éléments (en nombre de termes) ;
– *k1* et *b* : les paramètres classiques de la BM25standard.

D'autres auteurs considèrent que l'importance d'un terme dans un élément est l'agrégation de l'importance du terme dans le contenu du noeud même, dans le contenu de ses descendants, dans le contenu de ses voisins directs et dans le contenu des nœuds auquel il est relié. Le calcul du poids des termes est effectué au moment de l'indexation.
Plusieurs autres formules de pondérations ont été proposées dans la littérature (voir section II.7)

II-6 les langages d'interrogation :

Les utilisateurs devraient pouvoir exprimer leurs besoins selon deux catégories de Requêtes :
1. les requêtes orientées contenu CO (Content Only pour contenu seulement) : ce sont des requêtes composées de simples mots clés comme en RI. C'est le cas lorsque les utilisateurs n'ont pas d'idée précise de Ce qu'ils recherchent ou n'ont pas de connaissance concernant la structure des documents.
2. Les requêtes CAS (Content And Structure pour orienté contenu et structure) : sont des requêtes composées de contraintes sur le contenu (donc de mots clés) et de contraintes structurelles. C'est le cas lorsque les utilisateurs ont au moins une connaissance partielle de la structure de la collection qu'ils interrogent.

La majorité des langages de requêtes proposés dans la littérature sont issus de la communauté des bases de données. La profusion des différents langages peut être décryptée comme illustré dans la figure II.9.
D'une manière générale, les langages de requêtes doivent supporter à la fois des contraintes portant sur le contenu et sur la structure. De plus l'intégration de fonctions des systèmes de recherche nécessite la prise en compte de requêtes par liste de mots clés du type : *CONTAINS (<element>, collection de mots clés).*

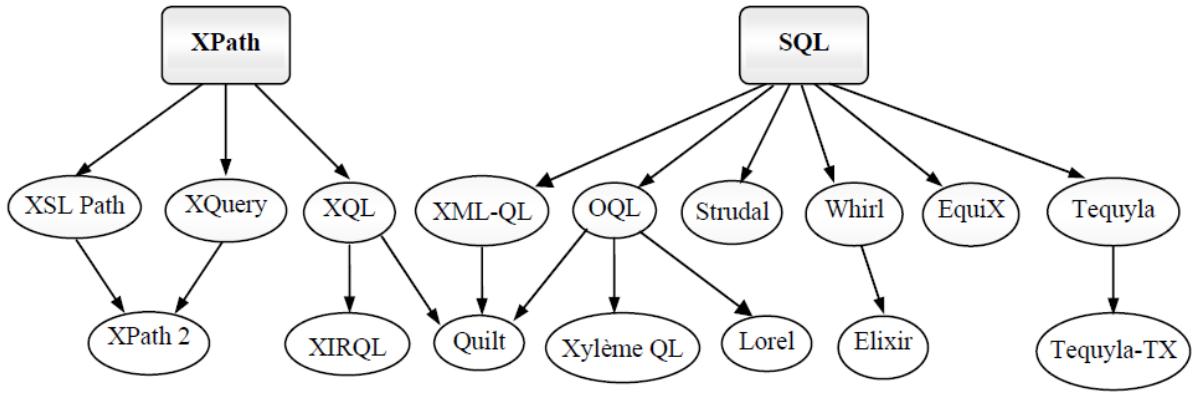


Figure -II.9- Historique des langages d'interrogation XML : Les liens représentent des relations d'inclusion (quelquefois partielle) des différents langages de requête [Piwowarski, 2003 a]

La section suivante traite de la prise en compte de cette structure dans le processus d'appariement

II.7Les modèles de recherche d'information structurée :

Introduction :

Plusieurs auteurs se sont intéressés à la recherche et l'interrogation de documents XML, certains se sont inspirés des modèles de RI existants qu'ils ont adaptés à XML pour tenir compte de l'information structurelle contenue dans les documents XML et des granularités variées de l'information alors que d'autres ont proposés de nouveaux modèles.

Dans certains de ces modèles, la recherche porte sur le contenu (CO pour Content Only) et dans d'autres, elle porte sur le contenu et la structure (CAS pour Content and Structure). Les requêtes portant sur le contenu sont formulées avec de simples mots clés. Les requêtes portant sur la structure contiennent des contraintes sur la structure du document.

Les modèles proposés peuvent être classés selon deux types d'approches permettant d'effectuer l'appariement élément requête :

– les approches de propagation de score(ou propagation de pertinence) : consistent à calculer des valeurs de pertinence pour les différents nœuds feuilles (c'est-à-dire les nœuds contenant du texte). Ces valeurs sont par la suite propagées et agrégées vers les nœuds ancêtres. Cette approche a été utilisée dans beaucoup de travaux de recherche.
– les approches de propagation de termes: consistent à propager le contenu des nœuds feuilles à ses ancêtres, et les scores sont calculés par la suite pour chaque nœud indépendamment. Peu d'auteurs ont utilisé cette approche nous citons par exemple [Fellag, 2006], [Ben Aouicha 2009].

II.7.1-Modèle vectoriel étendu :

Les adaptations proposées pour le modèle vectoriel consistent à mesurer une similarité vectorielle de chaque élément à la requête. Les éléments sont représentés par des vecteurs de termes pondérés. Ils sont renvoyés à l'utilisateur par ordre décroissant de pertinence.

On trouve dans **[Fuller et al, 1993]** une première adaptation du modèle vectoriel.
La mesure de similarité d'un nœud *n* à une requête $q = \{t1, t2, \ldots, tk\}$ est donnée par l'équation suivante :

$$sim(q,n) = \alpha(T) \cdot cosm(q,n) + \sum_{k=1}^{s} \frac{cosm(q,n_k)}{\beta^{k-1}}$$

Où α(T) est un facteur qui prend en compte le type du nœud.
s : est le nombre de nœuds enfants n_k de *n*.
β : est un paramètre permettant d'assurer que le nombre d'enfants n'introduit pas un biais dans la formule.

La fonction *cosm* est définie comme suit :

$$\mathrm{cosm}(q,n) = \sum_{i=1}^{T} \frac{w_i^q * w_i^n}{|n|}$$

Avec :

w_i^q et w_i^n sont respectivement le poids du terme t_i dans la requête q et dans le nœud n.

$|n|$: le nombre de termes dans le nœud n.

La pertinence d'un nœud peut ainsi être calculée à part, puis combinée avec la pertinence des nœuds descendants.

Dans **[Grabs et al, 2002],** les auteurs proposent d'évaluer l'importance d'un terme dans un élément donné en fonction de l'importance du terme dans les éléments du même type.

Lorsque la requête est composée d'une condition sur le type d'un élément (on nommera cat ce type) ainsi que d'une condition sur le contenu de cet élément (requête orientée contenu et structure), la similarité d'un élément ***e*** de type ***cat*** à la requête *q* est calculée selon l'équation suivante:

$$\boldsymbol{RSV}\ (\boldsymbol{e},\boldsymbol{q}) = \sum_{\boldsymbol{t}\in \boldsymbol{terms}(\boldsymbol{q})} \boldsymbol{tf}(\boldsymbol{t},\boldsymbol{e}).\boldsymbol{ief}_{\boldsymbol{cat}}(\boldsymbol{t})^2.\boldsymbol{tf}(\boldsymbol{t},\boldsymbol{q})$$

où ***tf(t, e)*** est la fréquence du terme ***t*** dans l'élément ***e.***

$$\boldsymbol{ief}_{\boldsymbol{cat}} = \boldsymbol{log}\frac{\boldsymbol{N}_{\boldsymbol{cat}}}{\boldsymbol{ef}_{\boldsymbol{cat}}(\boldsymbol{t})}$$

Avec

N_{cat} : le nombre d'éléments du type *cat*.

$ef_{cat}(t)$: la fréquence du terme *t* dans les éléments du type *cat*.

Les requêtes orientées contenu sont quant à elles traitées de la façon suivante :

Soit *SE(e)* l'ensemble des descendants de *e* incluant *e*. $\forall$ *se* $\in$ *SE(e), l*$\in$ *path(e, se)* est une étiquette appartenant au chemin reliant *e* à *se,* c'est-à-dire un type d'élément.

Soit enfin $aw_l \in [0, 1]$ un facteur modélisant l'importance de l'étiquette *l*. La similarité d'un élément *e* à une requête *q* composée de simples mots-clés est définie de la façon suivante :

$$\boldsymbol{RSV}(\boldsymbol{e},\boldsymbol{q}) = \sum_{\boldsymbol{se}\in \boldsymbol{SE(e)}} \sum_{\boldsymbol{t}\in \boldsymbol{terms(q}} \boldsymbol{tf}(\boldsymbol{t},\boldsymbol{se}).\left(\prod_{\boldsymbol{l}\in \boldsymbol{path(e,se)}} \boldsymbol{aw}_{\boldsymbol{l}}\right).\boldsymbol{ief}_{\boldsymbol{cat(se)}}(\boldsymbol{t})^2.\boldsymbol{tf}(\boldsymbol{t},\boldsymbol{q})$$

Le modèle JuruXML [Mass et al, 03] propose d'indexer les éléments selon leur type (un index par type d'élément) et d'appliquer ensuite le modèle vectoriel pour la pondération des éléments. Les requêtes orientées contenu sont évaluées sur chacun des index, et les résultats, qui ont été normalisés, sont ensuite fusionnés afin de fournir à l'utilisateur une liste unique de résultats. Une requête structurée est quant à elle évaluée en trois phases. Tout d'abord, la requête originale est décomposée en un ensemble de conditions de la forme (*chemin, terme*). Ensuite, une correspondance vague entre les chemins est calculée par la fonction cr(c_i^q,c_i^e) :

$$\mathbf{C_r}(c_i^q, c_i^e) = \begin{cases} \frac{1+|c_i^q|}{1+|c_i^e|} & \textbf{si } c_i^q \textbf{ est une sous séquence de } c_i^e \\ \mathbf{0} & \textbf{sinon} \end{cases}$$

Où : C_i^q est la condition de chemin pour un terme *ti* de la requête *q* et C_i^e le XPath de ce terme dans l'élément *e*.
Enfin, on a :

$$RSV(e,q) = \frac{\sum_{(t,c_i^q)\in q} \sum_{(t,c_i^e)\in e} w_q(t) * w_e(t) * cr(c_i^q, c_i^e)}{|q| * |e|}$$

Où $w_q(t)$ et $w_e(t)$ sont respectivement le poids du terme t_i dans la requête *q* et dans l'élément *e*.

Et $|q|$ et $|e|$ sont les nombres de termes dans *q* et *e*.

dans **[Yang et al., 2007]** les auteurs proposent d'étendre le modèle vectoriel pour représenter un document XML par une matrice. Ainsi, un élément (sous arbre) XML et la requête sont représentés respectivement par deux matrices. La Figure II.10 illustre la représentation matricielle d'un document XML. Les valeurs de cette matrice sont les poids des termes dans les différents éléments.

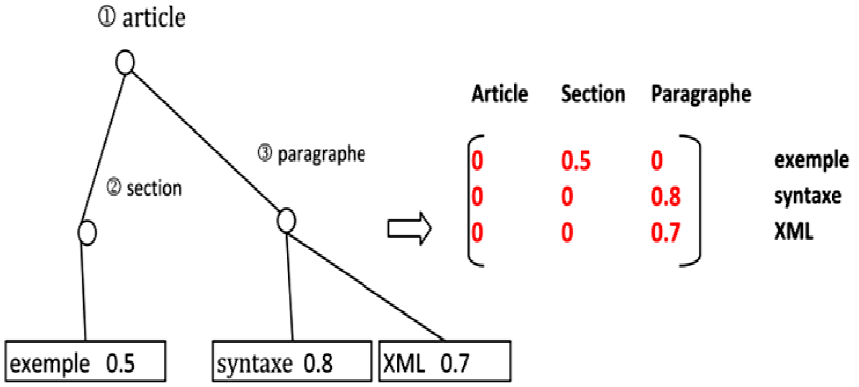


Figure II.10 Représentation matricielle d'un document [Yang et al., 2007]

La fonction de similarité est calculée en effectuant le cosinus entre les deux matrices comme suit :

$$RSV(Q,E) = \frac{M_Q * M_E}{\|M_Q\| * \|M_E\|}$$

II.7.2Le modèle booléen étendu :

Hatano et al dans [Hatano et al, 02] se sont basés sur des approches portant sur l'extension des modèles booléens aux documents structurés grâce aux p-normes.
Une requête est représentée par un vecteur *q* dont les composantes sont un (1) si le terme apparaît dans la requête, zéro (0) sinon.
L'unité d'information à renvoyer à l'utilisateur est définie comme étant tout élément contenant au moins un élément feuille. Pour calculer le score d'un élément vis-à-vis d'une requête, les auteurs propagent des scores, depuis les feuilles de l'arbre documentaire jusqu'à l'élément considéré. Pour ce faire, chaque élément feuille *ej*(*j=1, …, N*)est représenté par un vecteur :

$$F(e_j) = (w_{t_1}^{e_j}, w_{t_2}^{e_j}, ..., w_{t_n}^{e_j})$$

Où *n* représente le nombre de termes indexés dans la collection et W_{ti}^{ej} représente le poids d'un terme t_i dans un élément feuille qui est donné par :

$$w_{t_i}^{e_j} = \frac{tf(t_i, e_j)}{\sum_{j=1}^{N}\sum_{k=1}^{n} tf(t_k, e_j)} . \log \frac{N_d}{df(t_i)}$$

Où : $tf(t_i,e_j)$:est la fréquence du terme t_i dans l'élément feuille e_j .
$tf(t_k ,e_j)$:est la fréquence du terme t_k dans e_j (la double sommation calcule la somme des fréquence de tout les termes de *ej*.
Nd : est le nombre de documents dans la collection.
df(ti) : est le nombre de documents contenant t_i.
ainsi la similarité entre un élément feuille *ej*(représenté par le vecteur *F(ej)*) et une requête *q* est calculée grâce au cosinus :

$$sim(F(e_j), q) = \frac{F(e_j).q}{|F(e_j)|.|q|}$$

Ils utilisent ensuite la p-norme pour calculer le score final d'un élément. Ce score est calculé de manière récursive car il prend en compte le score des sous-éléments :

$$RSV(q,e) = 1 - \sqrt[\frac{1}{p}]{\frac{1}{|enf(e)|}\sum_{é \in enf(e)}(1 - RSV(q,é))^p}$$

Où : *p* est généralement choisi avec une valeur égale à 2.
enf(e) : Nombre de termes de l'élément enfant de *e*.
Lorsqu'un élément *e* est une feuille :

$$RSV(q, e) = sim(F(e_j),q).$$

Dans [Larson ,2002], les auteurs proposent de combiner des méthodes probabilistes utilisant une régression logistique avec une approche basée sur le modèle booléen, pour évaluer la pertinence des documents et des éléments.
La valeur de probabilité de pertinence R d'un composant C (élément) est calculée comme étant le produit des probabilités de la pertinence de C vis-à-vis la requête (Qbool)

présentée par un modèle booléen et de la pertinence de C vis-à-vis la requête (Qprob) présentée par un modèle probabiliste la formule est présenté ci-dessous :

P(R/Q,C)=P(R/Qbool,C)xP(R/Qprob,C)

Cette combinaison permet de restreindre l'ensemble des documents pertinents aux documents ayant une valeur booléenne égale à 1 tout en leur attribuant un rang. Ces deux types d'extension permettent de surmonter les limites des modèles booléens au niveau du tri des résultats.

Dans (Trotman et al., 2003), les auteurs ont proposé d'étendre le modèle booléen avec un nouvel opérateur binaire non commutatif **contains**. Le premier opérande est de type XPath et le second est une expression booléenne. Ce modèle permet aux requêtes d'être complètement spécifiées en termes de contenu et d'information structurelle basée sur le langage d'interrogation XPath. La recherche consiste à extraire le titre et le convertir en requête booléenne, les éléments considérés comme pertinents sont par la suite classés selon la formule okapi BM25.

II.7.3 Modèle probabiliste :

Le modèle probabiliste est étendu pour tenir compte de la structure des documents XML dans les probabilités. Plusieurs approches ont été proposés nous présentons dans ce qui suit les plus utilisés :

II.7.3.1 Le modèle d'inférence probabiliste :

Dans le modèle probabiliste inférentiel, le classement des documents est basé sur la probabilité que le document retrouvé « *d* » implique la requête donnée « *q* »

($P(d \rightarrow q)$).

Pour étendre ce modèle au document XML, les probabilités doivent tenir compte de l'information structurelle.

L'une des approches d'extension du modèle probabiliste inférentiel aux documents structurés se résume en l'utilisation de probabilités conditionnelles de jointure, avec par exemple $P(d|t)$ devenant $P(d|p \text{ contains } t)$, où *d* représente un document ou une partie de document, *t* est un terme et *p* est un chemin dans l'arbre structurel de *d*.

Une autre approche est issue des travaux, **Fuhr & al. [Fuhr & al, 2001][Gövert & al, 2002]** qui ont proposés un modèle probabiliste inférentiel basé sur une méthode d'augmentation (multiplication par un facteur donné). L'approche adopte une indexation par unités disjointes (sauf pour les nœuds feuilles d'une granularité trop fine, dont les termes sont propagés au nœud indexable le plus proche dans la hiérarchie). De plus pour préserver des unités disjointes, on ne peut associer à un nœud que les termes non reliés à ses descendants. Les calculs de pertinence se font comme suit :

-*Dans le cas des requêtes orientées contenu*, le poids de pertinence d'un nœud est calculé par propagation des poids des termes les plus spécifiques dans l'arbre du document. Les

poids sont de plus diminués par multiplication par un facteur d'augmentation défini par les auteurs. Nous illustrerons cette propagation par l'exemple suivant :
Soit la structure de document XML suivante, contenant des termes pondérés :

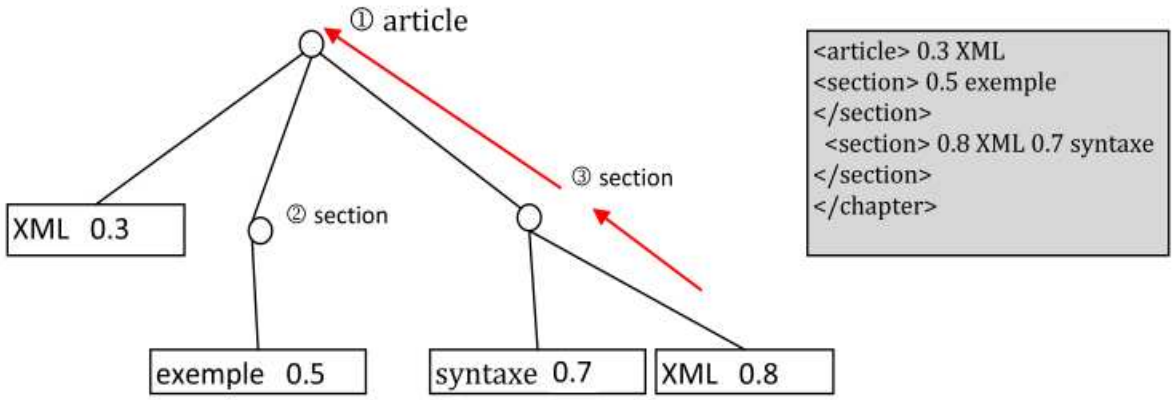


Figure II.11 Modèle d'augmentation [Fuhr et al., 2001]

Et soit à rechercher le terme « XML »
Le nœud article répond à la requête "XML" car il y deux index où le terme "XML" apparait.
En utilisant un facteur d'augmentation égal à 0.6, le poids de la pertinence de l'élément «article » est calculé comme suit :
P([article,XML]) + 0.6*P(section[3],XML) –P([article,XML]) * 0.6*P([section[3],XML)
=0.3+0.6*0.8-0.3*0.6*0.8=0.636

Le nœud **« section »** (ayant une pertinence de 0.8 à la requête) sera donc mieux classé que le nœud « article ».

-*Pour les requêtes orientées contenu et structure*, des probabilités d'apparition de chaque terme de la condition de contenu dans les éléments répondant aux conditions de structure sont calculées, et des sommes pondérées de ces probabilités sont ensuite effectuées. Une adaptation de ce modèle est proposée par [DeCampos et al., 2009].

II.7.3.2 les modèles de langue :

Dans **[Sigurbjaornsson et al., 2003],** les auteurs ont proposé une approche pour la recherche d'information dans des documents XML basée sur les modèles de langue. L'idée de base est de combiner le modèle de langage de l'élément, et de la collection.
Pour estimer les modèles de langage, les auteurs ont proposé d'utiliser l'approche d'indexation basée sur les sous arbres imbriqués. Puis, pour chaque requête (orientée contenu) les éléments sont triés par rapport à la probabilité que le modèle de langage de l'élément génère la requête :

$$P(e|q) \propto P(e) \times P(q|e)$$

avec $P(e)$: la probabilité a priori de l'élément *e*.
$P(q|e)$:la probabilité que l'élément *e* génère la requête .

Les auteurs considèrent l'indépendance entre les termes de la requête, pour une requête composée de k termes $Q = \{t1, t2,\ldots, tk\}$, la probabilité $P(Q\backslash E)$ est calculée selon cette formule :

$$P(Q\backslash E) = \prod_{i=1}^{i=k} (\lambda * P_{mle}(t_i \backslash E) + (1-\lambda) * P_{mle}(t_i \backslash C))$$

C : est la collection.
λ : est le paramètre d'interpolation ou de lissage entre le modèle de l'élément e et celui de la collection,

$P_{mle}(ti\backslash E)$:est la probabilité de ti dans le modèle de langue de l'élément *E*.

$P_{mle}(ti\backslash C)$ est la probabilité de ti dans le modèle de langue de la collection *C*.

Le modèle de langue de la collection et de l'élément est estimé par des statistiques en utilisant les fréquences des termes dans l'élément. L'estimation de la probabilité est calculée en utilisant une fonction de score S(*E*,Q) . Pour chaque élément *E* on a:

$$S(E,Q) = \beta . \log(\sum_{t} tf(t,e)) + \sum_{i=1}^{k} \log\left(1 + \frac{\lambda . tf(t_i, e) . \sum df(t)}{(1-\lambda) df(t_i) . (\sum tf(t,e))}\right)$$

Où *tf(t,e)* :est la fréquence du terme *t* dans l'élément *e* .
df(t) : est le nombre d'éléménts contenant *t* dans toutes 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.

II.7.3.3 les réseaux bayésien :

Plusieurs travaux ont été faits dans le cadre de la recherche d'information basé sur les réseaux bayésien. Ces différentes approches seront détaillées dans le chapitre suivant.

II.7.4 Maximum des poids(le modèle proposé par Moffat et al) :

Ahn & Moffat [**Ahn & Moffat, 2002]** proposent deux méthodes de propagation de score utilisant une combinaison linéaire des scores des enfants appelées « maximum- by-category» et « summation ».

Le calcul du score de la méthode « maximum-by-category » (similarité d'une requête q vis à-vis d'un élément e) est :

$$RSV(q,e) = RSV_0(q,e) + \alpha \max_{e' \in enf(e)} |RSV(q,e')|$$

enf(e) **:**enfant de l'élémént e.

le score le plus élevé des enfants est propagé vers le nœud(élément)parent.

Le RSV peut être négatif ce qui explique le valeur absolue.

α :est une constante avec une valeur de 0.8.

le calcule du score de la méthode summation se fait ainsi :

$$RSV(q,e) = RSV_0(q,e) + \left(\frac{\beta}{|enf(e)|} + \frac{\gamma}{|\{e' \in enf(e) / RSV(q,e') > 0\}|} \right) \sum_{e' \in enf(e)} RSV(q,e')$$

Où |enf(e)|:nombre de termes de l'élément enfant de e.

βet γ sont des constantes (dans leur article β=γ=0.5).

la contribution des enfants va donc être d'autant plus importante que le nombre d'enfant est restreint(lié à β) et que le nombre d'enfants ayant un score supérieur à zéro est important(terme lié à γ).

II.7.5 Entropie et sélection de termes (approche de Cui et al) :

Dans **[Cui& al, 2003],** les auteurs proposent une approche originale : un mot n'apparaîtra qu'une fois sur tout chemin entre un nœud et la racine du document.

Le poids d'un terme t dans un élément e, (w(t,e)) est calculé ainsi :

$$w(t,e) = \log(1 + tf_t(e)) \times entropie(t,e)$$

avec

$$entropie(t,e) = \frac{-\sum_{e' \in enf(e)} tf_t(e') \times \log \frac{tf_t(e')}{tf_t(e)}}{-tf_t(e) \times \log \frac{1}{|enf(e)|}}$$

Pour les feuilles de l'arbre, ils définissent :

$$w(t,e) = \log(1 + tf_t(e)) \log \frac{N}{n_t}$$

La *« remontée »* des termes s'effectue des feuilles de l'arbre jusqu'à la racine. A chaque nœud, si :

$$w(t,e) \geq \underset{e' \in enf(e)}{moyenne}(w(t,e')) + \underset{e' \in enf(e)}{variance}(w(t,e'))$$

Alors le terme *t* est affecté à l'élément *e* et est supprimé pour tous ses descendants. Ce procédé est répété pour tous les éléments jusqu'à la racine.
La pertinence entre un élément *e* et une requête *q* se calcule ainsi :

$$\mathbf{RSV(q,e)}=\sum_{t\in termes(q)} w(t,e)\,x\,log(\frac{N}{n_t})$$

Où :
$tf_t(e)$=fréquence du terme *t* dans l'élément *e*
N : nombre total de documents dans le corpus (collection)
n_t : nombre total de documents contenant le terme *t*.
|*enf(e)*| : nombre de termes de l'enfant *e*.

II.7.6 Le modèle XFIRM (XML Flexible Information Retrieval Model : modèle flexible pour la recherche dans des documents semi-structurés) [Sauvagnat, 2005]

Le but de ce modèle est de répondre au mieux au critère de spécificité et exhaustivité demandé par l'utilisateur. Ce modèle évalue les requêtes grâce à une technique de propagation de la pertinence des nœuds dans l'arbre des documents.

La démarche consiste à calculer un poids pour les nœuds feuilles de l'arbre, d'utiliser ce calcul pour la mesure de la pertinence des nœuds parents, en tenant compte de la distance qui sépare ce nœud ainsi considéré du nœud feuille. La pertinence d'un nœud *n* notée P_n est calculée comme suit :

$$\mathbf{Pn} = |\boldsymbol{F_n^P}| \sum_{nf_k \epsilon F_n} \alpha^{dist(n,nf_k)-1} * RSV(q, nf_k)$$

Avec :
-$|F_n^p|$: nombre de nœuds feuilles descendants de *n* ayant un score non nul.
- *Fn* : est le nombre total de nœuds feuilles nf_k descendants de *n*.
- α∈[0,1] est un paramètre qui quantifie l'importance de la structure dans l'évaluation du score de pertinence.

- *dist(n,nfk)* : représente la distance entre le nœud *n* et le nœud feuille nf_k (nombre d'arc séparant les deux nœuds)dans l'arbre du document
- *RSV (q, nf_k)* : poids du nœud feuille *nfk* dans le document. Il est exprimé selon la formule suivante :

$$RSV(q, nf_k) = \sum_{i=1}^{N} W_i^q * W_i^{nf}$$

Avec w_i^q : le poids du terme *i* dans la requête *q* ($w_i^q = tf_i^q$).
w_i^{nf} : Le poids du terme *i* dans nœud feuille *nf* ($w_i^{nf} = tf_i^{nf} * ief_i * idf_i$).
Où tf_i^q et tf_i^{nf} sont respectivement la fréquence du terme *i* dans la requête *q* et dans le nœud feuille *nf*.
ief_i = $\log(\frac{|Fc|}{|nfi|})$ permet d'évaluer l'importance du terme i dans la collection de nœuds feuilles.
où |*Fc*| est le nombre total de nœuds feuilles de la collection,
et |nfi| est le nombre de nœuds feuilles contenant *i*.

