



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT INFORMATIQUE

Mémoire de fin d'étude En vue de l'obtention du diplôme de MASTER

Domaine : informatique

Option : Systèmes Informatiques

Thème :

**Implémentation et expérimentation d'un modèle de recherche
d'information dans les documents XML basé sur les réseaux bayésiens**

Réalisé par :

Melle HAMDAD Nassima

Mr RAHEM Mohamed Amine

Proposé et dirigé par :

Mme FELLAG Samia

Année universitaire : 2014/2015

Remerciements

En premier lieu, nous remercions le bon Dieu, le créateur qui nous a facilité le chemin et donné la persévérance pour travailler.

*Nous tenons à exprimer notre profonde gratitude à notre promotrice, **M^{me} FELLAG** pour ses prestigieux conseils et orientations.*

D'autre part, nos vifs remerciements vont droit à tous nos enseignants qui nous ont transmis leur savoir.

Nous tenons enfin à remercier d'avance, Monsieur le président et tous les membres du jury qui nous feront l'honneur de juger notre travail.

A cette liste, il faudra ajouter tous ceux et celle qui, tout au long de ce travail, nous ont soutenu moralement sans les nommer explicitement, on les remercie de leurs encouragements et de nous avoir aidé à surmonter maintes difficultés

Dédicaces

Je dédie ce travail:

En premier lieu, à mon père et à ma très chère mère, qui m'ont soutenu le long de ma vie en témoignage de leurs affection et sacrifices.

A mes grands frères Rachid et Yazid.

A mes grandes sœurs Samia et Bahia.

A mes beaux-frères.

A mes bellessœurs.

A mes neveux et nièces.

A ma très chère cousine Thiziri.

A tous mes amis sans oublier Fifi.

Nassima

Dédicaces

Je dédie ce travail:

En premier lieu, à mon père et à ma très chère mère, qui m'ont soutenu le long de ma vie en témoignage de leurs affection et sacrifices.

***A** mon grand frère.*

***A** ma petite sœur.*

***A** tous mes amis.*

Amine

Liste des figures

LISTE DES FIGURES

Figure 1.1 : processus en u de la recherche d'information [Sauvagnat,2005].....	4
Figure 1.2 : Exemple d'arbre DOM.....	9
Figure 1.3 : représentation matricielle d'un document [Yang et al., 2007].....	13
Figure 1.4 : modèle d'augmentation [Fuhr& al,2001].....	15
Figure 1.5 : Exemple de requête CO, issue du jeu de test 2003.....	28
Figure 1.6 : Exemple de requête CAS (Topic 205 d'INEX 2005).....	29
Figure 1.7 : Taxonomie des modèles de recherche d'information [Mataoui, 2007].....	111
Figure 1.8 : un modèle de réseau de neurones pour la RI [Sauvagnat, 2006].....	115
Figure 2.1 : connexion en série.....	35
Figure 2.2 : connexion divergente.....	35
Figure 2.3 : connexion convergente.....	36
Figure 2.4 : d-séparation cas des variables en série.....	37
Figure 2.5 : d-séparation cas des variables divergente.....	38
Figure 2.6 : d-séparation cas des variables convergente.....	39
Figure 2.7 : Architecture simplifiée du réseau d'inférence.....	41
Figure 2.8 : Architecture générale.....	44
Figure 2.9 : Architecture globale.....	46
Figure 2.10 : Architecture globale du réseau multi connecté.....	48
Figure 2.11 : Modèle de base.....	50
Figure 2.12 : autre réseau bayésien étendu.....	51
Figure 2.13 : Architecture générale.....	51
Figure 2.14 :l'approche de Myaeng.....	55
Figure 2.15 : une vue locale du RB : utilisation de deux modèles classiques M1 et M2.....	57

Figure 2.16 : d'un document indexé (RI) à un document structuré indexé (RIS).....	58
Figure 2.17 : réseaux bayésien multi niveau.....	59
Figure 2.18 : un fragment de réseau bayésien contenant une unité virtuelle.....	62
Figure 2.19 : exemple de recherche avec structure.....	64
Figure 2.20 : architecture du modèle.....	65
Figure 2.21 : architecture générale du modèle proposé par [Naffakhinajeh et al, 2009].....	68
Figure 3.1 : topologie du réseau bayésien.....	75
Figure 4.1 : Interface de la VMware Workstation 9.....	85
Figure 4.2 : Architecture de base du model XFIRM [Sauvagnat, 2005] système.....	86
Figure 4.3 :L'interface de l'IDE Eclipse sous linux.....	87
Figure 4.4 : Illustration de l'interface SQL*PLUS sous linux.....	89
Figure 4.5 : Illustration des commandes de lancement du SGBD du système en lignes de commande.....	93
Figure 4.6 : Utilisation de la commande « netca » à partir d'un terminal.....	94
Figure 4.7 : Illustration des étapes à suivre pour la configuration d'un <i>listener</i> sous Linux...	96
Figure 4.8 : Exemple d'une requête de test par le modèle bayésien proposé.....	97
Figure 4.9 :L'histogramme associé aux valeurs du gain pour les différents niveaux pour le système XFIRM.....	100
Figure 4.10 :L'histogramme associé aux valeurs du gain pour les différents niveaux pour le modèle de BELKACEM Thiziri.....	102
Figure 4.11 : L'histogramme associé aux valeurs du gain pour les différents niveaux pour notre modèle.....	103
Figure 4.12 :L'histogramme associé au tableau comparatif.....	104

Liste des tableaux

LISTE DES TABLEAUX

Tableau 4.1 : Les 20 requêtes utilisées dans les tests des formules de pondération.....	91
Tableau 4.2 : Caractéristiques de la collection INEX 2006.....	92
Tableau 4.3 : Caractéristiques de la collection de test.....	92
Tableau 4.4 : Les résultats de test obtenus pour le système XFIRM.....	100
Tableau 4.5 : Les résultats de test obtenus pour le modèle de BELKACEM Thiziri.....	100
Tableau 4.6 : Les résultats de test obtenus par le modèle bayésien.....	102
Tableau 4.7 : Tableau comparatif des résultats obtenu	104

Sommaire

Introduction générale

Introduction générale	2
-----------------------------	---

Chapitre I : La recherche d'information et la structure

Partie I : la recherche d'information standard

Introduction.....	3
1. Définition de la RI.....	3
2. Définition d'un SRI avec le processus en U.....	4

Partie II : Recherche d'information structuré

Introduction.....	5
2. XML.....	5
2.1. Qu'est-ce qu'XML.....	5
2.2. Structure d'un document XML	6
2.2.1. Le Prologue	6
2.2.2. Les Commentaires	7
2.2.3. L'arbre d'Eléments	7
2.3. La DTD	7
2.4. Les parseurs XML :	9
2.4.1. DOM (Document Object Model)	9
2.4.2. SAX (Simple API for XML).....	10
2.5. Recherche d'information dans les documents XML	10
2.5.1. Modèle vectoriel étendu	11
2.5.2. Modèle booléen étendu	13
2.5.2. Modèle probabiliste	14
2.5.3. Le modèle d'inférence probabiliste.....	15
2.5.4. Les modèles de langue	16
2.5.5. Les réseaux bayésiens	18
2.5.6. Autres approches	19
2.6. Synthèse	26
2.7. La campagne d'évaluation INEX	27
Conclusion.....	32

Chapitre II : Les réseaux bayésiens et la recherche d'information

Introduction.....	33
-------------------	----

Partie I : Les réseaux bayésiens

1. Présentation	33
2. Définition	33
3. Loi de Bayes	34
4. Les relations de dépendance dans un réseau bayésien	35
5. La notion de D-séparation	36
a. Relation en série.....	37
b. Relation divergente	38
c. Relation convergente	39
6. Apprentissage des réseaux bayésiens	39
a. Apprentissage de structure	40
b. Apprentissage par paramètre	40
7. L'inférence dans un réseau bayésien	40

Partie II : Les modèles de recherche d'information basée sur les réseaux bayésiens

1. Modèle de réseau bayésien d'inférence de base	41
2. Le modèle de Ribeiro-Neto [Ribeiro-Neto et al., 1996]	44
3. Autres modèle de recherche basé sur les réseaux bayésiens	46
3.1.Le modèle d'Indrawan [Indrawan et al, 1996]	46
3.2.Le réseau multi connectés pour la RI	47
3.3.Le modèle de Acid [Acid et al, 2003]	50
3.4.Modèle de Brini [Brini et al , 2005] basé sur les réseau possibilistes.....	51
4. Synthèse	54

Partie III : Les modèles de RIS basés sur les réseaux bayésiens

1. Les travaux de Myaeng [Myaeng et al, 1998]	55
2. les travaux de Piwowarski [Piwowarski et al, 2002, 2003, 2005]	56
3. Les travaux de Crestani [Crestani et al, 2003] et [Crestani et al, 2004]	58
4. Les travaux de De Campos [L.M. de camps and al, 2003]	61
5. L'approche proposée par Alimazighi [Alimazighi et al, 2005]	62
6. Les travaux de Bessai [Bessai et al, 2010]	64
7. L'approche proposée par Najeh [Najehnaffakhi, 2009]	67
8. Synthèse	71
Conclusion.....	73

Chapitre III : Un modèle proposée basée sur les réseaux bayésiens

Introduction.....	74
1. Approche proposée par [Djebbar&Lamrous, 2013]	74
1.1.Description du modèle	75
1.2.pertinence d'un document vis-à-vis d'une requête	76

1.3. Configuration d'élément répondant à la requête	76
1.4. Apprentissage du modèle	76
1.4.1. Calcul de la probabilité à priori $P(d_i)$	77
1.4.2. Calcul $P(Q \theta^i)$	77
1.4.3. Calcul de la valeur de $P(\theta^e d_i)$	79
1.4.4. Calcul de $P(\theta^i \theta^e) = P(t_1, \dots, t_n \theta^e)$	80
Conclusion.....	81

Chapitre IV : Expérimentation et résultats

Introduction.....	82
-------------------	----

Partie I : Environnement et outils d'implémentation

1. VMware Workstation	82
2. Le système de recherche Xfirm	83
2.1 Présentation	83
2.2 Modules exploités	84
3. L'environnement de développement Eclipse	84
3.1 Le langage Java	85
4. Le SGBD Oracle10g EX	86
4.1 Définition	86
4.2 Les fonctionnalités d'Oracle	86
4.3 Les outils de connexion au SGBD et à la BDD	87
4.3.1 Le pilote de connexion au SGBD Oracle	87
4.3.2 L'ODBC/JDBC	87

Partie 2 : Tests et évaluation des résultats

1. Protocole d'évaluation	88
1.1 Les requêtes de tests	88
1.2 La collection de test	90
2. Procédure d'implémentation	90
2.1 Phase d'indexation	90
2.2 Phase de recherche.....	91
3. La mesure d'évaluation	95
4. Tests préliminaires	95
5. Tests et résultats	97
Conclusion.....	103

Conclusion générale

Conclusion générale	104
---------------------------	-----

ANNEXE

Annexe A.....	106
Annexe B.....	117

Introduction

Introduction générale :

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

En général, l'utilisateur se contente de donner quelques mots clés. Ces derniers sont issus d'une connaissance générale sur un domaine donné. Par conséquent, les documents renvoyés par le système de recherche peuvent appartenir à des domaines et disciplines différents par lesquels l'utilisateur n'est pas concerné, c'est pour cela qu'il trouve l'information voulu (ou recherché) noyé dans les documents retournés par le système de recherche d'information. Ces systèmes, popularisés par des moteurs de recherche telle que « Google ou Yahoo », permettent un accès à une information dont l'unité indivisible la plus petite est le document entier.

La récente explosion du Web donne accès à une quantité d'informations énorme et sans cesse croissante en termes d'échange d'information et de communication, ce qui a donné l'essor à une nouvelle technologie qui est « XML » qu'on trouve utilisée dans la plupart des domaines (Recherche d'Information Structurée, Bases de Données, finance, e-commerce, e-Learning, statistique, web sémantique, ...). Dans le cadre de la Recherche d'Information Structurée (RIS), le granule documentaire considéré n'est plus le document entier mais une partie ou un fragment de celui-ci.

Notre contribution s'inscrit dans le cadre des SRI utilisant un modèle de recherche basé sur les réseaux bayésiens qui est un modèle graphique issu d'une rencontre entre la théorie probabiliste et la théorie des graphes dans lequel les documents et les requêtes sont représentés par des nœuds, et leurs liens sont graphiquement modélisés par des arcs. Cette structure est annotée par des tables de probabilités.

Notre travail se divise en 4 chapitres :

- Le premier présente un état de l'art sur la Recherche d'information structurée tout en définissant les notions générales du langage XML, ainsi qu'une présentation des différents modèles de recherche dans les documents structurée et les différentes mesures utilisées pour l'évaluation des SRIS et nous finissons avec une synthèse.
- Le deuxième chapitre 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 avec une synthèse en guise de conclusion. Et enfin nous avons présenté et synthétisé les différents modèles de RIS basé sur les réseaux bayésiens.
- Le troisième chapitre porte sur la présentation de l'approche que nous avons implémentée
- Le quatrième et dernier chapitre présente en partie 1 l'environnement et les outils d'implémentation et en partie 2 les tests et évaluation des résultats et enfin une

comparaison avec un modèle basé sur approche probabiliste, le modèle de langue en l'occurrence.

Nous clôturons notre rapport par une conclusion générale et des perspectives à entreprendre pour enrichir notre travail.

Chapitre I :

La recherche d'information et la structure

Partie I : la recherche d'information standard

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 en information 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.

1. Définition de la RI :

La recherche d'information est un ensemble de programmes et d'opérations permettant la gestion, la représentation, l'interrogation, la recherche, le stockage et la sélection des informations répondant à une requête. A l'aide de la requête on effectue l'interrogation de la collection de documents et celles-ci exigent un appariement entre la requête et les documents.

Le but principal de la recherche d'information est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information.

On distingue 4 types génériques de besoin d'information et donc de stratégies de recherche :

- *La recherche d'un élément connu* : l'utilisateur sait exactement quels éléments il recherche. Il sait reconnaître les éléments désirés s'il les voit. Par exemple, il recherche une citation bibliographique précise et possède assez d'information pour l'identifier sans ambiguïté.

- *La recherche d'information spécifique* : l'utilisateur recherche une information spécifique mais ignore sous quelle forme elle se présente. Cette information peut d'ailleurs être présentée à plusieurs reprises. Exemple : A quelle date le président

Kennedy a-t-il été assassiné ? La réponse sera trouvée ou non, mais ne peut être partielle.

- *La recherche d'une information générale* : l'utilisateur recherche une information sur un sujet en général. Il existe de nombreuses façons de décrire le sujet. Il est possible que l'information pertinente ne soit pas reconnue, et cette information peut ne satisfaire l'utilisateur que de façon partielle.

- *L'exploration* : le but n'est pas de répondre à une question en particulier, mais de parcourir l'ensemble des données pour découvrir quels types d'informations concernant un sujet ou un domaine sont présents.

La recherche d'un élément connu nécessite la mise en place de langage de requête structuré tel que SQL pour les bases de données ou XQuery pour XML, comme illustré dans les trois premières stratégies qui sont le sujet d'étude du domaine scientifique distinct. En ce qui concerne le troisième type de besoin est l'apanage de la recherche d'information.

2. Définition d'un SRI avec le processus en U :

Le processus de recherche d'information consiste à satisfaire les besoins en information des utilisateurs. La réalisation de cette tâche consiste principalement à mettre en œuvre un processus appelé processus en U illustré dans la figure 1.1 ci-dessous, dont le but est de retourner à l'utilisateur le maximum de documents pertinents par rapport à son besoin en information et le minimum de documents non-pertinents.

Dans ce processus, les utilisateurs doivent formuler leurs besoins en information sous forme d'une requête. Avant la mise en correspondance des besoins utilisateurs (requêtes) et des informations (documents), une phase préalable d'analyse des documents, indépendante du besoin, est nécessaire.

Le but du SRI est alors de faire correspondre au mieux la pertinence système avec la pertinence utilisateur.

Processus en U de Recherche d'Information [Salton et al, 1984] est schématiquement représenté sur la figure 1.1 :

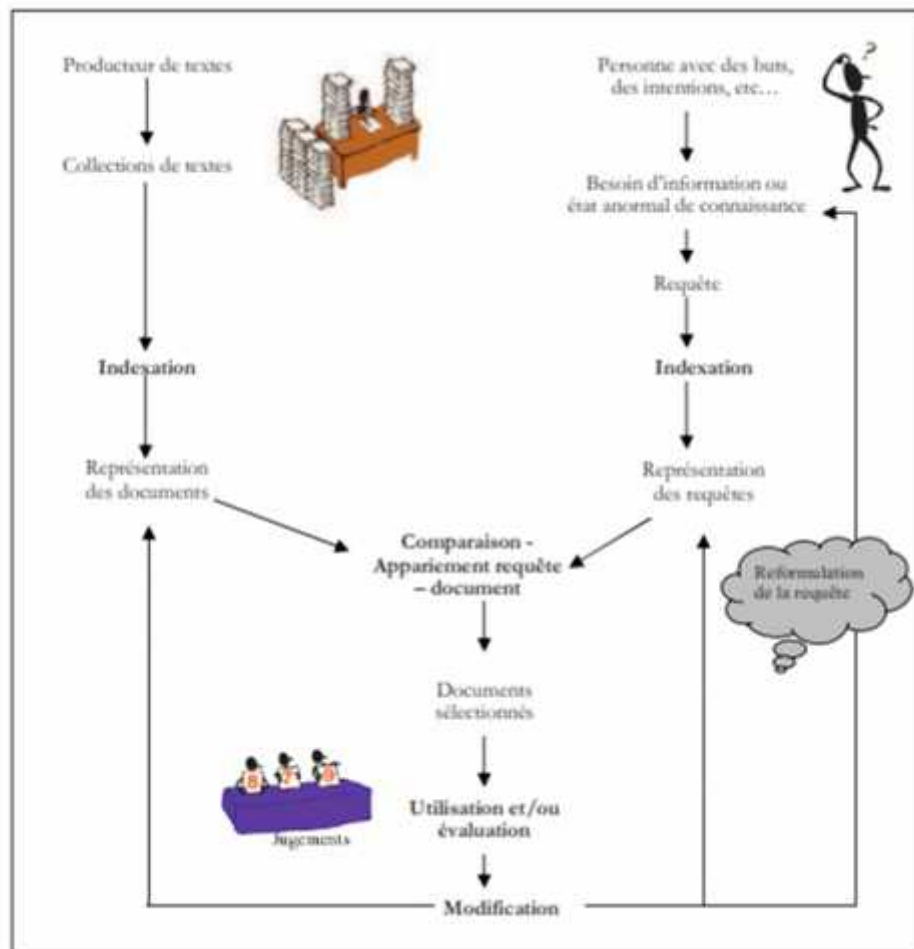


Figure 1.1 : processus en U de la recherche d'information [Sauvagnat,2005]

Ce processus est composé de trois fonctions principales:

- l'indexation des documents et des requêtes ;
- l'appariement requête-document, qui permet de comparer la requête et le document ;

- et la fonction de modification, qui intervient en réponse aux résultats obtenus. Les modifications éventuelles concernent les documents (ajout ou suppression éventuels de la base de données) ou la requête. Les modifications les plus courantes concernent la requête seulement.

Pour plus de détails concernant les concepts de base de la recherche d'information standard voir annexe A.

Partie II : Recherche d'information structuré

Introduction :

Ce n'est que récemment, que l'utilisation de l'information apportée par la structure, a commencé à être utilisée dans le domaine du traitement automatique de l'information textuelle. Internet, et plus récemment XML, ont favorisé l'essor des documents structurés par les liens entre les documents et par la structure interne des documents.

La communauté en RI s'intéresse actuellement à cette évolution, car elle voit dans la structure, un moyen intéressant permettant d'affiner aussi bien la représentation des documents, que la notion d'unité d'information, qui était jusque-là liée au document entier. Ceci a engendré, une problématique majeure, liée à la granularité de l'information à retourner. En effet, la prise en compte de l'information structurelle, devrait permettre, de mieux cibler l'information requise par les utilisateurs. Cette problématique sera détaillée dans le présent chapitre, ainsi que les différentes solutions proposées dans la littérature pour sa résolution. Ce chapitre sera donc consacré, à la présentation d'un aperçu des travaux effectués en RI structurée.

2. XML

2.1. Qu'est-ce qu'XML :

Le développement du document électronique et du web ont conduit à l'émergence des formats de donnée structurées, tel que SGML, HTML et XML, permettant de représenter les documents sous forme plus riche que le simple contenu [L.Denoyer et al., 2004]. L'avantage de ces documents est qu'ils possèdent une structure qui facilite leur présentation, ainsi que leur interprétation et leur exploitation dans des concepts présentant différents besoins.

Présentation :

XML (eXtensible Markup Language) ou langage à balises extensible est apparu comme nouveau standard pour la représentation et l'échange de données sur Internet. Il sépare le contenu d'un document des instructions de sa représentation, car comme l'HTML, XML utilise des balises et des attributs.

La différence entre les deux est que l'HTML définit la signification de chaque balise et chaque attribut, et XML utilise le balisage pour délimiter les éléments de données tant dis que leurs interprétations s'effectuent par l'application qui les lit.

Parmi les points remarquables d'XML [Allorge, 2000] : XML apparaît comme un format d'échange de données universel. Il garantit à ses utilisateurs l'indépendance de leurs documents de toute technologie propriétaire.

2.2. Structure d'un document XML :

Tout document XML se compose [Allorge, 2000]:

- d'un *Prologue*, dont la présence est facultative mais conseillée. Il contiendra un certain nombre de déclarations.
- d'un *arbre d'éléments*. Il forme le contenu du document.
- de *commentaires* et d'*instructions de traitement*, dont la présence est facultative. Ils pourront, moyennant certaines restrictions, apparaître aussi bien dans le prologue que dans l'arbre d'éléments.