$idfi = \log(\frac{|D|}{|d_i|})$ permet d'évaluer l'importance du terme *i* dans la collection de documents, où |*D*| est le nombre total de documents de la collection.
|*di*| est le nombre de documents contenant le terme *i*.

II.7.7L'approche proposé par Trotman :

[Trotman ,2005] a proposé d'attribuer des degrés d'importance pour chaque structure du document et de remplacer le *tf* par la fréquence du terme en tenant compte du poids de la structure.
Dans le modèle vectoriel une telle approche se traduit dans le calcul de fréquence d'un terme en remplaçant la formule :

$$tf_{id} = \sum_{p=1}^{n} tf_{ipd}$$

par la formule :

$$tf'_{id} = \sum_{p=1}^{n} (C_p \times tf_{ipd})$$

Ou tf_{ipd} : est le nombre d'occurrences du terme *t* à la position *p* du document *d*.
Cp : est le poids de chaque structure du document qui doit être fixé.
Cette méthode d'indexation et de recherche des données structurées permet de donner un poids aux structures. Un algorithme génétique est employé pour l'apprentissage des poids.

II.7.6 Le système GPX :

Le système GPX (*Gardens Point XML IR*) dans [s,Geva, 2005]et [S,Geva,2006] consiste à calculer des scores de pertinence pour les nœuds feuilles, et ensuite, propager ces scores vers les nœuds internes.

– Calcul du score des nœuds feuilles : Les fréquences des termes dans les nœuds feuilles et dans la collection sont utilisées avec un paramètre permettant de privilégier les éléments ayant des termes multiples de la requête. La formule utilisée dans le calcul des scores des nœuds feuilles pénalise les éléments ayant des termes très fréquents dans la collection et récompense les éléments ayant le plus de termes uniques de la requête.

La formule utilisée pour le calcule des scores des nœuds feuilles *nf* est la suivante :

$$\boldsymbol{RSV\ (q,\ nf)=K^{n-1}\sum_{i=1}^{n}\frac{t_i}{f_i}}$$

Avec *n* est le nombre de termes uniques de la requête *q* dans l'élément, *K* est une constante (*K = 5*). K^{n-1} permet d'augmenter le score des éléments ayant des termes distincts multiples de la requête.

ti est la fréquence du ième terme de la requête dans l'élément *nf*.

fi est la fréquence du ième terme de la requête *q* dans la collection.

– **Propagation de la pertinence des nœuds feuilles:** Une fois que les scores de tous les éléments textuels de la collection sont calculés, ces scores sont propagés vers le haut d'une manière récursive dans l'arbre de document XML comme suit :

$$\boldsymbol{RSV\ (q,\ n)=D\ (n)\sum_{l=1}^{n} RSV(q, nc_l)}$$

Avec

$RSV\ (q,\ nc_l)$: est le score de pertinence de $l^{ème}$ élément fils.

n est le nombre d'éléments fils, $D(\boldsymbol{n})$ =0.49 si $n = 1$ et 0.99 sinon.

La valeur du facteur *D* dépend du nombre de fils pertinents contenus dans le nœud interne *n*. Si le nœud interne possède un seul fils pertinent, la constante *D* est 0.49. Un élément ayant un seul fils pertinent est classé après son fils. Cependant, si l'élément possède plusieurs fils pertinents le facteur *D* est 0.99. Un élément possédant plusieurs fils pertinents est classé avant tous ses descendants.

II.7.7L'approche proposée par [Fellag ,2006] :

Ce modèle évalue les requêtes grâce à une technique de propagation des termes. Tout d'abord un score entre un nœud *ni* et la requête *q* est calculé comme suit :

$$\mathbf{RSV(q,n_i)=\sum_{k=1}^{M} p_{qk} \times p_{nik}}$$

Avec

P_{qk} et P_{nik} sont respectivement le poids du terme *k* dans la requête *q* et le nœud *ni*.

Le poids *Pk* d'un terme *k* dans les nœuds feuilles, et dans la requête est calculé comme suit :

$$P_k = tf_k \times Idf_k \times ief_k \qquad \textbf{(1)}$$

Avec

tfk : term frequency (fréquence d'un terme *k* dans le nœud feuille ou dans la requête)

Idfk : Inverted document frequency = fréquence inverse de document=$\boldsymbol{log\frac{N}{n}}$

N : Nombre total de documents dans la collection

n : Nombre de documents contenant le terme *k*

ief k: Inverted element frequency = fréquence inverse d'éléments qui est une adaptation de *Idf* à la granularité de l'information à traiter qui n'est plus le document mais l'élément ; elle est calculé comme suit :

$$\log\left(\frac{Ne}{ne}+\alpha\right) \text{ avec } 0.5 \le \alpha \le 1$$

Ne : Nombre total de nœuds feuilles dans le document.

ne : Nombre de nœuds feuilles contenant le terme *k* dans le document.

Ensuite afin d'identifier la partie du document qui répond le mieux à la requête utilisateur, une méthode de propagation des termes, en partant des nœuds feuilles jusqu'à la racine du document a été proposé. Cette méthode consiste à fixer un seuil en le nombre minimal des termes qu'un nœud doit avoir pour qu'il soit considéré comme informatif.

La remonté des termes s'effectue des feuilles de l'arbre jusqu'à la racine. Deux cas peuvent se présenté :

1-Si le nœud *e* possède *un seul* nœud fils *e'*. Soit *t* un terme du nœud *e'*.

t peut être remonté vers *e*, s'il vérifie la condition suivante :

$$\mathbf{P_{moy} \le P(t,e') \le P_{max}}$$

Avec :

$$P_{moy} : \mathbf{P_{moy}} = \frac{\sum_{i=1}^{Nte'} \mathbf{P(t_i,e')}}{\mathbf{Nte'}}$$

P_{moy} : le poids moyen des termes dans *e'*

P_{max} : le poids maximum des termes dans *e'*

$N_{te'}$: nombre de termes dans le nœud *e'*

P(t,e') le poids du terme *t* dans le nœud *e'*(calculé avec la formule 1).

Le terme *t* sera supprimé du nœud fils *e'* et remonté vers le nœud père *e* en conservant son poids, donc : **P(t,e)= P(t, e')**

1-si le nœud *e* possède *plusieurs* nœuds fils *e'*.Soit *t* un terme d'un nœud fils *e'* de *e*.
t peut être remonté vers *e*, si *t* existe dans au moins un nœud frère de *e'* et si la moyenne du poids de *t* dans les nœuds fils de *e* ou il apparaît, vérifie la condition suivante :

$$P_{moy} \leq \underset{e' \in enf(e)}{moy}(P(t,e')) \leq P_{max}$$

Avec

$$P_{moy} = \frac{\sum_{e' \in enf(e)} \sum_{i=1}^{Nte'} p(t_i, e')}{Nt}$$

P_{moy} : le poids moyen des termes dans les nœuds *e'* fils de *e*
Nt : nombre de termes dans tous les nœuds *e'* enfants de *e*
Nte' : nombre de termes dans le nœud *e'*

$$\underset{e' \in enf(e)}{moy}\left(P(t,e')\right) = \frac{\sum_{e' \in enf(e) / t \in e'} P(t,e')}{Ne'}$$

Ne' : nombre de nœuds *e'* contenant le terme *t.*
Pmax : le poids maximum des termes dans tous les nœuds *e'* enfants de *e.*
Le terme *t* sera supprimé des nœuds fils *e',* et remonté vers le nœud père *e,* et son poids dans *e* sera :

$$P(t,e) = \underset{e' \in enf(e)}{moy}(P(t,e'))$$

II.7.8 Le modèle XIVIR [BenAouicha, 2009]: (*XML Information retrieval based on Virtual links*) :

Ce modèle permet la recherche d'information dans des documents XML par la structure, par le contenu et par la combinaison des deux.

a- Recherche par le contenu :

Pour propager le texte se situant dans un noeud feuille à ces ancêtres, deux manières sont proposées :

a.1La propagation de texte par profondeur : qui consiste à traduire le contenu de chaque noeud feuille par un ensemble de termes pondérés, ceux-ci seront propagés vers les ancêtres de ce noeud tout en diminuant leurs poids en fonction de la distance parcourue au moment de la propagation. Le poids d'un terme *t* dans un nœud *n* est calculé comme suit :

$$w(t,n) = ief_t \times \sum_{n \to c_1 \to c_2 \dots c_k} \frac{tf(t,c_k)}{d(n,c_k)+1}$$

Où **w(t, n)** est le poids du terme t dans le nœud n,

tf(t, ck) est la fréquence du terme *t* dans le nœud *ck*

$\boldsymbol{ief_t}$ est inversement proportionnel au nombre de nœuds de la collection contenant le terme *t* est calculé comme suit : $\boldsymbol{ief_i}=\frac{\boldsymbol{N}}{\boldsymbol{N_t}}$

Où *N* est le nombre d'éléments dans la collection

N_t est le nombre d'éléments contenant le terme t.

$n \to c1 \to c2 \dots ck$ est une branche de l'arbre en question

$d(n,c_k)$: représente la distance entre les nœuds *n* et c_k calculé en nombre d'arcs les séparant

a.2La propagation du texte par la profondeur et la largeur : est réalisée en fonction de la distance qui sépare l'élément feuille qui contient du texte et l'élément noeud interne qui est censé recevoir le texte. Le facteur de propagation est calculé en fonction de cette distance. Le poids d'un terme *t* dans un nœud *n* est calculé comme suit :

$$w(t,n) = ief_t \times \sum_{n \to c_1 \to c_2 \dots c_k} \frac{tf(t,c_k)}{\prod_{e \in \{n,c_1,c_2 \dots c_{k-1}\}} |Children(e)|}$$

Avec |Children(e)| est le nombre total des nœuds fils de l'élément *e*.

Pour la recherche par contenu, le score relatif aux contenus entre un nœud *n′*de la requête et un nœud *n* d'un document, (*RSVc*), est calculé comme suit :

$$\boldsymbol{rsv_c(n,n') = \sum_{p \in t \cap t'} w(p,n)}\textbf{x w(p,n')}$$

où *n* est un nœud dans le document.

n′ est un nœud de la requête.

w(p,n) : le poids du terme *p* dans le nœud du document.

W(p, n′) : le poids du terme *p* dans le nœud de la requête.

b-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 :

$$\omega = \exp(\lambda * (1 - d(A, B)))$$

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 :

$$RSV_s = \sum_{A_q \xrightarrow{w_q} B_q \in E_q \equiv A_d \xrightarrow{w_d} B_d \in E_d} w_q * w_d$$

où Eq (resp. Ed) est l'ensemble de tous les chemins pondérés de la requête (resp. du document).

c- Recherche par le contenu et la structure : Le score final de l'élément XML vis à vis de la requête est calculé en combinant les scores estimés selon les deux critères contenu et structure comme suit :

$$\mathbf{Rsv\ (n, q) = rsv\ (n, q) = \alpha\ x rsv_c(n, ń) + (1 - \alpha)\ x rsv_s(n, ń)}$$

où q est la requête et $\alpha \in [0, 1]$ est un paramètre permettant de renforcer la recherche selon le contenu ou la structure il est définit soit par expérimentation ou bien par l'utilisateur.

II.7.8 Conclusion :

Nous venons de présenter un certain nombre de travaux qui ont tenté d'adapter les modèles de la RI standard, à la RI structurée. Une des adaptations a été la prise en compte de la nouvelle granularité de l'information dans le calcul de pondération en utilisant Ief (Inverse Element Frequency) [Grabs & Scheck, 2002] [Sauvagnat,2005].

Les autres adaptations sont spécifiques au modèle pris en compte par exemple définition d'une probabilité tenant compte de la présence d'un terme dans une structure donnée d'un document, pour le modèle probabiliste dans [Fuhr& al, 2001].

Dans le but de retourner le fragment de document répondant de manière pertinente à la requête utilisateur, le score de pertinence d'un nœud (élément) est calculé.
Pour ce faire, certaines approches, adoptent la méthode de propagation des termes [Cui& al, 2003], d'autres, la méthode de propagation de pertinence (ou de score) [Ahn & Moffat, 2002][Sauvagnat, 2005], ou encore la méthode propagation de poids [Fuhr & al, 2001][Gövert &al, 2002].
Il en découle que, quelque soit l'approche adoptée, le score de pertinence d'un nœud dépend fortement du score de ses descendants.
La section suivante est consacrée à la présentation de la démarche d'évaluation des systèmes de RI structurée, en l'occurrence INEX.

II.8. La Campagne d'Evaluation INEX :

INEX (INitiative for the Evaluation of XML Retrieval) est considérée comme étant la seule campagne d'évaluation des SRI dans des documents XML (depuis 2002 à ce jour). L'évaluation de l'efficacité des SRI dans des documents XML nécessite une collection de test (documents XML, requêtes et jugements de pertinence), cette collection est fournie aux différents participants pour arriver à effectuer un classement de leurs systèmes au niveau de la campagne.
Dans la suite, nous décrivons brièvement le corpus INEX, les requêtes, les tâches, les jugements de pertinence et les mesures d'évaluation.

II.8.1Description des campagnes INEX 2002 à 2006 :

II.8.1. 1 La Collection de test :

Les collections de test préparées dans le cadre d'INEX ne cessent d'évoluer dans le but d'améliorer la qualité de l'évaluation. De 2002 à 2004 la collection de documents était composée d'articles scientifiques provenant de *l'IEEE Computer Society*, balisés au format *XML*. La collection, d'environ *500 Mo*, contenait plus de *12000* articles, publiés de *1995* à *2002*, et provenant de *18* magazines ou revues différents.
Les articles sont généralement structurés de la façon suivante :
- Chaque article est composé d'un en-tête (<fm>), un corps (<body>) et d'annexe <bm>.

Un élément, parmi ceux cités est conçu de la manière suivante :
- Le corps est composé de sections (<sec>)
- Une section est composée de paragraphes (<p>)
- Les annexes sont composées de références bibliographiques (<bibl>) et de curriculum vitae (<vt>)

Un article moyen est composé d'environ 1500 éléments, et la profondeur moyenne des documents est de 6.9 (nombre de niveaux). Au total, la collection contient 8 millions de nœuds et 192 balises différentes.
A partir de 2006, la collection de base utilisée pour les tests étant la collection *Wikipedia*, qui est 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 multimédia, elle contient environ 246.730 images.

II.8.1.2 Les requêtes *(Topic) :*

Elles sont créées par des différents participants et représentent les demandes de l'utilisateur sur la collection. Les requêtes se divisent en deux catégories principales :

- **Les CO (Content Only) :** Se sont des requêtes en langage naturel formulé avec de simples mots clés. Les mots-clés de ces requêtes sont reliés par (+) pour impliquer l'obligation de présence du mot et (-) pour impliquer l'interdiction de sa présence, comme le montre l'exemple suivant :

```
<inex_topic topic_id="98" query type="CO">

<title> "Information Exchange" +"XML" "Information Integration"</title>

<description> How to use XML to solve the information exchange (information
integration) problem, especially in heterogeneous data sources ? </description>

<narrative> Relevant documents/components must talk about techniques of
using XML to solve information exchange (information integration) among
heterogeneous data sources where the structures of participating data sources
are different although they might use the same ontologies about the same
content. </narrative>

<keywords> Information exchange, XML, information integration,
heterogeneous data sources </keywords>
</inex topic>
```

Figure II-12: Exemple de requête CO, issue du jeu de test 2003

- **Les CAS (Content And Structure) :**

Ces requêtes contiennent de plus des contraintes sur la structure des documents. Pour chaque Topic, différents champs permettent d'expliciter le besoin de l'utilisateur
- le champ Title : donne la définition de la forme générale (CO)
- le champ Keywords : donne l'ensemble des mots clés de la requête qui ont permit l'exploration du corpus.
- Les champs Description et Narrative : expression en langage naturel des intentions de l'utilisateur.
- Le champ Castitle pour la forme structurée CAS.

Et voici un exemple de requête orientée structure et contenu :

```
<inex_topic topic_id="205" query_type="CO+S" ct_no="12">
<InitialTopicStatement>McLuhan</InitialTopicStatement>
<title>marshall mcluhan</title>
<castitle>//bdy//*[about(., "Marshall McLuhan")]</castitle>
<description>
Find information about the relevance of Marshall McLuhan's ideas for current
digital technologies.
</description>
<narrative>
I am writing an essay on the inuence of new media icon Marshall McLuhan
on digital technologies. I'm seeking information describing how McLuhan's views
have inuenced current digital technologies. To be relevant, a retrieved item should
discuss some aspect of Marshall McLuhan's visionary ideas or famous one-liners
in the context of current digital technologies. Retrieved elements that merely cite
some of McLuhan's work are non-relevant, as are elements that discuss ideas not
originating from McLuhan.
</narrative>
```

Figure II-13 : Exemple de requête CAS (Topic 205 d'INEX 2005)

II.8.1.3 Les tâches:

II.8.1.3.1 la tâche ad hoc :

La campagne *INEX* regroupe plusieurs tâches dévaluation. Parmi ces tâches figure la tâche *adhoc*. Elle est considérée comme une simulation de l'utilisation d'une bibliothèque électronique *(Digital Library)* composée de documents XML, qui est interrogé par des requêtes utilisateur conçues dans la compagne et portent sur le contenu et la structure.
La tâche ad-hoc est à son tour composée de sous-tâches divisées selon soit :

- Le type de requêtes : on distingue :

a- La tâche CO: (Content Only task):
Sert à répondre aux requêtes utilisateur de type CO par des granules d'information XML. Dans cette tâche, aucune indication de structure n'aide les SRI à savoir le type d'unité à retourner.

b- La tâche SCAS : (Strict Content And Structure task) :
Répondre aux requêtes CAS avec des granules XML de manière stricte, ie : obéir aux exigences de structure et contenu à la fois, dont le champ Title des requêtes est basé sur une syntaxe XPath.

c- La tâche VCAS : (Vague Content And Structure task) :
Répondre aux requêtes CAS de manière vague. ie : avec des granules satisfaisant globalement les requêtes.

- Ou bien la stratégie de recherche, c'est à dire le critère sur lequel est jugée la performance d'un système. On distingue trois sous-tâches :

1- La tâche Thorough: dans laquelle on suppose qu'un utilisateur préfère retrouver tous les éléments fortement pertinents.
2-La tâche Focused : dans laquelle on suppose qu'un utilisateur préfère ne pas avoir d'éléments imbriqués dans ses réponses.
3-La tâche Fetch and Browse : appelé aussi All in Context, qui 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 donnée.
En 2006 une nouvelle tâche appelé **Best in Context** a été définie. Cette tâche permet d'évaluer les meilleurs points d'entrée dans un article donnée.

II.8.1.3.2Autres tâches:

En 2004, quatre nouvelles tâches ont été proposées:
- La tâche « Relevance Feedback », usage des exigences de structure et contenu pour la reformulation des requêtes.
- La tâche « Language Natural », formulation des requêtes en langage naturel.
- La tâche « Interactive », étude du comportement utilisateur face aux corpus XML pour cerner son besoin.
-La tâche hétérogène, proposé aux participants de nouvelles collections pour qu'ils développent des approches indépendantes des DTDs.

II.8.1.4 L'évaluation :

Pour juger la pertinence des documents vis-à-vis des requêtes, une échelle de pertinence à deux dimensions a été proposée en 2002 basé sur le degré de pertinence et de couverture, qui ont été remplacée par les notions d'exhaustivité et de spécificité, depuis 2003. Telles que :

1) la dimension d'exhaustivité :

Décrit jusqu'à quel point un élément discute du sujet de la requête. Pour cela, 4 niveaux sont définis sur cette échelle :
- *Pas exhaustif* : il ne traite pas du tout du sujet de la requête.
- *Marginalement exhaustif* : il traite peu d'aspects du sujet de la requête.
- *Assez exhaustif* : il traite de nombreux aspects du sujet de la requête.
- *Très exhaustif* : il traite la plus part ou tout les aspects du sujet de la requête.

2) la dimension de spécificité :

La dimension de spécificité décrit jusqu'à quel point un élément se focalise sur le sujet de la requête, ie : la couverture du document/élément au sujet et elle est aussi mesurée sous quatre niveaux, l'élément peut être à :
- *Pas de couverture* : le thème traité n'a rien à avoir avec celui de la requête, ie : n'est pas du tout spécifique.
- *Couverture trop large* : ou marginalement spécifique où le thème de la requête est traité exactement dans un sous élément.

- *Petite couverture* : si l'élément renvoyé contient juste une partie de l'information pertinente.
- *Couverture exacte* : ou l'élément est très spécifique et le seul sujet qui traite est celui de la requête.

L'usage de l'échelle à deux dimensions, est impliqué par le besoin de mesurer la pertinence d'un élément par rapport à son descendant, et lorsqu'un élément n'est pas pertinent, il n'a pas de couverture et inversement.

❖ **Mesures d'évaluation :**

Les mesures qui ont été proposées par la compagne d'évaluation INEX étendent les mesures traditionnelles utilisées dans la RI. Le but est de prendre en considération la nature structurée des documents XML. Plusieurs mesures ont étés proposées pour évaluer les différents critères d'un système de recherche.

Dans la suite, nous décrivons quelques une des mesures les plus connues proposées dans INEX.

✓ **Mesure de precal :**

Elle a été utilisée lors de la compagnie d'évaluation 2002 pour définir la probabilité qu'un élément retrouvé et retourné à l'utilisateur soit pertinent, est calculée par :

$$\mathrm{P(pert/retr)(x)}\frac{\mathrm{x.n}}{\mathrm{x.n} + \mathrm{esl_{x.n}}}$$

Tel que : *pert* : document *x* pertinent.

retr : document retrouvé.

eslx.n : nombre attendu d'éléments non pertinents retrouvés jusqu'à ce qu'un point de rappel *x* soit atteint.

n : le nombre de documents pertinents dans la collection par rapport à une certaine requête.

L'inconvénient majeur de la métrique INEX 2002 est qu'elle ignore l'imbrication (*Overlap*) inhérente aux éléments XML et évalue le retour d'un élément pertinent sans prendre en compte le fait qu'il ait été déjà peut-être vu entièrement ou en partie par l'utilisateur. Par exemple, un premier système renvoyant une section pertinente et un de ses paragraphes pertinents obtient les mêmes performances qu'un second système renvoyant deux éléments pertinents non imbriqués.

✓ **La mesure INEX 2003 (dite inex-ng) :**

Dans INEX 2003, une nouvelle mesure est proposée. Cette mesure incorpore la taille des éléments et le concept d'imbrication dans les mesures de rappel et de précision comme donné dans les équations ci-dessous :

$$rappel_o = \frac{\sum_{i=1}^{k} e(c_i).\frac{|c_i'|}{|c_i|}}{\sum_{i=1}^{N} e(c_i)}$$

$$precision_o = \frac{\sum_{i=1}^{k} s(c_i).|c_i'|}{\sum_{i=1}^{k} |c_i'|}$$

Où les éléments c_1, ..., c_k forment une liste triée de résultats,
N est le nombre total d'éléments dans la collection,
e(*ci*) et *s*(*ci*) sont les valeurs d'exhaustivité et de spécificité de l'élément *ci*,
|*ci*| est la taille de l'élément
|*c'i*| est la taille d'un élément c' qui n'a pas été précédemment vu par l'utilisateur.

Jusqu'à 2004, les seules mesures appliquées dans l'évaluation des SRI étaient le rappel et la précision, par la suite dans les compagnies INEX 2005 et 2006 d'autres mesures ont été définies pour mettre une meilleure évaluation des SRI en RI structurée [Xavier Tannier]

✓ **Le gain cumule (xCG) :**
Cumulation des scores de pertinence des éléments de la liste des résultats. Etant donnée une liste d'éléments triée par ordre décroissant dans laquelle, les éléments sont présentés par leurs scores de pertinence :

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

i : le rang de l'élément dans la liste
xCG(i) : somme des scores de pertinence des documents j (j = 1,i)
xG(j) : le score du document de rang j
Après avoir calculé le gain cumule des éléments, pour chaque requête on calcule un vecteur de gain idéal xCGI à partir de la base de rappel, et le xCG peut être alors comparé au xCI avec
le nxCG :

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

tel que : pour l'élément de rang i, le score de pertinence cumulé acquis sur le score de pertinence idéale voulu nous donne une valeur de norme comprise entre 0 et 1 tel que :
si nxCG(i)⟶ 0 élément non pertinent
si nxCG(i)⟶ 1 élément pertinent
Et il reflète le gain relatif que l'utilisateur accumule jusqu'à ce rang si le système avait produit une liste triée optimale.

✓ **L'effort précision (ep) :**
Elle représente l'effort (en nombre de liens à visiter) qu'un utilisateur doit fournir pour parvenir à un gain donné *r*, *esystem* (respectivement *eideal*) est le rang auquel le gain *r* est atteint par le système (respectivement par la liste optimale).
Cette mesure dépend du gain, car elle est calculée par :

$$\mathrm{ep(r)} = \frac{\mathrm{e_{idéal}}}{\mathrm{e_{system}}}$$

Tel que : *r* : c'est le gain
e : le rang correspondant au gain
eidéal : le rang auquel le gain est idéal
esystem : le rang auquel le gain est celui retourné par le système évalué.

❖ **Jugement de pertinence :**

Pour juger la pertinence d'un SRI, il faut l'évaluer suivant les mesures d'évaluation conçues dans les compagnies. Mesurer la performance d'un SRI structuré, revient à mesurer sa capacité de retrouver et restituer les documents à la fois *exhaustifs* et *spécifiques* à la requête.

II.8.2INEX 2007 : [INEX 2007]

Le changement principal dans INEX 2007 concerne la permission de retourner des parties arbitraires d'un document et l'évaluation de la pertinence d'un texte d'un élément en fournissant des requêtes diverses telles que :

```
<inex_topic topic_id="414" ct_no="3">
  <title>hip hop beat</title>
  <castitle>//*[about(., hip hop beat)]</castitle>
  <description>what is a hip hop beat?</description>
  <narrative>
    To solve an argument with a friend about hip hop music and beats, I
    want to learn all there is to know about hip hop beats. I want to know
    what is meant by hip hop beats, what is considered a hip hop beat,
    what distinguishes a hip hop beat from other beats, when it was
    introduced and by whom. I consider elements relevant if they
    specifically mention beats or rythm. Any element mentioning hip hop
    music or style but doesn't discuss abything about beats or rythm is
    considered not relevant. Also, elements discussing beats and rythm,
    but not hip hop music in particular, are considered not relevant.
  </narrative>
</inex_topic>
```

Figure II-14 : Exemple d'une requête INEX 2007

L'évaluation s'effectue par l'étude des trois tâches ad hoc suivantes :

II.8.2.1 Focused tasck (la tâche concentrée) :

Demande aux systèmes participants de renvoyer une liste triée de tout les éléments ou unités d'informations sans chevauchement, par exemple dans le cas du renvoie des éléments XML, un paragraphe et sa section conteneur ne devraient pas être renvoyés à la fois. Pour cette tâche, à partir de toutes les parties pertinentes estimées pour un document, les systèmes participants doivent choisir les éléments non-chevauchants qui représentent les unités les plus appropriées à la recherche.

II.8.2.2 In context tasck (tâche de mise en contexte) :

Elle correspond à la tâche « User » dont les réponses focalisées sur la recherche sont groupés par document dans leur ordre original et permettant à l'utilisateur d'y accéder via d'autre moyens de navigation. Ceci, en supposant que l'utilisateur considère le document comme l'unité de recherche la plus naturelle et souhaite avoir un aperçu de celle-ci. Cette tâche est composée de deux sous tâches principales :

- **Relvant in context :** (appropriée dans le contexte), demande aux systèmes le renvoi des parties non chevauchées des documents (éléments XML ou passages) groupés par documents où elles sont contenues.
- **Best in context :** (mieux dans le contexte), demande aux SRI de renvoyer une simple unité XML par document, qui doit correspondre au meilleur point d'entrée pour commencer la lecture du texte approprié.