2.2.1. Le Prologue :

Il contient :

- Une déclaration XML
- Des instructions de traitement
- Une déclaration de type document (DTD)
- **Déclaration XML :**

Syntaxe d'une déclaration :

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
```

Cette déclaration indique au programme qui va traiter le document :

- la version du langage XML utilisée. Ici la version est 1.0
- Encoding : le codage de caractères utilisé dans le document qui est ISO-8859-1. Ce codage de caractère contient le code ASCII (American Standard Code for Information Interchange) ainsi que les lettres et les caractères accentué nécessaire a présenté la plus part des alphabets latins comme l'anglais, le français... etc.
- Standalone : l'existence ou non de déclarations extérieures au document qui doivent être prise en compte pour le traitement de celui-ci et dans l'exemple précédent le document contient des déclarations extérieures

Remarque :

La déclaration est facultative mais fortement conseillée. Chacune des trois informations est elle aussi facultative mais si elles apparaissent c'est obligatoirement dans cet ordre.

- **Instructions de traitement :**

Elles sont facultatives et ne font pas partie du document. Leur contenu est simplement transmis à une application en vue de déclencher certains traitements.

- Déclaration de Type de Document (DTD) :

XML fournit un moyen de vérifier la syntaxe d'un document grâce au DTD. La DTD permet d'établir des règles de définition d'un document XML, les éléments à inclure, leur type et les valeurs par défaut à leur attribuer. Elle peut être déclarée au sein du document ou sous forme d'un document externe. Il n'est pas obligatoire pour un document XML de se conformer à une DTD. Cependant il doit être *bien formé*, c'est-à-dire qu'il doit respecter les règles générales de syntaxe d'XML.

Lorsqu'une DTD est associée à un document XML, on dit qu'il est *valide*.

2.2.2. Les Commentaires :

Syntaxe : 

Il se positionnent n'importe où après le prologue et peuvent figurer sur plusieurs lignes. Le corps du commentaire peut contenir n'importe quel caractère à l'exception de la chaîne « -> ». On ne peut donc pas inclure un commentaire dans un autre. Le commentaire ne peut pas non plus être inclus à l'intérieur d'une balise.

2.2.3. L'arbre d'Eléments :

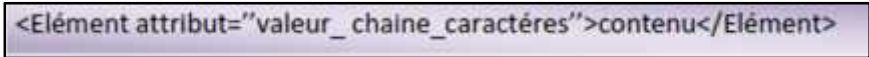
C'est une hiérarchie qui exprime la partie cruciale du document et construit sa structure sémantique, constituée essentiellement d'un unique élément racine qui contient l'ensemble des éléments ayant ou non des attributs.

a- Les Eléments

Syntaxe : 

Un élément est une unité sémantique composée des balises d'ouverture et de fermeture portant le nom de l'élément et son contenu. Un élément vide est représenté par `<nom_element/>`

b- Les attributs

Syntaxe : 

Un attribut est une information secondaire associée à une balise d'ouverture

2.3. La 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.

Syntaxe : `< !DOCTYPE nom_type [déclaration]>`

Nom_type : c'est un nom choisit arbitrairement, il sert à indiquer la portée des déclarations définit entre crochets. Elles seront valides pour tout document dont l'élément racine sera du type Nom_type.

Afin d'être valide tout document doit inclure dans son prologue une déclaration de type, et cette dernière doit contenir, soit directement soit par référence à des entités externes, des déclarations pour tous les éléments, attributs, notations et entités qui seront utilisés dans le document.

Une DTD peut être définit de deux façons :

a. DTD interne :

Un document valide contiendra directement sa DTD au sein de la déclaration type

Exemple :

```
<?xml version="1.0" standalone="yes" ?  
<!DOCTYPE carte [<!ELEMENT carte (#PCDATA)>]>  
<carte>As de pique</carte>
```

Standalone doit prendre la valeur de yes, indiquant que le document est complet et qu'aucune déclaration externe ne doit être recherchée.

b. DTD externe :

L'avantage de la DTD externe est la possibilité de la réutiliser. En effet cette DTD est stocké dans une entité externe.

Exemple :

```
<?xml version="1.0" standalone="no" ?>  
<!DOCTYPE carte SYSTEM "carte.dtd">  
<carte>As de pique</carte>
```

Fichier jeux.xml