II.8.2.3 L'évaluation de pertinence :

Pour chaque partie pertinente d'un document (passage ou élément XML), l'évaluation de pertinence enregistre la taille du texte mis en évidence contenu dans cette partie aussi bien que la partie texte de tout le document. Dans cette procédure, il est demandé de mettre en évidence les phrases représentants l'information pertinente dans un ensemble de documents XML de la collection Wikipedia utilisée. Un programme d'évaluation calcule ensuite la pertinence des parties jugées du document (y compris le document tout entier), ainsi les valeurs de pertinence associées aux parties sont triées dans une échelle continue de 0 à 1 où :
0 correspond à une partie documentaire ne contenant aucune information pertinente,
1 correspond à une partie pleine d'information pertinente et enregistrer la taille du texte pertinent correspondant.

❖ **Les mesures d'évaluation :**

Depuis 2007, les mesures officielles sont basées sur l'interpolation de Rappel/Précision sur 101 niveaux. Soit *pr* la partie (élément XML ou passage) correspondante au rang *r* de la liste triée des parties $\mathcal{R}$ d'un document retournée par un SRI :
Avec $|\mathcal{R}| = 1500$ éléments ou passages. Et soient *rsize*(*pr*) la quantité du texte pertinent contenu dans *pr* (si pas de texte pertinent, *rsize*(*pr*) = 0 : mise en évidence de la pertinence du texte), *Trel* la taille totale (par nombre de caractères) d'un texte pertinent répondant à une requête INEX 2007 et *size*(*pr*) la taille de *pr* en nombre de caractères.

• Pour la tâche concentrée, les systèmes sont demandés de retourner une liste triée des parties-document par leurs valeurs estimées de pertinence et de même décider quelles sont les moins chevauchées :
Mesurer la pertinence de la fraction de texte recouvré au rang *r* par la fonction suivante comme mesure de précision :

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

Pour réaliser un score de haute précision au rang *r*, on doit mesurer la pertinence de la fraction en tenant compte de la quantité du texte non-pertinent contenu dans cette fraction par la fonction suivante comme fonction de rappel :

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

Et puis appliquer la précision moyenne (*AP : Average Precision*), en calculant la précision à chaque niveau de rappel (après avoir retrouvé une partie pertinente d'un document) :

$$AP = \frac{\sum_{r=1}^{|R|} rel(p_r).\, p[r]}{\sum_{r=1}^{|R|} rel(p_r)} .\, R[|\mathcal{R}|]$$

Avec : $[|\mathcal{R}|]$ = nombre total de parties du document correspondant

$$re(pr) = \begin{cases} 0 \; si\ la\ partie\ p\ du\ document\ ne\ contient\ pas\ d'\ information\ \textit{Pertinente soulignée} \\ 1 \; sinon \end{cases}$$

et indique la pertinence de la partie *pr* du document.
D'autres mesures sont utilisées pour évaluer les résultats des autres tâches :
Soit $\mathcal{P}$d l'ensemble des parties d'un document et *p* une partie de cet ensemble,

- La précision-document :

$$P(d) = \frac{\sum_{p \in \mathcal{P}\mathrm{d}} rsize(p)}{\sum_{p \in \mathcal{P}\mathrm{d}} size(p)}$$

Pour montrer que pour atteindre la haute précision, il faut qu'il contienne le moindre possible de textes non-pertinent dans un document.

- Le rappel-document :

$$R(d) = \frac{\sum_{p \in \mathcal{P}\mathrm{d}} rsize(p)}{Trel(d)}$$

Pour montrer que pour atteindre le haut rappel dans un document il faut qu'il contienne le maximum possible de texte pertinent.

- Le F-score du document :

$$F(d) = \frac{2.\, P(d).\, R(d)}{P(d) + R(d)}$$

Cette mesure est aussi utilisée dans la tâche in-context pour mesurer le score *S* d'un document *d* par la fonction suivante : *S(d) = F(d)* ∈ [0,1] (0 : document sans texte pertinent et 1 : tout les textes pertinents sont recouverts sans avoir à recouvrir du texte non-pertinent).
Il y a aussi la mesure d'évaluation de la tâche *Best in context* où le score du document d est calculé par une mesure de distance de similarité :

$$S(x,b) = \frac{A.L}{A.L + d(x,b)}$$

Avec :

L : la longueur du document *d*
A > 0 est un paramètre de contrôle
b : est le meilleur point d'entrée (élément) pour lire le texte pertinent nommé BEP (Best Entry Point)
x : est un point d'entrée quelconque
Par la suite, on aura une liste triée de documents par leurs scores de pertinence et calculer d'autres valeurs telles que la précision généralisée et le rappel généralisé.

II.8.3INEX 2009 :

Durant l'année 2009, une extension de la collection Wikipedia est fournie : elle est composée de 2.666.190 articles de Wikipedia annotés et elle a une taille de 50.7GB. Cette collection est utilisée dans la tâche ad-hoc ainsi que dans d'autres tâches.

D'autres collections sont aussi fournies par la compagnie d'évaluation pour évaluer d'autre tâche telles que la collection mmWikipedia pour une sous tâche de la tâche multimédia. Et avec le développement continu des SRI et l'apparition des nouvelles techniques dans le domaine de la RI, la compagnie INEX est vue s'améliorée jusqu'à INEX 2012 où de nouvelles mesures sont apparues et de nouvelles collections et tâches sont fournies aux SRI.

II.8.4INEX 2012 : [INEX 2012] :

En 2012, INEX et en collaboration avec CLEF, ont étudié les différents aspects pour l'accès à l'information concentrée et ils ont établi les tâches de base pour l'évaluation des SRI en se basant sur des collections appropriées:

• **La tâche des données liées LDT (Linked Data Track) :** examinant la récupération sur une collection de documents fortement structurés, ses dernier sont tirés de DBpedia et Wikipedia. La tâche ad hoc a des requêtes à satisfaire avec des entités de la collection composée des articles Wikipedia et des propriétés RDF[5] de DBpedia et [6]YAGO2. La tâche de recherche de facettes demande de retourner une liste limitée d'éléments qui guideront l'utilisateur de façon optimale vers des informations pertinentes.

---

[5] **RDF :** (Ressource Description Framework) est un standard décrivant les ressources : personnes, lieux, documents… et est un framework contenant un modèle de données : langage et syntaxe. Il a été créé en 1999 en tant que norme sur XML pour les métadonnées. Les métadonnées peuvent être : l'auteur d'une page web, à quelle date une entrée de blog a été publiée, … etc.

[6] **YAGO2 :** est une base de connaissance sémantique à plus de dix millions entités, en effet c'est une énorme base de connaissance sémantique, issu de Wikipedia WordNet et GeoNames. Actuellement, YAGO2 a connaissance de plus de 10 millions d'entités (comme les personnes, les organisations, les villes, etc) et contient plus de 120 millions de faits au sujet de ces entités, sa précision a été évaluée manuellement prouvant une précision de 95%.

• **Relevance feedback track (RFT) :** qui examine l'utilité du niveau de passage progressif du relevance feedback en simulant l'interaction d'un chercheur. Une évaluation non conventionnelle suit la trace où les soumissions sont des programmes informatiques exécutables plutôt que des résultats de recherche, en se basant sur la collection INEX Wikipedia de 50.7GB de 2.666.190 articles XML de Wikipedia.

• **Snippet Retrieval Track (SRT) :** (tache de récupération des petits bouts) qui examine la façon de génération des petits bouts informatifs (unités informatives) et qui devrai fournir des informations pertinentes permettant à l'utilisateur de déterminer la pertinence de chaque document sans avoir à consulter toute la collection. En utilisant la collection Wikipedia d'INEX 2009 qui est une version XML d'English Wikipedia.

• **Social book search track (SBST) :** la tâche de recherche des livres sociaux, examine des techniques pour soutenir les utilisateurs dans la recherche et la navigation dans des collections numériques des livres (à des livres numérisés), des métadonnées et des médias sociaux complémentaires. Elle étudie la valeur relative des métadonnées autorisées et le contenu créé par l'utilisateur en se basant sur une collection de test à des données de l'Amazone et Library Thing et la tâche demande des pages confirmant ou réfutant l'utilisation d'un corpus de textes pleins.

• **Tweet contextualization track (TCT) :** sert à répondre à des requêtes de forme « What is this tweet about ? » avec un résumé synthétique d'informations contextuelles saisies de Wikipedia et évaluées selon la pertinence du texte retourné « le dernier point d'intérêt » en se basant sur une collection tirée de Wikipedia 2011 ainsi que 1000 requêtes en anglais.

Introduction :

Une des grandes problématiques dans le domaine de la recherche d'information est de traiter une grande quantité de données pour en extraire de l'information. Il serait donc intéressant d'avoir un (ou plusieurs) modèle(s) effectuant le lien entre les observations et la réalité pour un objectif précis, et cela, même lorsque les observations sont incomplètes et/ou imprécises.

L'approche Bayésienne est connue comme une bonne solution aux problèmes contenant de l'incertitude. Comme la RI est aussi un processus incertain, l'application de l'approche Bayésienne dans la RI a été étudié.

Avant d'aborder les modèles de recherche d'information basés sur les réseaux bayésiens, nous avons jugé utile de définir, en section 1, quelques notions importantes sur les réseaux bayésiens, les probabilités conditionnelles dans ces réseaux et les méthodes d'apprentissage des RBs. La section 2 sera consacrée à la présentation des travaux utilisant une modélisation de type réseaux bayésiens et ayant eu plus ou moins du succès. Nous conclurons par une comparaison entre ces différents modèles

III-1Les réseaux bayésiens

III-1-1 Présentation :

La représentation des connaissances et le raisonnement à partir de ces représentations a donné naissance à de nombreux modèles. Les modèles graphiques probabilistes, et plus précisément les réseaux bayésien (RB), initiés par Judea Pearl dans les années 80[Pearl,88] se sont révélés des outils très pratiques pour la représentation de connaissances incertaines et le raisonnement à partir d'informations incomplètes, dans de nombreux domaines tel que le diagnostic (médical et industriel), l'analyse de risques, et le data mining.

III-1-2 Définition :

Un réseau bayésien est un graphe orienté acyclique DAG (Directed Acyclic Graph) permettant de représenter les influences (relations de causalité, dépendances directes ou conditionnelles) entre les variables du domaine et muni d'un ensemble de tables de probabilités conditionnelles pour quantifier le déterminisme (ou l'incertitude) dans les relations de causalité. Un réseau bayésien représente d'une façon compacte la distribution de probabilités jointe globale associée à toutes les variables du domaine modélisé.

Un réseau bayésien revêt donc deux aspects, le premier qualitatif (graphique) et l'autre quantitatif :

a) L'aspect qualitatif :

Un graphe orienté acyclique DAG noté S(**X**, E) où
X : est l'ensemble des nœuds (chaque nœud représente une variable aléatoire, discrète ou continue, décrivant le domaine),

E : est l'ensemble des arcs orientés représentant les relations de cause à effet ou de dépendance entre les nœuds qu'ils relient (par exemple un arc du nœud A vers le nœud B représente une relation de causalité ou de dépendance directe). A est dit parent de B, B fils de A. Les arcs sont des relations binaires irreflexives. Les parents d'un nœud A sont tous les nœuds ayant des arcs entrant vers A.

b) **L'aspect quantitatif :**

Est défini par une collection de tables de probabilités conditionnelles associées aux variables aléatoires *Xi*. Pour chaque variable *Xi*, on spécifie les probabilités de chaque valeur de *Xi* sachant toutes les valeurs possibles des parents de *Xi*.

La distribution de probabilités jointe globale (associée à l'ensemble des variables aléatoires) est décomposée sous forme de produit des distributions de probabilités locales par la règle de chaînage (également appelée formule de factorisation ou de chaînage [Chain Rule]):

$$P(X_1,..,X_n) = \prod_{i=1}^{n} P(X_i / Pa(X_i))$$

Où *Pa* (x_i) représente l'ensemble des parents (causes directes) de *Xi*. La loi de factorisation peut être expliquée par la loi de Bayes et la notion de d-séparation (abordée dans la section III-2-5).

III-1-3 Loi de Bayes :

La loi de Bayes permet de calculer la probabilité d'une variable sachant la valeur d'une autre variable de la manière suivante :

$$\mathbf{P\ (A|B) = \frac{P(B|A)\times P(A)}{P(B)}}$$

P(A|B) : est dite probabilité a posteriori, probabilité de *A* étant donné (sachant) *B*.
P(A) : est dite probabilité à priori (probabilité marginale) de *A*.
P(B) : est dite probabilité à priori (probabilité marginale) de *B*.
P(B|A) : pour un *B* connu est appelée la fonction de vraisemblance de *A*. *A* est dite l'évidence[7].

[7] a été traduit mot à mot de l'anglais evidence qui signifie en réalité dans notre contexte la preuve.

III-1-4 Les relations de dépendance dans un réseau bayésien :

Dans un réseau bayésien les nœuds peuvent être reliés en différentes topologies comme suit :

- **Connexion en série :**

Selon la figure III.1, *A* a une influence sur *C* qui a une influence sur B. L'information peut circuler de *A* vers *B* ou de *B* vers *A* à travers *C* dans les deux cas. Par contre, si *C* est connue ou instanciée, la voie est bloquée et *A* et *B* deviennent indépendants. On dit dans ce cas, que *A* et *B* sont **indépendant (ou d_séparés)** étant donnée *C.*

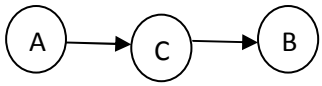


Figure III.1 connexion série

- **Connexion divergente:**

Dans la figure III.2, L'information peut passer entre les enfants de *C* lorsque la variable *C* est non instanciée. les enfants *A*, *B* sont **indépendants** étant donné *C*.

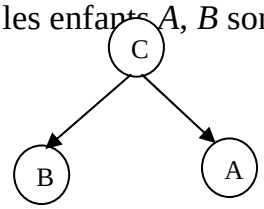


Figure III.2 Connexion divergente

- **Connexion convergente (ou la V_ structure):**

Dans ce type de connexion décrite dans la figure III.3, lorsqu'aucune information n'est donnée sur le nœud fils *C* mis à part l'information apportée par les parents *A* et *B*, les parents sont dits dans ce cas indépendants. Par contre, si l'état de *C* est connu alors *A* et *B* sont dépendants (l'état de *A* donnera une information sur l'état de *B*).

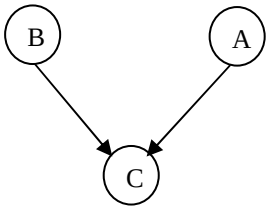


Figure III.3 Connexion convergente

III-1-5 la notion de D-séparation :

La d-séparation est une propriété graphique qui renseigne sur la circulation de l'information dans un réseau causal. La d-séparation explicite les conditions dans lesquelles l'information peut circuler entre deux sous-ensembles de variables. Ainsi, le calcul d'une probabilité d'intérêt se trouve énormément simplifié. La d-séparation étend la notion d'indépendance conditionnelle aux variables qui ne sont pas parents directes de la variable d'intérêt.

On dit que deux variables distinctes A, B d'un réseau sont d_séparées, si pour tout chemin entre A et B, il existe une variable intermédiaire C, distincte de A et de B telle que :

- soit la connexion est en série ou divergente et C est instanciée ;
- ou la connexion est convergente et ni C, ni ses descendants ne sont instanciés.

Ainsi, si les deux variables A et B sont d_séparées, alors tout changement d'état dans A n'aura pas d'impact sur l'état de B.

Exemple : Nous reprenons ici un exemple classique extrait de [PEARL, 1988] *:*

Ce matin-là le temps est clair et sec, Mr Holmes sort de sa maison. Il s'aperçoit que la pelouse de son jardin est humide. Il se demande s'il a plu la nuit, ou s'il a simplement oublié de débrancher son arroseur automatique. Il jette un coup d'œil à la pelouse de son voisin, et s'aperçoit qu'elle est également humide. Il en déduit alors qu'il a plu, et il décide de partir au travail sans vérifier son arroseur automatique.

Soient quatre variables aléatoires booléennes associées aux prédicats suivants :

A : Mr Holmes a oublié de débrancher son Arroseur automatique
L : il a Plu pendant cette nuit
J : la pelouse du Jardin de Mr Holmes est humide
V : la pelouse du jardin du Voisin de Mr Holmes est humide

La figure III.4 représente le réseau bayésien correspondant.

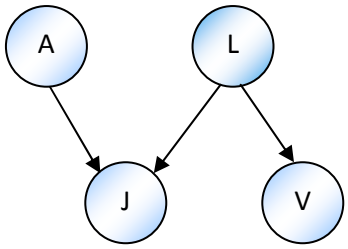


Figure III.4. Réseau bayésien de l'exemple

La distribution jointe globale de ce réseau est:

$P(A,L,J,V)=\prod_{X_i=A,P,J,V} P(X_i/pa(X_i)$

P(A,L,J,V)=P(A)*P(L)*P(J/A,L)*P(V,L)

Pour l'évaluer on doit déterminer chacun des facteurs du produit à savoir $P(X_i/pa(X_i))$ (avec X_i=A,L,J,L) dont chacun est représenté par une des tables ci_dessous.

θ_A= P(A)

A	**P(A)**
t	0.4
f	0.6

θ_L=P(L)

L	**P(L)**
t	0.7
f	0.3

θ_V =P(V/L)

V	**p (V|L=t)**	**p (V|L=f)**
t	0.58	0.42
f	0.5	0.5

θ_J=P(J/A,L)

J	**P (J| A=t, L=t)**	**P (J| A=t, L=f)**	**P (J|A=f, L=t)**	**P (J|A=f, L=f)**
t	0.7	0.95	0.6	0.22
f	0.3	0.05	0.4	0.78

Tableaux III.1 : Exemple de tables de probabilités conditionnelles (dits paramètres)

III-1-6 Apprentissage des réseaux bayésien :

Un réseau bayésien est construit soit sur la base de la connaissance d'un expert humain, soit automatiquement à l'aide d'outils d'apprentissage automatiques ou bien conjointement avec l'apport d'un expert et d'autres informations obtenues par des techniques d'apprentissage automatique. L'apprentissage d'un réseau bayésien désigne l'élaboration de la structure graphique du réseau et des tables de probabilités conditionnelles. Il concerne l'apprentissage de structure et de paramètres.

III-1-6-1 Apprentissage des paramètres :

Ici on suppose que la structure du réseau a été fixée, et où il faudra estimer les probabilités conditionnelles de chaque nœud du réseau.

- **À partir de données complètes :**

Il s'agit d'estimer les distributions de probabilités (ou les paramètres des lois correspondantes) à partir de données disponibles.
Dans le cas où toutes les variables sont observées, la méthode la plus simple et la plus utilisée est l'estimation statistique qui consiste à estimer la probabilité d'un événement par la fréquence d'apparition de l'événement dans la base de données. Cette approche, appelée maximum de vraisemblance (MV), nous donne alors :

$$\mathbf{P(X_i=x_k|parent(X_i)=c_j)=}\boldsymbol{\theta_{i,j,k}}\mathbf{=}\frac{\boldsymbol{N_{i,j,k}}}{\sum_{\boldsymbol{k}}\boldsymbol{N_{i,j,k}}}$$

Où $N_{i,j,k}$ = nombre d'événement dans la base de donné ou $\{X_i = x_k$ et $Pa(X_i) = c_j\}$

➢ **À partir de données incomplètes :**

Dans la pratique, Certaines variables ne sont observées que partiellement ou même jamais. La méthode d'estimation de paramètres avec des données incomplètes la plus couramment utilisée est fondée sur l'algorithme itératif EM (Expectation-Maximisation) proposé par Dempster [Dempster et al, 1977] Soit :

- Xv= $\{X_V^{(l)}\}_{l=1....N}$ l'ensemble des données observées.
- $\theta^{(t)}$= { $\theta_{i,j,k}^{(t)}$} les paramètres du réseau bayésien à l'itération t.

L'algorithme EM s'applique à la recherche des paramètres en répétant jusqu'à convergence des deux étapes Espérance et Maximisation décrites ci-dessous :

❖ **Espérance** : estimation des $N_{i,j,k}$ manquants en calculant leur moyenne conditionnellement aux données et aux paramètres courants du réseau

$$N_{i,j,k}^{*} = E[N_{i,j,k}] = \sum_{i=1}^{N} p(X_i = x_k \mid parent(X_i) = c_j, X_v^{(1)}, \theta^{(t)})$$

❖ **Maximisation** : en remplaçant les $N_{i,j,k}$ manquants par leur valeur moyenne calculée précédemment, il devient possible de calculer de nouveaux paramètres $\theta_{i,j,k}^{(t+1)}$ par le maximum de vraisemblance comme suit :

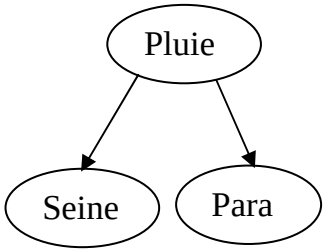
$$\theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^{*}}{\sum_{k} N_{i,j,k}^{*}}$$

L'algorithme EM est défini ainsi :

• Initialiser $\theta^{(0)}$, t=0
• Répéter
  t=t+1
  calculer $N^*_{i,j,k}$
  calculer $\theta^{(t)}_{i,j,k}$
jusqu'à ce que $|\theta^{(t)} - \theta^{(t-1)}| < \varepsilon$

Exemple d'apprentissage de paramètre avec l'algorithme EM [NGUYEN,2005] :

Prenons le réseau bayésien et la base d'exemples définis ci-dessous (où «?» représente une donnée manquante) :



Pluie	Seine
0	?
n	?
0	N
n	N
0	0

Les propositions associés aux variables aléatoires (nœuds)sont :
Pluie = «il pleut à Rouen»
Seine = «la Sein déborde»
Para = «j'ai sorti mon parapluie»

Estimation des paramètres $\theta_{S|P=o}$ et $\theta_{S|P=n}$ avec l'algorithme EM :

- ***Initialisation*** : Les valeurs initiales des paramètres sont choisies aléatoirement, on choit $\theta^{(0)}_{S|P=o} = 0.3$ et $\theta^{(0)}_{S|P=n} = 0.4$

- ***Première itération*** : Le calcul de l'étape E est résumé dans le tableau ci-après.

		P(S|P=o)		P(S|P=n)	
Pluie	Seine	S=o	S=n	S=o	S=n
o	?	0.3	0.7	0	0
n	?	0	0	0.4	0.6
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	N*	1.3	1.7	0.4	1.6

Cette étape nous donne $\theta^{(1)}_{S|P=o}=1.3/(1.3+1.7)=0.433$ et $\theta^{(1)}_{S|P=n}=0.4/(0.4+1.6)=0.2$

- **Deuxième itération** : Le calcul de l'étape E est résumé dans le tableau ci-après.

		P(S|P=o)		P(S|P=n)	
Pluie	Seine	S=o	S=n	S=o	S=n
o	?	0.433	0.567	0	0
n	?	0	0	0.2	0.8
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	N*	1.433	1.567	0.2	1.8

Cette étape nous donne $\theta^{(2)}_{S|P=o} = 1.433/3 = 0.478$ et $\theta^{(2)}_{S|P=o} = 0.2/2 = 0.1$

- ***Convergence*** : Après quelques itérations de l'algorithme EM, les valeurs de paramètres convergent vers $\theta^{(t)}_{S|P=o}= 0.5$ et $\theta^{(t)}_{S|P=n}= 0$.

III-1-6-2 Apprentissage de la structure :

Dans certaines situations, la structure est fournie par un expert. Si ce n'est pas le cas, on fait l'apprentissage à partir de données complètes ou incomplètes. La recherche de la structure est un problème difficile principalement à cause du fait que l'espace de recherche est de taille super-exponentielle en fonction du nombre de variables. Le problème confronté est : comment choisir la meilleure structure d'un Réseau Bayésien?

Il y a deux approches générales de construction de la structure d'un Réseau Bayésien par apprentissage. L'une est basée sur la recherche et des méthodes de marquage (search and scoring), l'autre est basée sur des méthodes d'analyses de dépendances.

La première approche est de nature heuristique, elle consiste à chercher la meilleure structure qui s'adapte aux données. Elle commence avec un graphe déconnecté, utilise des méthodes de recherche pour ajouter des arcs et teste par l'usage d'un score si la nouvelle structure est meilleure que l'ancienne.

Dans la deuxième approche, leurs algorithmes essayent de découvrir les dépendances entre les variables dans le réseau puis emploient ces dépendances pour impliquer la structure.

III-1-7 L'Inférence dans un réseau bayésien :

L'inférence est le calcul de la probabilité de n'importe quelle variable du modèle probabiliste à partir de l'observation d'une ou plusieurs autres variables. Il consiste à propager une ou plusieurs informations certaines au sein de ce réseau pour en déduire comment sont modifiés les croyances concernant d'autres nœuds.

La structure du graphe joue un rôle important dans la complexité de ces calcules ainsi que le choix de la méthode d'inférence. Les algorithmes d'inférence, de part la nature du résultat qu'ils fournissent, se répartissent en algorithmes d'inférence exacte qui fournissent des résultats exactes et des algorithmes d'inférence approchée dont les résultats sont des approximations des probabilités réelles. En effet, L'inférence exacte dans un réseau bayésien n'est pas toujours possible d'un point de vue calculatoire, du fait du très grand nombre de variables. Des méthodes d'inférences approchées ont été alors proposées pour les réseaux très complexes.

III-2 Les modèles de recherche d'informations basés sur les réseaux Bayésiens:

Les Réseaux Bayésiens (RB) ont été utilisés en RI depuis les années 1990 avec [Pearl,1988], [Buntine, 1994], [Jensen, 2000]. Ils fournissent un formalisme pour fusionner des informations provenant de différentes sources (requêtes passées, réinjection de pertinence,..), pour le calcul de la correspondance entre la requête et les documents .La composante qualitative du réseau permet de représenter les documents de la collection, les termes d'indexation ou concepts des documents et du besoin utilisateur ou requête par des nœuds ainsi les relations de dépendance (ou d'indépendance) existant entre ces variables par des arcs. L'aspect quantitatif du réseau permet d'évaluer les arcs reliant toute paire de nœuds au moyen de calcul de probabilités.

Les modèles les plus connus en RI utilisant les réseaux bayésien sont les Réseaux d'Inférence [Turtle et Croft, 1990] et les Réseaux de Croyance [Ribeiro-Neto et al., 1996].

III-2.1 Modèle à base de Réseaux Bayésiens d'Inférence :

Un réseau d'inférence en RI est matérialisé par un graphe orienté sans cycle. Les nœuds du graphe correspondent à des concepts, à des groupes de mots ou à des documents (des variables propositionnelles). Un nœud particulier va représenter le besoin de l'utilisateur. Les arcs du graphe représentent des relations sémantiques entre les nœuds. A ces nœuds sont associés des probabilités de croyance.

une architecture simplifié du modèle de recherche d'information basé sur les réseaux bayésien d'inférence proposé par [Turtle 1991]est représenté par la figure ci_dessous :

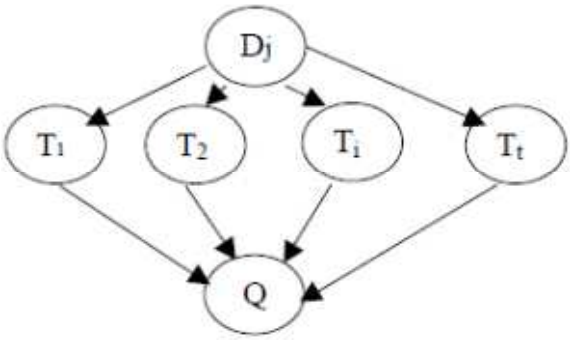


Figure III.5– Architecture simplifiée du réseau d'inférence

Le nœud document ***Dj*** correspond à l'événement qu'un document donné de la collection est observé. Le nœud T_i (i=1,…t) correspond au terme T_i indexant le document et correspond à l'événement que ce terme T_i du document est rencontré.
La dépendance entre un document et un terme est symbolisée par un arc entre les nœuds document et terme. Les domaines de tous les nœuds sont binaires {vrai, faux} désignant le fait que le nœud est instancié ou non (exemple: un nœud représentant le terme aura pour instanciation vrai uniquement lorsque son nœud parent, document, est aussi instancié à vrai). Le nœud Q représente le besoin utilisateur. Le domaine du nœud Q est vrai pour désigner que la requête est satisfaite.