```
<?xml version="1.0" encoding="ISO8859-1" ?>  
<!ELEMENT carte (#PCDATA)>
```

fichier carte.dtd

standalone doit prendre la valeur de no, indiquant au processeur que des déclarations sont contenue dans des entités externes.

2.4. Les parseurs XML :

Le rôle des parseurs est de choisir la façon de stocker les documents XML en mémoire et de les parcourir.

Cet analyseur (ou parser en anglais) permet de valider ou pas un document XML, permet aussi de récupérer des balises, leur contenu, leurs attributs et de les rendre accessibles.

XML dispose de deux types de parser : DOM (Document Object Model) orienté hiérarchie et SAX (Simple API for XML) orienté événement.

2.4.1. DOM (Document Object Model) :

Le DOM (Document Object Model ou modèle objet de document) est une API (Application Programming Interface ou interface de programmation d'application) permettant de représenter un document XML sous forme d'arbre (structure arborescente). Chaque objet représente une entité (document, élément, attribut, texte,...) d'un document XML.

Le document XML après avoir été chargé en mémoire, est accessible au travers d'un ensemble d'objet 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.

Le modèle DOM est une recommandation du W3C depuis octobre 1998.

Exemple de document XML et l'arbre DOM qui lui est associé [Amann, 2003]

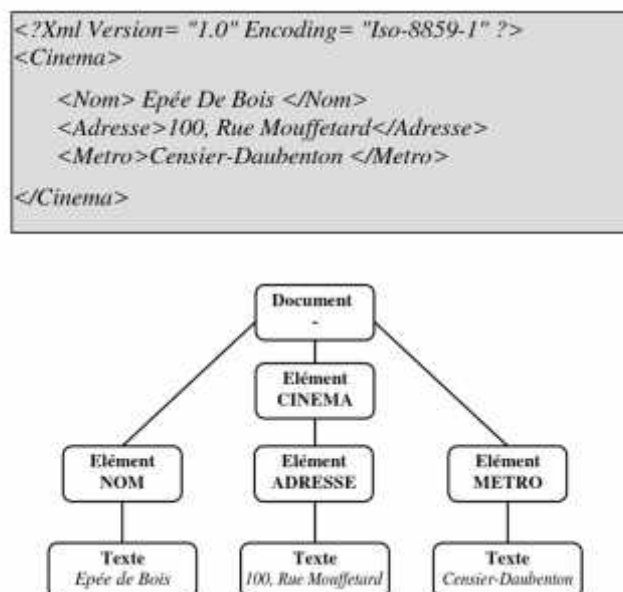


Figure 1.2 : Exemple d'arbre DOM

Il existe dans pratiquement tous les langages des implémentations de DOM pouvant lire des documents XML.

2.4.2. SAX (Simple API for XML):

SAX fournit une interface événementielle pour parcourir un document XML. Cette API renvoie des événements à l'application qui manipule le document XML (ouverture de balise, fermeture de balise, contenu textuel...). Elle permet donc, de traiter à la volée l'occurrence de telle ou telle balise.

SAX est bien adapté pour les traitements qui ne nécessitent qu'une seule passe sur le document, ou dans le cas de gros volumes de données, s'il n'est pas nécessaire d'avoir une représentation complète des données en mémoire.

Sax utilise des méthodes qui permettent la gestion des événements dont : startDocument(), endDocument(), startElement(..), endElement(..) ...

C'est grâce à cela que SAX génère un événement à chaque fois qu'une partie d'un document précis est rencontrée (début ou fin de document, début d'un élément...).

2.5. Recherche d'information dans les documents XML :

Afin de tenir compte de l'information structurelle contenue dans les documents XML, plusieurs approches orienté RIS présenté dans la littérature sont des adaptations des modèles traditionnels. Bien que, il existe d'autres auteurs qui ont proposé d'autres modèles. Parmi tous ces auteurs on distingue ceux qui ont utilisé pour leurs modèle, des recherches portant sur le contenu (CO(content only) : les requêtes sont formulées avec de simple mot clé) et d'autre sur le contenu et la structure (CAS(content and structure) : contiennent des contraintes sur la structure du document).

En parallèle de nouveaux modèles pour la RIS ont été proposés, on en distingue deux catégories :

Modèles de propagation de score : Ils 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. Nous pouvons citer dans cette catégorie le modèle XFIRM [Sauvagnat, 2005].

Modèles de propagation de termes : Ils consistent à propager le contenu des nœuds feuilles à ses ancêtres, et les scores sont calculés par la suite pour chaque nœud indépendamment. [Fellag, 2006], [Cui & al, 2003] et XIVIR [Benaouicha, 2009] sont des exemples de ces modèles.

Dans ce qui suit nous allons présenter quelques approches qui ont été proposés.

2.5.1. 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.

Les approches proposées sont :

- **Modèle de Fuller [Fuller & al, 1993] :**

Ici on trouve une des premières adaptations du modèle vectoriel.

Dans ce modèle la pertinence d'un nœud peut être calculée à part, puis combinée avec la pertinence des nœuds descendants.

La mesure de similarité d'un nœud n par rapport à une requête $q = t_1, t_2, \dots, t_k$ est représentée par l'équation suivante :

$$sim\ q, n = \alpha\ T \cdot cosm\ q, n + \frac{S}{\beta^{k-1}} \frac{\prod_{l=1}^T w_l^q * w_l^n}{|n|}$$

Avec :

$\alpha\ T$: 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.

w_l^q et w_l^n : sont respectivement le poids du terme t_l dans la requête q et dans le nœud n .

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

- **Modèle de Grabs [Grabs & Scheck, 2002] :**

Les auteurs proposent donc de mesurer l'importance d'un terme dans un élément en fonction de son importance dans les éléments de même type.

Exemple :

Si la requête porte sur les paragraphes qui parlent des chiens alors l'ensemble des éléments considérés va être l'ensemble des paragraphes.

La similarité d'un élément e de type cat (requête portant sur le contenu et la structure) à la requête q est calculée de la manière suivante :

$$RSV\ e, q = \sum_{t \in \overline{terms}(q)} tf\ t, e \cdot ief_{cat}\ t^2 \cdot tf(t, q)$$

Où $tf\ t, e$ est la fréquence du terme t dans l'élément e .

$$ief_{cat} = \log \frac{N_{cat}}{ef_{cat}(t)}$$

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 seront traitées comme suit :

- Soit $SE(e)$: l'ensemble des descendants de e incluant e ,
- $\forall se \in SE(e), l \in \text{path}(e, se)$ 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 :

$$RSV(e, q) = \frac{\sum_{se \in SE(e)} \sum_{t \in \text{terms}(q)} tf(t, se) \cdot aw_l \cdot \text{ie}_{cat se} \cdot t^2 \cdot tf(t, q)}{\sum_{l \in \text{path}(e, se)}$$

- **Modèle JuruXml [Carmel & al, 2003] ainsi que [Mass & al, 2003] :**

Les auteurs proposent d'indexer les éléments selon leur type et d'appliquer ensuite le modèle vectoriel.

Ils exploitent l'idée d'utiliser la paire (terme, contexte), le contexte est le chemin où le terme apparaît.

Les requêtes orientées contenu : elles sont évaluées sur chacun des index, et les résultats une fois normalisés sont ensuite fusionnés afin de fournir à l'utilisateur une liste unique de résultats.

Les requêtes structurées : elles sont évaluées 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 chemins est calculée par la fonction $Cr(c_l^q, c_l^e)$

$$Cr(c_l^q, c_l^e) = \begin{cases} \frac{1 + c_l^q}{1 + c_l^e} & \text{si } c_l^q \text{ est une sous séquence de } c_l^e \\ 0 & \text{Sinon} \end{cases}$$

Où

c_l^q est la condition de chemin pour un terme t_l dans la requête q .

c_l^e est le XPath de ce terme t_l dans l'élément e .

On a aussi :

$$RSV(e, q) = \frac{\sum_{(t, c_l^q) \in q} \sum_{t, c_l^e \in e} w_q(t) \cdot w_e(t) \cdot Cr(c_l^q, c_l^e)}{|q| \cdot |e|}$$

Où :

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

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

- **Modèle de Yang [Yang & al, 2007] :**

Les auteurs proposent de représenter dans ce modèle un document XML par une matrice. Ils représentent ainsi un élément et une requête par deux matrices. La figure 1.3 illustre la représentation matricielle d'un document XML.

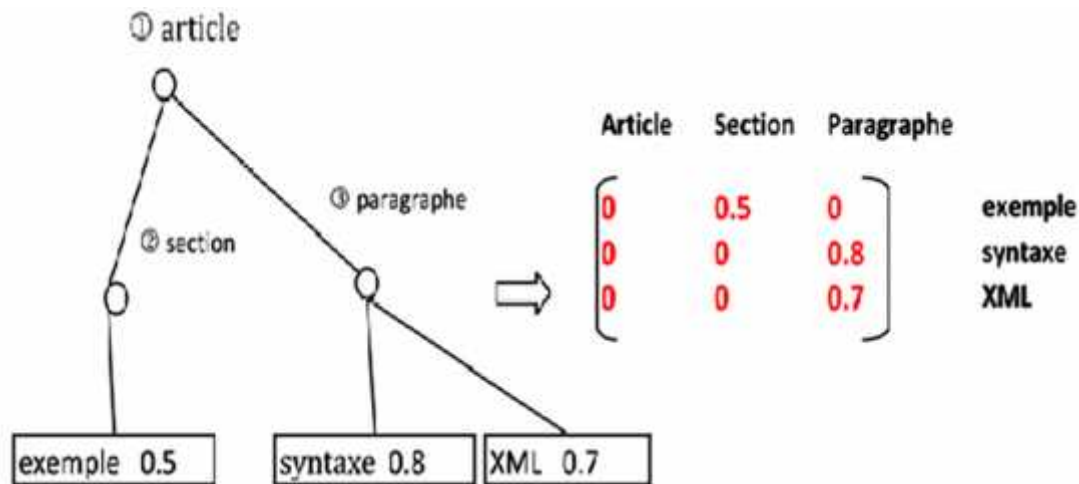


Figure 1.3 : représentation matricielle d'un document [Yang et al., 2007]

Où :

Les valeurs de cette matrice sont les poids des termes dans les différents éléments.

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

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

2.5.2. Modèle booléen étendu :

- Modèle de Hatano [Hatano & al, 2002] :

Ils 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 :

$$q_i = \begin{cases} 1 & \text{si le terme apparaît dans la requête} \\ 0 & \text{sinon} \end{cases}$$

L'unité d'information à renvoyer à l'utilisateur est définie comme étant tout élément contenant au moins un élément feuille.

Une première similarité est calculée avec :

$$RSV^{(0)}(q, e) = \cos(\vec{q}, \vec{e})$$

Avec :

$$w(t, e) = \frac{tf_t(e)}{\sum_{t' \in T} \sum_{e' \in E} tf_{t'}(e')} \times \log \frac{N}{n_t}$$

$w(t, e)$: représente le poids du terme t dans l'élément e

$tf_t(e)$: Fréquence du terme t dans l'élément e

T : ensemble du vocabulaire

E : ensemble des éléments du document considéré

N : Nombre total de documents dans le corpus (collection)

n_t : Nombre total de documents contenant le terme t .

Le score final est calculé en utilisant la p-norme comme suit :

$$RSV_{q,e} = 1 - \frac{1}{|enf(e)|^{\frac{1}{p}}} \sum_{e' \in enf(e)} 1 - RSV_{q,e'}^p$$

Où :

P est généralement choisi avec une valeur égale à 2.

Lorsqu'un élément e est une feuille $RSV = RSV^{(0)}$

$|enf(e)|$: Nombre de terme de l'élément enfant de e .

- **Modèle de Larson [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 présentée ci-dessous :

$$P(R/Q,C)=P(R/Qbool,C)*P(R/Qprob,C)$$

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.

- **Modèle de Trotman [Trotman, 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 [Braga & al, 2002]. 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 (« Best Matching » ou meilleur arrangement) [Rob & al, 1976].

2.5.2. Modèle probabiliste :

- **Modèle de Lalmas [Lalmas, 1997] :**

Cet auteur se base sur la théorie de Dempster-Shafer [Shafer, 1976] pour proposer un modèle qui permet d'indexer et de retrouver des documents structurés. L'intérêt de la théorie de Dempster-Shafer en RIS tient à l'existence d'un opérateur permettant la combinaison de croyance provenant de plusieurs éléments.

2.5.3. Le modèle d'inférence probabiliste :

Pour étendre le modèle probabiliste aux documents XML, les probabilités doivent tenir compte de l'information structurale. Une approche est d'utiliser des probabilités conditionnelles sur les chemins des documents,

Avec par exemple :

$$P(d/t) \text{ devenant } P(d/p \text{ contains } t)$$

où :

d : représente un document ou une partie de document

t : est un terme

p : est un chemin dans l'arbre structurel d .

- Modèle de Fuhr [Fuhr & al, 2001] :

Les auteurs ont proposé une méthode d'augmentation basée sur le modèle probabiliste. Cette méthode est basée sur le langage de requêtes XIRQL, et a été implémentée au sein du moteur de recherche HyRex [Gövert & al, 2002].

Dans cette approche, les nœuds sont considérés comme des unités disjointes. Tous les nœuds feuilles ne sont cependant pas indexés (car d'une granularité trop fine). Dans ce cas-là les termes sont propagés jusqu'au nœud indexable le plus proche. Afin de préserver des unités disjointes, on ne peut associer à un nœud que des termes non reliés à ses nœuds descendants.

Le poids de pertinence des nœuds dans le cas de requêtes orientées contenu est calculé grâce à la propagation des poids des termes les plus spécifiques dans l'arbre du document. Les poids sont cependant diminués par multiplication par un facteur, nommé facteur « d'augmentation ».

Exemple :

Considérons la structure de document de la figure 1.4 ci-dessous, contenant un certain nombre de termes pondérés (par leur probabilité d'apparition dans l'élément), et la requête « XML ».

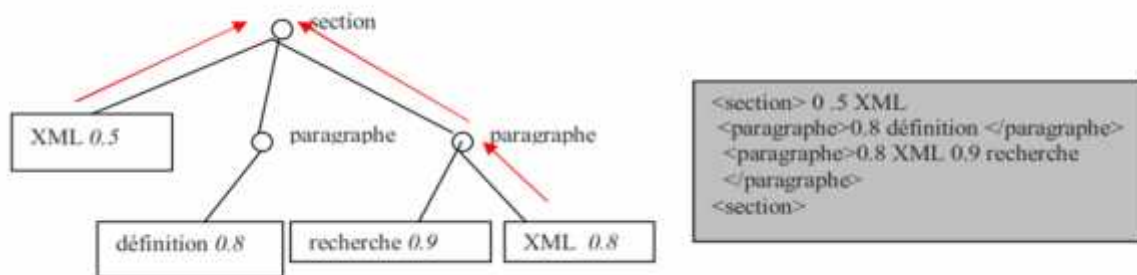


Figure 1.4 : modèle d'augmentation [Fuhr & al,2001]

Le poids de pertinence de l'élément section est calculé comme suit, en utilisant un facteur d'augmentation égale à 0.7 :

$$P([section, XML]) + P([paragraphe[2]]).P([paragraphe[2], XML]) - P([section, XML]).P([paragraphe[2]]).P([paragraphe[2], XML]) = 0.5 + 0.7 * 0.8 - 0.5 * 0.7 * 0.8 = 0.68.$$

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

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.

2.5.4. Les modèles de langue :

- Modèle de Sigurbjaornsson [Sigurbjaornsson & 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 est de combiner le modèle de langage de l'élément, du document et de la collection. Les auteurs ont utilisé deux types d'index : un index pour les éléments du document XML qui assure la même fonction qu'un fichier inverse en RI classique et un autre (index article) pour tout le document utilisé pour des calculs statistiques. L'arbre XML est indexé en se basant sur le post et le pre-ordre des nœuds.

Pour chaque élément e , on estime le modèle de langage (score) pour une requête donnée q selon la formule suivante :

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

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 et la formule précédente devient alors :

$$P(e|q) \propto P(e) \cdot \prod_{l=1}^k P(t_l|e)$$

Avec

t_l : Terme de la requête.

La probabilité de $P(t_l|e)$ est une interpolation linéaire des trois modèles de langage (élément, article et collection) :

$$P(t_l|e) = \lambda_e \cdot P_{mle}(t_l|e) + \lambda_d \cdot P_{mle}(t_l|d) + (1 - \lambda_e - \lambda_d) \cdot P_{mle}(t_l)$$

Avec :

$P_{mle}(t_l|e)$ est la probabilité de t_l dans le modèle de langage de l'élément estimée par les statistiques à partir de l'index des éléments.

$P_{mle}(t_l|d)$ est la probabilité de t_l dans le modèle de langage du document estimée par les statistiques à partir de l'index article

$P_{mle}(t_l)$ est la probabilité de t_l dans le modèle de langage de la collection.

λ_e et λ_d sont des constantes.

- Modèle de Laitang [Laitang & al, 2013] :

La faible utilisation de la structure dans la RIS pour les modèles de langue, d'après Cyril Laitang s'explique par le fait que l'expression du besoin en information de l'utilisateur

sous forme de requête, porte en priorité sur le contenu souhaité plutôt que sur l'emplacement de ce dernier dans l'arbre du document XML.

L'auteur conduit sa contribution dans l'objectif de proposer un modèle qui prend en compte l'utilité de la structure d'un document et l'utiliser efficacement. Donc, une requête cible un élément à balise (label) spécifique, et le score de pertinence du nœud répondu est donné par la formule suivante :

$$RSV_{n,Q} = \begin{cases} 0 & \text{si } l(n) \neq EC(Q) \\ \lambda \times score_c_{n,Q} + (1 - \lambda) \times score_s_{n,Q} & \text{si non} \end{cases}$$

Avec :

$score_c$ est score de contenu pour le nœud n

$score_s$ est le score de similarité structurelle pour n

$\lambda \in [0,1]$ est le paramètre d'apport du $score_c$ et $score_s$

$EC(Q)$ est la fonction qui retourne la balise spécifique par la requête Q

$l(n)$ est la balise du nœud n

Cette approche commence par l'évaluation des nœuds feuilles qui contient éventuellement l'information textuelle, ceci par le calcul d'une mesure de score dite « score de contenu » basé sur l'approche $tf*idf$ [Sparck Jones, 1973], ainsi :

$$score_c x = \frac{tf(t_i|x) \times idf_{t_i}}{t_i \in Q}$$

Avec x est un nœud feuille de la collection considérée.

Un score est attribué ensuite aux nœuds interne, par une double propagation [Laitang & Pinel-Sauvagnat, 2011] [Laitang & al., 2011] :

- Bottom up : propagation du score $P(x)$ à la racine, ainsi chaque nœud interne n , obtiendra un score $P(n)$ par cumulation des scores qui le traverse (du bas vers le haut).
- Top down : propagation des scores des ancêtres jusqu'aux nœuds feuilles par intuition qu'un nœud pertinent appartient à un document qui l'est aussi.

D'où, le score de contenu brut d'un nœud n est donné par la formule de propagation suivante :

$$bs_c n = \begin{cases} I + V + A & \text{si } n \neq \text{racine} \\ I & \text{si non} \end{cases}$$

$$I = \frac{P(n)}{|feuilles()|} \quad \text{score intermédiaire de contenu}$$

$$V = \frac{P(a_1) - P(u)}{|feuilles(a_1)|} \quad \text{score de voisinage}$$

$$A = \frac{bs_c(a_1, Q) - \frac{P(a_1)}{|feuilles(a_1)|}}{|fils(a_1)|} \quad \text{score des noeuds ancêtres}$$

$bs_c(a_1, Q)$ est le score brut du nœud a_1 à la requête Q

Avec :

$$P(n) = \frac{\sum_{x \in \text{feuilles}(n)} P(x)}{|\text{feuilles}(n)|}$$

$|\text{feuilles}(n)|$: est le nombre de nœuds feuilles du document

a_1 : nœud père

$P(a_1)$: score intermédiaire de a_1

$|\text{feuilles}(a_1)|$: nombre de nœuds feuilles de a_1

Après normalisation, le score final est obtenu par le maximum $\max()$ des scores de contenu brut des nœuds $n_i \in C$ obtenus en réponse à la requête Q :

$$\text{score}_c(n, Q) = \frac{bs_c(n, Q)}{\max(bs_c(n_i, Q) \mid n_i \in C)}$$

Avec :

$$\text{score}_c(n, Q) = \sum_{q_i \in Q} \text{score}_c(n, q_i) \quad \text{et} \quad Q = q_1, q_2, \dots, q_i, \dots, q_k$$

Le score de contenu brut d'un nœud interne est calculé par la somme des scores de contenu pour chaque sou-requête :

$$bs_c(n, Q) = \sum_{q_i \in Q} \frac{\text{score}_c(n, q_i)}{\max(\text{score}_c(n_i, q_i) \mid n_i \in C)}$$

Ce qui est du score de similarité structurelle : $\text{score}_c(n, Q)$ du nœud n est donnée par :

$$\text{score}_c(n, Q) = bs_c(n, Q) = \text{sim}_s(S, Q)$$

Avec :

S est un sous-arbre extrait du document pour répondre à la contrainte structurelle.

$\text{sim}_s(S, Q)$ est la probabilité entre la requête Q et le sous-arbre S calculée par modèle de langue en utilisant l'une des formules proposées par l'auteur.

2.5.5. Les réseaux bayésiens :

Les réseaux bayésiens font l'objet de notre travail, pour cela nous avons préféré leur consacrer un chapitre à part.

2.5.6. Autres approches :

- Modèle maximum des poids [Moffat & al, 2002] :

Les auteurs proposent ici 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 :

$$RSV_{q,e} = RSV_0_{q,e} + \alpha \max_{e' \in enf(e)} |RSV_{q,e'}|$$

Avec : $enf(e)$: enfant de l'élément e .