Le calcul de la pertinence revient dans ce modèle à instancier chaque document de la collection et à calculer la probabilité de satisfaire la requête étant donnée le document instancié par la formule suivante :

$$\mathbf{P(Q|d_j)=\sum_{\forall \theta^k \epsilon \theta} \left(P(Q|\theta^k) . \prod_{T_i \epsilon Q \wedge D_j} P(\theta_i^k | d_j) \right) . P(d_j)}$$

Où θ est l'ensemble des configurations possibles des parents de *Q*,
θ_i^j est une instance d'un nœud particulier *Ti* telle que dans la configuration de θ^j de θ.

La quantification totale de la pertinence revient à quantifier chaque membre de la formule précédente.

Remarques :

1- A chaque instanciation d'un document *Dj*, sa probabilités à priori P (D_j=d_j)= $\frac{1}{n}$ ce calcul sera donc supprimé de la propagation globale parce que ce terme est considéré comme un coefficient uniforme appliqué à tous les documents de la collection.

2- Turtle a proposé cinq formes canoniques pouvant répondre à tout type de recherche. La requête peut être agrégée par les opérateurs booléens (ET, OU, et NON). Pour évaluer les probabilités conditionnelles *P (Q | θ)* d'un nœud Q ayant n parents, {θ_1, .., θ_n}, et, P (θ_1 = t_1) = p_1, .., P (θ_n = t_n) = p_n les agrégations suivantes sont définies :

$$\mathbf{P_{Ou}(Q|\theta) = 1-(1-p_1)-..-(1-p_n)}$$
$$\mathbf{P_{Et}(Q|\ \theta) = p_1 \times .. \times p_n}$$
$$\mathbf{P_{Non}(Q|\ \theta_1) = 1- p_1}$$
$$\mathbf{P_{Somme}(Q|\ \theta) = \frac{p_1+..+p_n}{n}}$$
$$\mathbf{P_{SommePondéree}(Q|\ \theta) = \frac{(w_1p_1+..+w_np_n)\ w_q}{w_1+..+w_n}}$$

La somme pondéré mesure la configuration positive en fonction du poids de chaque parent instancié positivement, ainsi que du poids de la requête w_q. Le poids utilisé peut être le facteur de discrimination *idf* ou une de ses variantes ou un poids attribué par l'utilisateur.

1- Les arcs reliant les termes d'indexation aux documents sont pondérés par des variantes de tf – idf comme suit :

P(ti | dj) =0.5 + 0.5 ∗ ntfij ∗ ndfi

Avec

$$\mathbf{ntf_{ij}}=\frac{tf_{ij}}{max_{\ t_k \in d_j}\ tf_{kj}} \qquad \mathbf{ndfi}=\frac{log\left(\frac{N}{n_i}\right)}{log(N)}$$

Tf_{ij} : est la fréquence du terme *ti* dans le document *dj*.

$max_{\ t_k \in d_j}\ tf_{kj}$: est la fréquence maximal dans le document *dj*.

N : est le nombre de documents de la collection.

ni : est le nombre de documents contenant le terme ti.

on estime que : P(t_i|$\overline{\mathbf{PARENTS}}_{\mathbf{Ti}}$)=0

Où $\overline{PARENTS}_{Ti}$ signifie que tous les parents de Ti sont instanciés à faux.

$\mathbf{P(\bar{t}_i|d_j)=1-P(t_i|d_j)}$

Parmi les systèmes qui ont utilisé les réseaux d'inférence on peut citer le système INQUERY proposé dans [Turtle et Croft, 1990] [Turtle, 1991] [Turtle et Croft, 1991] et d'autres travaux basés sur ces réseaux ont été aussi proposés pour les systèmes hypertextes[8] [Savoy et al., 1991].

III-2-2 Le modèle de croyance : [Ribeiro-Neto et al., 1996].

Ce modèle est basé sur la définition préalable d'un espace d'échantillonnage qui permet de séparer clairement les portions de documents des portions de requêtes et donc de calculer d'une manière« efficace» les degrés de croyance. L'architecture générale du modèle est présentée dans la figure 6.
L'univers de discours est donné par l'ensemble des termes d'indexation utilisés dans le système, noté *U*, et $U = \{T_1, .., T_T\}$ où $_T$ est le nombre de termes manipulés dans le système (pour représenter les documents ou la requête).

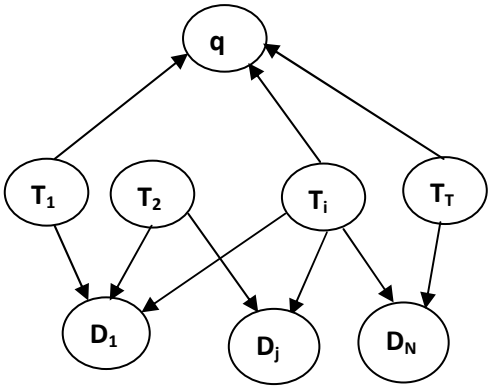


Figure III.6 – Architecture générale

A la réception d'un besoin utilisateur, la requête Q est instanciée et le processus de propagation est déclenché.

D'après la topologie du modèle, les nœuds documents et requête sont d_séparées par les termes d'indexation. Ainsi :

$$\mathbf{P(D_j|\ Q)} \propto \sum_{\boldsymbol{\theta}} \boldsymbol{P(D_j\,|\theta\ )}\times\mathbf{P(Q|\boldsymbol{\theta})}\times\mathbf{P(\boldsymbol{\theta})}$$

Où θ est l'ensemble des configurations possibles sur *U*, donc 2^T configurations dans *U* sont possibles, mais en réalité uniquement les termes indexant la requête sont considérés.

Pour calculer la probabilité d'un document étant donnée une configuration d'un concept $P(Dj=1/\theta) = P(dj|\theta)$, le modèle propose d'utiliser la fonction cosinus du modèle vectoriel, comme suit :

[8] Système qui exploite les liens hyper text (internes ou externes) figurant dans un documents dans la recherche d'information.

$$P(d_j \mid \theta^Q) = \frac{\sum_{\theta_i^Q=1}^{T} w_{ij} \times w_{iq}}{\sqrt{\sum_{\theta_i^Q=1}^{T} w_{ij}^2} \times \sqrt{\sum_{\theta_i^Q=1}^{T} w_{iq}^2}}$$

Ou θ^Q_i : est la configuration des termes telle que donnée dans la requête *Q*.
w_i^j, w_i^q : les poids du terme *ti* dans le document *Dj* et la requête *Q* respectivement.
Les poids *wij* sont des variantes de la pondération par *tf* idf*.

Ainsi :

$$P(\overline{d_j} \mid \theta^Q) = 1 - P(d_j \mid \theta^Q)$$

La probabilité d'une requête étant donnée une configuration $P(Q \mid \theta)$ est calculée comme suit :

$$P(Q \mid \theta) = \{1 \text{ si } \forall\, T_i,\ \ \theta_i^Q = \theta_i$$
$$= 0 \text{ sinon } \}$$

Ou θ^Q_i , θ_i l'instanciation du terme *Ti* dans la requête et dans une configuration **θ** respectivement.
Ainsi que :

$$P(\overline{Q} \mid \theta) = 1 - P(Q \mid \theta)$$

et

$$P(\theta) = (\frac{1}{2})^T$$

Le modèle propose aussi de calculer La probabilité *P (dj)* qui donne le degré au quelle le document *Dj* couvre complètement l'espace des termes *U*. Cette probabilité est calculée comme suit:

$$P(d_j) = \sum_{\theta} P(d_j \mid \theta) P(\theta)$$

Ainsi que la probabilité *P(Q)* qui donne le degré auquel la requête couvre complètement l'espace des termes *U*. Cette probabilité répondrait à la croyance associée à la proposition Est-il vrai que *Q* couvre complètement U ?

$$P(Q) = \sum_{\theta} P(Q \mid \theta) P(\theta) \qquad \textbf{Avec :} \qquad P(\theta) = (\frac{1}{2})^T$$

Le calcul de cette équation nécessiterait 2^T calculs, ou*T* est le nombre de termes manipulés par le système, mais en réalité uniquement les termes indexant la requête sont considérés.

III-2-3Autres modèle de recherche basé sur les réseaux bayésien :

Des extensions des deux modèles basés sur les réseaux Bayésiens ont été proposées aussi bien pour résoudre des problèmes d'optimisation de calculs nécessités par les topologies des réseaux ([indrawan et al, 1996] et [De Campus et al, 2003], etc..) que pour les appliquer à des collections de documents de types hétérogènes et contenant des liens hypertextes[Crestani et al, 2003, Silva et al, 2000, Denoyer et al,2004],nous présentons dans ce qui suit les modèles plus connus notamment celui de Indrawan et celui de l'équipe de De Campos:

III-2-3-1 Le Modèle d'Indrawan [indrawan et al, 1996] **:**

Dans ce modèle, Le réseau document constitue le noyau constant du réseau. Le second réseau est composé de la requête et il est dynamique.

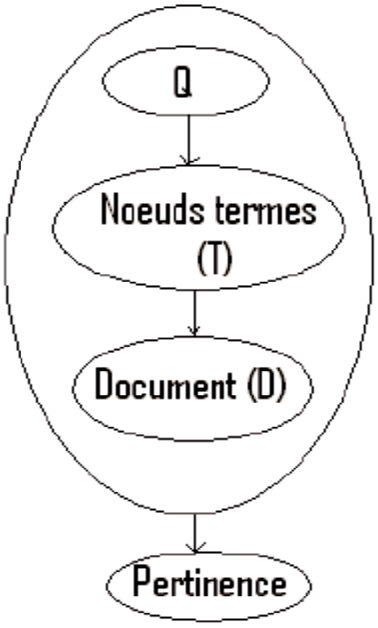


Figure III.7 – Architecture globale

Dans l'architecture générale (simplifiée) présentée dans la figure III.7, on remarque que le nœud requête est parent des nœuds termes qui sont aussi parents des nœuds documents. Les arcs sont orientés de *Q* vers les termes *T* et des nœuds termes vers les nœuds documents *D*. On remarque aussi l'introduction d'un nœud « pertinence ».

❖ **Représentation des documents :**

Le réseau document est composé de deux couches de nœuds : une couche supérieure composée des mots clés de la collection et une couche inférieure contenant les concepts, ou toute entité générée par la combinaison des mots clés.
Le nœud terme implique l'existence des documents.

❖ **Représentation de la requête :**

Le réseau requête (*Q*), qui est temporaire, contient un nœud racine qui symbolise le besoin utilisateur, un ensemble de mots clés qui forment la description de la requête, relié à *Q*.

❖ **Calcul de la pertinence :**

Le processus de propagation est déclenché par la requête et l'information est propagée sur les nœuds documents. La probabilité de pertinence d'un document étant donnée une requête, selon la topologie du graphe (figure III.8) est mesurée par :

$$P(Pert \mid Dj, Q) = P(Pert \mid Dj, Ti, Q)$$

Où *Pert* est un nœud virtuel, nœud fils final du réseau, et décrit l'événement « la pertinence d'un document donnée en réponse à une requête ». *Dj* est un document de la collection et *Q* une requête.

Une simplification de la propagation suivant la règle de chaînage, calcule la probabilité de pertinence d'un document étant donné la requête par *P(Dj|Q).*

III-2-3.2 Le réseaux multi connectés pour la RI :

Le modèle de réseau multi connectés pour la RI ainsi proposé prend en compte les relations entre les termes. Ce modèle ne considère pas tous les termes de la collection : un apprentissage sur le réseau initial permet de produire un nouvel arbre où il n'y a pas plus d'un chemin orienté reliant chaque paire de nœuds. Ce réseau contient un ensemble plus petit de termes et leurs relations de dépendance.

Ce réseau comprend deux ensembles de nœuds, comme montré dans la figure III.8: L'ensemble des nœuds termes, *T* ; une variable *Ti* associée à un terme prend ses valeurs dans le domaine {ti, $\overline{\text{ti}}$ } où *ti* désigne le terme *Ti* comme étant pertinent (donc figure dans sa requête) et $\overline{ti}$ comme n'étant pas pertinent.

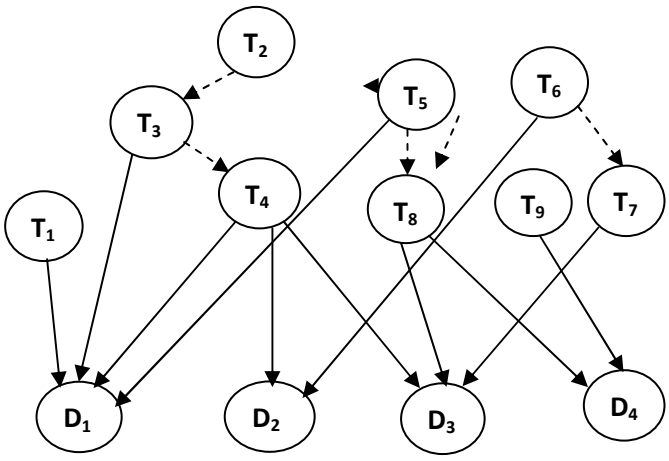


Figure III.8 Architecture globale du réseau multi connecté

- L'ensemble des nœuds documents; une variable *Dj* prend ses valeurs dans {dj, $\overline{dj}$}, où dj signifie « le document *Dj* est pertinent » (s'il répond au besoin utilisateur).

Dans la topologie proposée, le degré de pertinence d'un document *Dj* peut être complètement déterminé par la connaissance de la pertinence de tous les termes d'indexation du document *Dj*. Lorsque cette information est absente, la connaissance de la pertinence pour la même requête d'un autre document Dk peut avoir une influence sur celle de *Dj* (c.à.d tout document *Dj* est conditionnellement indépendant de tout document *Dk* lorsque les valeurs de pertinence de tous les termes indexant le document *Dj* sont connues).

Les termes présents dans la requête propagent l'information à travers le réseau pour calculer la pertinence d'un document étant donnée la requête, *P (dj/Q)* :

- les probabilités à priori des nœuds termes racine sont données par :

  $P(ti) = \frac{1}{M}$ avec *M* le nombre de termes de la collection.

  $P(\overline{ti}) = 1 - P(ti)$

- Pour calculer les probabilités conditionnelle de chaque terme étant donné ses parents P(ti|pa(Ti)) :

$$P(ti|pa(Ti))=\frac{n(t_i,pa(T_i))}{n(pa(T_i))+|T_i|}$$

Où $|T_i|$ est le nombre de valeurs que T_i peut avoir, $n(t_i, pa(T_i))$ est le nombre de fois dans la collection où le terme *Ti* figure dans *pa(Ti)* avec la valeur pertinente . $n(pa(T_i))$ donne le nombre de fois, dans le document, où toutes les variables de pa(T_i) sont pertinents.

- **Pour l'estimation de la probabilité des documents :**

$$P(d_j|\theta_{Dj})=\sum_{T_i \in R(\theta_{Dj})} w_{ij}$$

$$P(d_j|\theta_{Dj})=1\text{-}P\ (d_j|\overline{\theta}_{Dj})$$

avec

θ_{Dj} l'ensemble des configurations possibles des parents de Dj ; certaines sont pertinentes, notées $R(\theta_{Dj})$, et d'autres pas. Une configuration est non pertinente lorsque les instanciations des variables qu'elle contient ne sont pas conformes à la présence des termes dans le document.

w_{ij} le poids du terme Ti dans le document Dj .Ce poids est compris entre 0et 1 est obtenu par des variations normalisées de tf $*$ idf.

Remarque : La somme des poids des termes présents dans le document et absents de la requête sont aussi considérés dans le calcul de la pertinence.

- Enfin, la probabilité de pertinence d'un document étant donnée une requête $P(d_j/Q)$ est égale à la somme des produits des poids des termes de la requête et des documents. Les termes considérés sont ceux de la requête présents dans le document.

L'inférence : l'inférence consiste à utiliser la requête comme évidence en instanciant ces termes comme valeurs pertinents, calculer $P(t_i|Q)$ ∀ Ti, puis propager cette information jusqu'au nœud document.

III-2.3.4 Un modèle de réseaux bayésien basique [Acid et al, 2003] :

L'ensemble des variables V_B est constitué de deux ensembles $V_B = T \cup D$: l'ensemble $T=\{T_1, \ldots, T_M\}$ des M termes de la collection et l'ensemble $D=\{D_1, \ldots, D_N\}$ des N documents de la collection. Le domaine de chaque variable terme T_i est $\{t_i, \overline{t_1}\}$ où l'instanciation $T_i = t_i$ signifie que le terme T_i est pertinent et $T_i=\overline{t_1}$ signifie que le terme T_i n'est pas pertinent. Le domaine de chaque variable D_j *est* $\{d_j, \overline{d_j}\}$ où $D_j=d_j$ $(\overline{d_j})$ signifie que le document D_j est pertinent (non pertinent)étant donné la requête. La figure 1 représente les termes et les documents de la collection. Chaque terme est relié aux documents dans lesquels il apparait.

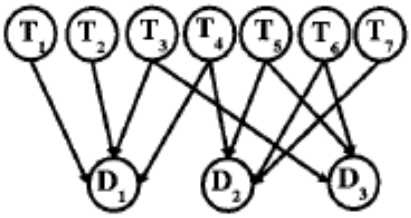


Figure III.9 Modèle de base

Remarquons que $Pa(Dj) = \{Ti \in T \mid Ti \in Dj\}$. De plus les auteurs estiment que :

- $\mathbf{P(t_i)= \frac{1}{M}}$ **et** $\mathbf{P(\overline{t_1}) = \frac{M-1}{M}}$ où M est le nombre de termes de la collection.

- $\mathbf{P(d_j|pa(D_j))=\sum_{T_i \in D_j,\, t_i \in pa(D_j)} w_{ij}}$

Inférence et recherche

Etant donné la requête Q, l'évidence est représentée en plaçant l'état de chaque terme T_{iq} appartenant à la requête à t_{iq}(pertinent). Le processus d'inférence consiste alors à calculer $P(d_j|Q)$:

$$\mathbf{P(d_j|Q) = \sum_{T_i \in D_j} w_{ij}.P(t_i|Q)}$$

Par ailleurs, les termes sont marginalement indépendants, alors :

$$\mathbf{P(t_i|Q)=1 \text{ si } T_i \in Q \; et \; P(t_i|Q)= \frac{1}{M} \text{ si } T_i \notin Q}$$

on aura donc :

$$P(d_j|Q) = \sum_{T_i \in (D_j \cap Q)} w_{ij} + \frac{1}{M} \sum_{T_i \in (D_j \setminus Q)} w_{ij}$$

Pour tenir compte de la fréquence des termes dans la requête, on a dupliqué qf_i fois dans le réseau pour chaque terme Ti apparaissant dans la requête et l'équation deviendra :

$$P(d_j|Q) = \sum_{T_i \in (D_j \cap Q)} w_{ij}\, qf_i + \frac{1}{M} \sum_{T_i \in (D_j \setminus Q)} w_{ij}$$

les auteurs on présenté une autre extension à ce modèle, illustrée par la figure 2, qui consiste à établir des relations fondées sur des mesures de similitudes entre les documents en estimant les probabilités conditionnelles de pertinence de chaque document étant donné que l'autre document (avec lequel il est dépendant) est jugée pertinent.

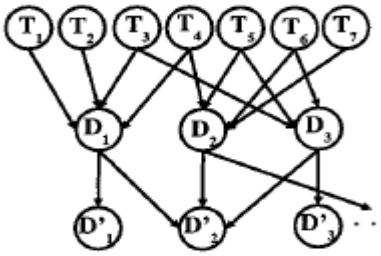
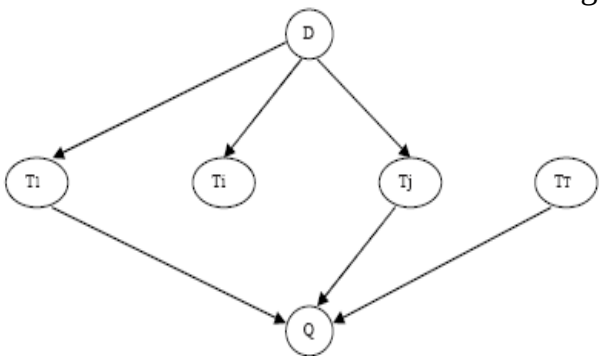


Figure III.10 Autre Réseau bayésien étendu

III.2.3.5 Un modèle de RI basé sur les réseaux possibilistes [Brini et al ,2005] :

L'architecture générale de ce modèle est illustrée dans la figure (11).



FigureIII.11 Architecture générale

Un document *Dj* est instancié ou pas, prenant ses valeurs dans le domaine *{dj, $\bar{d}j$}*. L'instanciation d'un nœud document, *Dj = dj* (resp.$\bar{d}j$) signifie que le document est pertinent (resp. non).
Une requête *Q* prend ses valeurs dans le domaine *{q, $\bar{q}$}*. Seule l'instanciation positive est prise en considération,
Le domaine d'un nœud terme d'indexation *Ti*, est *{ti,$\bar{t}i$ }*. *Ti = ti* signifie que le terme *ti* est présent dans le document (ou dans la requête) et est donc *représentatif* du contenu en information du document (ou de la requête) à un certain degré.

Soit *T* (*Dj*) (resp. *T* (*Q*)) l'ensemble des termes d'indexation du document *Dj* (resp. de la requête). La requête exprime la demande de documents contenant certains termes et peut également en exclure d'autres. Il existe une instanciation de l'ensemble des parents de la requête (*Par*(*Q*)) qui représente la requête dans sa forme la plus stricte (conjonctive). Soit θ^Q cette instanciation. Toute instanciation des parents de *Q* est notée θ.

Les auteurs adoptent une approche possibiliste dans le but de mesurer par deux évaluations le score de pertinence d'un document étant donnée une requête. Ce modèle devrait être capable d'inférer des propositions telles que :
– Il est possible à un certain degré que le document soit pertinent étant donnée la requête, notée par $\Pi(D \mid Q)$;
– Il est certain (dans le sens possibiliste) que le document soit pertinent étant donnée la requête, notée par $N(D \mid Q)$.

Le premier type de proposition est censé éliminer les documents non pertinents.
Le second se focalise sur le renforcement de la certitude de la pertinence.

Ainsi, le processus de propagation évalue les degrés de possibilité, $\Pi(dj \mid Q)$, et de nécessité, $N(dj \mid Q)$ comme suit :

$$\Pi(d_j \mid Q) = \frac{\Pi(Q \wedge d_j)}{\Pi(Q)}$$

$$N(d_j \mid Q) = 1 - \Pi(\overline{d_j} \mid Q) = 1 - \frac{\Pi(Q \wedge \overline{d_j})}{\Pi(Q)}$$

La possibilité de *Q* est donnée par :

$$\Pi(Q) = \max(\Pi(Q \wedge d_j), \Pi(Q \wedge \overline{d_j}))$$

$$\Pi(d_j \mid Q) = \min(1, \frac{\Pi(Q \wedge d_j)}{\Pi(Q \wedge \overline{d_j})});$$

$$\Pi(\overline{d_j} \mid Q) = \min(1, \frac{\Pi(Q \wedge \overline{d_j})}{\Pi(Q \wedge d_j)})$$

$$\Pi(Q \wedge D_j) = \max_{\theta^l \in \theta} \Pi(Q \mid \theta^l) . \prod_{T_i \in T(Q) \wedge T(Dj)} \Pi(\theta^l_i \mid D_j) . \Pi(D_j) . \prod_{T_k \in T(Q) \setminus T(Dj)} \Pi(\theta^l_k))$$

Avec :
θ : les configurations possibles de l'ensemble des parents de *Q*,
θ^l: une configuration possible de θ.
θ^l_i *:* l'instanciation de *Ti* dans la configuration θ^l ;

Ainsi que :

$$\Pi(Q \mid \theta^l) = 0 \text{ si } \nexists T_i \in PAR_Q \text{ tel que } \theta^l_i = \theta^Q_i$$
$$= \frac{1 - \prod_{i:\theta^l_i = \theta^Q_i}(1 - \pi(Q \mid t_i))}{1 - \prod_{T_k \in Par_Q}(1 - \pi(Q \mid t_k))} \text{ sinon}$$

$$\Pi(Q \mid t_i \wedge_{k \neq i} \overline{t_k}) = \frac{idf_i}{logN} = nidf_i$$

Avec
$idf_i=\log\frac{N}{ni}$
ni : est le nombre de documents contenant le terme t_i .
N : le nombre de documents de la collection.

1. **Possibilité *a priori* des documents :**

$$\Pi(d_j) = \frac{l_j}{\max_{k=1,..,N} l_k} = nl_{d_j}$$

avec *lj* la longueur du document *dj* en terme de fréquence ; *lj* = $\sum_i tfij$.
Plus le document est court, moins il est pertinent. Dans tous les cas, Π (*dj*) =1, si on ne veut pas favoriser le document de manière exagérée.

2. **Pondération des termes indexant les documents :**

Cette mesure est calculée comme suit :

$$\Pi(t_i/d_j) = ntf_{ij} = \frac{tf_{ij}}{\max_{\forall_{t_k \in d_j}}(tf_{kj})}$$

Avec
tf_{ij} : la fréquence du terme ti dans le document dj.
$\boldsymbol{max}_{t_k \in d_j}\ \boldsymbol{tf_{kj}}$: est la fréquence maximal dans le document dj.

III-3. Synthèse:

Dans ce chapitre, nous avons présenté l'approche Bayésienne et l'état de l'art des modèles Bayésien dans la RI, voici une étude comparative entre les modèles les plus connus présentés précédemment :

Le modèle inférentiel instancie le document à la réception d'une requête tandis que le modèle de croyance instancie la requête. Une différence majeure dans la topologie de ces deux réseaux concerne le sens de la dépendance des termes d'indexation avec les documents. Alors que pour le modèle inférentiel cette dépendance, quantifiée par *P(ti| dj)*, dans le modèle de croyance la relation de dépendance est quantifiable par *P(dj | ti)*.

L'apport le plus important du modèle inférentiel a été de pouvoir combiner l'information provenant des représentations différentes de documents ainsi que de combiner différentes formulations de la requête.

Le système INQUERY[Turtle et Croft,1991], même s'il ne présente pas de capacités d'apprentissage, est d'une part largement utilisé et a permis de construire des systèmes au meilleur niveau de l'état de l'art, et d'autre part, il constitue le point de départ du développement de tout un ensemble de modèles de recherche d'information.

Dans le modèle d'Indrawan [indrawan et al, 1996], contrairement au modèle de Turtle[Turtle 1991], les auteurs considèrent que les termes impliquent le document. La requête instancie le système dans le modèle d'Indrawan alors que les documents sont instanciés dans le modèle de Turtle. Son apport principal concerne l'optimisation proposée pour la propagation dans les réseaux. Ainsi, le nombre de probabilités conditionnelles à calculer à diminué grâce à l'utilisation des nœuds virtuels et uniquement les termes de la requête sont pris en compte dans les calculs.

Le réseau multi connecté pour la RI représente les relations de dépendance entre les termes d'indexation d'un document ainsi que des techniques qui réduisent le temps de calcul. Une autre variante de modèle multi connecté qui relie les documents a été proposée estimant que deux documents sont similaires s'ils contiennent des termes similaires.

D'autres auteurs ont exploité la théorie des possibilités pour proposer le modèle possibiliste qui utilise deux mesures de calcul de l'incertain à savoir les mesures de plausibilité d'un événement et celui de sa certitude. L'utilisation de la théorie des Possibilités permet au modèle de séparer les motifs du rejet d'un document comme non pertinentes (en tenant en compte des valeurs de possibilité) des motifs de la sélection d'un document pertinent (au moyen des valeurs de nécessité). Cette dichotomie est obtenu par distinction entre les termes qui sont peut-être représentatifs (en général, les termes apparaissant fréquemment dans un document) et ceux qui sont nécessairement représentatifs (un terme dans un document de grande valeur discriminante, c'est à dire apparaissant dans quelques documents dans la collection entière).

III.4 Implémentation de la relevance feedback par les réseaux bayésiens :

La technique de Relevance feedback(retour de pertinence) consiste à formuler automatiquement une nouvelle requête en fonction des jugements de pertinence retournés par l'utilisateur après évaluation d'un ensemble de document retournés, et cela en modifiant les poids des termes ou en ajoutant de nouveaux termes qui sont considérés comme utiles pour retourner des documents plus pertinents. Ce processus est répété jusqu'à satisfaction complète de l'ensemble des documents retournés. La figure présente le modèle de système de recherche d'information utilisant les réseaux bayésiens.

L'équipe de De compos [de campos,2003]a introduit plusieurs méthodes de relevance pour la RI par les RBs. Ces méthodes sont basés sur le concept des évidences partiels représentant les nouvelles informations obtenue après évaluation des résultats de la requête originale. Ces évidences seront introduites à travers le réseau et un nouveau processus d'inférence est exécuté pour calculer les probabilités à postériori de pertinence de documents de la collection. La qualité des méthodes proposées a été testée en effectuant des expérimentations avec différentes collections standards.

La figure suivante illustre le système de recherche d'information basé sur les réseaux bayésiens.

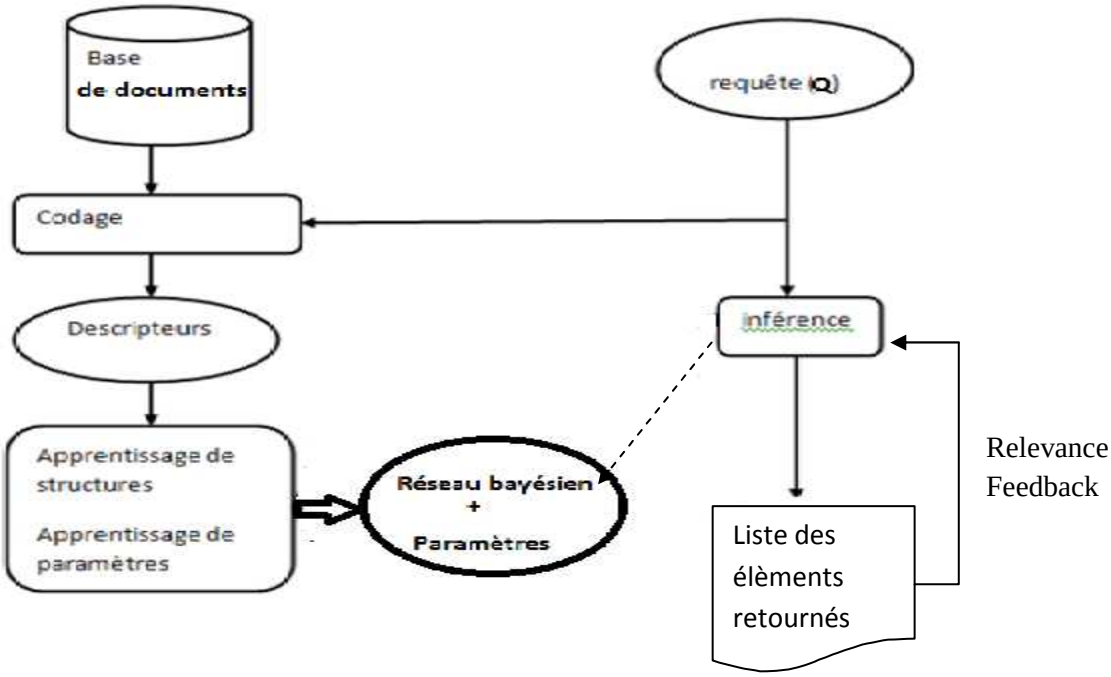


Figure III.12: système de recherche d'information utilisant les réseaux bayésiens

III-5 Conclusion

Les réseaux bayésiens sont des outils très pratiques et très efficaces pour la représentation de connaissances incertaines et le raisonnement à partir d'informations complètes ou incomplètes.

Un des apports majeurs du modèle de RI basé sur les réseaux bayésiens a été de généraliser les modèles classiques, à savoir les modèles booléens, probabilistes, et vectoriels.

L'utilisation des réseaux bayésiens en RI a révélé deux principaux problèmes:

1- 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 ;

2- 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.

Les réseaux bayésien ont été introduit aussi dans la recherche d'information structuré ces différents travaux seront décrits dans le chapitre suivant.

IV.1 Introduction :

Aujourd'hui, l'approche dominante pour manipuler les probabilités dans le domaine de l'intelligence artificielle est basé sur l'utilisation des réseaux bayésiens, et ceux-ci ont également été utilisés dans les RI comme des extensions des modèles classiques probabilistes.

Les réseaux bayésiens ont déjà été utilisés pour la recherche sur des documents plats et ont montré en particulier avec le système Inquery [Callan et al 1992] qu'ils permettaient de concevoir des systèmes efficaces.

Plusieurs auteurs se sont intéressés à leur introduction dans la recherche d'information structurées, estimant qu'ils offrent des modèles simples et naturels pour, à la fois, représenter la structure hiérarchique des documents XML et pour manipuler l'information incertaine inhérente à la recherche d'information de manière générale.
La section suivante présente les différents travaux ayant utilisé les réseaux bayésiens dans la recherche d'information structurée :

IV-2 Les modèles de RIS basés dur les réseaux bayésiens

IV-2-1 Les travaux de (Myaeng et al, 1998):

Les auteurs ont proposé une extension du modèle Inquery tel que chaque document[9] est représentée par une arborescence où chaque nœud représente une unité structurelle du document dont les termes sont les feuilles de l'arbre et le document entier est la racine (figure VI.1).

Le modèle permet à la fois de prendre en compte la structure dans une recherche documentaire simple aussi bien qu'une recherche documentaire à partir des questions structurées (comme par exemple trouver tous les documents comportant un titre et une section traitant de tel ou tel thème).

[9] L'approche s'est portée sur la recherche dans les documents structurés de type SGML

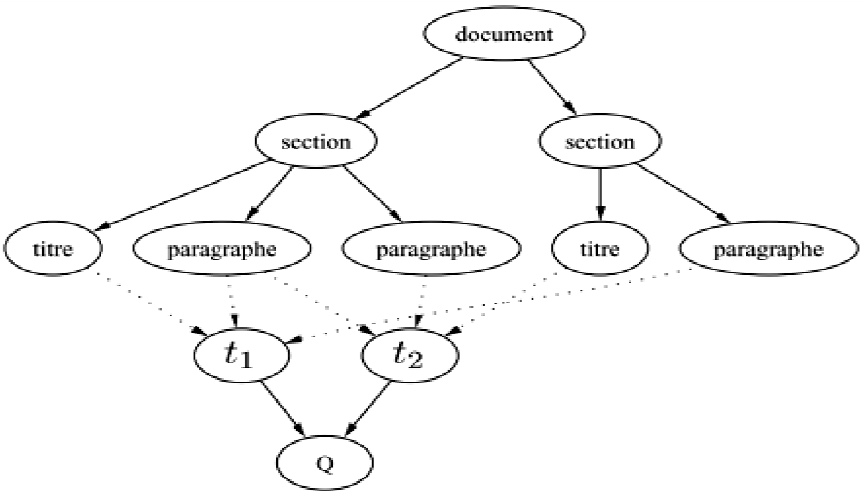


Figure VI.1-l'approche de Myaeng

L'évaluation démarre du nœud document jusqu'aux termes. A la réception d'une requête, on calcule pour tout document d la façon dont les termes de la requête "représentent bien" le document *D*, à savoir P(q/d) : la probabilité de chaque terme de la requête représentant le document est calculée, Pour obtenir cette probabilité, on calcule d'abord la probabilité qu'une section représente le document, puis la probabilité qu'un terme représente cette section et enfin la probabilité qu'une requête représente ce terme. Pour réduire la complexité du modèle, les auteurs ont émis des hypothèses de simplification.

En utilisant le formalisme bayésien, il suffit de connaitre les distributions $P(t/e_{1,\ldots,}e_n)$ *où* $e_1,..,e_n$ sont un ensemble d'éléments et $P(e|pa(e))$ est la probabilité d'observer l'élément *e* si on observe son parent *pa(e).*

Myaeng et al proposent dévaluer ces différentes probabilités de la façon suivante.
La probabilité qu'un terme t représente un élément e est proportionnelle à la fréquence du terme dans l'élément et à l'inverse de la fréquence documentaire comme suit :

$$P(\mathbf{t}|\mathbf{e}) \propto \frac{\mathbf{1}}{\mathrm{ef}_t(\mathcal{E})} \times \mathrm{tf}_t(e)$$

et

$$P(\mathbf{e}|\mathrm{pa}(\mathbf{e})) \propto \lambda(e, \mathrm{pa}(e)) \times \mathrm{l}(e) \times \cos(\overrightarrow{e}, \overrightarrow{\mathrm{pa}(e)})$$

Où λ *(e, pa(e))* : est un paramètre liée a l'importance du sous-élément structurel (fixée a 1 dans les expériences de Myaeng).
l(e) : est la longueur du texte associée à un élément structurel (nombre de mot dans l'élément structurel).

L'avantage de ce modèle est de permettre le traitement des requêtes structurées (CAS) qui posent des contraintes sur l'endroit de la structure de chaque document où doivent se trouver certains termes (titre de section, paragraphe, etc.).

IV-2-2 Les travaux (Piwowarski et al, 2002, 2003,2005) :

Dans [Piwowarski 02], [Piwowarski 03a] et [piwowarski,2005], la structure de réseau bayésien utilisée reflète directement la hiérarchie des documents. A chaque élément de la hiérarchie est associée une variable aléatoire qui peut prendre trois valeurs différentes dans l'ensemble V= {N, G, E}, où :

N: (pour Non pertinent) lorsque l'élément n'est pas pertinent;

G: (trop grand) lorsque l'élément est légèrement ou moyennement spécifique;

E :(pour exacte) lorsque l'élément a une spécificité élevée (pertinent).

Deux autres types de variables aléatoires sont considérés. Le premier est la requête, qui est représentée par un vecteur de fréquences de termes. Le second est associé aux modèles de pertinence utilisés pour évaluer la similarité locale de l'élément à la requête et peut prendre deux valeurs: *pertinent(R)* ou *non pertinent ($\bar{R}$)*

Topologie du modèle

Elle est illustrée par la figure VI.2.

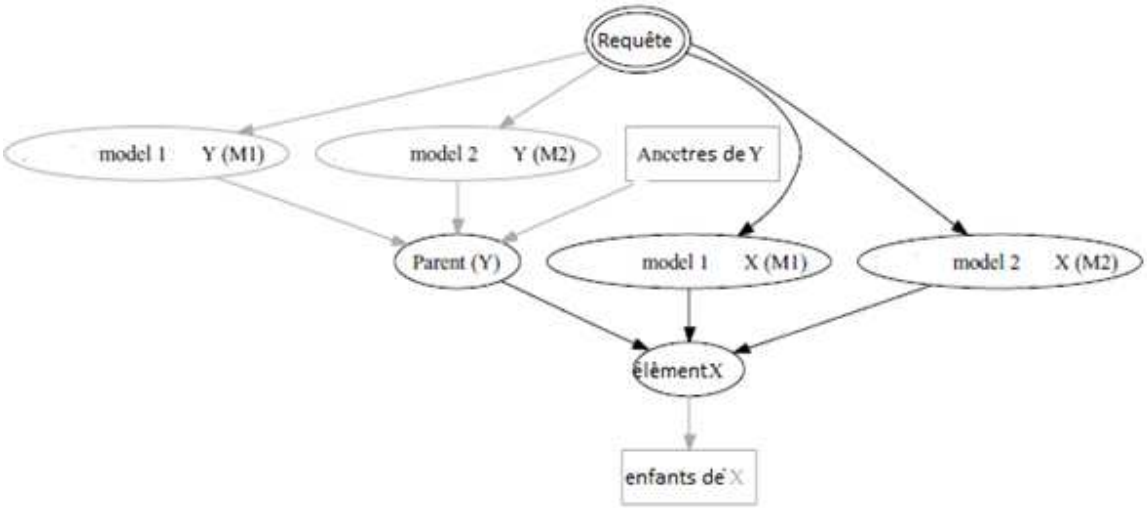


Figure IV.2 : une vue locale du RB : utilisation de deux modèles classiques M1 et M2

L'évaluation de la pertinence d'un elèments vis-à-vis d'une requête

Pour une requête donnée, un score local de pertinence est calculé pour chaque élément. Ce score dépend uniquement de la requête et du contenu de l'élément. Pour

calculer ce score local, plusieurs modèles peuvent être utilisés. Les auteurs ont proposés d'utiliser le modèle OKAPI et le modèle différentiel.

La probabilité qu'un élément soit dans l'état *N, G ou E* dépend ensuite de l'état de l'élément parent, et du jugement par le(s) modèle(s) de pondération utilisé(s) que l'élément est pertinent ou non pertinent

On a alors (si on considère deux modèles de baseM1 etM2 pour le calcul du score local de l'élément) :

$$P(X = v_x \mid q) = \sum_{\substack{v_Y \in V \\ m_1,\ldots,m_n \in \{R,\neg R\}}} P(X = v_X \mid Y = v_Y, M_1(X) = m_1, \ldots, M_n(X) = m_n) \times P(Y = v_y \mid q) \times P(M_1(X) = m_1) \times \cdots \times P(M_1(X) = m_1) \quad (1)$$

Où, $v_x \in V$, q est une requête composée de simples termes, et θ est un paramètre obtenu par apprentissage. Il dépend des différents états des quatre variables aléatoires (état de l'élément, état du parent, pertinence des modèles de base *M1 etM2*), et de la catégorie $c(e)$[10] de l'élément.

Les scores de pertinence sont calculés récursivement dans le réseau bayésien en commençant par la racine des documents. Le modèle est étendu au traitement des requêtes orientées contenu et structure.

IV-2-3 les travaux de l'equipe de Crestani dans [Crestani et al ,2003] et [Crestani et al ,2004]:

Les auteurs ont supposés que chaque document est composé d'une structure hiérarchique de niveaux abstraits *L1, L2. .. L_L* chacun représentant une association structurelle d'éléments dans le texte. Le niveau dans lequel le document lui-même est inclus est le niveau L1 et le niveau le plus spécifique est le niveau L_L.

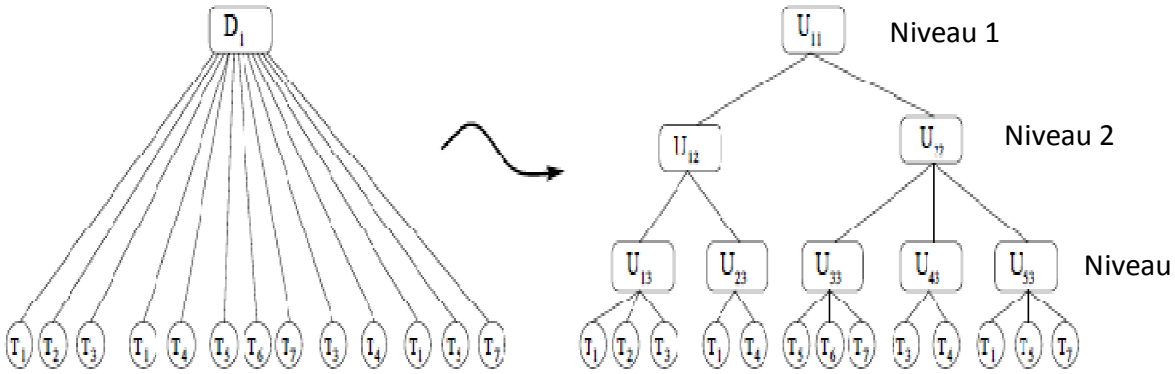
Chaque niveau contient des unités structurelles notée $U_{i,j}$ où i est l'identifiant de l'unité au niveau j. le nombre d'unités dans un niveau j est noté $|L_j|$ et $L_j=\{U_{1j},U_{2j},\ldots U_{|Lj|j}\}$les unités sont organisés suivant la structure du document.

Chaque unité *Uij* (excepté pour j=1) est inclue dans une seule unité $U_{z(i,j)^{11},j-1}$ du niveau *j-1*.

[10] Afin de limiter le nombre de paramètres à calculer, les éléments sont regroupés dans des catégories-chaque catégorie correspond à un ensemble de balises avec une sémantique similaire.

[11] $z(i; j)$ est la fonction qui retourne l'index de l'unité de niveau j contenant l'unité d'index i et de niveau j.

Topologie du modèle :
En tenant compte du modèle proposé en RI, les auteurs ont déduit le modèle présenté dans la figure (VI.3). Où seules les unités de niveau bas U_{1l},… $U_{|L_l|l}$ seront reliées aux termes qui les indexent.



FigureVI.3 :D'un document indexé (RI) à un document structuré indexé (RIS)

Ce graphe représente deux types de nœuds correspondant à deux types de variables aléatoires binaires :U_{ij} dont le domaine de valeurs est $\{u_{ij}$, $\bar{u}_{ij}\}$ désignant le fait que l'unité soit pertinente ou non. Il en est de même pour la variable T_k représentant un terme Tk qui prend ces valeurs dans l'ensemble{tk, $\overline{\text{tk}}$}.

les termes sont marginalement indépendants entre eux et les unités sont conditionnellement indépendantes. Ce modèle réspecte la topologie bayésienne avec *l*+1 niveaux et dont les arcs partent du niveau des termes vers celui des unités au niveau l et d'autres arcs partent des unités de niveau j vers des unités de niveau *j-1* pour *j=2…..l.* ainsi :

- $\forall T_k \in \mathcal{T}$, $Pa(T_k) = \emptyset$.
- $\forall U_{il} \in L_l$, $Pa(U_{il}) = \{T_k \in \mathcal{T} \mid U_{il}$ **est indexé par** $T_k\}$.
- $\forall j = 1, \ldots, l-1, \forall U_{ij} \in L_j$, $Pa(U_{ij}) = \{U_{kj+1} \in L_{j+1} \mid U_{kj+1} \subseteq U_{ij}\}$.

Un exemple de réseau bayésien multi niveaux à 3 niveaux est illustré dans la figureVI.4 :

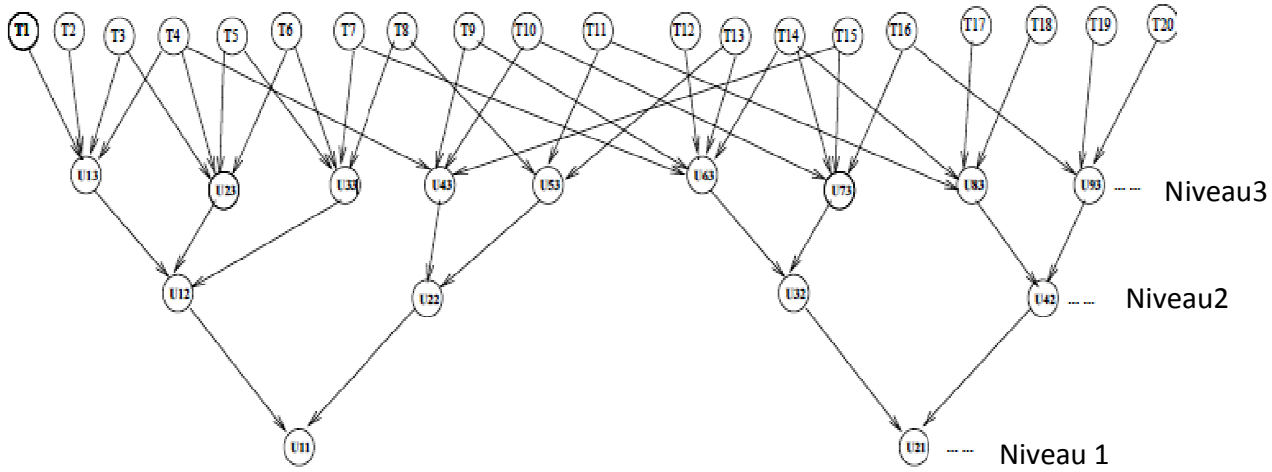


Figure VI.4 réseaux bayésien multi niveau

Les probabilités conditionnelles :

- Les noeuds termes T_k ont les mêmes probabilités marginales $\mathbf{P(t_k)=\frac{1}{M}}$ où M est le nombre des termes de la collection.

- **Les unités U_{il} de niveau l** :

$$\mathbf{P(u_{il}|\ pa(U_{il}))=\sum_{T_k \in R(pa(U_{il}))} w_{ki}}$$

Où

$R(pa(U_{il}))$ ={$T_k \in pa(U_{il})|t_k \in pa(U_{il})$} i.e les termes de $pa(U_{il})$ instanciés comme pertinents dans la configuration *de* $pa(U_{il})$.

W_{ki} : est le poids associé à chaque terme indexant l'unité U_{il}

- **Les unités structurelles de niveau j (j≠*l*)** :

pour estimer $P(u_{ij}|pa(U_{ij}))$ on a utilisé une mesure de similarité entre deux ensembles de termes, l'un l'unité U_{ij} et l'autre est associé aux unités contenues dans U_{ij} instanciés comme pertinents dans la configuration $pa(U_{ij})$.

Ainsi que $P(u_{ij}|pa(U_{ij}))$ est définie ainsi :

$$\mathbf{P(u_{ij}|pa(U_{ij}))=\sum_{U_{hj+1} \in R(pa(U_{ij}))} P^{j}_{hi}}$$

Où le poids, P^{j}_{hi} de l'unité U_{hj+1} dans l'unité Uij est définie par :

$$P^{j}_{hi} = \frac{\sum_{T_k \in A(U_{hj+1})} w^{j+1}_{kh}}{\sum_{T_k \in A(U_{ij})} w^{j}_{ki}}$$

Où $A(U_{ij})$ et $A(U_{hj+1})$ sont les ensembles de termes utilisés pour indexer U_{ij} et $pa(U_{ij})$ respectivement[12] .

$$w_{ki}^{j}=tf_{ki}^{j}.\mathbf{idf_k}$$

Avec

tf_{ki}^{j} : est la fréquence du terme t_k dans l'unité U_{ij}

idf_k : est la fréquence documentaire inverse du terme tk dans toutes la collection.

$w_{kh}^{j+1}=tf_{kh}^{j+1}.\mathrm{idf_k}$

$avec\ tf_{kh}^{j+1}$: est la fréquence du terme tk dans $pa(U_{ij})$

L'inférence dans le modèle :

Le processus d'inférence consiste à calculer la probabilité à posteriori de pertinence de toutes les unités structurelles étant donnés la requête $P(u_{ij}|Q)$. On distingue les deux probabilités suivantes:

- Pour les unités de niveau L_l :

$$\mathbf{P(u_{il}|Q)}=\sum_{T_k\in pa(U_{il})\cap Q} w_{ki} + \frac{1}{M}\sum_{T_k\in(pa(U_{il}\setminus Q)} w_{ki}$$

- Pour les unités de niveau Lj, $\mathbf{j\neq l}$:

$$\mathbf{P(u_{ij}|Q)} =\sum_{U_{hj+1}\in pa(U_{ij})} P_{hi}^{j}.P(u_{hj+1}|Q)$$

Et ainsi on peut calculer les probabilités niveau après niveau en commençant par le niveau ***l*** et aller jusqu'au niveau 1.

IV-2-4 Les travaux de L'équipe de De Campos[L. M. de Campos and al,2003]

L'équipe a implémenté le système Garnata qui est un système de recherche d'information conçu pour implémenter des modèles graphiques probabilistes. Même s'il traite des requêtes orientés contenu, il peut tout autant utiliser des requêtes orientés structure et contenu. Son système d'indexation utlise l'approche classique des fichiers inverses contenant les occurrences des termes des balises xml ou toute autre donnée structurelle pour représenter l'organisation interne des documents.

[12] En réalité l'unité de niveau j≠1 n'est indexé par aucun terme, on se réfère aux termes indexant l'unité de niveau l incluant soit l'unité U_{ijj} ou les unités de $R(U_{ij})$.

Topologie du modèle

Le modèle utilisé est celui décrit dans [**[Crestani et al ,2003].**

- **Les nœuds virtuels**

Initialement, on a supposé que chaque unité structurelle était composée d'autres unités structurelles sauf l'unité terminale qui ne contient que du texte mais réellement on peut trouver des unités contenant du texte et d'autres unités.

Exemple : Soit l'élèment mixte suivant :
<Paragraphe> un exemple de paragraphe avec <Remarque> une information supplémentaire<\Remarque> **<\paragraphe>**

Pour résoudre ce cas, le modèle a inclus des nœuds speciaux dits nœuds virtuels qui seront des parents de l'unité concernée par ce cas(« paragraphe » dans l'exemple) avec les unités inclus come l'illustre la figure IV.5. Ces nœuds virtuels seront des noeuds terminaux contenant du texte.

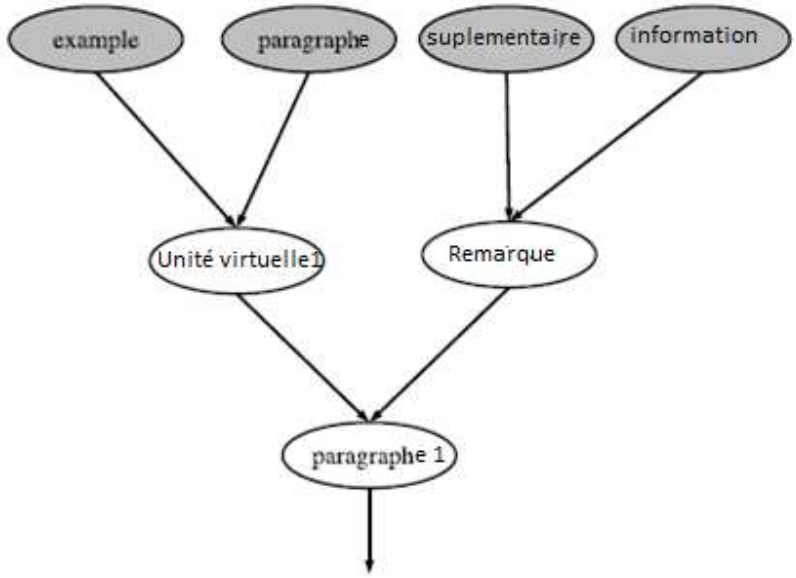


Figure IV.5: un fragment de reseau bayesien contenant une unité virtuelle.

Remarques

- Les unités virtuels ne devraient pas être retournés à l'utilisateur
- L'expérimentation a été faite sur la collection « Shakespeare »[13] contenant les pièces de Shakespeare et la collection inex . L'expérimentation sur le temps

[13] http://qmir.dcs.qmw.ac.uk/Focus/collbuilding.htm

d'exécution et la capacité de stockage de l'indexation ainsi que le temps des recherches d'information a révélé une performance acceptable.

- A cause de l'aspect experimental du système plusieurs formules de pondération peuvent etre appliqués. Ces differentes valeurs de pondération précalculées sont stockés dans des fichiers dits fichiers de pondération. un moyen rapide pour en insérer une dans l'index lui-même ce qui consommera un temps de lecture sur disque.

IV-2-5 l'approche proposée par Alimazighi dans [alimazighi et al 2005]

Topologie du modèle:
Le modèle propose d'utiliser la structure d'un document XML afin de définir la structure du réseau bayésien. Par conséquent, les nœuds ou variables de ce réseau sont les éléments structurels. Une autre particularité dans ce modèle est l'association à chaque nœud du réseau une matrice contenant les termes et leurs poids.

Calcul des paramètres du réseau:
Le but de cette étape est de calculer la probabilité conditionnelle de chaque nœud connaissant ses Parents $\mathbf{P}(v_n/pa(v_n))$

Avec *Vn*: variable aléatoire nœud n.
Vpa (n): variable aléatoire du parent du nœud n.