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

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

α : est une constante avec une valeur de 0.8.

Le calcul du score de la méthode summation se fait ainsi :

$$RSV_{q,e} = RSV_0_{q,e} + \frac{\beta}{|enf(e)|} + \frac{\gamma}{\left| \frac{e' \in enf(e)}{RSV_{q,e'}} > 0 \right|} RSV(q, e')$$

Où

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

β et γ sont des constantes (dans leur article $\beta = \gamma = 0.5$).

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).

- Modèle Entropie et sélection de termes [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

$tf_t(e)$ = fréquence du terme t dans l'élément e .

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

Où :

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

Pour les feuilles de l'arbre, ils définissent :

$$w_{t, e} = \log 1 + tf_{t, e} \log \frac{N}{n_t}$$

Où :

N : nombre total de documents dans le corpus (collection)

n_t : Nombre total de documents contenant le terme t .

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

$$w_{t, e} \geq \text{moyenne}_{e' \in enf(e)} w_{t, e'} + \text{variance}_{e' \in enf(e)} w_{t, e'}$$

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 :

$$RSV_{q, e} = \sum_{t \in \text{termes}(q)} w_{t, e} \times \log \frac{N}{n_t}$$

- Modèle XFIRM [Sauvagnat, 2005] :

Le modèle XFIRM (XML Flexible Information Retrieval Model : modèle flexible pour la recherche dans des document semi-structurés) 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 :

$$P_n = F_n^p \alpha^{dist(n, nf_k) - 1} \times RSV_{q, nf_k}$$

$nf_k \in F_n$

Avec :

- $RSV(q, nf_k)$: poids du nœud feuille nf_k dans le document.

Il est exprimé selon la formule suivante :

$$RSV_{q, nf_k} = \frac{N}{l=1} w_l^q \square w_l^{nf}$$

- $|F_n^p|$: nombre de nœuds feuilles descendants de n ayant un score non nul.

- F_n : est le nombre total de nœuds feuilles nf_k descendants de n .

- $\alpha \square [0,1]$ est un paramètre qui quantifie l'importance de la structure dans l'évaluation du score de pertinence.

- $dist(n, nf_k)$: 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.

w_l^q : le poids du terme i dans la requête q ($w_l^q = tf_l^q$).

w_l^{nf} : le poids du terme i dans nœud feuille nf ($w_l^{nf} = tf_l^{nf} \square ief_l \square idf_l$).

Où :

tf_l^q et tf_l^{nf} sont respectivement la fréquence du terme i dans la requête q dans le nœud feuille nf .

$ief_l = \log(\frac{|Fc|}{|nfi|})$: 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 .

$idf_l = \log(\frac{|D|}{|d_i|})$: permet d'évaluer l'importance du terme i dans la collection de documents,

Où :

$|D|$ est le nombre total de documents de la collection.

$|d_i|$ est le nombre de documents contenant le terme i .

- **Modèle Trotman [Trotman, 2005] :**

Trotman 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 :

$$tf_{id} = \frac{n}{p=1} tf_{ipd}$$

Par la formule :

$$tf'_{id} = \frac{n}{p=1} C_p \times tf_{ipd}$$

Où :

tf_{tpd} : est le nombre d'occurrences du terme t à la position p du document d .

C_p : 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.

- **Modèle Geva [Geva, 2005] :**

Le système GPX (Gardens Point XML IR) consiste à calculer des scores de pertinence pour les nœuds feuilles, et ensuite, propager ces scores vers les nœuds internes.

a. Calcul du score des nœuds feuilles :

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

$$RSV_{q,nf} = K^{n-1} \frac{t_i}{f_i}$$

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.

t_i est la fréquence du i ème terme de la requête dans l'élément nf .

f_i est la fréquence du i ème terme de la requête q dans la collection.

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.

b. 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 :

$$RSV_{q,n} = D_n \frac{RSV_{q,nc_i}}{i}$$

Avec

$RSV(q, nc_i)$: est le score de la pertinence de l' i ème élément fils.

n : est le nombre d'élément fils, $D(n)=0.49$ 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.

- **Modèle de Fellag [Fellag, 2006] :**

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

$$RSV_{q, n_i} = \prod_{k=1}^M P_{qk} \times P_{nik}$$

Avec

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

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

$$P_k = tf_k \times idf_k \times ief_k \quad 1$$

Avec

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

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

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 :

$$\log \frac{Ne}{ne} + \alpha \quad \text{avec } 0.5 \leq \alpha \leq 1$$

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 présenter :

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 :

$$P_{moy} \leq P_{t, e'} \leq P_{max}$$

Avec

$$P_{moy} : P_{moy} = \frac{\sum_{i=1}^{N_{te'}} P_{t_i, e'}}{N_{te'}}$$

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 :

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

Avec

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

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

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

$$moy_{e' \in enf(e)} P(t, e') = \frac{\sum_{e' \in enf(e)} P(t, e')}{Ne'}$$

Ne' : nombre de nœuds e' contenant le terme t .

P_{max} : 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 :

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

- Modèle XIVIR [BenAouicha, 2009] :

Le modèle XIVIR (XML Information Retrieval based on Virtual links) permet la recherche d'information dans des documents XML par la structure, par le contenu et par la combinaison des deux.

1. Recherche par contenu :

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

1.1. la propagation de texte par profondeur :

Qui consiste à traduire le contenu de chaque nœud feuille par un ensemble de termes pondérés, ceux-ci seront propagés vers les ancêtres de ce nœud 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 :

$$w(t, n) = ief_t \times \frac{tf(t, c_k)}{d(n, c_k) + 1}$$

$n \rightarrow c_1 \rightarrow c_2 \rightarrow \dots \rightarrow c_k$

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

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

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

ief_t : est inversement proportionnel au nombre de nœuds de la collection contenant le terme t est calculé comme suit :

$$ief_t = \frac{N}{N_t}$$

Où :

N : est le nombre d'éléments dans la collection

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

1.2. la 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 nœud 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 :

$$w_{t,n} = ief_t \times \frac{tf_{t,ck}}{\prod_{n \rightarrow c1 \rightarrow c2 \dots ck} |Children(e)|}$$

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,

(RSV_c) , est calculé comme suit :

$$rsv_c(n, n') = \sum_{p \in t(n')} w_{p,n} \times w_{p,n'}$$

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.

2. Recherche par la structure :

Le document XML est représenté sous forme d'un arbre défini comme un ensemble de chemins entre deux nœuds $A \rightarrow B$ où A est le nœud parent du nœud B . La relation entre A et B peut être directe (parent/ fils-direct) ou indirecte (parent/descendant). Afin de refléter l'importance de la relation entre les nœuds A et B , un poids est calculé pour chaque chemin. Si la relation est directe, le poids est égal à 1, sinon, le poids w est calculé comme suit :

$$w = \exp \lambda \square 1 - d(A, B)$$

Où $d(A, B)$ est la distance qui sépare les deux nœuds A et B , et λ est un document d , (RSVs), est calculé comme suit :

$$RSV_0 = \frac{w_q \square w_d}{Aq \square w_q Bq \square Eq = Ad \square w_d Bd \square Ed}$$

Où

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

- 3. 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 :

$$RSV_{n,q} = rsv_{n,q} = \alpha xrsv_{c,n,\hat{n}} + 1 - \alpha xrsv_{s,n,\hat{n}}$$

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 défini soit par expérimentation ou bien par l'utilisateur.

2.6. Synthèse :

Nous venons de présenter un certain nombre de travaux qui ont tenté d'adapter les modèles de la RI standard, à la RI structurée.

Dans [Grabs & al, 2002] et [Sauvagnat, 2005] 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).

Dans [Carmel & al, 2003] et [Grabs & scheck, 2002] l'adaptation est l'utilisation d'un vecteur pour la structure dans le cas du vectoriel étendu.

Pour le modèle probabiliste dans [Fuhr & al, 2001] l'adaptation est spécifique 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.

Dans [Cui & al, 2003] adopte la méthode de propagation des termes 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.

Dans [ahn & Moffat, 2002] et [Sauvagnat, 2005] adoptent la méthode de propagation de pertinence (ou de score).

Dans [Fuhr & al, 2001] et [Govert & al, 2002] eux adoptent la méthode de propagation de poids.

Dans les différents travaux que nous avons présentés au préalable, les auteurs ont fait adapter différentes approches à la nouvelle granularité d'information, cela en utilisant différents paramètres intégrés aux formules de calcul de score, dont on trouve :

- La fréquence d'un terme de la requête dans l'élément ;
- La taille du document en nombre de nœuds feuilles ;
- Le poids du terme dans la requête
- Le nombre de terme dans l'élément e .

En conclusion quelques soit l'approche adoptée, le score de pertinence d'un nœud dépend fortement du score de ses descendants

Dans ce qui suit nous allons présenter la compagne d'évaluation des systèmes de RI structurée, en l'occurrence INEX

2.7. La compagne d'évaluation INEX :

INEX (*INitiative for the Evaluation of XML Retrieval*) a vu le jour en 2002 pour répondre à la demande des chercheurs en RI structurée qui ne bénéficiaient pas jusqu'à alors d'un forum d'évaluation spécifique.

Le but d'INEX est de développer une méthodologie et des outils (collections de test, ensemble de requêtes, mesures d'évaluation), pour tester des approches et systèmes particuliers de recherche pour XML, en utilisant un langage de requête qui est NEXI¹ (Narrowed Extend XPath I) et un ensemble de tâches telles que : ad-hoc, la tâche multimédia, la tâche « relevance feedback » et la tâche hétérogène, et par la suite plusieurs compagnies INEX dérivées de celle-ci à plusieurs évolutions et améliorations sont évaluer les SRI participant, telles que :

- INEX 2005-2006 : En 2005 la collection de base est une collection d'outils provenant de la « IEEE Computer Society », ciblée au format XML. Elle est de volume d'environ 500 à 700 Mo composée de 8 millions éléments, équivalent à 12000 articles, d'environ 1500 éléments à profondeur moyenne de 6,9 (nombre de niveaux), provenant de 18 magazines ou revues 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 moyenne de 6.72. Le nombre moyen de nœuds XML par document est 161,35.
- 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. L'évaluation s'effectue par l'étude des trois tâches ad hoc : Focused task (la tâche concentrée), Incontext task (tâche de mise en contexte) composée des deux tâches : Relev in context et Best in context. [INEX 2007 Evaluation Measures (Draft)]

¹ On trouvera une description générale d'INEX dans : <http://inex.is.informatik.uni-duisburg.de/2006>

- 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.
[INEX repport, Repport on INEX 2012]

1. la collection de test :

La collection de documents INEX se compose d'articles scientifiques provenant de la IEEE Computer Society, transcrits au format XML. La collection, d'environ 494 Mo, contient plus de 12000 articles, publiés de 1995 à 2002, et provenant de 18 magazines ou revues différents. Chaque année, depuis 2002, un nouvel ensemble de 60 *requêtes* est défini par les participants et utilisé pour l'évaluation. En moyenne un article contient 1532 nœuds XML.

Les articles sont généralement composés d'un en-tête (*<fm>*), d'un corps (*<body>*) et d'annexes (*<bm>*). Chacun de ces éléments peut à son tour être restructuré : exemple, le corps est composé de section *<sec>* elles-mêmes composées de paragraphes *<p>* et les annexes sont composées de référence bibliographiques *<bibl>* et éventuellement de curriculum vitae *<vt>*.

2. Les requêtes :

Les requêtes (ou Topics) sont créés par les différents participants et doivent être représentatives des demandes de l'utilisateur moyen sur la collection. Les requêtes se divisent en deux catégories principales:

Les CO (Content Only) : Des requêtes en langage naturel dont les mots-clés 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 1.5 : Exemple de requête CO, issue du jeu de test 2003

- Les CAS (Content And Structure) : A l'indication du contenu bref voulu, on ajoute des contraintes de structure telles que les exigences de contenu des éléments du document XML

Pour chaque Topic, est construite une présentation sous forme de différents champs pour expliciter sa portée :

- Title : définition de la forme générale (CO)
- Key words : 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 désires 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 1.6 : Exemple de requête CAS (Topic 205 d'INEX 2005)

Les taches [INEX 2005] :

La tâche ad-hoc :

C'est une recherche qui se fait en simulation de l'utilisation d'une bibliothèque composée de documents XML, qui est interrogé par des requêtes utilisateur conçues dans la compagnie et portent sur le contenu et la structure. Cette principale tâche est fragmentée en trois sous tâches distinctes :

- 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.

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

L'évaluation :

L'évaluation de pertinence des SRI XML passe premièrement par une validation des éléments/documents qui sont jugés à la main. Par la suite, en 2002 une échelle à deux dimensions a été proposée et basée sur les degrés de pertinence et couverture, qui ont été remplacée par les notions d'exhaustivité et de spécificité, depuis 2003. Telles que :

L'exhaustivité est mesurée selon une échelle à quatre niveaux, un élément est :

- 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 tous les aspects du sujet de la requête.

La spécificité décrit à quel point l'élément est focalisé sur 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.

Mesure d'évaluation :

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

a. 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 :

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

i : le rang de l'élément dans la liste

$xCG(i)$: somme des scores de pertinence des documents j ($j = \overline{1, i}$)

$xG(j)$: 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 $nxCG$:

$$nxCG_i = \frac{xCG(i)}{xCI(i)}$$

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 $nxCG(i) \longrightarrow 0$ élément non pertinent

si $nxCG(i) \longrightarrow 1$ 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.

b. 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 , e_{system} (respectivement e_{ideal}) 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 :

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

Tel que : r : c'est le gain

e : le rang correspondant au gain

e_{ideal} : le rang auquel le gain est idéal

e_{system} : le rang auquel le gain est celui retourné par le système évalué.

c. Mesure de precal :

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

$$P_{\text{pert/retr}}(x) = \frac{x \cdot n}{x \cdot n + \text{esl}_{x,n}}$$

Tel que :

pert : document x pertinent

retr : document retrouvée

$\text{esl}_{x,n}$: 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.

Conclusion :

Plusieurs collections de documents XML commencent à se constituer, développant dans le même temps des langages et des systèmes pour chercher l'information dans ces collections.

L'aspect structurel des documents XML offre un double avantage, celui de permettre à l'utilisateur de spécifier précisément ce qu'il cherche, et celui de permettre au système de retourner des documents ou parties de documents les plus spécifiques et les plus exhaustifs. En effet, les documents XML permettent aux utilisateurs de formuler des requêtes non seulement sur leur contenu, mais aussi d'utiliser leur structure pour des recherches plus précises.

Chapitre II :

les réseaux bayésiens

Introduction :

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 bayésiens, initiés par Judea Pearl dans les années 1980, se sont révélés des outils très pratiques pour la représentation de connaissances incertaines, et le raisonnement à partir d'informations incomplètes.

Nous avons réparti ce chapitre en 3 parties :

- la partie 1 : consiste à définir quelques notions importantes sur les réseaux bayésiens, les probabilités conditionnelles et les méthodes d'apprentissage des réseaux bayésiens
- partie 2 : présentations des travaux utilisant une modélisation de type réseaux bayésiens ayant eu plus ou moins de succès
- partie 3 : différents travaux ayant utilisé les réseaux bayésiens dans la recherche d'information structurée

Partie I : Les réseaux bayésiens :**1. Présentation :**

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

2. Définition :