En utilisant la loi de probabilité de BAYES nous avons :

$$\mathbf{P(A/B)}=\frac{(A\cap B)}{P(B)}$$

Et si $A\subset B$ alors :

$$P(A\cap B) = P(A)$$

$$P(A/B) = \frac{P(A\cap B)}{P(B)} = \frac{P(A)}{P(B)}$$

Dans ce cas chaque noeud est inclus dans son parent alors la formule précédente s'écrit comme suit :

$$\mathbf{P}(v_n/pa(v_n)) = \frac{P(v_n)}{P(pa(v_n)}$$

Donc :

$$\mathbf{P}(v_n/pa(v_n)) = \frac{Card(v_n)}{Card(pa(v_n)}$$

Avec *card(v_n)* : le nombre de terme ti dans le nœud n.
card(pa(v_n)) : le nombre de terme dans le parent du nœud n.

L'inférence dans ce réseau :

Le modèle proposé est capable de faire face aux exigences d'information reliant structure et contenu.

❖ **Recherche sur le contenu :**

Les auteurs ont proposé un algorithme de recherche est décrit comme suit :

1- commencer la recherche au niveau du nœud racine. Si le terme recherché est dans la matrice associé au nœud racine aller dans la recherche, sinon arrêter.

2- Passer aux nœuds suivants et lancer la recherche. Si le terme est dans la matrice associée à chaque nœud, sélectionnez le nœud (s) répondant mieux à la requête (sélectionnez le nœud (s) ayant la probabilité conditionnelle maximale.

3- Répétez 2 pour le reste des nœuds du réseau jusqu'à ce que le but soit atteint.

❖ **Recherche avec structure** :

La particularité de cette recherche est que, dans la structure du réseau bayésien, la requête peut dépendre d'un nœud de ce réseau.

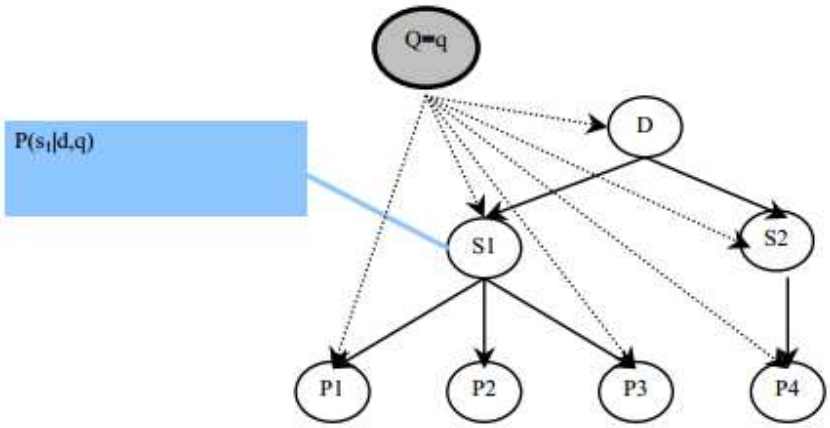


Figure VI.7 : Exemple de recherche avec structure

L'Algorithme de recherche proposé est le suivant:

1- Définir les nœuds liés à l'élément de structure à retourner.
2- Rechercher le terme dans la matrice associé à chaque nœud.
   - Si le terme recherché se trouve dans la matrice associé, sélectionnez donc les nœuds rependant mieux à la requête (les nœuds ayant la probabilité conditionnelle maximale).
   - sinon aucun nœud ne répond à la requête.

IV-2-6 Le modèle possibiliste où Les travaux de [BESSAI et al] :

Topologie du modèle proposé :

Le modèle proposé est représenté par un réseau possibiliste d'architecture illustrée par la figure (VI.) sachant qu'à chaque nœud est associé une variable aléatoire binaire.

Les nœuds documents représentent les documents de la collection. L'instantiation Di=di signifie que 'le document *Di* est pertinent pour une requête donnée', et ¬*di* signifie que le document *Di* n'est pas pertinent pour la requête donnée.

Les nœuds E1, E2, …En, représentent les balises du document Di. L'instanciation *Ei = ei* signifie que l'élément *Ei* est pertinent pour la requête; *Ei = ¬ei* signifie que l'élément '*Ei*' est non pertinent pour la requête.
Les nœuds *T1, T2, …, Tm* sont les nœuds termes où L'instanciation *Ti = ti* signifie que le terme '*Ti*' est représentatif du nœud père auquel il est rattaché,
Ti = ¬ti signifie que le terme '*Ti*' est non représentatif du nœud père auquel il est relié.
Un terme est relié aussi bien au nœud qui le comporte ainsi qu'à tous les ascendants de ce dernier

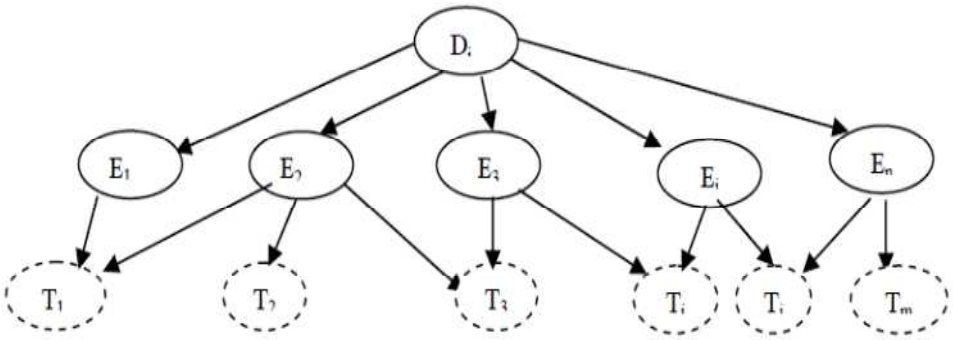


Figure VI.8 Architecture du modèle

Evaluation d'une requête par propagation

L'évaluation d'une requête consiste à injecter une nouvelle évidence à travers les arcs activés du réseau pour rechercher les documents et les éléments pertinents par rapport à la requête. La pertinence est modélisée selon deux dimensions : la nécessité et la possibilité de pertinence notée ainsi :

Π(di/Q) : est il possible à un certain degré ou non que le document *Di* soit pertinent pour la la requête *Q*.

N(di/Q) :est il certain ou non que le document di soit pertinent pour la requête Q

Ici les auteurs ont supposés les hypothèses suivantes :

- Un document a autant de possibilité d'être pertinent que non pertinent pour un utilisateur donné, soit Π(di) = Π (¬ di) = 1, quelque soit i.

- La requête est composée d'une simple liste de mots-clés (interprétée de manière conjonctive) Q= {t1∧t2∧ .. ∧tl}. L'importance relative des termes entre eux dans la requête est ignorée.

$$\Pi(d_i/Q)=\frac{\Pi(Q\wedge d_i)}{\Pi(Q)} \quad et \quad N(d_i/Q)=1-\Pi(\neg d_i/Q)$$

En considérant la quantité *Π* (*Q*) indépendante des documents et étant donné l'architecture du modèle:

$$\Pi(Q\wedge d_i)=\max_{\theta^e\in\theta^E}(\Pi(Q/\theta^e)*\Pi(\theta^e/d_i)*\Pi(d_i)) \quad [1]$$

Ou θ^e représente une instanciation possible des variables balises, c'est-à-dire les parents des termes de la requête parmi toutes les configurations possibles définies par θ^E. De plus les auteurs supposent que les variables termes et les variables balises sont indépendantes. L'équation [1] devient alors:

$$\Pi(Q\wedge d_i)=\max_{\forall\theta^e\in\theta^E}(Prod(\underset{E_j\in\theta^e}{Prod}(\underset{T_i\in T(E)\wedge T(Q)}{Prod}(\Pi(t_i/\theta_j^e)))*\underset{E_j\in\theta^e}{Prod}(\Pi(\theta_j^e/d_i))*\Pi(d_i)) \quad [2]$$

Où *Prod*: signifie produit
Π : désigne la possibilité.
ti ∈*T(E)* ∧ *T(Q)* : représente les termes de la requêtes qui indexent les balises.
θ_j^e: Représente l'instance de Ej dans la configuration θ^e

Détermination des Valeurs des arcs noeud balise- noeud terme :

- La possibilité de pertinence d'un terme (ti) pour représenter un élément (ej), notée Π *(ti /ej)*, est calculée comme suit:

$$\Pi(t_i/e_j)=tf_{ij}/\max_{\forall t_k\in e_j}(tf_{kj})$$

Tf_{ij}: est la fréquence du terme ti dans l'élément ej.

Max(tf $_{kj}$) : est la fréquence maximal des termes dans l'élément ej.

Par contre un degré de nécessaire pertinence, βij, du terme ti pour représenter l'élément ej, sera défini par :

$$N(t_i \rightarrow e_j) \geq \beta_{ij} = \mu(tf_{ij} * ief_{ij}) * idf = \mu(tf_{ij} * \log(\frac{Ne}{ne_i}) * \log(\frac{N}{n_i}))$$

Avec, *N et Ne* : respectivement le nombre de document et d'éléments dans la collection ; *ni et nei* : respectivement le nombre le document et le nombre d'éléments contenant le terme ti.
μ: est une fonction de normalisation

Valeur de l'arc noeud document – noeud balise (propagation de pertinence) :
Le degré de possibilité de propagation d'une balise (ej) vers le noeud document di est défini par Π (ej / di) et est quantifié comme suit:

$$\prod(e_j / d_i) = \alpha^{dist(di,\ ej)-1}$$

Avec :
- *dist(di, ej)* la distance de la balise *ej* à la racine *di* conformément à la structure hiérarchique du document.
- $\alpha \in]0..1]$ est un paramètre permettant de quantifier l'importance de la distance séparant

Les auteurs ont proposés de calculer la nécessité de propagation de pertinence d'un élément *ej* vers le nœud racine *dj* ainsi :

$$N(e_j \rightarrow d_i) = 1 - \frac{le_j}{dl}$$

Avec le_j : la taille du noeud balise (élément structurel) e_j
dl : la taille d'un document (en nombre de termes).

D'après la formule, plus un terme est de taille petite plus la nécessité de le propager est grande. Par conséquent

$$\prod(e_j / \neg d_i) = le_j / dl$$

IV-2-7le modèle d'agrégation où l'approche proposé dans [Najeh naffakhi 2010] :

Topologie du modèle proposé

On a proposé un modèle pour la RIS basé sur les réseaux bayésien dont la topologie illustré par la figure(VI.9).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 d'un document XML.

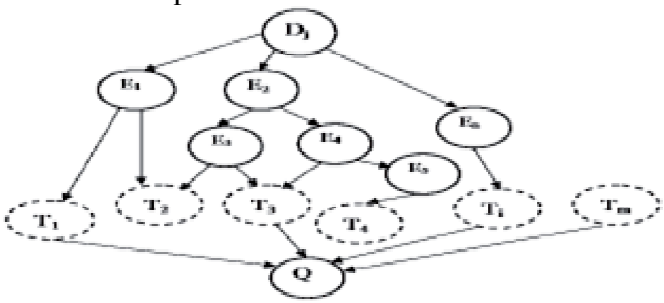


Figure VI.9 *Architecture générale du modèle proposé par (***Naffakhi najeh et al ,2009***)*

Dans ce modèle dit modèle d'agrégation,, la requête de l'utilisateur démarre un processus de propagation pour récupérer et agréger les éléments XML. Ainsi, au lieu de récupérer un document entier ou une liste d'éléments disjoints qui sont susceptibles de répondre partiellement à la requête, on essaye de construire un document virtuel(dit document composite) qui regroupe un ensemble d'éléments, qui sont pertinentes tous ensemble et complémentaires(non redondants)

T*opologie du modèle bayésien*

Le nœud Dj représente un document de la collection C. Chaque nœud *Dj* représente une variable aléatoire binaire. L'instanciation *Dj = 1* signifie que le document est activé (choisi).

Les nœuds E_1, E_2,..., E_n représentent les éléments du document Dj.
Chaque nœud *Ej* représente une variable aléatoire prenant des valeurs binaires dans l'ensemble dom(*Ej) = {1, 0}.*
L'instanciation *Ej = 1* signifie que l'élément *Ej* est indexé par au moins un nœud terme.

Les nœuds *T1, T2, ..., Tm* sont les nœuds termes. Chaque nœud terme *Ti* représente une variable aléatoire binaire prenant des valeurs dans l'ensemble dom(Ti) = {1, 0} où l'instanciation *Ti =1* signifie que le terme *Ti* est présent dans nœud père auquel il est relié(le nœud balise *Ej* contient ce terme *Ti* ou requête Q). Un terme est relié aussi bien au nœud qui le comporte qu'à tous les ascendants de ce dernier.

Les auteurs supposent que la requête Q est composée d'une simple liste de mots-clés : Q = {T1, ..., Tm}. Et la notation suivante est utilisée :

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

Les termes de la requête qui indexent les éléments de documents, Ti∈(T(Q) ∧T(E)), sont évalués dans le contexte de leurs parents par P(Ti|Ej), et séparés des termes de la requête absents des éléments de documents.

Processus génératif du modèle :

Le processus génératif correspondant est une application récursive du processus suivant :

1) 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 représente la requête dans sa forme la plus stricte (exactement telle que formulée par l'utilisateur).
Soit $\theta^T(Q)$ cette instanciation. L'ensemble des instances possibles des parents de la requête est noté θ^Q.

2) Générer l'ensemble des configurations possibles θ d'un résultat de recherche en identifiant une configuration θi pertinente. Cette configuration est définie par : $\theta_i = \{\theta_i^{j,id}\}_{j=1,...,z}$, $\theta_i^{j,id}$ est l'élément Ej d'un document identifié par son attribut id, j est la valeur pré-ordre assignée en effectuant un parcours séquentiel préfixé de la représentation en arbre du document structuré et i le numéro de la configuration θi, sous les trois contraintes qui permettent de vérifier la non redondance d'information dans une configuration et la complémentarité de leurs élèments:

- $P(\theta_i^{j,id}, Pa(\theta_i^{j,id})) = 0$ (1)
- $(T(\theta_i^{j,id}) \wedge T(\theta_i^{j',id'}) \wedge T(Q)) \subset T(Q)$ (2)
- $(T(\theta_i^{j,id}) \wedge T(Q)) \wedge (T(\theta_i^{j',id'}) \wedge T(Q)) = \oslash$ (3)

Avec Pa($\theta_i^{j,id}$) est l'élément parent du nœud associé à $\theta_i^{j,id}$

3) la probabilité jointe d'observer une requête Q et sa réponse dans un document Dj est donnée par :

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

- En considérant le premier facteur, P(Q|T(Q)), est la probabilité de la requête étant donnée ces termes. Cette probabilité est calculée suivant cette formule :

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

Avec P(Q|Tk)=1 pour tout terme∀Tk ∈(T(Q) ∧T(E)) et 0 sinon.

- Le deuxième facteur de la formule [1] mesure la pertinence est calculé comme suit :

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

avec

$$P(T_k = 1|\theta_i) = \frac{\sum_{\forall \theta_i^{j,id} \in \theta_i} tf_k^{j,id}}{\boldsymbol{tf_d}}$$

Où :

$\sum_{\forall \theta_i^{j,id} \in \theta_i} tf_k^{j,id}$: est la fréquence du terme tk dans une configuration θ_i.

$\boldsymbol{tf_d}$: est la fréquence du terme dans un document d.

- Le troisième facteur de la formule [1] est P(θi|Dj=1), mesure la complémentarité entre les éléments d'une configuration possible. Ces éléments sont indépendants, la formule précédente s'écrit comme suit :

$$P(\theta_i|D_j=1)= \prod_{j=1}^{|\theta_i|} P(\theta_i^j|D_j=1)$$

Avec

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

(D_j,θ_i^j): est la distance entre le nœud racine Dj et un de ces nœuds descendant θ_i^j du document (relativement à une configuration donnée θi).

dist(D_j ,élément plus profond(θ_i^k)):la distance entre le nœud racine Dj et le plus profond élément muni du nœud θ_i^j est noté θ_i^k

La distance entre deux nœuds quelconques est déterminée par le nombre d'arcs les séparant.

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

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

4) Finalement, générer le document composite qui correspond à la configuration qui sera celle qui comporte les termes de la requête et celle qui maximise la pertinence et la complémentarité en termes de probabilité donc celle qui optimise la formule suivante :

$$\underset{\theta_i^* \in \theta}{argmax} \left(\prod_{T_k \in T(Q)} P(Q|T_k) \times \prod_{T_k \in (T(Q) \wedge T(\theta_i))} P(T_k|\theta_i) \times \prod_{j=1}^{|\theta_i|} P(\theta_i^j|D_j=1) \right)$$

- Cette proposition, permet de générer un seul document composite, mais sans restriction sur le nombre de nœuds résultats que ce document peut contenir.

IV-3 Synthèse:

Bien que les réseaux bayésiens ont d'abord été appliquées à la RI vers la fin des années 80, ils ont été plus largement utilisé dans la prochaine décennie qui a suivi la naissance du modèle de réseau d'inférence [turtle et croft, 1991]. Depuis lors, de nombreux modèles et applications ont été développées, en montrant que ces modèles graphiques et probabiliste sont adaptés pour être utilisés dans la RI. Les travaux portant sur la RIS sont assez récents.

Cette section a montré qu'une modélisation du processus de RIS peut être obtenue en utilisant le formalisme des réseaux bayésiens. En associant des probabilités initiales pour les racines du graphe, on calcule de proche en proche le degré de croyance associé à chacun des nœuds restants. Les réseaux bayésiens offrent ainsi un cadre naturel pour modéliser des données structurées et pour faire de l'inférence sur des corpus de données structurées.

Les modèles de RIS, utilisant les Réseaux bayésiens, proposés différent par :

- le nombre de sous réseaux :
- l'orientation des arcs
- La modélisation de l'indépendance ou dépendance entre les nœuds
- le type de calcul des paramètres(les probabilités conditionnelles)
- les types de requêtes traités (CO ou CAS)

- les modèles étudiés représentent des arcs allant des documents jusqu'à leurs termes à l'exception de ceux de Crestani [Crestani et al, 2004] dont les arcs des réseaux sont dirigés des termes vers les documents. les relations de dépendance ou d'indépendances entre les nœuds sont exprimées au moyen de présence ou d'absence de liaisons entre les nœuds dans le graphe.
- Dans les travaux que nous avons présenté, Certains auteurs ont modélisé chaque document par un réseau bayésien (piwowarski et al 2002,2003a) d'autres ont représenté par un réseau bayésien tout le corpus (le système Granata). Le système garnata [De camposand al,2003] a introduit des nœuds virtuels pour représenter les éléments mixtes.
- Toutes les approches ont utilisé des mesures de probabilités à l'exception du modèle possibiliste qui a utilisé deux mesures de calcul de l'incertain qui sont les mesures de plausibilité d'un événement et celui de sa certitude. L'utilisation de la théorie des Possibilités permet au modèle de séparer les motifs du rejet d'un document comme

non pertinentes (en tenant en compte des valeurs de possibilité) des motifs de la sélection d'un document pertinent (au moyen des valeurs de nécessité). Cette dichotomie est obtenu par distinction entre les termes qui sont peut-être représentantatifs (en général, les termes apparaissant fréquemment dans un document) et ceux qui sont nécessairement représentatifs (un terme dans un document de grande valeur discriminante, c'est à dire apparaissant dans quelques documents dans la collection entière).

- Dans le modèle de (piwowarski, 2002), on a utilisé trois types de scores (ou trois mesures de probabilités : élément exact, élément trop grand et élément non pertinent) qui peuvent être de nouveaux critères d'évaluation, mieux adaptés aux SRI pour filtrer les listes résultats retournant les éléments.

- Pour le calcul des paramètres L'autre particularité du modèle est l'apprentissage des paramètres qui permet d'adapter ce modèle par n'importe quelle collection. Malgré son efficacité, il présente une complexité dans l'application de l'algorithme d'apprentissage et de l'inférence. Ces difficultés sont liées d'une part au fait que les documents ayant en général des structures différentes, les arbres qui les représentent sont d'arité et de profondeur variables, ce qui rend complexe le calcul des probabilités conditionnelles en chaque noeud et le partage de paramètres. Dans l'approche de (Alimazighi, 2005), ces paramètres sont stockés dans des tables de calcul probabilités conditionnelle ce qui optimisera le temps pendant le processus d'inférence au détriment de l'espace mémoire qui sera nécessaire pour le stockage des paramètres. D'autres modèles ont utilisé les fonctions de calcul de probabilités tel que le modèle d'agrégation. Ces paramètres sont calculés au cours du processus de l'inférence pour optimiser l'espace de stockage au détriment d'une consommation du temps de leur calcul durant la propagation de l'information.
- Pour le traitement de la requête, certaines approches ont pris en considération la formulation de la requête par le contenu et la structure telle que dans (piwowarski, Alimazighi). D'autres approches n'ont traité que les requêtes de type CO (ex modèle d'agrégation).
- Une nouvelle approche de sortie de résultats a été appréhendée dans le modèle d'agrégation qui consiste à rassembler l'ensemble des éléments pertinents dans un document dit document composite qui sera retourné à l'utilisateur (au lieu retourner les éléments individuellement). L'avantage, pour l'utilisateur, est non seulement de pouvoir obtenir les paries d'informations demandées mais de les voir affiché tous à la fois (le document composite). L'inconvénient est que la combinaison d'éléments retournés peut contenir des éléments figurant dans des classes différentes; ce qui risque d'embrouiller l'utilisateur lors de sa lecture.

- Enfin nous avons remarqué qu'aucun des travaux n'a pris en considération les liens entre termes ou document ou traité le processus de relevance Feedback où les RBs serait d'un grand apport

Conclusion

Dans l'ensemble, les réseaux bayésien fournissent un outil graphique très intuitif pour représenter les connaissances disponibles. En effet, les réseaux bayésien montrent une grande capacité de représentation, face à des problèmes imprégnée de l'incertitude comme la RI.

Les réseaux bayésien permettent également d'effectuer l'inférence probabiliste (calcul des probabilités a posteriori) de manière efficace dans de nombreux cas. De plus le développement des techniques d'apprentissage offre également la possibilité de régler automatiquement les modèles de paramètres ou de détecter les dépendances entre les variables du modèle.

Un autre avantage des réseaux bayésien est que nous avons plus d'informations sur la pertinence de chaque élément qu'avec n'importe quel autre modèle (ex le modèle proposé piwowarski a introduit trois valeurs de mesure différentes de pertinence)

Mais à cause du volume très important d'informations représentés, l'inférence reste une tâche très lourde[14] et ce, malgré l'existence de mécanisme d'inférence très puissants qui sont développés. Les nombreux travaux de RIS par les RBs, et selon leur contexte, ont apporté leur contribution pour surmonter cet inconvénient.

[14] Consulter le rapport (work shop) 2005 sur les modèles graphique en recherche d'information

IV.4 Contribution :

La topologie du réseau bayésien que nous proposons est :

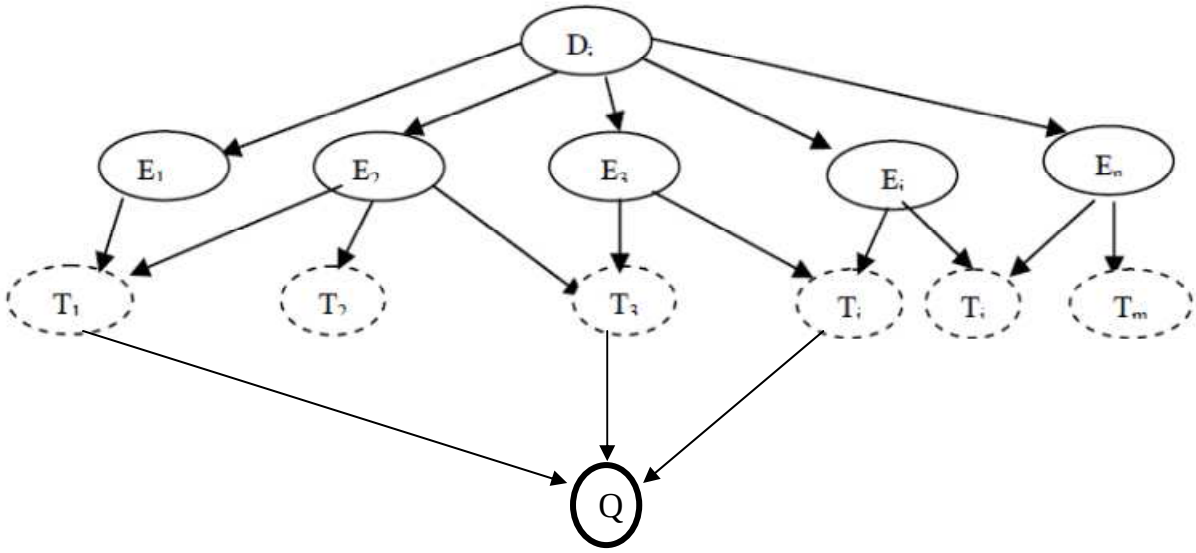


Figure IV.1 topologie du réseau bayésien

Cette topologie est justifiée par le fait que :

- chaque élément dans un document a été défini pour décrire ou expliquer une notion sémantique propre à l'élément lui-même.

  Exemple : un article sur la **recherche d'information** peut contenir deux paragraphes : **définition de la recherche et définition de l'information** dont chacun a un contenu qui est propre à lui. En d'autres termes la sémantique du contenu véhiculé par paragraphe 1 est différente de celle de paragraphe2 et même de section. D'où leur indépendance.

- Ce modèle permettra de réduire le temps d'inférence de propagation de l'information sachant que l'inconvénient dans les réseaux bayésiens est le nombre important de calculs qu'il implique.

Remarque : Nous utiliserons la structure hiérarchique du document (son arbre DOM) pour évaluer , précisément, plusieurs calculs de probabilités.

IV.4.1 Description du modèle :

A chaque nœud du réseau est associé une variable aléatoire binaire prenant leurs valeur dans leur domaine {1,0} respectifs .Le nœud Dj représente un document de la collection C. L'instanciation Dj= 1 signifie que le document est activé (choisi). Nous

ne nous intéressons qu'au cas où le document Dj est activé, et nous le notons dj .Les nœuds E_1, E_2,..., E_n représentent les éléments du document Dj. L'instanciation Ej = 1 signifie que l'élément Ej est pertinent par rapport à la requête Q.

Les nœuds *T1, T2, ..., Tm* sont les nœuds termes où l'instanciation Ti = 1 signifie que le terme Ti est présent dans l'objet (nœud père auquel il est relié c.-à-d. le nœud balise Ej, ou requête Q, contient ce terme Ti) et donc représentatif de l'objet.

Le passage du document vers la représentation sous forme de réseau bayésien consiste à ramener tous les nœuds (balises du document) au niveau des variables Ei.
Nous notons par T(E) (resp. T (Q)) l'ensemble des termes d'indexation des éléments du document (resp. de la requête).

Nous considérons l'hypothèse que les termes ainsi que les éléments sont indépendants entre eux. La requête étant exprimée sous forme d'une liste de mots clés (requête orientée CO).

IV.4.2 Pertinence d'un document vis-à-vis d'une requête :

$$P(di/Q) = \frac{P(Q \wedge di)}{P(Q)}$$

La valeur du numérateur est constant (c'est la même valeur quelque soit le document choisi), donc :

$$\mathbf{P(di/Q_i)} \ \boldsymbol{\alpha}\ \mathbf{P(Q \wedge d_i)}$$

$$\mathbf{P(Q \wedge d_i)} = \sum_{\boldsymbol{\theta^l}, \boldsymbol{\theta^e}} \mathbf{P}(\mathbf{Q}/\boldsymbol{\theta^l})\mathbf{P}(\boldsymbol{\theta^l}/\boldsymbol{\theta^e}) * \mathbf{P}(\boldsymbol{\theta^e}/\mathbf{di}) * \mathbf{P}(\mathbf{di}) \qquad [\mathbf{1}]$$

$\boldsymbol{\theta^l}$ est une configuration possible des termes de la requête

$\boldsymbol{\theta^e}$ est une combinaison parmi toutes les combinaisons possible des variables éléments (les parents des termes de la requête qui indexent les balises). Ti ∈T(E) ^ T(Q) : représente les termes de la requêtes qui indexent les balises),
$\boldsymbol{\theta_j^e}$ représente un élément E_j dans la configuration $\boldsymbol{\theta^e}$

Remarque: l'inconvénient des RBs est le nombre important de calculs nécessaire pour déterminer les différents paramètres et calculer la pertinence d'un élément vis-à-vis d'une requête.