Un réseau bayésien représente d'une façon compacte la distribution de probabilités jointe globale associée à toutes les variables du domaine modélisé. [Tabia, 2005]

Un réseau bayésien est un graphe orienté acyclique DAG (Directed Acyclic Graph) qui permet de représenter les influences (relation de causalité, dépendances directes ou conditionnelles) entre les variables du domaine et munie d'un ensemble de tables de probabilités conditionnelles pour quantifier l'incertitude (ou le déterminisme) dans les relations de causalité.

Ainsi un réseau bayésien $\beta = G, \theta$ peut être formellement défini par :

$G = V, E$, un graphe acyclique orienté où :

- V est l'ensemble des nœuds de G ,
- E est l'ensemble des arcs de G

$\theta = P(V_i | Pa(V_i))$ ensemble des probabilités de chaque nœud V_i conditionnellement à l'état de ses parents $Pa(V_i)$ dans G .

Un réseau est appréhendé selon un aspect qualitatif et quantitatif :

➤ **Aspect qualitatif :**

L'aspect qualitatif du graphe indique les dépendances (ou indépendances) entre les variables et donne un outil visuel de représentation des connaissances, outil plus facilement appréhendable par ses utilisateurs.

Exemple : Un graphe orienté acyclique DAG comme nous l'avons vu précédemment $G = (V, E)$ où V est l'ensemble des nœuds où chaque nœud représente une variable aléatoire, discrète ou continue, décrivant le domaine. E est l'ensemble des arcs où un arc de A vers 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 entrants vers A .

➤ **Aspect quantitatif :**

Est défini par une collection de tables de probabilités conditionnelles θ_i tel qu'à chaque nœud X_i est attachée une table de probabilités conditionnelles (TPC) comme suit :

Soit Pa_i les parents du nœud X_i

La TPC du nœud X_i contient les probabilités $P(X_i = x_i | Pa_i = pa_i)$ pour toute valeur x_i de X_i et toute combinaison de valeurs pa_i de Pa_i

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

$$P(X_1 \dots X_n) = \prod_{i=1}^n P(X_i | Pa(X_i))$$

Où :

$Pa(X_i)$ représente l'ensemble des parents (cause directes) de X_i . La loi de factorisation peut être expliquée par la loi de Bayes et la notion de d-séparation (abordée dans la section qui suit)

3. Loi de Bayes :

La formule de Bayes est définie comme suit :

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

Elle permet de calculer la probabilité d'une variable sachant la valeur d'une autre variable.

$P(A|B)$: est dite probabilité a posteriori, probabilité de A étant donné (sachant) B.

$P(B|A)$: Pour un B connu est appelée la fonction de vraisemblance de A. A est dite l'évidence¹.

$P(A)$: est dite probabilité à priori (probabilité marginale) de A.

$P(B)$: est dite probabilité à priori (probabilité marginale) de B.

4. Les relations de dépendance dans un réseau bayésien :

Il existe différentes connexion dans un réseau bayésien, les nœuds peuvent être reliés en différentes topologies comme suit :

i. Connexion en série :

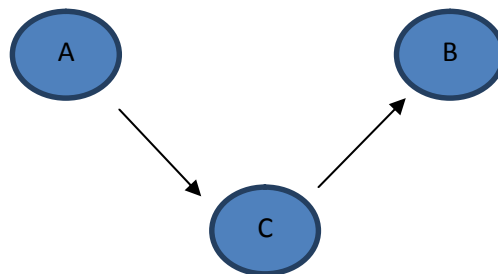


Figure 2.1 : connexion en série

Dans ce cas (figure 2.1) on dit que A et B sont indépendant (ou d_séparés) étant donné C.

$\left. \begin{array}{l} A \text{ a une influence sur } C \\ C \text{ a une influence sur } B \end{array} \right\} \Rightarrow$ 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.

ii. Connexion divergente :

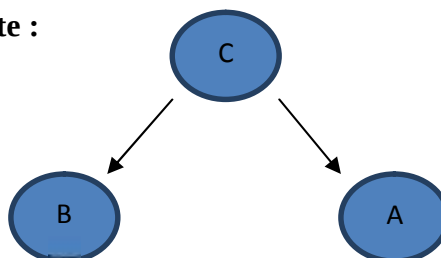


Figure 2.2 : connexion divergente

Les enfants A et B sont indépendant étant donné C.

L'information dans ce cas peut passer entre les enfants de C lorsque la variable C est non instanciée.

¹ A été traduit mot à mot de l'anglais evidence qui signifie en réalité dans notre contexte la preuve.

iii. Connexion convergente (ou la V_structure)

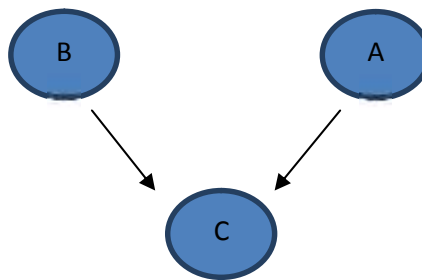


Figure 2.3 : connexion convergente.

Dans ce type de connexion décrite dans la figure 2.3, 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 indépendants (l'état de A donnera une information sur l'état de B).

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 simplifiés.

La d-séparation étend la notion d'indépendance conditionnelle aux variables qui ne sont pas parents directes de la variable d'intérêt.

L'hypothèse de base qui différencie un réseau bayésien d'un système de déduction est que :

- Dans le système de déduction, on applique l'implication $A \Rightarrow B$ uniquement en transmettant l'information dans le sens de ce qui est connu de A implique ce qui est connu de B.
- Dans un réseau bayésien, une part est faite à l'abduction², en ce sens qu'on admet qu'une certaine connaissance sur B implique une certaine connaissance de A.

Si $P(A)$ est une constante connue alors une variation de $P(B)$ ne modifie pas $P(A)$, évidemment, mais elle modifie une quantité bayésienne : $P(A|B)=P(A)/P(B)$.

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 :

²Mouvement qui consiste à écarter un membre ou un segment de membre de l'axe du corps. Raisonement par lequel on restreint dès le départ le nombre des hypothèses susceptibles d'expliquer un phénomène donné.

- Soit la connexion est en série ou divergente et C est instancié.
- 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.

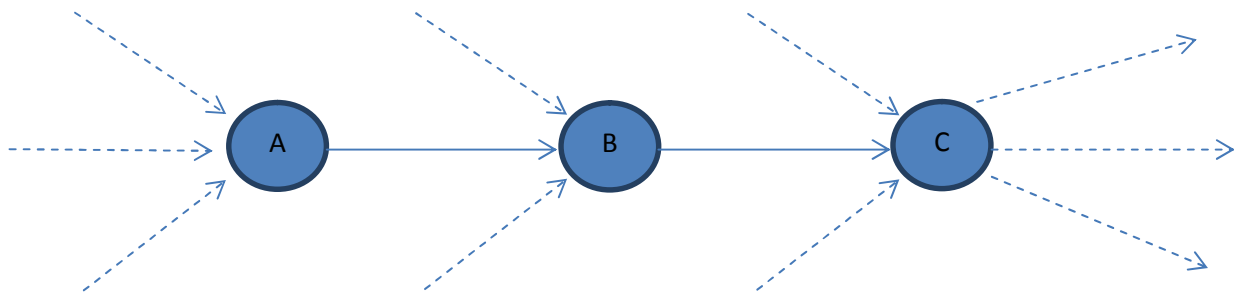
Si A et B ne sont pas d-séparés, ils sont d-connectés.

Etant donnés trois sommets A, B et C, liés deux à deux par des flèches, et quelle que soit la disposition des flèches qui les lient, les trois sommets dépendent les uns des autres. Les trois dispositions possibles sont :

- Relation en série
- Relation divergente
- Relation convergente

a. Relation en série

Soit le cas où A, B et C sont en série. $P(A)$ peut être fixée ou bien dépendre d'autres valeurs.



$P(A)$ dépend d'autres valeurs

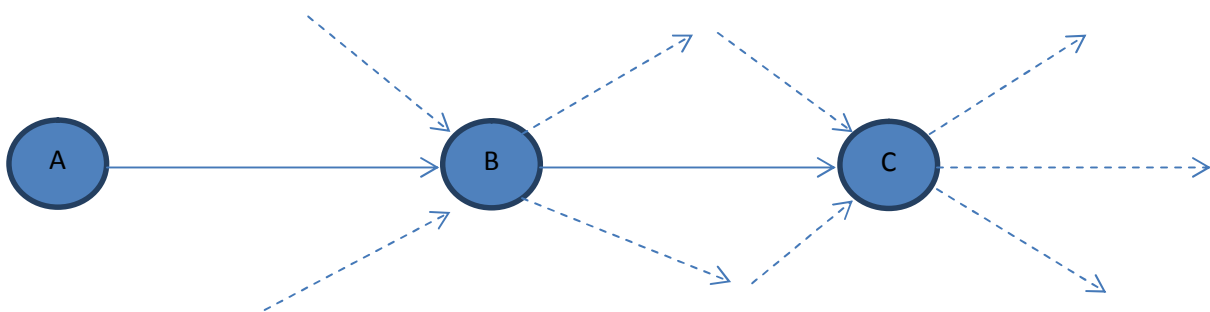


Figure 2.4 : d-séparation cas des variables en série

$P(A)$ est fixée

Dans les deux cas, $P(C)$ dépend de $p(A)$ via $P(B)$

- Si on ne sait rien de $P(B)$, $P(C)$ dépend toujours de $p(A)$
- Par contre, si $P(B)$ est connu (ou instancié) alors $P(C)$ ne dépend plus de $P(A)$

Exemple :

Il pleut $\longrightarrow$ Terre Mouillée $\longrightarrow$ Herbe poussée

- Si on sait que la terre est mouillée, l'herbe poussera, qu'il pleuve ou non.
- Si on ne sait rien sur le fait que la terre soit mouillée ou non, alors sa croissance dépend quand même de la pluie.

b. Relation divergente :

Soit le cas où A, B et C sont en "chapeau" ou une relation divergente.

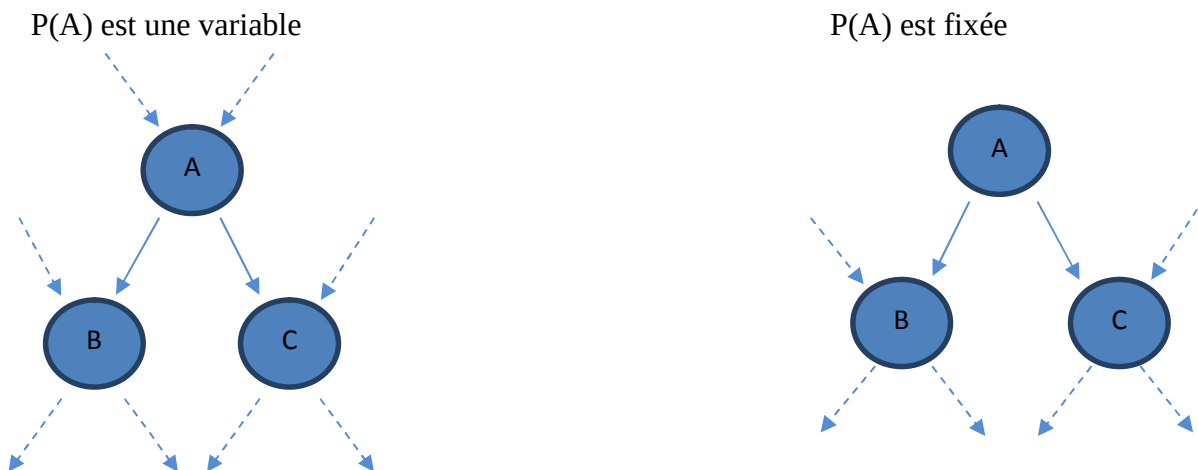
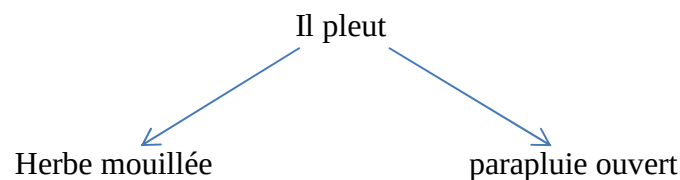


Figure 2.5 : d-séparation cas des variables divergente

$P(B)$ et $P(C)$ dépendent l'une de l'autre via A.

- Si $P(A)$ est variable ou inconnu, alors $P(B)$ et $P(C)$ dépendent l'un de l'autre.
- Si $P(A)$ est fixée alors $P(B)$ et $P(C)$ ne dépendent pas l'un de l'autre

Exemple :



- Si on sait qu'il pleut, alors que l'herbe soit mouillée ou non, les parapluies seront ouverts.
- Si on ne sait rien sur la pluie, alors plus on croira que l'herbe est mouillée, plus on croira que les parapluies sont ouverts

c. Relation convergente :

Soit le cas où A, B et C sont "en V"

$P(B)$ et $P(C)$ sont des variables $P(B)$ et $P(C)$ sont fixés

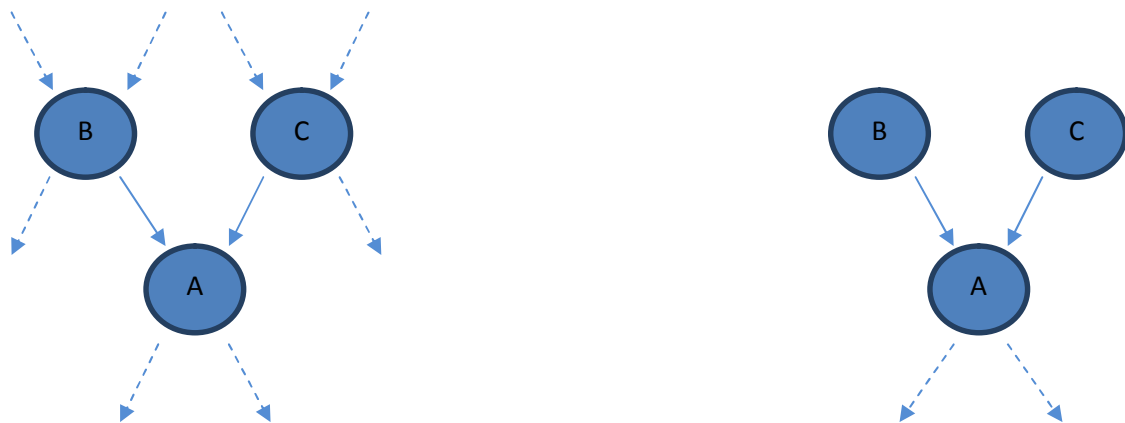


Figure 2.6 : d-séparation cas des variables convergente

$P(B)$ et $P(C)$ dépendent l'un de l'autre via A.

- Si A est connu ou fixé alors $P(B)$ et $P(C)$ dépendent l'une de l'autre
- Si A est inconnu alors $P(B)$ et $P(C)$ ne dépendent pas l'un de l'autre

Exemple :



- Si on sait qu'il y a eu un accident, alors si on croit que la route est dangereuse, la croyance que le conducteur était ivre décroît fortement.
- Inversement, si on sait qu'il y a eu un accident sur une route sans danger, alors on suppose que le conducteur n'était pas dans un état normal.
- Si on ne sait rien la probabilité d'accident, alors les routes dangereuses sont indépendantes des conducteurs ivres.

6. Apprentissage des réseaux bayésiens :

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 paramètre.

a. Apprentissage de 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 existe deux approches générales de construction de la structure d'un réseau bayésien par apprentissage :

- L'une basée sur la recherche et des méthodes de marquage (search and scoring). Elle 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.
- L'autre est basée sur des méthodes d'analyses de dépendances, leurs algorithmes essaient de découvrir les dépendances entre les variables dans le réseau puis emploient ces dépendances pour impliquer la structure.

b. Apprentissage par paramètre :

L'apprentissage de paramètre (ou paramétrisation) d'un réseau bayésien consiste en l'annotation de la structure, préalablement apprise ou connue, des tables de probabilités conditionnelles relatives à chaque nœud du réseau.

Les paramètres peuvent provenir directement de la connaissance "subjective" d'un expert ou calculé à partir de données d'apprentissage. Etant donné une structure et des données d'apprentissage, il s'agit de construire les tables de probabilités conditionnelles pour cette structure.

7. L'inférence dans un réseau bayésien :

L'inférence est la principale utilisation des réseaux bayésiens. Elle concerne le calcul de probabilités conditionnelles d'événements reliés par des relations de causalité. Généralement l'inférence 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.

Grace à la structure graphique et les tables de probabilités conditionnelles, les réseaux bayésiens réduisent le problème d'inférence à un problème de théorie de graphe et de probabilité. C'est alors que la structure du graphe joue un rôle important dans la complexité de ces calculs ainsi que le choix de la méthode d'inférence.

Les algorithmes d'inférence, se répartissent en algorithme d'inférence exacte qui fournissent des résultats exactes et des algorithmes d'inférence approchée dont les résultats

sont des approximations des probabilités réelles. Sachant qu'ils se répartissent de par la nature des résultats qu'ils fournissent.

En effet du point de vue calculatoire, l'inférence exacte dans un réseau bayésien n'est pas toujours possible et cela par rapport au grand nombre de variable. Ainsi des méthodes d'inférence approchées ont été proposées pour les réseaux très complexes.

Partie II : Les modèles de recherche d'information basée sur les réseaux bayésiens :

Dans ce qui suit nous allons voir quelques travaux utilisant une modélisation de type réseaux bayésiens et ayant eu plus ou moins de succès

Introduction :

Les premiers auteurs à avoir utilisé les réseaux bayésiens en RI sont : [Pearl, 1988], [Buntine, 1994] et [Jensen, 2000]. Afin de calculer la correspondance entre la requête et les documents, les auteurs fournissent un formalisme pour fusionner des informations provenant de différentes sources (requêtes passées, réinjection de pertinence...).

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 probabilité.

Les modèles les plus connus en RI utilisant les réseaux bayésiens sont :

- les réseaux d'inférence [Turtle et Croft, 1990]
- les réseaux de croyance [Ribeiro-Neto et al., 1996]

1. Modèle de réseau bayésien d'inférence de base :

L'architecture proposée par [Turtle, 1991] est une architecture simple comme représenté ci-dessous :

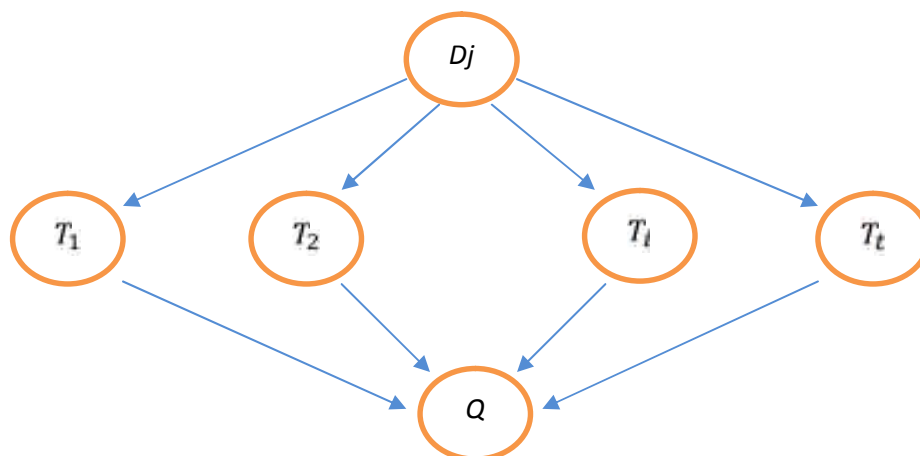


Figure 2.7 : Architecture simplifiée du réseau d'inférence

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 en information de l'utilisateur.
- Les arcs du graphe représentent des relations sémantiques entre les nœuds. A ces nœuds sont associées des probabilités de croyance.

Dans l'architecture on a :

- Le nœud document D_j : correspond à l'évènement qu'un document donné de la collection est observé.
- Le nœud T_i ($i=1, \dots, t$) : correspond au terme T_i indexant le document et correspond à l'évènement 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 :

$$P(Q, d_j) = \sum_{\theta^k \in \theta} P(Q | \theta^k) \cdot \prod_{T_i \in Q \wedge d_j} P(\theta_i^k | d_j) \cdot P(d_j)$$

Où :

θ : est l'ensemble des configurations possibles des parents de Q .

θ_i^j : est une instance d'un nœud particulier T_i 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.

Remarque :

- 1- A chaque instanciation d'un document D_j , sa probabilité à priori $P(D_j) = d_j = \frac{1}{n}$ ce calcul sera donc supprimé de la propagation globale car 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 $P(Q|\theta)$ d'un nœud ayant n parent, $\theta_1, \dots, \theta_n$, et $P(\theta_1 = t_1) = p_1, \dots, P(\theta_n = t_n) = p_n$ les agrégations suivantes sont définies :

$$P_{Ou}(Q|\theta) = 1 - (1 - p_1) \dots (1 - p_n)$$

$$P_{Et}(Q|\theta) = p_1 \times \dots \times p_n$$

$$P_{Non}(Q|\theta_1) = 1 - p_1$$

$$P_{Somme}(Q|\theta) = \frac{p_1 + \dots + p_n}{n}$$

$$P_{SommePondérée}(Q|\theta) = \frac{w_1 p_1 + \dots + w_n p_n}{w_1 + \dots + w_n}$$

La somme pondérée 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.

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

$$P(t_i | d_j) = 0.5 + 0.5 \cdot \frac{tf_{ij}}{ndfi}$$

Avec :

$$tf_{ij} = \frac{tf_{ij}}{\max_{t_k \in d_j} tf_{kj}}, \quad ndfi = \frac{\log \frac{N}{n_i}}{\log N}$$

tf_{ij} : est la fréquence du terme t_i dans le document d_j .

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

N : est le nombre de documents de la collection

n_i : est le nombre de documents contenant le terme t_i .

On estime que : $P(t_i | \overline{PARENTST_i}) = 0$

Où $\overline{PARENTST_i}$ signifie que tous les parents de T_i sont instanciés à faux.

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

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 ont été aussi proposés pour les systèmes hypertextes³ [Savoy et al., 1991]

2. Le modèle de Ribeiro-Neto [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 2.8 suivante :

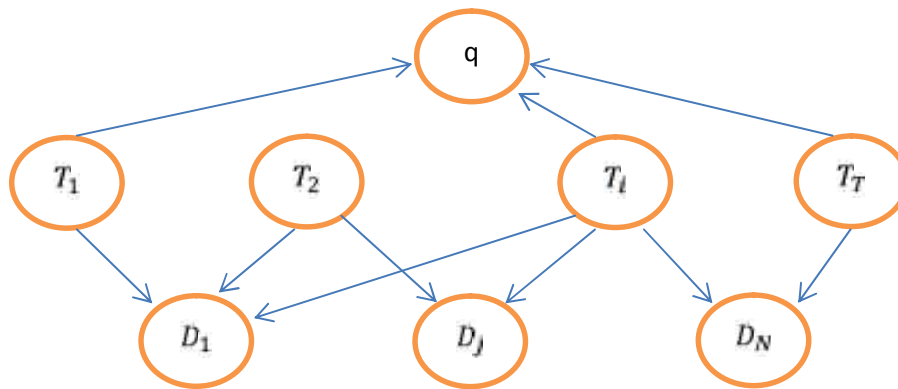


Figure 2.8 : Architecture générale

L'univers de discours est donné par l'ensemble des termes d'indexation utilisés dans le système, noté U , et $U = T_1, \dots, T_T$ où T est le nombre de termes manipulés dans le système (pour représenter les documents ou la requête).

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 comme suit :

$$P(D_j, Q) = \sum_{\theta} P(D_j, \theta) \times P(Q|\theta) \times P(\theta)$$

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.

Afin de calculer la probabilité d'un document étant donnée une configuration d'un concept : $P(D_j = 1 | \theta) = P(d_j | \theta)$

³ Système qui exploite les liens hyper text (internes ou externes) figurant dans un documents dans la recherche d'information.

Le modèle propose d'utiliser la fonction cosinus du modèle vectoriel, comme suit :

$$P(d_j | \theta^Q) = \frac{\sum_{\theta_l^Q=1}^T w_{lj} \times w_{lq}}{\sum_{\theta_l^Q=1}^T w_{lj}^2 \times \sum_{\theta_l^Q=1}^T w_{lq}^2}$$

Où

θ_l^Q : est la configuration des termes telle que donnée dans la requête Q.

w_{lj}, w_{lq} : les poids du terme t_l dans le document d_j et la requête Q respectivement.

Les poids w_{lj} , sont des variantes de la pondération par tf*idf.

Ainsi :

$$P(\bar{d}_j | \theta^Q) = 1 - P(d_j | \theta^Q)$$

La probabilité d'une requête étant donnée une configuration est calculée comme suit :

$$P(Q | \theta) = \begin{cases} 1 & \text{si } \exists T_i, \theta_i^Q = \theta_i \\ 0 & \text{sinon} \end{cases}$$

Où

θ_i^Q, θ_i : est l'instanciation du terme T_i dans la requête et dans une configuration θ respectivement.

Ainsi que :

$$P(\bar{Q} | \theta) = 1 - P(Q | \theta)$$

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

Le modèle propose ainsi de calculer la probabilité $P(d_j)$ qui donne le degré au quelle le document D_j couvre complètement l'espace des termes U. cette probabilité est calculée comme suit :

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

Ainsi que la probabilité $P(Q)$ qui donne le degré auquel la requête couvre complètement l'espace des termes U. cette probabilité répondait à la croyance associée à la proposition est-il vrai que Q couvre complètement U ?

$$P(Q) = \sum_{\theta} P(Q | \theta) P(\theta)$$

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

Le calcul de cette équation nécessiterait donc 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.

3. Autres modèle de recherche basé sur les réseaux bayésiens :

Afin de résoudre des problèmes d'optimisation de calculs nécessités par les topologies des réseaux ([Indrawan et al, 1996] et [De Campos et al, 2003]) et aussi pour les appliquer à des collections de documents de types hétérogènes et contenant des liens hypertextes ([Crestani et al, 2003], [Silva et al, 2000] et [Denoyer et al, 2004]). Des extensions des deux modèles basés sur les réseaux bayésiens ont été proposées.

Dans ce qui va suivre nous présentons les modèles les plus connus notamment celui de Indrawan et celui de De Campos

3.1.Le modèle d'Indrawan [Indrawan et al, 1996] :

L'architecture globale de ce modèle est représentée comme suit :

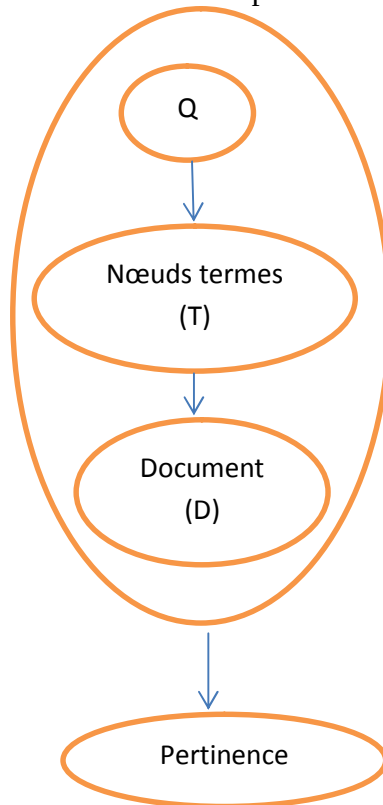


Figure 2.9: Architecture globale

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

Dans l'architecture générale (simplifiée) présenté dans la figure 2.9, 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 .