Au lieu de calculer la pertinence d'un élèment, Notre approche consiste à définir la meilleure configuration d'éléments (la plus pertinente) satisfaisant la requête de l'utilisateur à savoir :

Meilleur Configuration d'élèments répondant à la requête est :

Elle sera obtenue par la formule suivante :

$$\mathbf{P(Q \wedge d_i)} = \boldsymbol{arg\ max_{\theta^e}}\left(\left(\sum_{\boldsymbol{\theta^l}} \mathbf{P}(\mathbf{Q}/\boldsymbol{\theta^l})\mathbf{P}(\boldsymbol{\theta^l}/\boldsymbol{\theta^e})\right) * \mathbf{P}(\boldsymbol{\theta^e}/\mathbf{di}) * \mathbf{P}(\mathbf{di})\right) \quad \textbf{[3]}$$

IV.4.3 Apprentissage du modèle :

Dans notre modèle, la structure du RB étant connue (figure 1), L'évaluation de la requête est effectuée par la propagation de l'information apportée par la requête à travers le réseau.
le problème est d'estimer les paramètres de ce réseau, c'est-a-dire les tables de probabilités conditionnelles associée à chaque nœud suivant une configuration donnée. Cette tache est proche de celle consistant à estimer les paramètres $\mathbf{P}(\mathbf{Q}/\boldsymbol{\theta^l})$, $\mathbf{P}(\boldsymbol{\theta^l}/\boldsymbol{\theta^e})$, $\mathbf{P}(\boldsymbol{\theta^e}/\mathbf{di})$ de notre modèle. Il s'agit de déterminer les valeurs des relations requête-termes, termes éléments et éléments-racine.

- **Calcul de la probabilité à priori $\mathbf{P(d_i)}$** : **N**ous considérons qu'un document D_i instancié à 1 vérifie :

$$P(d_i)=1 \text{ et } P(\neg di)=0$$

Cette valeur étant la même pour chaque document Di du corpus, on pourra l'éliminer de la formule[3] (en plus du fait qu'elle soit neutre pour le produit)

$$\mathbf{P(Q \wedge d_i)} = \boldsymbol{arg\ max_{\theta^e}}\left(\left(\sum_{\boldsymbol{\theta^l}} \mathbf{P}(\mathbf{Q}/\boldsymbol{\theta^l})\mathbf{P}(\boldsymbol{\theta^l}/\boldsymbol{\theta^e})\right) * \mathbf{P}(\boldsymbol{\theta^e}/\mathbf{di})\right) \quad \textbf{[4]}$$

- **Calcul de $\mathbf{P}(\mathbf{Q}/\boldsymbol{\theta^Q})$** : elle sera estimée, selon la nature de la requête. Nous distinguons :

La requête conjonctive : dans le cas d'une requête booléenne ET, tous les termes d'une configuration doivent être instanciés comme dans la requête :

$$\mathbf{P}((\mathbf{Q}/\boldsymbol{\theta^l}) = \begin{cases} \mathbf{1} & \textbf{si } \boldsymbol{\theta_i^Q} = \boldsymbol{\theta_i^l}\ \forall\ \mathbf{T_i} \in \boldsymbol{parent(Q)} \\ \mathbf{0} & \textbf{sinon} \end{cases}$$

Exemple :

soit la requête conjonctive composée des termes T1 et T2

La probabilité P(Q/ $\boldsymbol{\theta^l}$)= P(Q/T1T2) est donnée par la table suivante :

T1T2	P(Q/T1T2)
t1t2	1
t1$\overline{t2}$	0
$\overline{t1}$t2	0
$\overline{t1t2}$	0

Ce type de requête est considéré comme étant trop strict.

La requête disjonctive :

Dans le cas d'une requête booléenne OU, au moins un terme de la configuration doit être instancié comme dans la requête :

$$\mathbf{P}((\mathbf{Q}/\boldsymbol{\theta^l}) = \begin{cases} \mathbf{1} & \exists \mathbf{T_i} \in \boldsymbol{parent}(\boldsymbol{Q}) \textbf{ telque } \boldsymbol{\theta_i^Q} = \boldsymbol{\theta_i^l} \\ \mathbf{0} & \textbf{Sinon} \end{cases}$$

Exemple :

soit la requête disjonctive composée termes T1 et T2

La probabilité P(Q/ $\boldsymbol{\theta^l}$)= P(Q/T1T2) est donnée par la table suivante :

T1T2	P(Q/T1T2)
t1t2	1
t1$\overline{t2}$	1
$\overline{t1}$t2	1
$\overline{t1t2}$	0

Ce type de requête est considéré comme étant trop large ou trop tolérant pour discriminer entre les éléments.

Un juste milieu entre la requête conjonctive et la requête disjonctive est de convenir qu'une requête est satisfaite par un document, si elle possède au moins un nombre K de termes communs avec le document :

$$\mathrm{P}(\frac{i}{n}) = \begin{cases} 1 \text{ si } & i \geq \frac{K}{n} \\ 0 \text{ sinon} & \end{cases}$$

Exemple :

soit la requête composée des termes T1 et T2 et T3

La probabilité P(Q/ $\boldsymbol{\theta^l}$)= P(Q/T1T2) est donnée par la table suivante (on considère K=2) :

T1T2T3	P(Q/T1T2T3)
t1 t2 t3	1
t1 t2 $\overline{t3}$	1
t1 $\overline{t2}$ t3	0
t1 $\overline{t2\ t3}$	0
$\overline{t1}$ t2 t3	1
$\overline{t1}$t2 $\overline{t3}$	0
$\overline{t1}$ $\overline{t2}$ $\overline{t3}$	0

Cette quantification, comme dans le cas d'une agrégation disjonctive de la requête, ne permet pas de bien discriminer entre les termes.

Noisy OR : On peut supposer que les probabilités conditionnelles P(Q|Tk) ne sont pas des booléens mais dépendent d'une évaluation appropriée des termes T_k. La combinaison des termes de la requête peut être basée sur le "Noisy-OR" [Pearl, 1988] où des poids peuvent être affectés, par l'utilisateur, à chaque terme selon son degré d'importance.

L'avantage majeur de ce type d'agrégation est qu'il permet d'atténuer le problème d'explosion combinatoire liée au calcul des probabilités conditionnelles.

Dans la littérature la spécificité d'un terme dans un document est mesurée par la fréquence inverse déterminée par l'une des valeurs df_i, idf_i ou $nidf_i$. Rappelons qu'un terme fréquent dans toute la collection n'augmente pas forcément la pertinence du document étant donnée la requête. Par contre, un terme spécifique peut apporter une plus value à cette pertinence.

Ainsi, plus un terme présent dans un document est spécifique, plus la pertinence du document en réponse à une requête qui contient ce terme augmente.

Il serait alors intéressant d'exploiter cette mesure en l'intégrant dans le calcul de P(Q/ $\boldsymbol{\theta^l}$)

Nous avons proposé la formule suivante :

$$P(Q/\ \boldsymbol{\theta^l}) = \frac{\sum_{\theta_i^l = \theta_i^Q} \boldsymbol{idf_i}}{|\boldsymbol{\theta^l}|}$$

Où $idf_i = \frac{N}{n_i}$

Avec N est le nombre de documents contenant le terme *ti*. N est le nombre de documents dans la collection. $|\boldsymbol{\theta^l}|$ est le nombre de termes de la requête.

- **Calcul de la valeur $P(\boldsymbol{\theta^e}/d_i)$** : les éléments ainsi que les termes étant considérés indépendants entre eux, on a alors :

$$\mathbf{P(\boldsymbol{\theta^e}/d_i)=\prod_{\boldsymbol{\theta_j^e}\in\boldsymbol{\theta^e}} P(E_j/di)}$$

Remarque : $\boldsymbol{\theta_j^e}$ $\boldsymbol{et}$ $\boldsymbol{E_j}$ désignent le même élèment.

Valeur de L'arc balise-document :

Partant de l'idée que plus un noeud est distant (éloigné) de la racine, plus il est porteur d'information, nous proposons la formule suivante :

$$\boldsymbol{P(\theta_j^e/di)=\frac{Dist(Di,\theta_j^e)}{Dist\left(Di,\text{élè}\boldsymbol{ment\ le\ plus\ profont}\left(\boldsymbol{\theta_j^e}\right)\right)+\alpha}}$$

Où dist(A,B) : donne le nombre d'arcs séparant le nœud A et le nœud B

$\boldsymbol{Dist\left(Di,\text{élè}ment\ le\ plus\ profont\left(\theta_j^e\right)\right)}$ détermine la distance entre le nœud Di et le nœud le plus profond dans le document passant par le nœud $\boldsymbol{\theta_j^e}$. $\boldsymbol{\alpha}$ est une valeur qui sera choisie proche de 0. Nous l'avons utilisé dans l'interêt de ne pas annuler des pertinences de configurations

- **Calcul de $P(Q/\boldsymbol{\theta^e})=P(t1,\ldots..,tn/\boldsymbol{\theta^e})$**

$$\mathbf{P(\theta^l/\boldsymbol{\theta^e})=\prod_{\boldsymbol{\theta_j^e}\in\boldsymbol{\theta^e}}\prod_{\boldsymbol{T_i}\in\boldsymbol{T(E)}\wedge\boldsymbol{T(\theta^Q)}}\boldsymbol{P(t_i/\theta_j^e)}}$$

Valeur de l'arc terme-balise

Intuitivement, plus le document est pertinents pour une requête plus ses éléments le sont et selon [K sauvagnat, 2005], La pondération peut dépendre de l'élément lui-même, ses parents et le document, nous propose alors la formule suivante :

$$P(ti/\theta_j^e) = \frac{tfij}{|\theta_j^e|} + (1-\rho)\, P(ti,di)$$

P(ti,dj) est le rapport de la fréquence de ti dans le document *Di* et le maximum des fréquences dans le document. $tfij$ est la fréquence du terme ti dans l'élèment θ_j^e. $|\theta_j^e|$ est le nombre de termes dans l'élèment . ρ est valeur fixée par expérimentation.

Nous avons aussi proposé les estimations suivantes :

$P(\bar{t}i/\overline{\theta_j^e}) = 0.8$ et $P(ti/\overline{\theta_j^e}) = 0.2$

Pour résumer, voici la liste des paramètres du réseau étudié :

Di	P(Di)
di	1
$\bar{d}\iota$	0

P(Ti/Ej)	ej	$\bar{e}j$
ti	$\frac{tfij}{\lvert\theta_j^e\rvert} + (1-\rho)\, P(ti,di)$	0.2
$\bar{t}\iota$	$1- P(ti/\theta_j^e) = \frac{tfij}{\lvert\theta_j^e\rvert} + (1-\rho)\, P(ti,di)$	0.8

P(Ej/Di)	dj
ej	$\frac{Dist(Di,\theta_j^e)}{Dist\left(Di, \text{élèment le plus profont}(\theta_j^e)\right) + \alpha}$
$\bar{e}j$	$1- \frac{Dist(Di,\theta_j^e)}{Dist\left(Di, \text{élèment le plus profont}(\theta_j^e)\right) + \alpha}$

$P(Q/\theta^l)$	Noisy Or
q	$\frac{\sum_{\theta_i^l = \theta_i^Q} idf_i}{\lvert\theta^l\rvert}$

IV.4.4 Réduction du temps d'inférence par notre approche :

Nous remarquons que dans l'ensemble des configuration possible à estimer, certaines peuvent ne pas être prises en considération à cause de la redondance d'information qu'elles engendrent :

Exemple :

Prenons la configuration (titre, paragraphe, section), si elle est amenée à être retourné à l'utilisateur, celui-ci consultera donc le contenu de *Titre,* le contenu de *paragraphe* et le contenu *de section dont une partie (à savoir* paragraphe) a été déjà lue. D'où la redondance d'informations.

Solution :

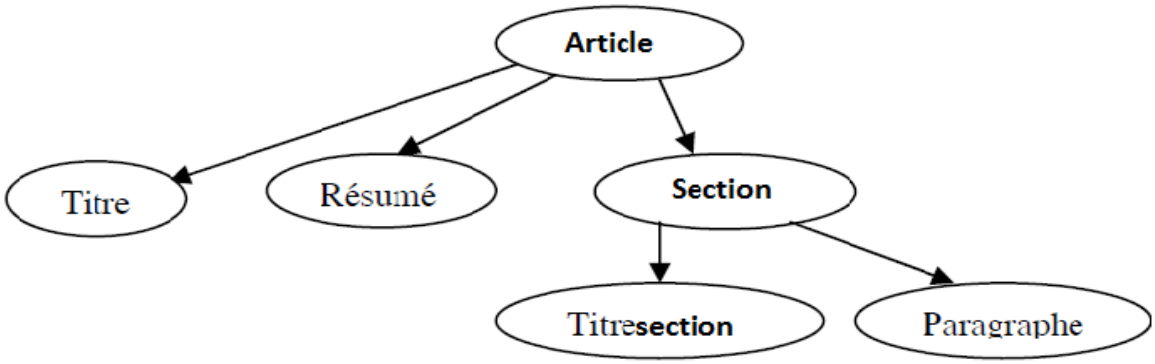
pour le calcule de la pertinence P(di/Q), on ne tiendra pas compte des configurations contenant des éléments imbriqués. Le nombre de configurations ainsi réduit permettera certainement de réduire le temps d'inférence

IV.4.5 Exemple illustratif :

Un exemple de document XML (un extrait d'un article) relatif à un un article va être utilisé pour illustrer notre approche. Le document XML exemple ainsi que le réseau bayésien qui lui est associé sont présentés ainsi :

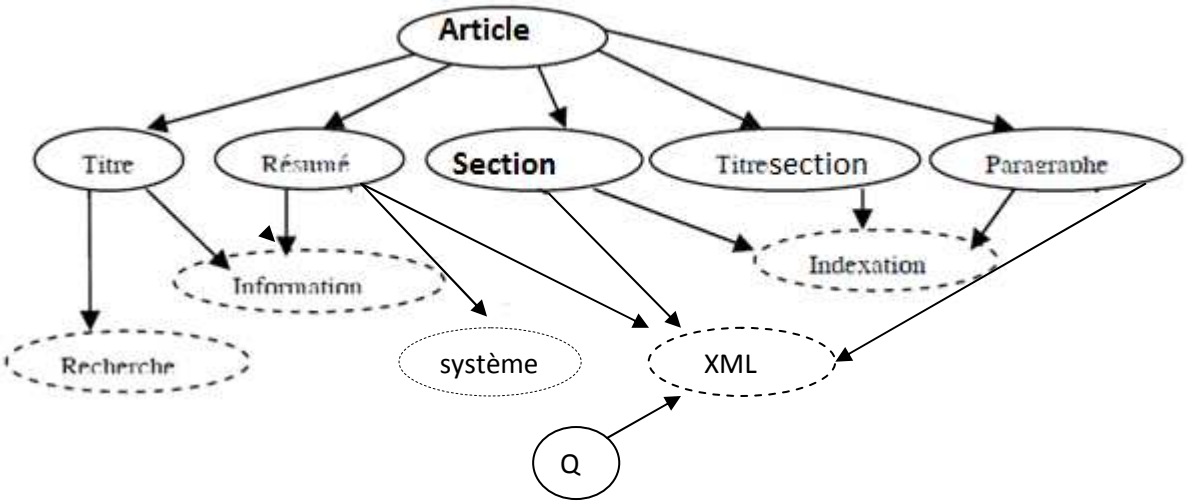
```
<Article>
    <Titre > Recherche d'Information </Titre >
    <Résumé>Devant la masse croissante de documents XML …</Résumé>
    <Section>
       <Titre section> Indexation </Titre section>
       <Paragraphe> L'indexation dans les documents XML est destinée à
       représenter … </Paragraphe>
   </Section>
</Article>
```

La structure hiérarchique du document XML 'Article' est comme suit :



Soit la requête : chercher les parties traitant de XML (Q={XML})

Le réseau bayésien correspondant est :



L'ensemble des éléments E= {e1=Titre, e2=Résumé, e3=Section, e4=Titre section, e5=Paragraphe}. L'ensemble des termes d'indexation des éléments, calculé en utilisant le contenu de chaque élément ainsi que celui de ses éléments fils dans le document, est tel que T(E) = {t1=Recherche, t2=Information, t3=Système, t4= XML, t5=indexation }.

Etant donnée la requête Q, le traitement de la formule de propagation [2] donné, on ne considère que les configurations de E qui comportent le terme de la requête 'XML', en l'occurrence seules les balises e2=Résumé, e3=Section, e5=paragraphe seront considérées. Les configurations qu'il faut donc considérer sont : {$\neg e_2 \Lambda e_3 \Lambda \neg e_5$, $\neg e_2 \Lambda \neg e_3 \Lambda e_5$, $e_2 \Lambda \neg e_3 \Lambda \neg e_5$, $e_2 \Lambda \neg e_3 \Lambda e_5$, $e_2 \Lambda e_3 \Lambda \neg e_5$,}.

Remarque : pour éviter les redondances d'information, nous avons éliminé les configurations contenant en même temps e3 et e5.

On supposera :

Le nombre de documents dans le corpus N=100,

Le nombre de documents contenant le terme t4, ni=25

On prendra $\rho = 0.6$, α=0.001

P(Q/t4) = $\log \frac{N}{n_i}$ = $\log\frac{100}{25}$ =0.6

P(t4/di)= $\frac{tf_{4i}}{Max(tf_{ki})} = \frac{10}{25}$ =0.4

P(t4/Ej)= $\frac{\boldsymbol{tf4j}}{|\boldsymbol{\theta_j^e}|}$ **+ (1-ρ) P(t4,di)=** $\frac{\boldsymbol{tf4j}}{|\boldsymbol{\theta_j^e}|}$ **+0.4*0.4=**$\frac{\boldsymbol{tf4j}}{|\boldsymbol{\theta_j^e}|}$ **+ 0.16**

e2 $\overline{e2}$	$\frac{3}{25}$ +0.16= 0.28 0.2
e3	$\frac{7}{50}$ +0.6 =0.14
$\overline{e3}$	0.2
e5	$\frac{7}{40}$ + 0.6 =0.775
$\overline{e5}$	0.2

P(Ej/di)

e2 $\overline{e2}$	$\frac{1}{1+0.001}$ =0.99 0 .01
e3	$\frac{1}{2.001}$ = 0.5
$\overline{e3}$	0.5
e5	$\frac{2}{2.001}$ = 0.99
$\overline{e5}$	0.01

On calcule la pertinence de chaque configuration vis-à-vis de la requête en utilisant la formule [4]:

Conf1=P(Q|t4)*P(t4|e2)*P(t4|e3) *P(t4|¬e5)*P(e2|di)* P(e3|di)* P(¬e5|di)

=0.6*0.28*0.14*0.2*0.99*0.5*0.01=0.00002

Conf2= P(Q|t4)*P(t4|¬e2)*P(t4|e3) *P(t4|¬e5)*P(¬e2|di)* P(e3|di)* P(¬e5|di)

=0.6*0.2*0.14*0.2*0.01*0.5*0.01=0.00000002

Conf3=P(Q|t4)*P(t4|e2)*P(t4|¬e3) *P(t4|¬e5)*P(e2|di)* P(¬e3|di)*(¬e5|di)

==0.6*0.28*0.2*0.2*0.99*0.5*0.01=0.0002

Conf4=P(Q|t4)*P(t4|e2)*P(t4|¬e3) *P(t4|e5)*P(e2|di)* P(¬e3|di)* P(e5|di)

=0.6*0.28*0.2*0.775*0.99*0.5*0.99=0.0127

Conf5= P(Q|t4)*P(t4|¬e2)*P(t4|¬e3) *P(t4|e5)*P(e2|di)* P(¬e3|di)* P(e5|di)

=0.6*0.2*0.2*0.775*0.99*0.5*0.99=0.045

Comme la configuration ayant la pertinence maximale est la configuration (¬e2, ¬e3, e5). On retournera donc l'élèment **paragraphe** à l'utilisateur.

CONCLUSION :

Les travaux présentés dans ce mémoire s'inscrivent dans le cadre de la Recherche d'Information (RI) et plus particulièrement la RIS. Nous nous sommes principalement intéressés aux différentes approches basés sur l'utilisation des réseaux bayésiens.

Les réseaux bayésiens permettent une « meilleure » modélisation de la notion de pertinence, élément fondamental en RI[Brini,2005] . Leur utilisation s'est avérée intéressante grâce notamment à leur puissance pour inférer la pertinence des documents ou des éléments vis à vis d'une requête ainsi qu'à leur capacité de représenter de manière naturelle les différents liens existants entre les objets manipulés, à savoir les termes, les documents les éléments et la requête.

Notre modeste contribution, en guise d'initiation à la recherche, a consisté à proposer une amélioration du modèle possibiliste [Brini 2005] en réduisant le temps d'inférence dans le réseau, qui est l'un des inconvénients des RB, en diminuant le nombre de configurations à estimer avant d'effectuer le calcul de pertinence.

Nous avons proposé une approche visant principalement à réduire le temps d'inférence. Notre modélisation est basée sur un réseau bayésien pour lequel les nœuds représentent les documents, leurs éléments, les termes d'indexation des élèments et de la requête. La topologie du réseau permet de prendre en compte naturellement les relations de dépendance entre ces nœuds.

L'évaluation de la pertinence d'un élément, d'une configuration ou d'un document vis à vis d'une requête est effectuée par un processus de propagation à travers le nœud requête.

Nous avons aussi proposé à chaque niveau de propagation des formules de calcul de probabilités.

Perspectives

De nombreuses perspectives découlent de notre travail, à savoir :.

- L'implémentation du modèle et son expérimentation
- Etablissement des comparaisons de résultats avec d'autres modèles de references.
- Expérimenter d'autres formules de calcul de probabilités(ex : celles proposées dans le modèle vectoriel)

- Exploiter la probabilité à priori P(Di) en proposant une probabilité utile qui permettera d'ajouter une plus value à la pertinence,

- l'intégration des relations entre paire de termes ou paires de documents. Les relations de dépendance entre paires de documents pourraient traduire des liens sémantiques ou statistiques évaluant les distributions des termes communs à des paires ou ensembles de documents.

- d'intégrer un processus itératif à la recherche pour la reformulation de requêtes. Pour ce faire, deux techniques existant dans les modèles basés sur les réseaux Bayésiens probabilistes pourraient être adaptées à notre approche. La première préconise l'ajout des noeuds 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 des documents jugés par l'utilisateur La seconde technique considère la requête reformulée comme une nouvelle information `a introduire dans le système ;

- **Autre outils de validation** : Bien que les DTD et les schémas soient les langages de validation les plus utilisés, il existe néanmoins :

Relax NG[1]: l'équivalent de XML Schéma mais proposé par OASIS[2](Organization for the Advancement of Structured Information Standards). Il sert de base au document de Open Office. Plus souple que XML Schema mais néanmoins moins implémenté ;

Schematron[3] : schematron n'est pas un langage validant «complet », il est généralement utilisé en complément d'un des précédents cités. Là où les autres sont principalement orientés vers des contraintes structurelles, lui est plus orienté « sémantique » en permettant d'en ajouter sur les valeurs des éléments et des attributs.

[1] Voir http://fr.wikipedia.org/wiki/Schematron

[2] Depuis son lancement en 1993, avec son produit phare :Standard Generalized Markup Language (SGML), OASIS a élaboré des normes à travers une variété de domaines technologiques.

[3] Voir http://fr.wikipedia.org/wiki/RELAX_NG

- **Approches pour la RI structurée :**

La RI structurée englobe deux approches qui tentent de proposer des méthodes pour l'indexation, l'interrogation, la recherche et le tri des documents XML. Ces deux approches sont :

- **L'approche orientée données** : Elle utilise des techniques développées par la communauté des bases de données, et voit les documents XML comme des collections de données, typées et relativement homogènes.
- **L'approche orientée document** : Cette approche est prise en charge par la communauté de la recherche d'information, et se focalise sur des applications considérant les documents XML d'une manière traditionnelle, c'est-à-dire que les balises servent uniquement à décrire la structure logiques des documents**.**

Le tableau ci-dessus résume quelques points propres à chaque approche pour chacune des phases du processus de recherche.

	Approches orientées BD	**Approches orientées RI**
Indexation	-Confondent les notions d'indexation et de stockage : toute l'information textuelle et structurelle des documents est stockée au sein de tables dans des BD. -Ceci pose un problème pour les recherches orientées contenu, puisque le contenu textuel est indexé en tant que chaîne de caractères. - Proposent des schémas de stockage optimaux pour la structure des documents.	-Utilise des techniques traditionnelles pour l'extraction des termes d'indexation. -De nouvelles problématiques sont soulevées concernant la structure : Que doit-on indexer de la structure des documents ? Comment relier cette structure au contenu même du document?
Langages d'interrogation	Historiquement les premiers à proposer des langages pour l'interrogation des documents XML. - Ces langages sont basés sur des syntaxes proches de SQL, et permettent à l'utilisateur d'exprimer des conditions très précises sur la structure des documents. - Les requêtes doivent toujours porter sur des conditions de structure bien définies. L'utilisateur doit de plus spécifier le type d'élément qu'il désire voir retourné par le système, alors qu'il n'a pas forcément d'idée précise sur la question.	-Cherchent à simplifier ces langages en ce qui concerne les conditions de structure. - Proposent de nouvelles fonctionnalités concernant la recherche sur le contenu (Utilisation du prédicat 'about' au lieu de 'contains', ou bien encore d'opérateurs booléens dans des conditions de contenu).

Traitement des requêtes	-Evaluent de façon exacte des expressions de type 'attribut=valeur'. - Le traitement est effectué d'une manière booléenne et il n'est pas possible de renvoyer à l'utilisateur une liste triée de résultats.	-Cherchent à évaluer le degré de pertinence entre la requête et les unités d'informations et attribuent à ces derniers un score de pertinence. - L'intérêt est double : tout d'abord sélectionner les unités d'informations qui répondent le mieux au besoin de l'utilisateur, et lui proposer ensuite une liste triée de résultats.

Tableau II.1: comparaison entre l'approche orientée RI et l'approche orientée BD [Mataoui, 2007]

❖ **Remarque :**

Les solutions proposées par la communauté de la RI peuvent être utilisées comme « surcouche » aux solutions orientées BD. Cette surcouche sert essentiellement à intégrer la notion de pertinence dans la recherche, en complétant les approches proposées par la communauté de BD pour le stockage et l'interrogation des documents [Sauvagnat, 2005].

Figure II.4 Domaines de compétence de la BD et de la RI [Sauvagnat 2005]

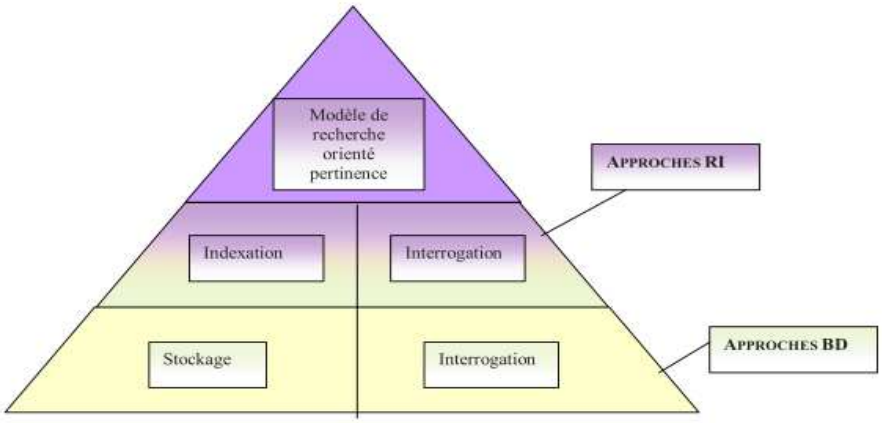


Figure II.4 Domaines de compétence de la BD et de la RI [Sauvagnat 2005]

II-5-2Le stockage des documents XML :

Avant de s'interroger sur la manière de stocker des données semi-structurées, il convient de savoir dans quel but on veut les stocker, et quelle utilisation on en fera. Ainsi, tel qu'illustré dans la figure II.5, si le stockage n'a qu'un but simplement documentaire, un simple stockage dans un système de fichier suffirait, par contre, une requête sur les valeurs internes de ce document sera impossible ou très coûteuse. Plusieurs façons de stocker des données XML existent : sous forme de BLOB dans un SGBD-R/O[4] ou dans un SGBD natif semi-structuré. Le stockage par BLOB convient à un stockage orienté texte, c'est-à-dire de type documentaire. Le stockage SGBD-R convient plutôt à un stockage orienté données et le stockage dans un SGBD natif peut convenir aux deux types de stockage suivant les spécificités du SGBD natif utilisé.