- 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
- Une couche inférieure contenant les concepts, ou toute entité générée par la combinaison des mots clés.

Les nœuds termes impliquent 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 clé 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 document.

La probabilité de pertinence d'un document étant donné la requête, selon la topologie du graphe (figure 2.9) est mesurée par :

$$P(\text{Pert } D_j, Q) = P(\text{Pert } D_j, T_i, Q)$$

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 ».

D_j : est un document de la collection.

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 $P(D_j, Q)$.

3.2. Le réseau multi connectés pour la RI :

Le modèle de réseau multi connecté 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 terme et leurs relations de dépendance.

L'architecture du réseau est présentée dans la figure 2.10 ci-dessus :

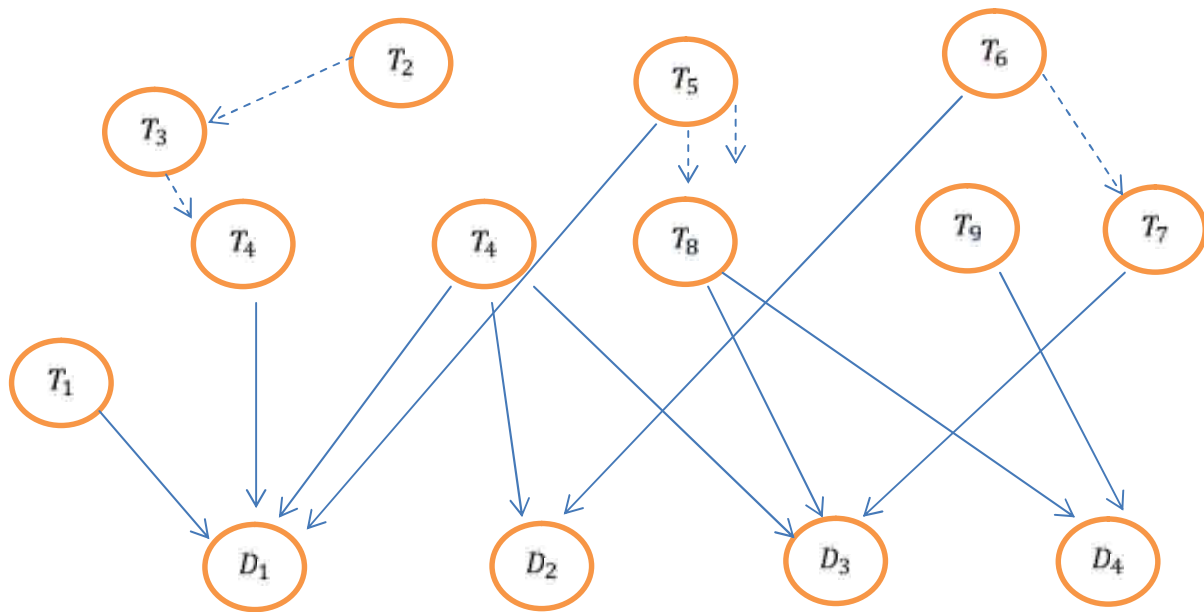


Figure 2.10 : Architecture globale du réseau multi connecté

Ce réseau comprend deux ensembles de nœuds, comme illustré dans la figure 2.10 : l'ensemble des nœuds termes, T ; une variable T_i associé à un terme prend ses valeurs dans le domaine $\{t_i, \bar{t}_i\}$ où t_i désigne le terme T_i comme étant pertinent (donc figure dans sa requête) et $\bar{t}_i$ comme n'étant pas pertinent.

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

Dans la topologie proposée :

- Le degré de pertinence d'un document D_j peut être complètement déterminé par la connaissance de la pertinence de tous les termes d'indexation du document D_j .
- Lorsque cette information est absente, la connaissance de la pertinence pour la même requête d'un autre document D_k peut avoir une influence sur celle de D_j (c-à-d tout document D_j est conditionnellement indépendant de tout document D_k lorsque les valeurs de pertinence de tous les termes indexant le document D_j sont connues).
- Les termes présent dans la requête propagent l'information à travers le réseau pour calculer la pertinence d'un document étant donnée la requête $P \ d_j \ Q$:
- Les probabilités à priori des nœuds termes racine sont données par :

$P \ t_i = \frac{1}{M}$ avec M le nombre de termes de la collection.

$$P \ \bar{t}_i = 1 - \frac{1}{M}$$

- Pour calculer les probabilités conditionnelles de chaque terme étant donné ses parents :

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

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 T_i figure dans $pa(t_i)$ 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.

- **L'estimation de la probabilité des documents :**

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

$$P(d_j | \theta_{D_j}) = 1 - P(d_j | \bar{\theta}_{D_j})$$

Avec :

θ_{D_j} : l'ensemble des configurations possibles des parents de D_j ; certaines sont pertinentes, notées $R(\theta_{D_j})$, 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 T_i dans le document D_j . Ce poids compris entre 0 et 1 est obtenu par des variations normalisées de $tf \cdot idf$.

Remarque : la somme des poids des termes présents dans le document et absents de la requête sont ainsi 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 : consiste à utiliser la requête comme évidence en instanciant ces termes comme valeurs pertinents, calculer $P(t_i | Q) \propto T_i$, puis propager cette information jusqu'au nœud document.

3.3. Le modèle de Acid [Acid et al, 2003] :

Ce modèle est un modèle de réseau bayésien basique. Architecture de ce réseau est présenté ci-après :

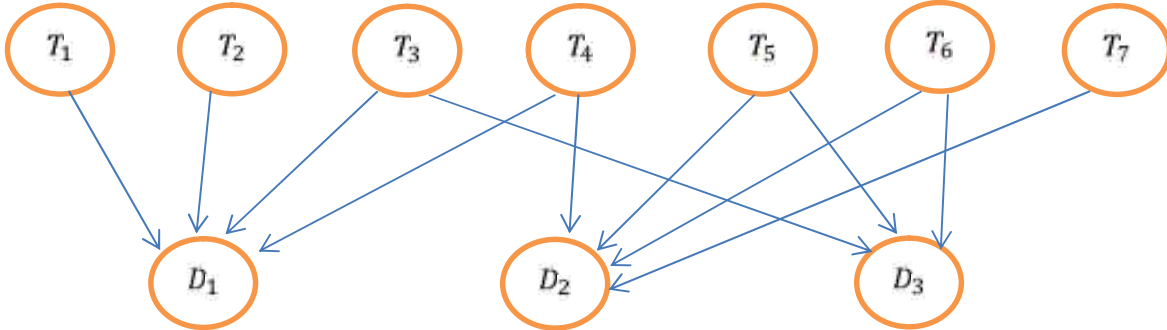


Figure 2.11 : Modèle de base

La figure 2.11 représente les termes et les documents de la collection. Chaque terme est relié aux documents dans lesquels il apparaît.

L'ensemble des variables V_B est constitué de deux ensembles $V_B = T \sqcup D$: l'ensemble $T = T_1, \dots, T_M$ des M termes de la collection et l'ensemble $D = D_1, \dots, D_N$ des N documents de la collection. Le domaine de chaque variable terme $T_i = t_i$ signifie que le terme T_i est pertinent et $T_i = \bar{t}_i$ signifie que le terme T_i n'est pas pertinent. Le domaine de chaque variable D_j est $d_j, \bar{d}_j$ où $D_j = d_j$ signifie que le document D_j est pertinent (non pertinent) étant donné la requête.

Remarquons que $Pa_{D_j} = T_i \sqcup T | T_i \sqcup D_j$. De plus les auteurs estiment que :

- $P t_i = \frac{1}{M}$ et $P \bar{t}_i = \frac{M-1}{M}$ où M est le nombre de termes de la collection.
- $P d_j | Pa_{D_j} = \sum_{T_i \sqcup D_j, t_i \sqcup Pa_{D_j}} w_{ij}$

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 :

$$P d_j | Q = \sum_{t_i \sqcup \bar{d}_j} w_{ij} \cdot P t_i | Q$$

Par ailleurs, les termes sont marginalement indépendants, alors :

$$P t_i | Q = 1 \text{ si } T_i \sqcup Q \text{ et } P t_i | Q = \frac{1}{M} \text{ si } T_i \sqcup Q$$

On aura donc :

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

Pour tenir compte de la fréquence des termes dans la requête, on a dupliqué qfi fois dans le réseau pour chaque terme T_i apparaissant dans la requête et l'équation deviendra :

$$P(d_j | Q) = \prod_{T_i \in \overline{D_j} \cap Q} w_{ij}^{qf_i} + \frac{1}{M} \prod_{T_i \in \overline{D_j} / Q} w_{ij}$$

Les auteurs ont présenté une autre extension à ce modèle, illustrée par la figure 2.12, 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.

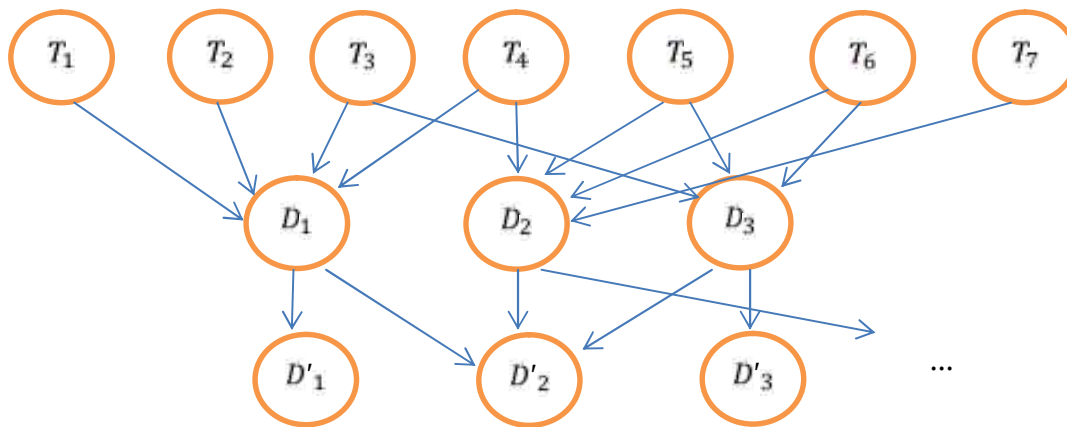


Figure 2.12 : autre réseau bayésien étendu

3.4.Modèle de Brini [Brini et al , 2005] basé sur les réseau possibilistes :

L'architecture générale de ce modèle est illustrée dans la figure 2.13 ci-dessous :

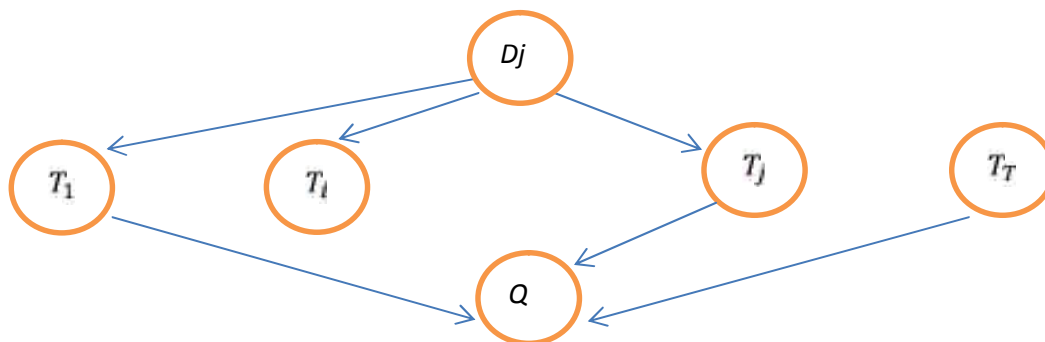


Figure 2.13 : Architecture générale.

Un document D_j est instancié ou pas, prenant ses valeurs dans le domaine $d_j, \bar{d}_j$. L'instanciation d'un nœud document, $D_j = d_j$ (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 T_i , est $t_i, \bar{t}_i$. $T_i = t_i$ signifie que le terme t_i 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 $T(D_j)$ (resp. $T(Q)$) l'ensemble des termes d'indexation du document D_j (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 ($\text{par}(Q)$) qui représente la requête dans sa forme la plus stricte (conjonctive). Soit θ^Q cette 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 telle que :

- Il est possible à un certain degré que le document soit pertinent étant donnée la requête, notée par $\Pi D|Q$;
- Il est certain (dans le sens possibiliste) que le document soit pertinent étant donnée la requête, notée par $N D|Q$.

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, $\Pi d_j Q$, et de nécessité, $N d_j Q$ comme suit :

$$d_j Q = \frac{\Pi Q \Delta d_j}{\Pi Q}$$

$$N d_j Q = 1 - \bar{d}_j Q = 1 - \frac{\Pi Q \Delta \bar{d}_j}{\Pi Q}$$

La possibilité de Q est donnée par :

$$Q = \max \quad Q \Delta d_j, \quad Q \Delta \bar{d}_j$$

$$d_j Q = \min \quad 1, \frac{\Pi Q \Delta d_j}{\Pi Q \Delta \bar{d}_j}$$

$$\bar{d}_j Q = \min \quad 1, \frac{\Pi Q \Delta \bar{d}_j}{\Pi Q \Delta d_j}$$

$$Q \Delta D_j = \max_{\theta^i \in \theta} Q|\theta^i \cdot \prod_{T_i \in T \setminus \bar{Q} \Delta T \setminus D_j} \theta_i^i D_j \cdot \prod_{T_k \in T(\bar{Q}) \setminus T(D_j)} \theta_k^i$$

Avec :

θ : Les configurations possibles de l'ensemble des parents de Q.

θ^i : est une configuration de θ

θ_i^i : L'instanciation de T_i dans la configuration

Ainsi que :

$$Q|\theta^i = \begin{cases} 0 & \text{si } \nexists T_i \in PAR_Q \text{ tel que } \theta_i^i = \theta_i^Q \\ \frac{1 - \prod_{t: \theta_i^t = \theta_i^Q} 1 - \pi Q|t_i}{1 - \prod_{t: \theta_i^t = PAR_Q} 1 - \pi Q|t_k} & \text{sinon} \end{cases}$$

$$Q|t_i \Delta k \neq i t_k = \frac{idf_i}{\log N} = nidf_i$$

Avec :

$$idf_i = \log \frac{N}{ni}$$

ni : est le nombre de document contenant le terme t_i .

N : le nombre de documents de la collection.

a. Possibilité a priori des documents :

$$d_j = \frac{l_j}{\max_{k=1 \dots N} l_k} = nld_j$$

Avec l_j la longueur du document d_j en terme de fréquence

$$l_j = \sum_i t f_{ij}.$$

Plus le document est court, moins il est pertinent. Dans tous les cas, $\prod d_j = 1$, si on ne veut pas favoriser le document de manière exagérée.

b. Pondération des termes indexant les documents :

Cette mesure est calculée comme suit :

$$t_i/d_j = nt f_{ij} = \frac{t f_{ij}}{\max_{t k \in d_j} (t f_{kj})}$$

Avec :

$t f_{ij}$: La fréquence du terme t_i dans le document d_j .

$\max_{t k \in d_j} (t f_{kj})$: est la fréquence maximal dans le document d_j .

4. Synthèse :

Après avoir présenté l'état de l'art des modèles bayésien dans la RI, nous avons effectué une comparaison voici en quelques lignes ce que nous avons donc conclue :

- Comparaison entre le modèle inférentiel et le modèle de croyance :

	Modèle inférentiel	Modèle de croyance
Instanciation	Instancie le document à la réception d'une requête	Instancie la requête
Le sens de la dépendance des termes d'indexation avec les documents	$P(t_i, d_j)$	$P(d_j, t_i)$

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], est d'une part largement utilisé malgré il ne présente pas de capacités d'apprentissage, et il 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
- Le modèle d'Indrawan [Indrawan et al, 1996] et de Turtle [Turtle, 1991] :

	Modèle d'Indrawan	Modèle de Turtle
Les termes impliquent le document	oui	non
instanciation	La requête instancie le système	Les documents sont instanciés

L'apport principal d'Indrawan concerne l'optimisation proposée pour la propagation dans les réseaux. Ainsi, le nombre de probabilités conditionnelles à calculer a 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)

Dans ce qui suit nous allons présenter les différents travaux ayant utilisé les réseaux bayésiens dans la recherche d'information structurée.

Partie III : Les modèles de RIS basés sur les réseaux bayésiens :

La section suivante présente les différents travaux ayant utilisé les réseaux bayésiens dans la recherche d'information structurée.

1. Les travaux de Myaeng [Myaeng et al, 1998] :

L'arborescence utilisée est :

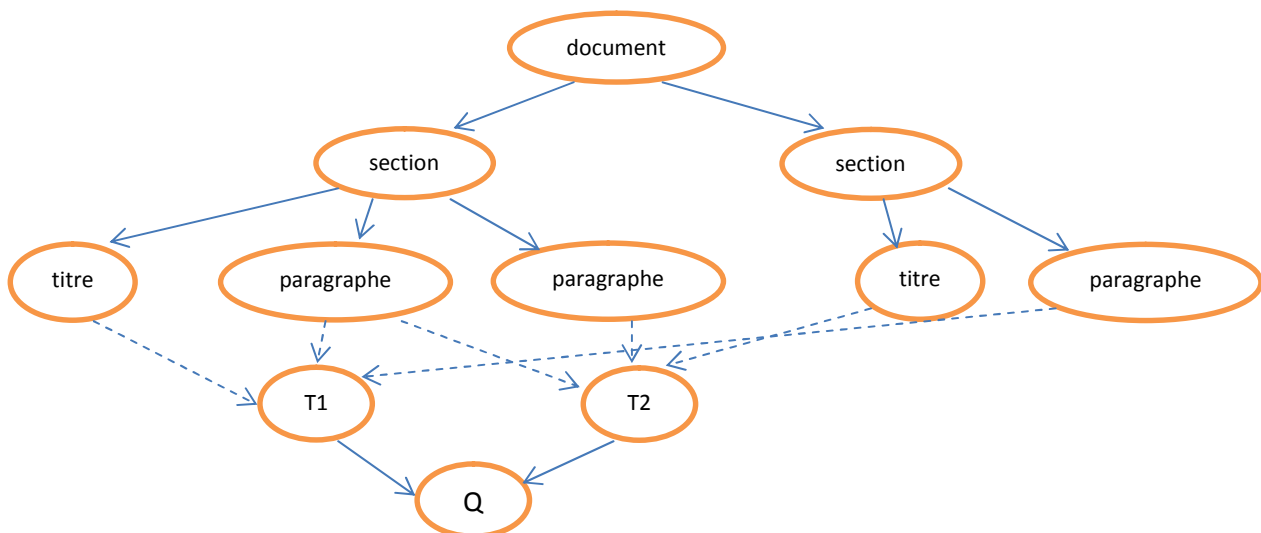


Figure 2.14 : l'approche de Myaeng

Les auteurs ont proposé une extension du modèle Inquiry [Turtle et Croft, 1991] tel que chaque document 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 2.14).

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).

Présentation de l'arborescence proposée :

- L'évaluation démarre du nœud document jusqu'aux termes.

- A la réception d'une requête, on calcul pour tout document d la façon dont les termes de la probabilité de chaque terme 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 calcul 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 connaître les distributions $P(t|e_1 \dots e_n)$ où $e_1 \dots e_n$ 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, 1998] 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 :

$$P(t|e) \propto \frac{1}{ef_t \cdot \epsilon} \times ef_t \cdot e$$

Et

$$P(e|pa(e)) \propto \lambda_{e,pa(e)} \times l_e \times \cos(e, pa(e))$$

Où :

$\lambda_{e,pa(e)}$: est un paramètre liée à l'importance du sous-élément structurel (fixée à 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)

Ce modèle permet 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).

2. les travaux de Piwowarski [Piwowarski et al, 2002, 2003, 2005] :

Topologie du modèle :

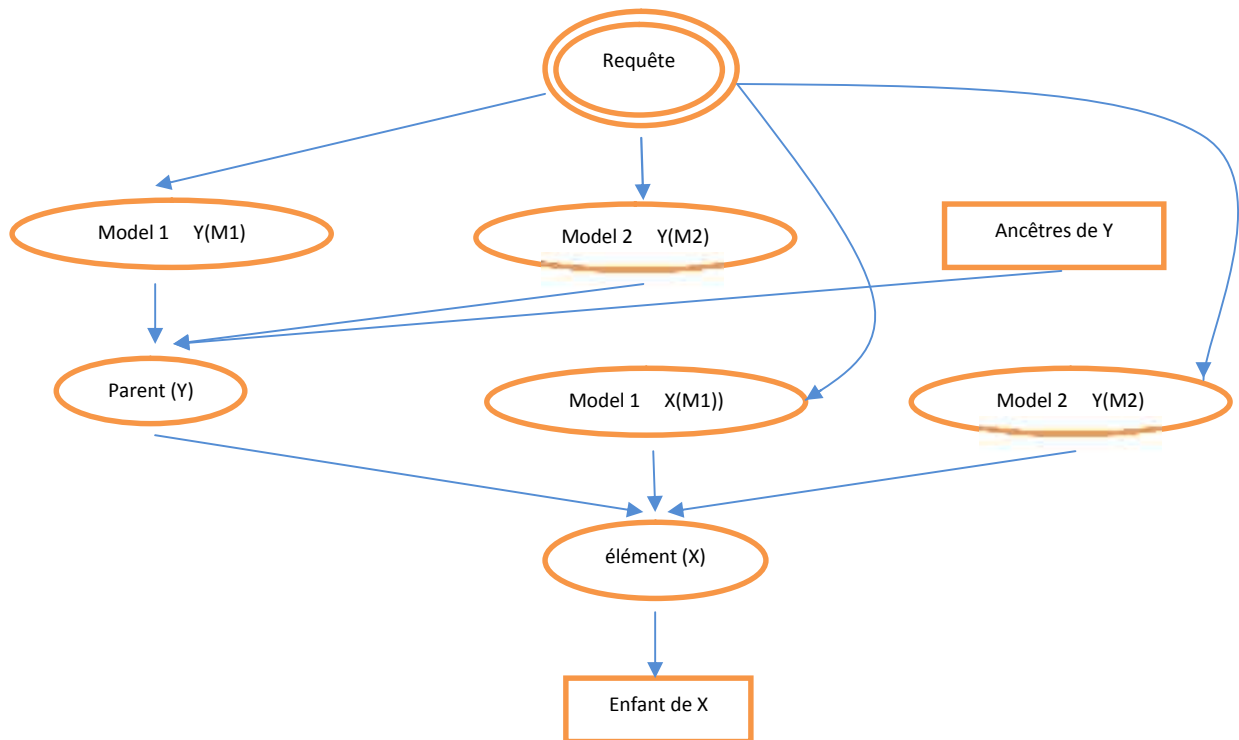


Figure 2.15 : une vue locale du RB : utilisation de deux modèles classiques M1 et M2

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 ($\bar{R}$)

L'évaluation de la pertinence d'un élément 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 base M_1 et M_2 pour le calcul du score local de l'élément) :

$$P(X = v_x | q) = \sum_{\substack{v_y \in V \\ m_1 \dots m_n \in \{R, -R\}}} P(X = v_x | Y = v_y, M_1(X) = m_1 \dots M_n(X) = m_n) \\ \times P(Y = v_y | q) \times P(M_1(X) = m_1) \times \dots \times P(M_n(X) = m_n) \quad (1)$$

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 M_1 et M_2), et de la catégorie $c(e)^4$ 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.

3. Les travaux de Crestani [Crestani et al, 2003] et [Crestani et al, 2004] :

Topologie du modèle : en tenant compte du modèle proposé en RI, les auteurs ont déduit le modèle présenté dans la figure 2.16. Où seules les unités de niveau bas $U_{1l} \dots U_{|L_l|l}$ seront reliées aux termes qui les indexent.

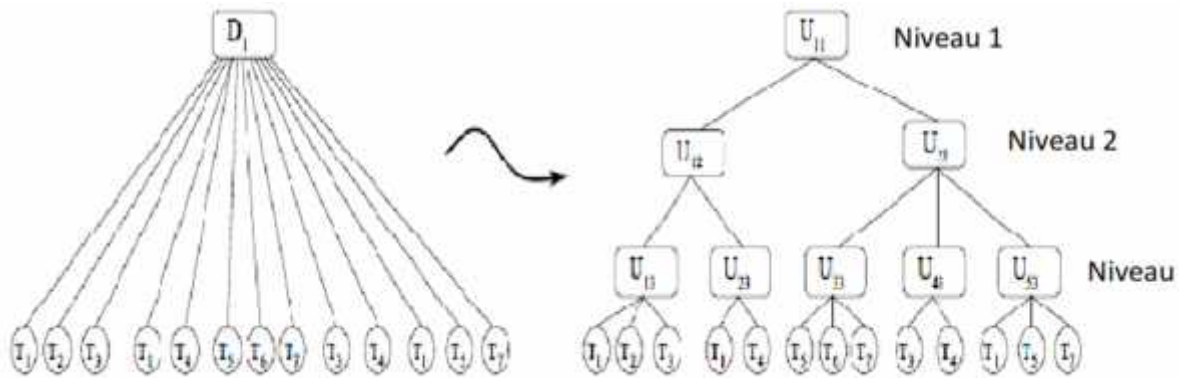


Figure 2.16 : d'un document indexé (RI) à un document structuré indexé (RIS)

Les auteurs ont supposés que :

- chaque document est composé d'une structure hiérarchique de niveaux abstraits $L_1, L_2, \dots, L_L$
- chacun représentant une association structurale d'éléments dans le texte. Le niveau dans lequel le document lui-même est inclus est le niveau L_1 et le niveau le plus spécifique est le niveau L_L .

⁴ 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

- Chaque niveau contient des unités structurales notée U_{ij} où i est l'identifiant de l'unité au niveau j . le nombre d'unités dans un niveau j est noté $|L_j|$ et $L_j = \{U_{1j}, U_{2j}, \dots, U_{|L_j|j}\}$ les unités sont organisés suivant la structure du document.
- Chaque unité U_{ij} (excepté pour $j=1$) est incluse dans une seule unité $U_{z(i,j)^5, j-1}$ au niveau $j-1$.

Information sur le graphe :

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 $\{tk, \bar{tk}\}$.

Les termes sont marginalement indépendants entre eux et les unités sont conditionnellement indépendantes. Ce modèle respecte la topologie bayésienne avec $l+1$ niveaux et dont les arcs partent du niveau des termes vers celui des unités au niveau 1 et d'autres arcs partent des unités de niveau j vers des unités de niveau $j-1$ pour $j = 2 \dots l$. Comme suit :

- $\square T_k \square T, Pa(T_k) = \square$
- $\square U_{il} \square L_l, Pa(U_{il}) = \{T_k \in T | U_{il} \text{ est indexé par } T_k\}$
- $\square j = 1, \dots, l-1, \square U_{ij} \square L_j, Pa(U_{ij}) = \{U_{k,j+1} \square L_{j+1} | U_{k,j+1} \square U_{ij}\}$

Un exemple de réseau bayésien multi niveaux à 3 niveaux est illustré dans la figure 2.17 :

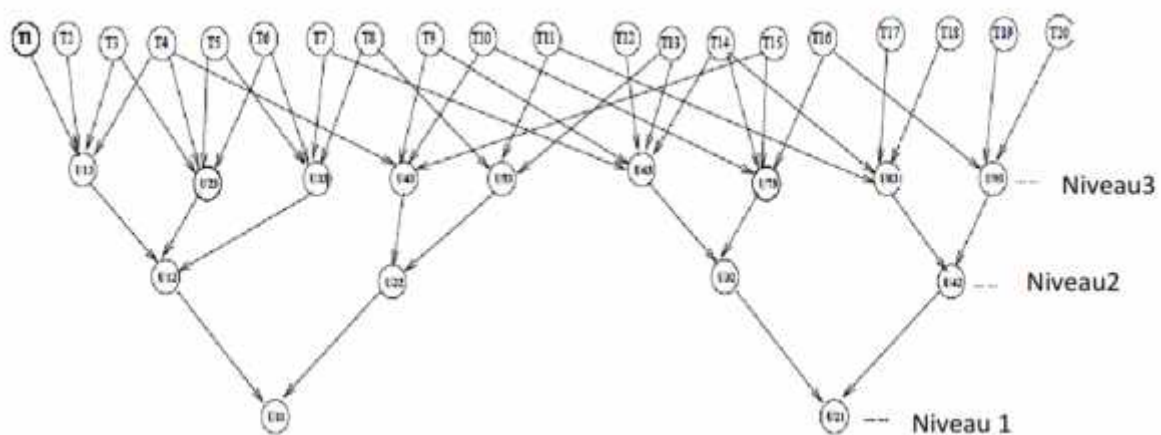


Figure 2.17 : réseaux bayésien multi niveau

⁵ $Z(i,j)$ est la fonction qui retourne l'index de l'unité de niveau j contenant l'unité d'index i et de niveau j .

Les probabilités conditionnelles :

- Les nœuds termes T_k ont les mêmes probabilités marginales

$$P(t_k) = \frac{1}{M}$$

Où : M est le nombre des termes de la collection.

- Les unités U_{il} de niveau l :

$$P(U_{il} | pa U_{il}) = \frac{w_{kl}}{\sum_{T_k \in R} w_{kl}}$$

Où :

$R \cap pa U_{il} = T_k \cap pa U_{il} | t_k \cap pa U_{il}$ i.e les termes de $pa U_{il}$ instanciés comme pertinents dans la configuration de $pa U_{il}$.

w_{kl} : est le poids associé à chaque terme indexant l'unité U_{il} .

- Les unités structurelles de niveau $j \neq l$

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 $P(U_{ij} | pa U_{ij})$ est définie ainsi :

$$P(U_{ij} | pa U_{ij}) = \frac{P_{hl}^j}{\sum_{U_{hj+1} \in R} P_{hl}^j}$$

Où le poids, P_{hl}^j de l'unité $U_{hj} + 1$ dans l'unité U_{ij} est définie par :

$$P_{hl}^j = \frac{\sum_{T_k \in A \cap U_{hj+1}} w_{kh}^{j+1}}{\sum_{T_k \in A \cap U_{ij}} w_{kh}^j}$$

Où :

$A \cap U_{ij}$ et $A \cap U_{hj} + 1$: sont les ensembles de termes utilisés pour indexer U_{ij} et $pa U_{ij}$ respectivement⁶.

$$w_{kh}^j = tf_{ki}^j \cdot idf_k$$

⁶ En réalité l'unité de niveau $j \neq 1$ n'est indexé par aucun terme, on se réfère aux termes indexant l'unité de niveau 1 incluant soit l'unité U_{ij} ou les unités de $R \cap U_{ij}$.

Avec :

tf_{kl}^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.

$$w_{kh}^{j+1} = tf_{kh}^{j+1} \cdot idf_k$$

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 priori de pertinence de toutes les unités structurales étant donné la requête $P u_{ij} Q$. On distingue les deux probabilités suivantes :

- Pour les unités de niveau L_l :

$$P u_{il}|Q = \frac{w_{kl}}{\tau_k \square pa \bar{u}_{il} \cap Q} + \frac{1}{M} \frac{w_{kl}}{\tau_k \square pa u_{il}|Q}$$

- Pour les unités de niveau $L_j, j \neq l$:

$$P u_{ij} Q = \frac{P_{hl}^j \cdot P u_{hj+1} Q}{u_{hj+1} \square pa u_{ij}}$$

Et ainsi on peut calculer les probabilités niveau après niveau en commençant par le niveau l et aller jusqu'au niveau 1.

4. Les travaux de De Campos [L.M. de camps 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 utilise l'approche classique des fichiers inverses contenant les occurrences des termes des balises xml ou toute autre donnée structurale pour représenter l'organisation interne des documents.

Topologie du modèle :

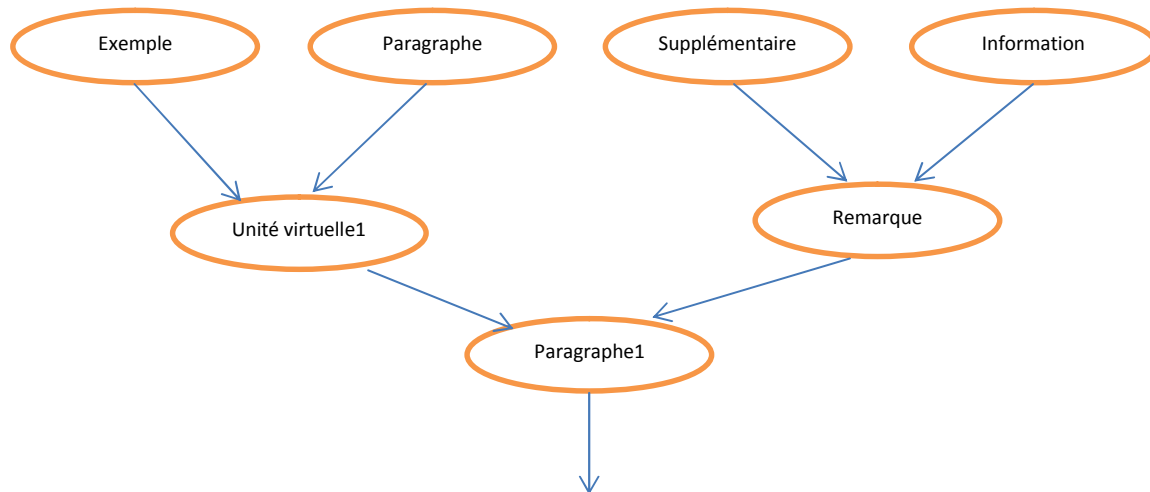


Figure 2.18 : un fragment de réseau bayésien contenant une unité virtuelle.

- 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'élément mixte suivant : <Paragraphe> un exemple de paragraphe avec <Remarque> une information supplémentaire<\Remarque><\paragraphe>

Pour résoudre ce cas, le modèle a inclus des nœuds spéciaux dits nœuds virtuels qui seront des parents de l'unité concernée par ce cas (« paragraphe » dans l'exemple) avec les unités inclus comme l'illustre la figure 2.18. Ces nœuds virtuels seront des nœuds terminaux contenant du texte.

Remarque :

- Les unités virtuelles ne devraient pas être retournées à l'utilisateur
- L'expérimentation a été faite sur la collection « Shakespeare »⁷ contenant les pièces de Shakespeare et la collection inex. L'expérimentation sur le temps 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 expérimental du système plusieurs formules de pondération peuvent être appliquées. Ces différentes valeurs de pondération pré calculées sont stockés dans des fichiers dits fichiers de pondération. un moyen rapide pour en insérer une dans l'index lui disque.

5. L'approche proposée par Alimazighi [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

⁷ <http://qmir.dcs.qmw.ac.uk/Focus/collbuilding.htm>

éléments structurels. Une autre particularité dans ce modèle est l'association à chaque nœud du réseau une matrice contenant les termes et leurs poids.

Calcule des paramètres du réseau :

Le but de cette étape est de calculer la probabilité conditionnelle de chaque nœud connaissant ses parents $P(v_n | pa(v_n))$

Avec : v_n : variable aléatoire nœud n.

$pa(v_n)$: variable aléatoire du parent du nœud n.

En utilisant la loi de probabilité de BAYES nous avons :

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

Et si $A \subseteq B$ 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 nœud est inclus dans son parent alors la formule précédente s'écrit comme suit :

$$P(v_n | pa(v_n)) = \frac{P(v_n)}{P(pa(v_n))}$$

Donc :

$$P(v_n | pa(v_n)) = \frac{\text{card}(v_n)}{\text{card}(pa(v_n))}$$

Avec :

$\text{card}(v_n)$: le nombre de terme t_i dans le nœud n.

$\text{card}(pa(v_n))$: 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.

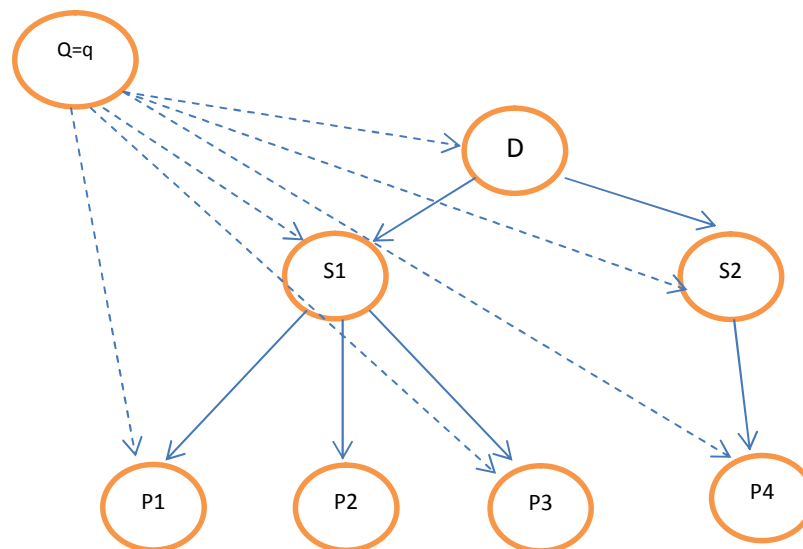


Figure 2.19 : exemple de recherche avec structure

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 répondant mieux à la requête (les nœuds ayant la probabilité conditionnelle maximale).
 - sinon aucun nœud ne répond à la requête.

6. Les travaux de Bessai [Bessai et al, 2010] :

Topologie du modèle :

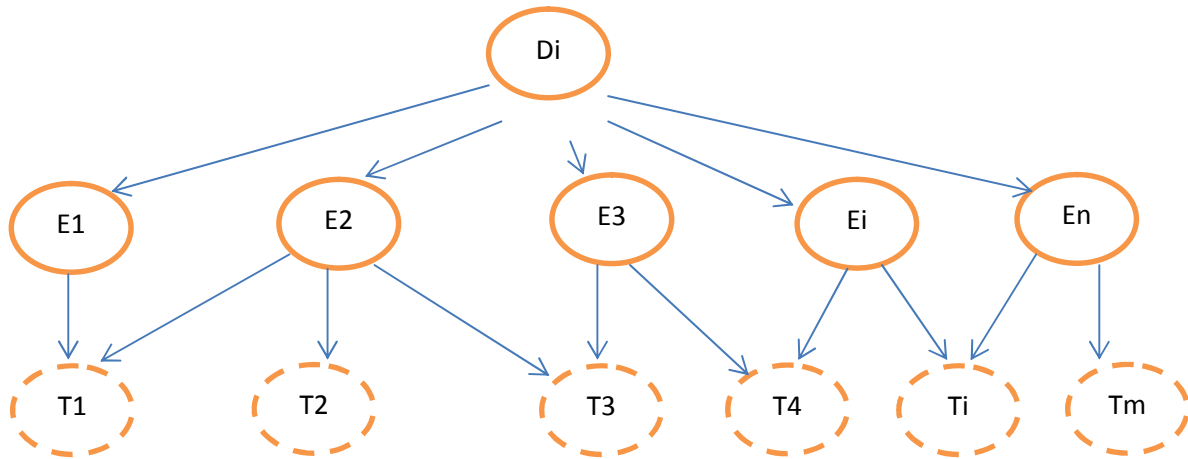


Figure 2.20 : architecture du modèle

Le modèle proposé est représenté par un réseau possibiliste d'architecture illustrée par la figure 2.20 sachant qu'à chaque nœud est associée une variable aléatoire binaire.

Les nœuds documents représentent les documents de la collection. L'instanciation $Di=di$ signifie que 'le document Di est pertinent pour une requête donnée', et $\neg di$ signifie que le document Di n'est pas pertinent pour la requête donnée.

Les nœuds $E1, E2, \dots, En$, représentent les balises du document Di . L'instanciation $Ei = ei$ signifie que l'élément Ei est pertinent pour la requête; $Ei = \neg ei$ signifie que l'élément ' Ei ' est non pertinent pour la requête. Les nœuds $T1, T2, \dots, Tm$ sont les nœuds termes où L'instanciation $Ti = ti$ signifie que le terme ' Ti ' est représentatif du nœud père auquel il est rattaché,

$Ti = \neg ti$ 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

- 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 :

$\Pi di|Q$: est-il possible à un certain degré ou non que le document Di soit pertinent pour 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 proposés les hypothèses suivantes :

- Un document a autant de possibilité d'être pertinent que non pertinent pour un utilisateur donné, soit $\Pi di = \Pi \neg di = 1$, quel que soit i .

- La requête est composée d'une simple liste de mots clés (interprétée de manière conjonctive) $Q = t_1 \wedge t_2 \wedge \dots \wedge t_l$. L'importance relative des termes entre eux dans la requête est ignorée.

$$P(di|Q) = \frac{P(Q|di)}{P(Q)} \text{ et } P(di|Q) = 1 - P(\neg di|Q)$$

En considérant la quantité $P(Q)$ indépendante des documents et étant donné l'architecture du modèle :

$$P(di|Q) = \max_{\theta^e \in \Theta^E} P(Q|\theta^e) \cdot P(\theta^e|di) \cdot P(di) \quad [1]$$

Où θ^e représente une instantiation possible des variables balises, c'est-à-dire les parents des termes de la requête parmi toutes les configurations possibles définie par Θ^E . De plus les auteurs supposent que les variables termes et les variables balises sont indépendantes.

L'équation [1] devient alors :

$$P(di|Q) = \max_{\theta^e \in \Theta^E} \text{Prod}_{E_j \in T_E \wedge T_Q} \text{Prod}_{T_i \in T_E \wedge T_Q} t_i \theta_j^e \\ \cdot \text{Prod}_{E_j \in \theta^e} \theta_j^e \cdot P(di) \quad [2]$$

Où :

Prod : signifie produit

P : désigne la possibilité.

$T_i \in T_E \wedge T_Q$: représente les termes de la requête qui indexent les balises.

θ_j^e : représente l'instance de E_j dans la configuration θ^e

Détermination des valeurs des arcs nœud balise- nœud terme :

- La possibilité de pertinence d'un terme (t_i) pour représenter un élément (e_j), notée $P(t_i|e_j)$ est calculé comme suit :

$$P(t_i|e_j) = \frac{tf_{ij}}{\max_{t_k \in e_j} tf_{kj}}$$

tf_{ij} : est la fréquence du terme t_i dans l'élément e_j .

$\max_{t_k \in e_j} tf_{kj}$: est la fréquence maximal des termes dans l'élément e_j .

Par contre un degré de nécessaire pertinence, β_{ij} , du terme t_i pour représenter l'élément e_j , sera défini par :

$$N(t_i \rightarrow e_j) \geq \beta_{ij} = \mu \cdot tf_{ij} \cdot idf_{ij} \cdot idf = \mu \cdot tf_{ij} \cdot \log \frac{Ne}{nei} \cdot \log \frac{N}{ni}$$

Avec :

N et N_e : respectivement le nombre de document et d'éléments dans la collection ;

n_i et nei : respectivement le nombre de document et le nombre d'élément contenant le terme t_i ;

μ : est une fonction de normalisation ;

Valeur de l'arc nœud document nœud balise (propagation de pertinence) :

Le degré de possibilité de propagation d'une balise (e_j) vers le nœud document d_i est défini par :

$$e_j \rightarrow d_i = \alpha^{dist(d_i, e_j) - 1}$$

Avec :

- $dist(d_i, e_j)$: la distance de la balise e_j à la racine d_i 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 e_j vers le nœud racine d_j ainsi :

$$N(e_j \rightarrow d_i) = 1 - \frac{le_j}{dl}$$

Avec :

- le_j : la taille du nœud 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 :

$$e_j \rightarrow d_i = \frac{le_j}{dl}$$

7. L'approche proposée par Najeh [Najehnaffakhi, 2009] :

Topologie du modèle :

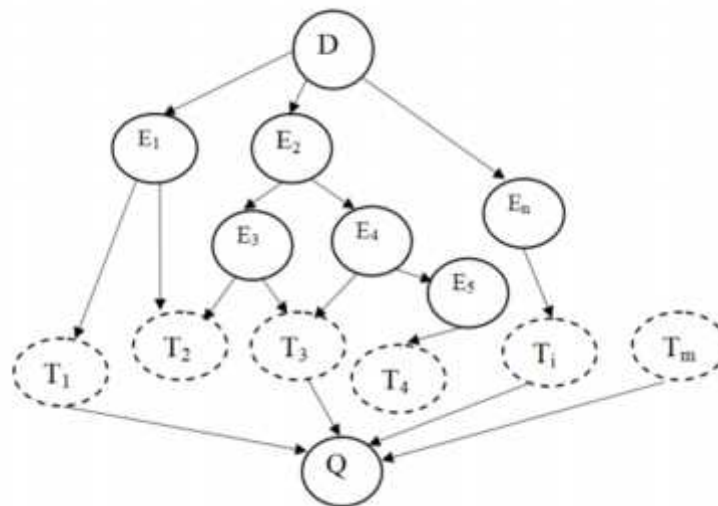


Figure 2.21 : architecture générale du modèle proposé par [Naffakhinajeh et al, 2009]

On a proposé un modèle pour la RIS basé sur les réseaux bayésien dont la topologie illustré par la figure 2.21. 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.

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).

Topologie du modèle bayésien :

- Le nœud D_j représente un document de la collection C . chaque nœud D_j représente une variable aléatoire binaire. L'instanciation $D_j=1$ signifie que le document est activé (choisi)
- Les nœuds $E_1, E_2, \dots, E_n$ représentent les éléments du document D_j .
- Chaque nœud E_j représente une variable aléatoire prenant des valeurs binaires dans l'ensemble $\text{dom}(E_j)=\{1,0\}$.
- L'instanciation $E_j=1$ signifie que l'élément E_j est indexé par au moins un nœud terme.
- Les nœuds $T_1, T_2, \dots, T_m$ sont les nœuds termes. Chaque nœuds terme T_i représente une variable aléatoire binaire prenant des valeurs dans l'ensemble $\text{dom}(T_i)=\{1,0\}$ où l'instanciation $T_i=1$ signifie que le terme T_i est présent dans le nœud père auquel il est relié (le nœud balise E_j contient ce terme T_i ou requête Q).
- Un terme est relié aussi bien au nœud qui le comporte qu'à tous les descendants de ce dernier.

Les auteurs supposent que la requête Q est composée d'une simple liste de mots-clés :

$Q=\{T_1...T_m\}$. Et la notation suivante est utilisée :

$T \ Q \ \text{resp.} \ T \ E$ 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, $T_i \in T \ Q \ \Delta T \ E$, sont évalués dans le contexte de leurs parents par $P \ T_i \ E_j$, et séparés des termes de la requête absente 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 instantiation 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 instantiation. L'ensemble des instances possibles des parents de la requête est noté θ^Q .
- 2- Générer l'ensemble des configurations possible θ d'un résultat de recherche en identifiant une configuration θ_i pertinente. Cette configuration est définie par :

$\theta_i = \theta_i^{j,id} \ j = 1 \dots z$, $\theta_i^{j,id}$ est l'élément E_j 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 θ_i , sous les trois contraintes qui permettent de vérifier la non redondance d'information dans une configuration et la complémentarité de leurs éléments :

$$- P \ \theta_i^{j,id}, Pa \ \theta_i^{j,id} = 0 \quad (1)$$

$$- T \ \theta_i^{j,id} \ \Delta T \ \theta_i^{j',id'} \ \Delta T \ Q \subseteq T \ Q \quad (2)$$

$$- T \ \theta_i^{j,id} \ \Delta T \ Q \ \Delta T \ \theta_i^{j',id'} \ \Delta T \ Q = \emptyset \quad (3)$$

Avec $Pa \ \theta_i^{j,id}$ est l'élément parent du nœud associé à $\theta_i^{j,id}$

- 3- La possibilité jointe d'observer une requête Q et sa réponse dans un document D_j est donnée par :

$$P \ Q, D_j = 1 = P \ Q \ T \ Q \times P \ T \ Q \ | \theta_i \times P \ \theta_i \ | D_j = 1 \quad [1]$$

- 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 :

$$P \ Q \ T \ Q = P \ Q \ | T_1 \dots T_m = \prod_{T_k \in T \ Q} P \ Q \ | T_k \quad [2]$$

Avec : $P(Q|T_k) = 1$ pour tout terme $T_k \in T \cap Q \cap T \cap E$ et 0 sinon.

➤ Le deuxième facteur de la formule [1] mesure la pertinence est calculé comme suit :

$$P(T, Q | \theta_l) = P(T_1 \dots T_m | \theta_l) = \prod_{T_k \in T \cap Q \cap T \cap \theta_l} P(T_k = 1 | \theta_l) \quad [3]$$

$$\text{Avec } P(T_k = 1 | \theta_l) = \frac{\sum_{\theta_l^{j,id} \in \theta_l} tf_k^{j,id}}{tf_d}$$

Où :

$\sum_{\theta_l^{j,id} \in \theta_l} tf_k^{j,id}$: est la fréquence du terme t_k dans la configuration θ_l .
 tf_d : est la fréquence du terme dans un document d .

➤ Le troisième facteur de la formule [1] est $P(\theta_l | Dj) = 1$, mesure la complémentarité entre les éléments d'une configuration possible. Ces éléments sont indépendants, la formule précédente s'écrit comme suit :

$$P(\theta_l | Dj) = 1 = \prod_{j=1}^{|\theta_l|} P(\theta_l^j | Dj) = 1$$

$$\text{Avec : } P(\theta_l^j | Dj) = 1 = \frac{\text{dist}(Dj, \theta_l^j)}{\text{dist}(Dj, \text{élément plus profond } \theta_l^k)}$$

Dj, θ_l^j : est la distance entre le nœud racine Dj et un de ces nœuds descendant θ_l^j du document (relativement à une configuration donnée θ_l).

$\text{dist}(Dj, \text{élément plus profond } \theta_l^k)$: la distance entre le nœud racine Dj et le plus profond élément muni du nœud θ_l^j est noté θ_l^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 :

$$P(Q|T_k) \times \prod_{T_k \in T \cap Q \cap T \cap \theta_l} P(T_k | \theta_l) \times \prod_{j=1}^{|\theta_l|} P(\theta_l^j | Dj) = 1$$

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 :

$$\arg \max_{\theta_l \in \Theta} P(Q|T_k) \times \prod_{T_k \in T \cap Q} P(T_k|\theta_l) \times \prod_{j=1}^{|\theta_l|} P(\theta_j^l|D_j) = 1$$

- Cette proposition, permet de générer un seul document composite, mais sans restriction sur le nombre de nœuds résultats que ce document peut contenir.

8. Synthèse :

En effet les réseaux bayésiens ont été appliquées à la RI vers la fin des années 80, ensuite ils ont connu une large utilisation 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.

Nous avons vu dans ce qui a précédé 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.

Nous avons conclu que ces modèles de RIS utilisant les réseaux bayésiens proposés diffèrent par rapport :

- 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 requête traités (CO ou CAS)

Etude comparatif :

- ❖ Pratiquement tous les modèles étudiés représentent des arcs allant des documents jusqu'à leurs termes mis à part le modèle 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 [Piwowarski et al, 2002, 2003] les auteurs ont modélisé chaque document par un réseau bayésien, ils ont 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. Tant dis que dans le système Granata [De campos et al, 2003] les auteurs représentent le corpus tout entier par un réseau bayésien et a introduit des nœuds virtuels pour représenter les éléments mixtes.

- ❖ On a remarqué dans toutes les approches qu'elles ont utilisées des mesures de probabilités à l'exception du modèle possibiliste [Bessai et al] 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 obtenue 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).

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. Le modèle d'agrégation lui a utilisé les fonctions de calcul de probabilités. 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 : dans [Piwowarski et al, 2002] et [Alimazighi, 2005] les auteurs ont pris en considération la formulation de la requête. Alors que pour [Najehnaffakhi, 2009] elle n'a traité que les requêtes de type CO
- ❖ On a vu aussi dans [Najehnaffakhi, 2009] qu'une nouvelle approche de sortie de résultats a été appréhendée 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 parties d'informations demandées mais de les voir affichées toutes 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 documents ou traité le processus de relevance Feedback où les réseaux bayésiens seraient 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 bayésiens permettent également d'effectuer l'inférence probabiliste (calcul des probabilités à 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 bayésiens 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)

Dans ce qui suit nous allons présenter en détails l'approche proposée par [Djebbar&Lamrous, 2013] que nous avons implémenté.

Chapitre III :

un modèle

proposé basé sur

les réseaux

bayésiens

Introduction :

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.

Plusieurs méthodes portant sur les réseaux bayésiens ont été proposé, nous avons décidé d'implémenté l'approche proposée par [Djebbar & Lamrous, 2013]. Dans ce chapitre nous allons détailler cette dernière avec quelques modifications que nous avons jugées nécessaires

1. Approche proposée par [Djebbar & Lamrous, 2013] :

Nous avons mis en œuvre une topologie du réseau bayésien dans le but d'avoir :

- Chaque élément dans un document 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. Autrement dit la sémantique du contenu véhiculé par paragraphe 1 est différente de celle de paragraphe 2 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.

La topologie proposée est la suivante :

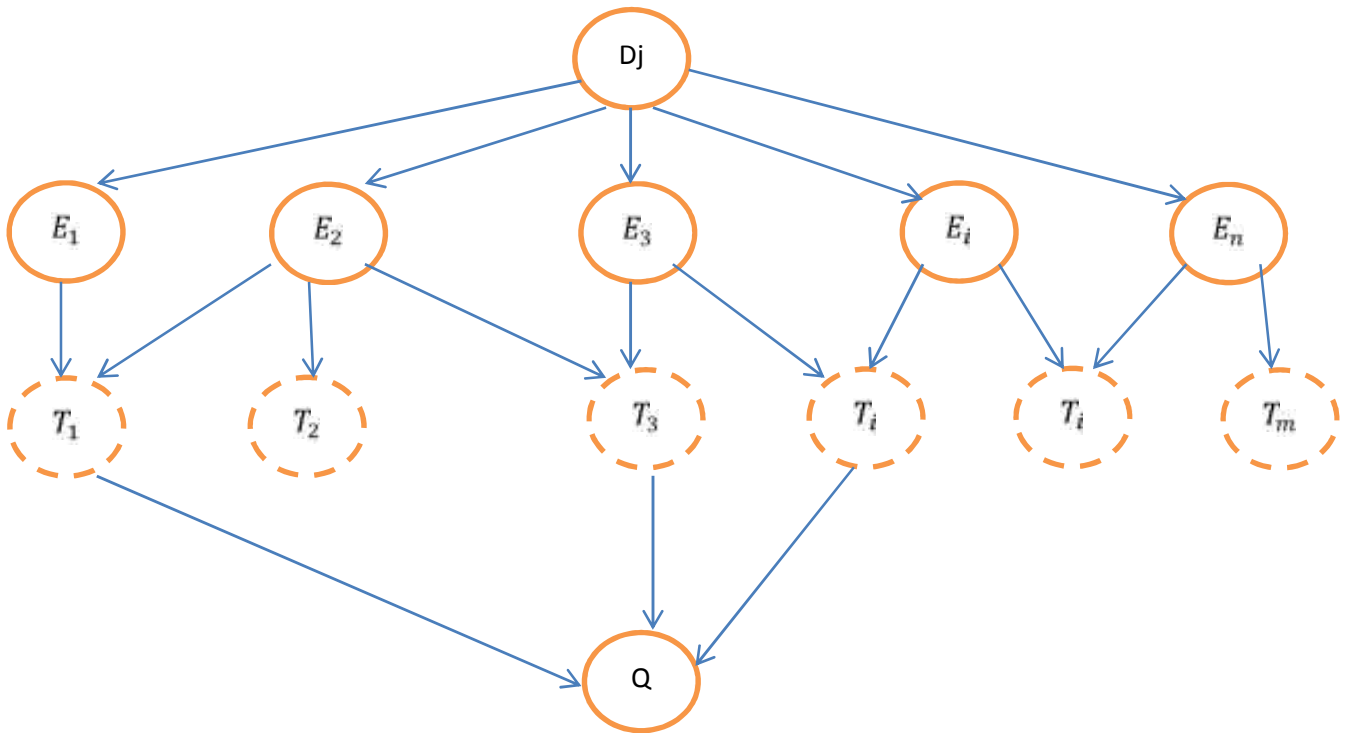


Figure 3.1 : topologie du réseau bayésien

1.1.Description du modèle :

- A chaque nœud du réseau est associée une variable aléatoire binaire prenant leurs valeurs dans le domaine $\{1, 0\}$ respectifs. Le nœud D_j représente un document de la collection C .
- L'instanciation $D_j=1$ signifie que le document est activé (choisi). (les auteurs ne s'intéressent aux documents activés noté d_j)
- Les nœuds $E_1, E_2 \dots E_n$ représentent les éléments du document D_j . L'instanciation $E_j=1$ signifie que l'élément E_j est pertinent par rapport à la requête Q et ne véhicule pas la même information par rapport aux autres éléments d'une configuration θ_i (une configuration θ_i est une réponse possible à la requête et composée par des éléments pertinents et non-redondant).
- Les nœuds $T_1, T_2 \dots T_m$ sont les nœuds terme où l'instanciation $T_i=1$ signifie que le terme T_i est présent dans l'objet (nœud père auquel il est relié c-à-d le nœud balise E_j , ou requête Q , qui contient ce terme T_i) et donc représentatif de l'objet.
- Le passage du document vers la représentation sous forme de réseau bayésien consiste à ramener tous les nœuds (balise du document) au niveau des variables E_i . On note par $T(E)$ (resp. $T(Q)$) l'ensemble des termes d'indexation des éléments du document (resp. de la requête).
- Les auteurs considèrent 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 (il s'agit d'une requête CO)

Remarque : l'inconvénient des réseaux bayésiens 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 élément, cette approche consiste à définir la meilleure configuration d'éléments (la plus pertinente) satisfaisant la requête de l'utilisateur à savoir :

1.2.pertinence d'un document vis-à-vis d'une requête :

$$P(d_i|Q) = \frac{P(Q \Delta d_i)}{P(Q)}$$

Remarque : la valeur du d'énomérateur est constant i.e c'est la même valeur quel que soit le document choisie

D'où la formule suivante :

$$P(d_i|Q) \propto P(Q \Delta d_i)$$

$$P(Q \Delta d_i) = \sum_{\theta^l} P(Q|\theta^l) P(\theta^l|\theta^e) * P(\theta^e|d_i) * P(d_i) \quad [1]$$

Avec :

θ^l : est une configuration possible des termes de la requête.

θ^e : est une combinaison parmi toutes les combinaisons possible des variables éléments.

Remarque : l'inconvénient des réseaux bayésiens 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 élément, nous avons opté à mettre en place des configurations et définir la meilleur configuration d'éléments (la plus pertinente) satisfaisant la requête de l'utilisateur.

1.3.Configuration d'élément répondant à la requête :