[4] SGBD relationnel ou SGBD orienté objet.

Cependant, les tailles des collections XML prennent souvent des proportions importantes, voire considérables. Si une recherche d'information dans une collection s'effectue de manière simplement séquentielle (c'est à dire en examinant toute la collection, du début jusqu'à la fin), le temps d'attente peut devenir prohibitif pour l'opérateur. L'indexation est l'outil qui permet de résoudre ce problème.

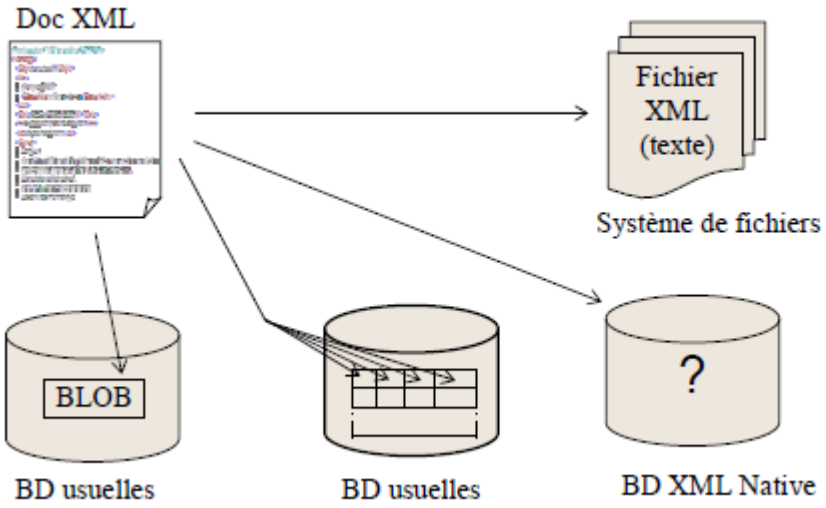


Figure II.5 : Les différents modes de stockage de documents XML [Alilaouar, 2007]

ANNEXE B

I. Les Réseaux Possibilistes :

I.1 La théorie des possibilités :

La théorie des possibilités introduite par Zadeh [Zadeh, 1978] et développée par Dubois et Prade [Dubois et al, 1988] [Dubois et Prade, 1998] traite l'incertitude sur l'intervalle [0,1], appelé échelle possibiliste, d'une manière qualitative ou quantitative. En fait, Lotfi Zadeh a formalisé la théorie des possibilités pour traiter l'incertitude permettant ainsi de traiter l'ignorance et de prendre en compte la pertinence d'une information incertaine. Dans cette théorie, l'information fournie par une source sur la valeur réelle d'une variable x est codée sous forme d'une distribution de possibilités dont les valeurs sont supposées être mutuellement exclusives, puisque x prend en définitive une seule valeur (sa vraie valeur), qui appartient à un ensemble W donné [Sandri, 1991]. La théorie des possibilités se base sur deux mesures de confiance : la mesure de possibilité et la mesure de nécessité [Fabiani, 1996].

I.2 Distribution de possibilité :

La théorie des possibilités [Dubois et al, 1988] est basée sur les distributions de possibilité. Une distribution de possibilité, notée par π, est une application d'un ensemble d'états possibles X vers l'échelle [0, 1] traduisant une connaissance partielle sur le monde.

$\pi(x) = 1$ correspond à un état possible, $\pi(x) = 0$ correspond à un état impossible.

Une distribution de possibilité normalisée exprime qu'un des états est totalement possible, ce qui se traduit par la condition :

$$\max_{x \in X} \pi(x) = 1$$

Si $\max_{x \in X} \pi(x) < 1$, ceci indique une contradiction interne dans la représentation, qui est alors partiellement incohérente.

Mesures de nécessité et de possibilité:

Dire qu'un événement est non possible n'implique pas seulement que l'événement contraire est possible mais aussi qu'il est certain. Deux mesures duales sont utilisées : la mesure de possibilité, et la mesure de nécessité. La possibilité d'un événement A, notée $\Pi(A)$ est obtenue par la formule $\Pi(A) = \max_{x \in A} \pi(x)$ et reflète la situation la plus normale dans laquelle A est vraie. Soit $\bar{A}$ le complémentaire de A.

La nécessité, notée $N(A)$, d'un événement A, définie par la formule :

$$N(A) = \min_{x \notin A}(1 - \pi(x)) = 1 - \Pi(A)$$

reflète la situation la plus normale dans laquelle A est faux. La distance entre $N(A)$ et $\Pi(A)$ évalue le niveau d'ignorance sur A.

I.3 Conditionnement possibiliste :

En logique possibiliste, le conditionnement consiste à modifier la distribution de possibilité initiale π à l'arrivée d'une nouvelle information. En fait, on doit restreindre les états possibles à ceux où la nouvelle information est vraie.

Soit C, une sous classe de X, représentant la nouvelle information.

La distribution initiale π est remplacée par $\pi' = \pi(.|C)$. Dans un cadre quantitatif, les degrés de possibilités des éléments de C sont proportionnellement modifiés.

Ainsi,

ANNEXE B

$$\pi(x \mid_p C) = \frac{\pi(x)}{\Pi(C)} \text{ si } x \in C$$
$$= 0 \text{ sinon}$$

Où $|p$ est le conditionnement basé sur le produit. Notons que c'est exactement la même définition qu'en théorie des probabilités : elle préserve la valeur relative des degrés de possibilités des éléments de C. La seule différence est que Π(*C*) est calculée avec la règle du maximum et non la somme.

I.4 Réseaux possibilistes :

Les travaux existants sur les réseaux possibilistes sont soit des adaptations directes de l'approche probabiliste [Benferhat et al., 1999] , ou des méthodes d'apprentissage à partir de données imprécises [Borgelt et al., 2000].Un graphe possibiliste orienté sur un ensemble de variables *V* = *V*1, *V*2, , , *VN* est caractérisé par une composante qualitative et une composante numérique. La première est un graphe acyclique orienté comme pour les réseaux Bayésiens. La structure du graphe représente l'ensemble des variables ainsi que l'ensemble des relations d'indépendance.

La seconde composante quantifie les liens du graphe en utilisant les distributions de possibilité conditionnelles de chaque noeud dans le contexte de ses parents. Ces distributions de possibilité doivent vérifier la contrainte de normalisation. Pour chaque variable *Vi* :

- Si *Vi* est un noeud racine et *domVi* le domaine de *Vi*, la possibilité *a priori* de *Vi* doit satisfaire max*vi* Π(*vi*) = 1, $\forall vi \in domVi$
- Si *Vi* n'est pas un noeud racine, la distribution conditionnelle de *Vi* dans le contexte de ses parents doit satisfaire max*vi* Π(*vi/PARVi*) = 1, $\forall vi \in domV\,i$ où *domVi* est le domaine de *Vi*, et *PARVi* est l'ensemble des parents de *Vi*.

I.4.1 Réseaux possibilistes basés sur le minimum :

Un graphe possibiliste basé sur le minimum, noté par *GPM*, est un graphe possibiliste où les possibilités conditionnelles sont obtenues par le conditionnement minimum.

La distribution de possibilité des réseaux possibilistes basée sur le minimum, notée par Π_M est obtenue par la règle de chaînage :

$$\pi_M(A_1, A_2, \ldots, A_N) = MIN_{i=1..N}\Pi(A_i \mid \theta_{A_i})$$

Avec MIN est l'opérateur minimum.

θ_{Ai}: L'ensemble des configurations possibles des parents de A_i .

I.4.2 Réseaux possibilistes basés sur le produit :

Un graphe possibiliste basé sur le produit, noté par *GPP*, est un graphe possibiliste Où les possibilités conditionnelles sont obtenues par le conditionnement de type produit. La distribution de possibilité des réseaux possibilistes basés sur le produit, notée par Π_P, est obtenue par la règle de chaînage

$$\Pi_P(V_1, .., V_N) = PROD_{i=1..N}\Pi(V_i / PAR_{V_i})$$

Où *PROD* est l'opérateur produit.

Bibliographie

[Abolhassani et al, 2004] : M. Abolhassani and N. Fuhr. Applying the divergence from randomness approach for content-only search in XML documents. In *Proceedings of ECIR 2004, Sunderland*, pages 409-419, 2004.

[Acid et al, 2003]: Acid, S., de Campos, L., Fernandez, J., and Huete, J. An information retrieval model based on simple bayesian networks. International Journal of Intelligent Systems 18, 2 (2003), 251–265.

[Ahn & Moffat, 2002]: Vo Ngoc Anh, Alistair Moffat, "Compression and an IR approach to XML Retrieval", In INEX 2002 Workshop Proceedings, p. 100-104, Germany, 2002.

[Alilaouar, 2007] : Contribution à l'interrogation flexible de données semi-structurées, thèse de doctorat de l'université Paul Sabatier de Toulouse, 2007.

[Alimazighi et al 2005] : Utilisation des réseaux bayésien pour la recherche d'information structuré Z. Alimazighi ,B.bessai zbessai,R. Djiroun ,2005.

[AMBRO, 1990]: Shachter, R. D, D'Ambrosio, B., & DelFabero, B. (1990). Symbolic probabilistic inference in belief networks. Dans Proceeding soft the Eighth National Conference on Artificial Intelligence, (pp. 126–131). 45

[AMBRO, 1994]: Li, Z. & D'Ambrosio, B. (1994). Efficient inference in bayes nets as a combinatorial optimization problem. International Journal of Approximate Reasoning, 11(1), 55–81. 45

[BAZ, 2005] : Mustapha BAZIZ. Indexation conceptuelle guidée par ontologie pour la recherche d'information, thèse Doctorat de l'Université PAUL SABATIER, 14/ 12/ 2005.

[Ben Aouicha,2009] : Mohammed ben Aouicha.Une approche algébrique pour la recherche d'information structurée. Thèse de doctorat en informatique, Université Paul Sabatier, Toulouse, France, janvier 2009.

[Benferhat et al., 1999]: Benferhat S., Dubois D., Garcia L., and Prade H, "Possibilistic logic bases and possibilistic graphs", In Proc. of the Conference on Uncertainty in Artificial Intelligence, pp. 57–64, 1999.

[BESSAI et al]: Recherche d'Information Structurée Vers un modèle possibiliste pour la recherche d'information dans des documents structurés Fatma-Zohra BESSAI MECHMACHE(1) et Mohand BOUGHANEM

[Borgelt et al., 2000]: Borgelt C., Gebhardt J., et Kruse R., "Possibilistic Graphical Models. Computational Intelligence in Data Mining",CISM Courses and Lectures 408, pp. 51-68, 2000.

Bibliographie

[Buntine, 1994]: Buntine W., “Representing Learning with graphical Models”. Technical Report, FIA 94-14, Artificial Intelligence Research Branch, NASA Armes Research Center, USA, 1994.

[Braga et al, 2002] : D. Braga, A. Campi, E. Damiani, P. Lanzi, and G. Pasi. FXpath : flexiblequerying of XML documents. In *Proceedings of Eurofuse 2002*, 2002.

[Brini et al, 2005] : Asma Brini, Mohand Boughanem, Didier Dubois Un modèle de réseau possibiliste pour la recherche d’information.

[Callan et al 1992]:J. P. Callan, W. B. Croft, and Hardings. The inquery retrieval system. DEXA, pages 78–83, 1992.

[Carmel & al, 2002]: D.Carmel, D.Efraty, G.Landau, Y.Maaker, and Y.Mass, "An Extension of the Vector space Model for querying XML documents via XML fragments", In R.Baeza Yates, N.Fuhr and Y.S.Maarek, editors, XML and Information Retrieval workshop of SIGIR 2002, Tampere, Finland, Aug 2002.

[Chi et al, 1996]:Y. Chiaramella, P. Mulhem, and F. Fourel. A model for multimedia information retrieval. Technical report, Technical report, FERMI ESPRIT BRA 8134, University of Glasgow, 1996.

[**Cui & al, 2003]**:H.cui, J-R.Wen, J-R.Chua, "Hierarchical indexing and flexible element retrieval for structured document", april 2003.

[Crestani et al, 2003]: Crestani, F., de Campos, L. M., Fernandez-Luna, J. M., and Huete, J. F. Ranking structured documents using utility theory in the bayesian network retrieval model. In Proc. of the symposium on String Processing and Information REtrieval (SPIRE) (2003), pp. 168–182.

[Crestani et al, 2004]: [Crestani, F. and de Campos, L. and Fernandez Luna, J. and Huete, J. (2004)]. A multi-layered Bayesian network model for structured document retrieval. In: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 7th European Conference, ECSQARU 2003, 2003-07-02 - 2003-07-05, Aalborg, Denmark.

[C.Chrisment, 2005] :C. Chrisment. Caractéristiques d’XML. Cours DEA 2IL, 2005

[De Campus et al, 2003]: de Campos, L., Fernandez-Luna, J., and Huete, J. Improving the efficiency of the bayesian network retrieval model by reducing relationships between terms. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 11, Supplement (2003), 101–116

Bibliographie

[Denoyer et al, 2004]: Denoyer, L., Wisniewski, G., and Gallinari, P. Document structure matching for heterogeneous corpora, 2004. Proc. Of the International ACM-SIGIR Conference : Workshop on XML and Information Retrieval.

[Dempster et al, 1977]: A.Dempster, N.Laird, et D.Rubin. Maximum likelihood from incomplete data via the EM algorithme. Journal of the Royal Statistical Society. B39:1-38.1977.

[DeCampos et al., 2009]: DeCampos L, Juan M., Fernández-Luna, Juan F., Huete , Carlos, Martín-Dancausa. Managing structured queries in probabilistic XML retrieval systems. Journal of, Information Processing & Management, December 2009.

[Dubois et al, 1988] : D. Dubois et H. Prade. *Possibility Theory*. Plenum,1988.

[Dubois et Prade, 1998]: Dubois D., and Prade H., "Possibility theory: qualitative and quantitative aspects", Dans: Quantified Representation of Uncertainty and Imprecision. Dov M. Gabbay, Philippe Smets (Eds.), KLUWER ACADEMIC PUBLISHERS, The Netherlands, p. 169-226, Vol. 1, Handbook of Defeasible Reasoning and Uncertainty Management Systems, 1998.

[Fabiani, 1996] : Fabiani P., “Représentation Dynamique de l’Incertain et stratégie de Prise d’Information pour un Système Autonome en Environnement Evolutif”, Thèse de Doctorat en Automatique et Informatique Industrielle, Ecole Nationale Supérieure de l’Aéronautique et de l’Espace, Toulouse, 1996.

[Fellag ,2006] : Mme Samia FELLAG. Recherche d’information dans les documents semi structuré XML. Thèse de magistère en informatique, Université Mouloud Mammeri, Tizi ouzou, septembre 2006.

[Florescu, 1999]: D. Florescu and D. Kossmann. Storing and querying XML data using an RDMBS. IEEE Data Engineering Bulletin, 22(3) : pages 27–34, 1999.

[FOX, 1983]: E.Fox, “Extending the Boolean and vector space models of information retrieval with pNorm queries and multiple concept types”, PhD Thesis Cornell University, 1983

[Fuller & al, 1993]: M. Fuller, E. Mackie, R. Sacks-Davis, R. Wilkinson, " Structural answers for a large structured document collection", In Proceedings of ACM SIGIR, pp. 204-213. Pitthsburgh, 1993.

[Fuhr & al, 2001]:N. Fuhr, K. Grossjohann, "XIRQL, a query language for information retrieval in XML documents", In proceedings of SIGIR 2001, Toronto Canada 2001.

Bibliographie

[Jensen, 2000]: Jensen F. V., “Bayesian Networks and Decision Graphs”, Wiley, 2000.
.

[JYN, 2004] : Jian-Yun Nie, cours hiver 2004 ; Module Recherche d’Information ; Université Montréal. Département d'informatique et de recherche opérationnelle (I.R.O.) Hiver 2004 http://www.iro.umontreal.ca/~nie/IFT6255

[HAR, 1992]: Donna Harman: Relevance Feedback Revisited, in the Proceedings of the ACM SIGIR Conference On Research and Development in Information Retrieval (SIGIR), pp 1-10, 1992.

[Hatano et al, 2002]: K.Hatano, H.Kinutani, M.Yoshikawa, and S.Uemra, "Information retrieval system for XML document", 2002

[Hernandez ,2006] : Hernandez N. Ontologie de domaine pour la modélisation du contexte en recherche d’information, thèse de doctorat en informatique, Université Paul Sabatier, (2006).
in Artificial Intelligence, (pp. 293–301)., San Francisco, CA, USA. Morgan Kaufmann Publishers. 79,88,90

[Indrawan et al, 1996]: Indrawan, M., Ghazion, D., and Srinivasan, B. Using bayesian networks as retrieval engines. In Proc. of the Text REtrieval Conference (TREC-6) (1996), pp. 437–444.

[INEX, 2007] « INEX 2007 Evaluation Measures (Draft) »
Jovan Pehcevski1, Jaap Kamps2, Gabriella Kazai3, Mounia Lalmas4, Paul Ogilvie5, Benjamin Piwowarski6, and Stephen Robertson3
[INEX, 2012] "Report on INEX 2012" P. Bellot, T. Chappell, A. Doucet, S. Geva, S. Gura jada, J. Kamps, G. Kazai, M. Koolen , M. Landoni, M. Marx, A. Mishra, V. Moriceau, J. Mothe, M. Preminger, G. Ramirez, M. Sanderson, E.Sanjuan, F. Scholer, A. Schuh, X. Tannier, M. Theobald, M. Trappett ,A. Trotman Q. Wang

[Géry et al , 2008]:Géry M., Largeron C., Thollard F., « Integrating structure in the probabilistic model for InformationRetrieval », *Web Intelligence*, p. 763-769, 2008

[Goldfarb, 1990]:C. Goldfarb. *The SGML Handbook*. Oxford University Press, Oxford, 1990.147*Bibliographie*

[Gövert & al, 2002]: N. Gövert, M. Abolhassanni, N. Fuhr, K. Grossjohann, “Content-Oriented XML Retrieval with HyreX”. In INEX 2002 Workshop Proceedings, p. 26-32, Germany, 2002.

Bibliographie

[Guo & Hsu(2001)]: Guo, H. & Hsu, W. (2001). A survey of algorithms for real-time bayesian network inference. (unpublished). 48

[Grabs & Scheck, 2002]:T.Grabs and H .J.Scheck, "Flexible information retrieval from XML with Power DB XML", In proceedings of the first annual workshop of INEX, pages 141-148, December 2002

[Grust, 2002]:T. Grust. Accelerating XPath location steps. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA*. In

[Larson, 2002]: Larson R.R. Cheshire ii at inex : Using a hybrid logistic regression and Boolean model for xml retrieval. In Proceedings of the First Workshop of the INiative for the Evaluation of XML REtrieval(INEX), pages 18-25. Dagsthul, Germany, December 2002.

[Lalmas, 1997]: M. Lalmas. Dempster-shafer's theory of evidence applied to structured documents: modeling uncertainty. In *Proceedings of SIGIR'97, Philadelphia, USA*, pages110–118, 1997.

[Lee, 1996]: Y. K. Lee, S.-J. Yoo, K. Yoon, and P. B. Berra. Index structures for structured documents. In *The first ACM international conference on Digital libraries, DL'96*, pages 91–99, 1996.

[Li et al, 2001]: Li Q. and Moon B. Indexing and Querying XML Data for Regular Path expressions. In Proceedings of 27th International Conference on Very Large Databases (VLDB'01), Rome, Italy, pp. 361-370, 2001.

[Luhn, 1958]: Luhn, H. The automatic creation of literature abstracts. IBM Journal of Research and Development 24, 2 (1958), 159–165.

[MAN, 2002] : Actualité des langages documentaires, les fondements théoriques de la recherche d'information .Maniez Jacques. 2002.

[Mass et al. 2002]: Y.Mass, M.Mandelboard, E.Amitay, and A.Soffer, " JuruxXML, an XML retrieval system", at INEX'02. In proceedings of INEX 2002, Dagstuhl, Germany, pages 73-80, 2002.

[Mataoui, 2007] : Reformulation de requêtes dans les systèmes de recherche d'information dans des documents XML, thèse de magistère, Université m'hamed Bougara de Boumerdes, 2007.

Bibliographie

[MOR, 2006] : Fabienne MOREAU. Revisiter le couplage traitement automatique des langues et recherche d'information, thèse Doctorat de l'Université de Rennes 1, 07 décembre 2006.

[Myaeng et al, 1998]: S. H. Myaeng, D.-H. Jang, M.-S. Kim, and Z.-C. Zhoo. A Flexible Model for Retrieval of SGML documents. In W. B. Croft, A. Mo_at, C. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 138{140, Melbourne, Australia, Aug. 1998. ACM Press, New York.

[Najeh naffakhi et al, 2009] :
Réseau bayésien pour un modèle de Recherche d'Information agrégée dans des documents structurés Najeh NAFFAKHI, Mohand BOUGHANEM, Rim FAIZ, 2009

[NGUYEN, 2005] : Rapport du Travail d'Intérêt Personnel Encadré Semestre II Réseaux Bayésiens, NGUYEN Trung Thanh, 2005

[PEARL, 1991]: Pearl, J. & Verma, T. (1991). A theory of inferred causation. Dans Allen, J. F., Fikes, R., & Sandewall, E. (Eds.), KR'91 : Principles of Knowledge Representation and Reasoning, (pp. 441–452)., San Mateo, California. Morgan Kaufmann. 71

[PEARL, 1988]: Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, second edition in 1991. 5,13,30,48,194

[Piwowarski et al, 2002]: Benjamin Piwowarski, Georges Etienne Faure, Patrick Gallinari.Bayesian Networks and INEX, 2002. *Proceedings in the First INEX Workshop*, décembre 2002.

[Piwowarski, 2003a] : Piwowarski B, Techniques d'apprentissage pour le traitement d'information structurées : application à la recherche d'information, Thèse de doctorat,Université Paris 6, 2003.

[PON & CRO, 1998]: Ponte, J. M., and Croft, W. B. A language modeling approach to information retrieval. research and development in information retrieval. In Proc. of the International ACM-SIGIR Conference (1998), Proc. Of the International ACM-SIGIR Conference, pp. 275–281.

[Ribeiro-Neto et al., 1996]: Ribeiro-Neto B., Silva I., et Muntz R., "A Belief Network Model for IR", Proc. of the 19th ACM-SIGIR Conf. on Research and Development in Information Retrieval, 253260, 1996.

[Ribeiro-Neto et al., 1999]: Baeza-Yates, R., and Ribeiro-Neto, B. Modern information re-trieval. ACM Press / Addison-Wesley, 1999.

Bibliographie

[ROB et al, 1976]: S.Robertson & K.Sparck Jones, *Relevance Weighting for Search Terms*. Journal of The American Society for Information Science, Vol 27, N°3, 1976.

[Salton & al., 1983]:G.Salton, E.Fox, and H.Wu, "Extended boolean information retrieval". Communications of the ACM, 26(11) :1022–1036, 1983

[Sandri, 1991] : Sandri S., "La Combinaison de l'Information Incertaine et ses Aspects Algorithmiques", Thèse de Doctorat en Informatique. Université de Paul Sabatier de Toulouse, 1991.

[Sauvagnat, 2005] : Sauvagnat k. Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés. Thèse de doctorat en informatique, Université Paul Sabatier, Toulouse, France, juin 2005.

[sauvagnat et M.Boughanem] : Sauvagnat k, Boughanem.M, Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée. Dans : Actes de CORIA 2006, Lyon, 15-17 mars (2006).

[Savoy et al., 1991] : Savoy J., Dubois D., et al., "Infor
mation Retrieval in Hypertext Systems an Approach Using Bayesian Networks", Electronic Pub., 4(2) : 87-108, 1991.

(Sigurbjaornsson et al., 2003): Sigurbjaornsson B., Kamps J., and de Rijke M. An element-based approach to XML retrieval. In Proceedings of INEX 2003 workshop, Dagstuhl, Germany, dec. 2003.

[Silva et al, 2000]: Silva, I., Ribeiro-Neto, B. A., Calado, P., de Moura, E., and Ziviani, N. Link-based and content-based evidential information in a belief network model. In Proc. of the International ACM-SIGIR Conference
(2000), pp. 96–103.

[SIN, 1995]:S. Robertson, S. Walker, M. Sparck Jones, and al. Okapi at trec-3. In *Second Text Retrieval Conf (TREC-3)*, pages 109.26, 1995.

[S.Geva,2005] : S.Geva. GPX-gardenspointXMLIRatINEX2005. In INEX'05, pages 240–253, 2005.

[S.Geva,2006] :S.Geva. GPX-gardenspointXMLIRatINEX2006. In INEX'06, pages 137–150, 2006.

Bibliographie

[Turtle 1991]: Turtle, H. Inference networks for document retrieval, 1991. Ph.D. thesis, University of Massachusetts.

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

[Turtle et Croft, 1991]: Turtle H. R., et Croft W. B., "Evaluation of an inference network-based retrieval model", In ACM Transaction on Information System, 9(3) : 187–222, 1991.

[Trotman et al., 2003]: Trotman A. and O'Keefe R. A. Identifing and ranking relevant document element. In Proceedings of INEX 2003 Workshop, pages 149,154. Dagsthul, Germany, December 2003.

[Trotman, **2004**] :Trotman A, and Sigurbjörnsson B. NEXI, Now and Next. In: Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 68, 2004, Revis .

[W3C, 1998a]: W3C. EXtensible Markup Language (XML) 1.0. Technical report, World Wide Web Consortium (W3C), Technical report, february 1998.

[W3C, 1998b] :W3C. DOM Level 1 (Document Object Model). Technical report, World Wide Web Consortium (W3C), W3C standard, october 1998.

[W3C, 2001a]:W3C. XML Schema. Technical report, World Wide Web Consortium (W3C), W3C Recommendation, 2001.

[W3C, 2001b] :eXtensible Stylesheet Language (XSL), version 1.0. Technical report, World Wide Web Consortium (W3C),W3C Recommendation, October 2001.

[Wolff et al, 2000] : J. Wolff, H. Flörke, and A. Cremers. Searching and browsing collections of structural information. In *Proceedings of IEEE advances in digital libraries, Washington, 2000,* pages 141–150, 2000.

[Xavier Tannier] : cours « Indexation et Recherche d'Information : Modèles de Recherche et Évaluation »Xavier Tannier Université Paris-sud 11

[Yang et al., 2007] : Yang J., Zhang F. XML Document Classification Using Extended VSM. INEX 2007: 234-244.

[Yuanhua Lv & ChengXiang Zhai, 2011]: "Adaptive Term Frequency Normalization for BM25" Yuanhua Lv, ChengXiang Zhai Urbana, IL 61801: 2011

Bibliographie

[Zadeh, 1978]: Zadeh L. A., “Fuzzy Sets as a basis for a theory of Possibility”, Fuzzy Sets and Systems, Vol. 1, pp. 3-28, 1978.

[Zargayouna, 2004] : Contexte et sémantique pour une indexation de documents semi-structurés.

[ZEM, 2004] : Zemirli W.Nesrine, Vers le développement d’un système de recherche d’information personnalisé intégrant le profil utilisateur. Formation Doctorale en informatique. Année 2003/2004.

[Zip, 1949] : G. K. Zipf. Human Behavior and the Principle of Least. Addison-Wesley Press, 1949.