$$P(Q \Delta d_i) = \arg \max_{\theta^e} \sum_{\theta^l} P(Q|\theta^l) P(\theta^l|\theta^e) * P(\theta^e|d_i) * P(d_i) \quad [2]$$

1.4.Apprentissage du modèle :

Dans notre modèle 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, i.e les tables de probabilités conditionnelles associée à chaque nœuds suivant une configuration donnée.

Cette tâche consiste à estimer pour nous les paramètres $P(Q|\theta^t)$ (valeur de relation requête-terme), $P(\theta^t|\theta^e)$ (valeur de relation terme-élément), $P(\theta^e|d_i)$ (valeur de relation élément-racine)

1.4.1. Calcul de la probabilité à priori $P(d_i)$:

Les auteurs considèrent qu'un document D_i instancié à 1 vérifie :

$$P(d_i) = 1 \text{ et } P(\neg d_i) = 0$$

Cette valeur étant la même pour chaque document D_i du corpus, en plus du fait qu'elle soit neutre pour le produit on pourra l'éliminer de la formule [2] :

$$P(Q|d_i) = \arg \max_{\theta^e} \prod_{\theta^t} P(Q|\theta^t) P(\theta^t|\theta^e) * P(\theta^e|d_i)$$

1.4.2. Calcul $P(Q|\theta^t)$:

Elle sera estimée selon la nature de la requête, nous distinguons :

- Requête conjonctive
- Requête disjonctive

Requête conjonctive : pour une requête booléenne ET, tous les termes d'une configuration doivent être instanciés comme dans la requête :

$$P(Q|\theta^t) = \begin{cases} 1 & \text{si } \theta_i^Q = \theta_i^t \square T_i \square \text{parent } Q \\ 0 & \text{sinon} \end{cases}$$

Exemple :

Soit la requête conjonctive composée des termes $T1$ et $T2$, la probabilité $P(Q|\theta^t) = P(Q|T1T2)$ est donnée par la table suivante :

$T1T2$	$P(Q T1T2)$
$t1t2$	1
$t1\bar{t2}$	0
$\bar{t1}t2$	0
$\bar{t1}\bar{t2}$	0

Conclusion : ce type de requête est considéré comme étant trop strict.

Requête disjonctive : pour une requête booléenne OU, au moins un terme de la configuration doit être instancié comme dans la requête.

$$P(Q|\theta^t) = \begin{cases} 1 \square T_i \square \text{parent } Q & \text{si } \theta_i^Q = \theta_i^t \\ 0 & \text{sinon} \end{cases}$$

Exemple :

Soit la requête disjonctive composée termes $T1$ et $T2$, la probabilité $P(Q|\theta^i) = P(Q|T1T2)$ est donnée par la table suivante :

$T1T2$	$P(Q T1T2)$
$t1t2$	1
$t1\bar{t2}$	1
$\bar{t1}t2$	1
$\bar{t1}\bar{t2}$	0

Conclusion : ce type de requête est considéré comme étant trop large ou trop tolérant pour discriminer entre les éléments

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 :

$$P_{\frac{i}{n}} = \begin{cases} 1 & \text{si } i \geq \frac{K}{n} \\ 0 & \text{sinon} \end{cases}$$

Exemple :

Soit la requête composée des termes $T1$, $T2$, et $T3$, la probabilité $P(Q|\theta^i) = P(Q|T1T2T3)$ est donnée par la table suivante :

On considère $k=2$

$T1T2T3$	$P(Q T1T2T3)$
$t1t2t3$	1
$t1t2\bar{t3}$	1
$t1\bar{t2}t3$	1
$t1\bar{t2}\bar{t3}$	0
$\bar{t1}t2t3$	1
$\bar{t1}t2\bar{t3}$	0
$\bar{t1}\bar{t2}t3$	0
$\bar{t1}\bar{t2}\bar{t3}$	0

Conclusion : 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 $P(Q|T_K)$ 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 grande valeur 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 $P(Q|\theta^i)$.

Les auteurs ont proposées suite à cela la formule suivante :

$$P(Q|\theta^i) = \frac{\sum_{\theta_j^i = \theta_i^q} idf_i}{|\theta^i|}$$

Où :

$$idf_i = \log \frac{N}{n_i}$$

Avec :

n_i : est le nombre de documents contenant le terme t_i .

N : est le nombre de document dans la collection

$|\theta^i|$: est le nombre de configuration possible des termes de la requête

1.4.3. Calcul de la valeur de $P(\theta^e|d_i)$:

Les éléments ainsi que les termes sont considérés comme étant indépendants entre eux, on a alors :

$$P(\theta^e|d_i) = \prod_{\theta_j^e \in \theta^e} P(E_j|d_i)$$

Où : θ_j^e : représente un élément E_j dans la configuration θ^e

Remarque : θ_j^e et E_j désignent le même élément.

Valeur de l'arc balise-document :

Partant de l'idée que plus un nœud est distant (éloigné) de la racine, plus il est porteur d'information, la formule suivante a été proposée :

$$P(E_j|d_i) = \frac{Dist(D_i, \theta_j^e)}{Dist(D_i, \text{l'élément le plus profond } \theta_j^e) + \alpha}$$

Où

$Dist D_i, \theta_j^e$: donne le nombre d'arcs séparant le nœud D_i au nœud θ_j^e .

$Dist D_i, l'élément le plus profond \theta_j^e$: détermine la distance entre le nœud D_i et le nœud le plus profond θ_j^e .

α : est une valeur qui sera choisie proche de 0. Son intérêt est de ne pas annuler des pertinences de configuration.

1.4.4. Calcul de $P(\theta^1, \theta^e) = P(t_1, \dots, t_n | \theta^e)$:

$$P(\theta^1 | \theta^e) = \frac{P(t_i | \theta_j^e)}{\prod_{\theta_j^e \in \theta^e} P(t_i | \theta_j^e) \prod_{T_i \in T \setminus \Delta T} P(t_i | \theta^Q)}$$

Valeur de l'arc terme-balise :

Intuitivement, plus le document est pertinent pour une requête plus ses éléments le sont, La pondération peut dépendre de l'élément lui-même, ses parents et le document.

$$P(t_i | \theta_j^e) = \rho \frac{tf_{ij}}{\theta_j^e} + (1 - \rho) P(t_i, d_i)$$

Avec :

tf_{ij} : est la fréquence du terme t_i dans l'élément θ_j^e .

θ_j^e : est le nombre de terme dans l'élément.

ρ : est une valeur fixée par expérimentation.

$P(t_i, d_i)$: est le rapport de la fréquence de t_i dans le document D_i et le maximum des fréquences dans le document.

Les auteurs ont proposé les estimations suivantes :

- $P(\bar{t}_i | \bar{\theta}_j^e) = 0.8$
- $P(t_i | \theta_j^e) = 0.2$

La formule [2] sera alors égale à

$$P(Q | d_i) = \arg \max_{\theta^e} \sum_{\theta^1} P(Q | \theta^1) \prod_{\theta_j^e \in \theta^e} \prod_{T_i \in T \setminus \Delta T} P(t_i | \theta_j^e) \prod_{T_i \in T \setminus \Delta T} P(t_i | \theta^Q) \prod_{\theta_j^e \in \theta^e} P(E_j | d_i)$$

Résumé :

- $P(D_i)$

D_i	$P(D_i)$
d_i	1
$\bar{d}_i$	0

- $P(T_i|E_j)$: terme-élément

$P(T_i E_j)$	e_j	$\bar{e}_j$
T_i	$\rho \frac{tf_{ij}}{\theta_j^e} + 1 - \rho P(t_i, d_i)$	0.2
$\bar{T}_i$	$1 - \rho \frac{tf_{ij}}{\theta_j^e} + 1 - \rho P(t_i, d_i)$	0.8

- $P(E_j|D_i)$: élément-racine

$P(E_j D_i)$	d_j
e_j	$\frac{Dist(D_i, \theta_j^e)}{Dist(D_i, \text{l'élément le plus profond } \theta_j^e) + \alpha}$
$\bar{e}_j$	$1 - \frac{Dist(D_i, \theta_j^e)}{Dist(D_i, \text{l'élément le plus profond } \theta_j^e) + \alpha}$

- $P(Q|\theta^l)$: requête-terme

$P(Q \theta^l)$	Noisy Or
Q	$\frac{\sum_{\theta_j^l = \theta_i^Q} idf_i}{ \theta^l }$

Conclusion :

Dans ce chapitre nous avons détaillé l'approche proposée sur les réseaux bayésiens. Dans le chapitre qui va suivre nous avons présenté l'expérimentation et les résultats des tests que nous avons effectués.

Chapitre IV : implémentation et résultats

Introduction :

Afin d'évaluer la pertinence de l'approche proposée nous avons procédé à son implémentation. Pour ce faire, nous avons utilisé un ensemble d'outils logiciel pour offrir une plateforme adéquate à nos tests. Dans la partie 1, nous présenterons les outils de développement ainsi que la plateforme d'accueil utilisée pour exécuter nos programmes de recherche dans le cadre du modèle bayésien proposé.

Cette application a été réalisée dans une plateforme Unix (UBUNTU 10.10) montée dans une VMware Workstation, en utilisant les outils suivants : module d'indexation Xfirm, Eclipse comme environnement de développement, Java comme langage de programmation, Oracle 10g XE comme SGBD pour la gestion de la BDD du système contenant l'index des documents de la collection et les pilotes ODBC & JDBC pour la connexion à la BDD du système Xfirm pour récupérer les données nécessaires à la recherche.

Dans le but d'évaluer la qualité d'un SRI ou d'une approche de recherche, cette dernière doit subir une implémentation et un ensemble de tests qui vont éclaircir sa qualité, sa pertinence ainsi que sa précision, et à quel point que cette approche pourra agir sur la satisfaction du besoin de en information de l'utilisateur. Dans cette optique, nous avons réalisé les tests que nous présenterons au long de la partie 2 qui montrera les éléments de tests utilisés, la procédure d'implémentation, les tests ainsi que les résultats obtenus et leur évaluation en utilisant des mesures de la compagne d'évaluation INEX.

Partie 1 : Environnement et outils d'implémentation**1. VMware Workstation :**

VMware Workstation est un environnement de test et de développement qui permet aux administrateurs système de créer et d'exécuter des machines virtuelles (VM) directement sur un bureau. Il permet aussi l'évaluation des hyperviseurs¹.

La version utilisée, VMware Workstation 9, est optimisée pour fonctionner avec les systèmes d'exploitation 64 bits et Windows 8. La VMware utilise une interface Web pour connecter les utilisateurs de machines virtuelles locales et serveur hébergé depuis un PC, un smartphone ou une tablette. La figure suivante montre l'interface de la VMware Workstation9 utilisée :

¹¹**Un hyperviseur** : est aussi appelé un gestionnaire de machine virtuelle, est un programme qui permet à plusieurs systèmes d'exploitation de partager une foule matérielle unique. Chaque système d'exploitation semble avoir le processeur de l'hôte, la mémoire et d'autres ressources à lui tout seul.

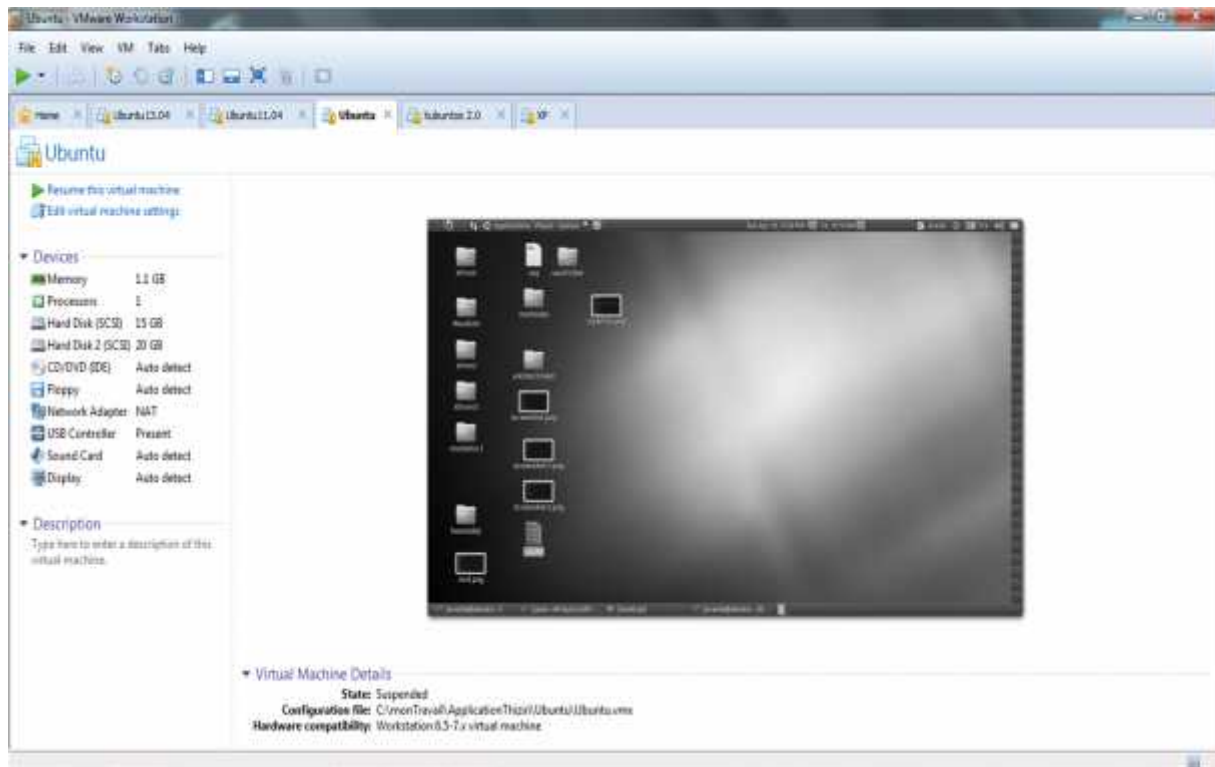


Figure 4.1 : Interface de la VMware Workstation 9

2. Le système de recherche Xfirm :

2.1 Présentation :

C'est un système de recherche d'informations développé en 2005 par Karen Sauvagnat, ce système se base sur le modèle vectoriel [Sauvagnat, 2005] et est orienté vers la RI structurée dans des corpus documentaires XML. Il a apporté de la flexibilité dans la recherche car il utilise une représentation des documents par un modèle générique de données (description des formats des documents dans une base de données) et un langage de requêtes en permettant l'expression du besoin en information de l'utilisateur avec des simples expressions du langage naturel puis une reformulation des requêtes.

XRIRM est un système orienté pertinence, il repose sur la propagation de pertinence à travers les éléments constituant un document : un premier score est calculé pour les nœuds feuilles ensuite il est propagé dans l'arbre du document et il est diminué durant la propagation pour répondre au critère de spécification d'une unité pour une requête reçue.

Pour assurer un fonctionnement cohérent, il présente un ensemble de modules qui ont donné lieu au développement d'un prototype permettant l'indexation de la collection documentaire XML. Ce prototype est entièrement réalisé en langage JAVA en utilisant des API (Application Programming Interface) telles que SAX (Simple API for XML) pour parser les documents XML et un JDBC pour l'accès aux BDD du système. Ses composantes principales ainsi que leurs liaisons et structures sont présentées par l'architecture du prototype présentée dans la figure suivante :

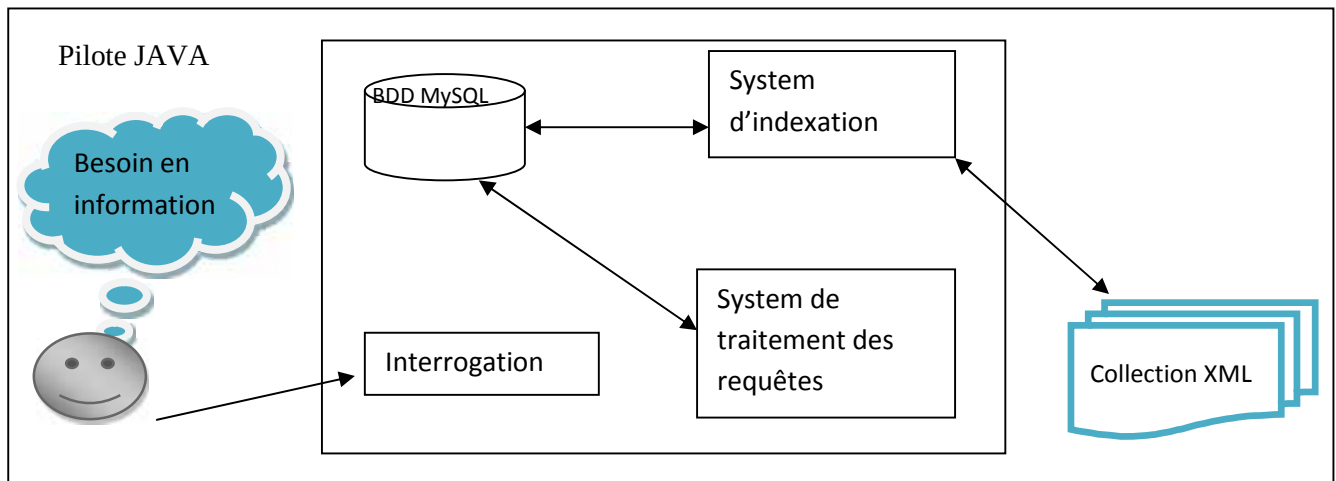


Figure 4.2: Architecture de base du model XFIRM [Sauvagnat, 2005] système

2.2 Modules exploités

Dans l'implémentation de notre travail, nous avons exploité le module d'indexation constitué du système d'indexation, la BDD et le parser SAX, tel c'est mentionné dans la figure ci-dessus, pour indexer notre collection de test constituée que de 1000 documents XML. Pour interroger notre système ci obtenu et effectuer la recherche, nous avons développé le module d'interrogation et de recherche, tel que ce dernier exploite la BDD contenant les index, en accédant aux tables suivantes :

- **Collection** : contient des informations générales sur toute la collection : nombre de documents, nombre de termes, nombre d'éléments ...
- **Document** : contient des informations sur tous les documents de la collection : nombre de termes, nombre de niveau ...
- **Terme** : contient des informations sur les termes de la collection : fréquence dans la collection, nombre d'éléments le contenant ...
- **Path**: contient des informations sur les nœuds de la collection : document conteneur, l'attribut, taille ...
- **NodeToTerm**: contient des informations sur les nœuds feuille de la collection : les termes présents dans un nœud donné ...

3. L'environnement de développement Eclipse :

C'est un environnement de développement intégré (IDE) développé par I.B.M. pour des applications telles que java, dans le site² d'I.B.M associé à Eclipse on trouvera la définition suivante : «*Le projet Eclipse est un projet de développement de logiciels open source dédié à fournir un robuste, complet, une qualité commerciale et plate-forme de l'industrie pour le développement d'outils hautement intégrés*». Donc, par définition, Eclipse est une plate-forme ouverte pour l'intégration de l'outil, par l'intégration des plugins à un IDE. La question a été confondue car une force industrielle complète, y compris la fonction IDE Java est fournie avec la plate-forme Eclipse, sous la forme de plugins qui étendent les installations du cadre de base de l'Eclipse. En outre, les plugins Eclipse peuvent s'étendre d'autres plug-ins. Quand une application basée sur Eclipse démarre, il découvre et active tous les plug-ins qui ont été configurés pour le poste de travail. La plate-forme Eclipse est littéralement la

²<http://www.eclipse.org>

partie, car il est capable d'effectuer n'importe quelle fonction qui a été ajoutée à elle par les plug-ins qu'il contient pour le moment.

Eclipse est dit universel et polyvalent, car il permet de créer des projets de développement, qui mettent en œuvre n'importe quel type de langage de programmation (Java, C++, PHP, JavaScript, ...). La figure suivante présente l'interface de l'IDE Eclipse sous linux :

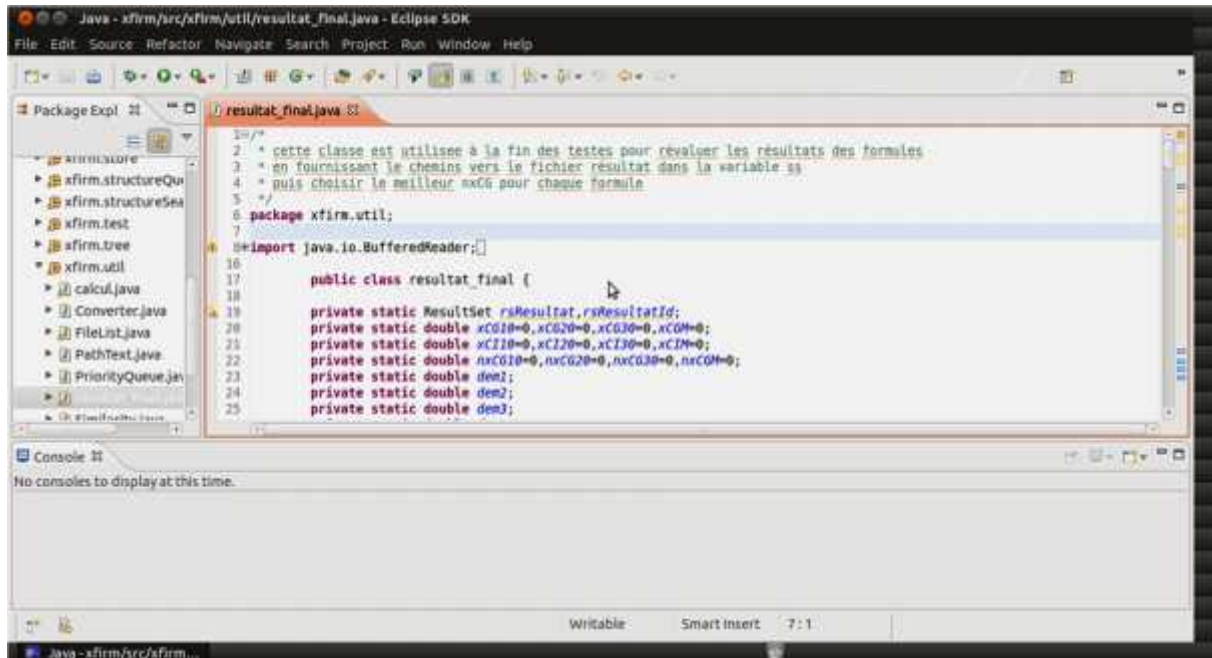


Figure 4.3 : L'interface de l'IDE Eclipse sous linux

3.1 Le langage Java :

Le choix de java comme langage de programmation s'est imposé vu que le code source d'XFIRM est entièrement écrit en java.

On peut faire remonter la naissance de Java à 1991. À cette époque, des ingénieurs de chez **Sun Microsystems** ont cherché à concevoir un langage applicable à de petits appareils électriques (*code embarqué*). Pour ce faire, ils se sont fondés sur une syntaxe très proche de celle de C++, en reprenant le concept de machine virtuelle déjà exploité auparavant par le PascalUCSD. L'idée consistait à traduire d'abord un programme source, non pas directement en langage machine, mais dans un pseudo langage universel, disposant des fonctionnalités communes à toutes les machines. Ce code intermédiaire, dont on dit qu'il est formé de *bytecodes*, se trouve ainsi compact et portable sur n'importe quelle machine, il suffit simplement que cette dernière dispose d'un programme approprié (on parle alors de *machine virtuelle*) permettant de l'interpréter dans le langage de la machine concernée. Il est dit aussi un langage multiplateforme selon la célèbre maxime «Write once, run everywhere».

Java est un langage de programmation à usage général mais avec des fonctionnalités qui le rendent plus simple :

- Création des applications web, les servlets,
- Les applets,
- Programmation procédurale, événementielle et implémentation flexible de l'orienté objet,
- Bibliothèques riches de méthodes.
- ...

Dans notre cas, on s'est basé sur la programmation procédurale par l'exploitation des différentes bibliothèques permettant l'implémentation des différentes phases de recherche.

Aujourd'hui, Java trouve une nouvelle niche dans la création d'applications RIA (Rich Internet Applications), des applications qui proposent des fonctionnalités, notamment des interfaces, plus évoluées à la fois sur Internet et sur les téléphones portables. Le langage JavaFx est un langage agile dérivé de Java, sous le contrôle de Sun Microsystems, qui met à profit la portabilité de Java ainsi que les vastes bibliothèques déjà disponibles dans le langage Java.

4. Le SGBD Oracle10g EX :

Un SGBD (Système de Gestion des Bases de Données) est un logiciel permettant la création, manipulation et modification des BDD, ainsi que le contrôle et la confidentialité des données, dans notre cas, le SGBD utilisé est Oracle 10g :

4.1 Définition :

Oracle est un [SGBD](#) écrit en langage [PL/SQL](#) et édité par la société³ du même nom, qui est un leader mondial des bases de données et a été créée en 1977 par Lawrence Ellison, Bob Miner, et Ed Oates. Elle s'appelle alors *Relational Software Incorporated (RSI)*² et commercialise un Système de Gestion de Bases de données relationnelles (SGBDR ou RDBMS pour *Relational Database Management System*) nommé *Oracle*.

En 1984 la première version d'Oracle (Oracle4) est commercialisée sur les machines IBM. Et en 1985 Oracle 5 permet une utilisation client-serveur grâce au middleware *SQL*Net*. Et en 1988 Oracle 6 est disponible sur un grand nombre de plates-formes et apporte de nombreuses nouvelles fonctionnalités ainsi qu'une amélioration notable des performances.

Et ce n'est qu'en 1992, qu'Oracle 7 sort sur les plates-formes UNIX (elle ne sortira sur les plates-formes Windows qu'à partir de 1995). Cette version permet une meilleure gestion de la mémoire, du CPU et des entrées-sorties. La base de données est accompagnée d'outils d'administration (*SQL*DBA*) permettant une exploitation plus aisée de la base.

4.2 Les fonctionnalités d'Oracle :

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents (transactions)

o L'interface *SQL*PLUS* :

Oracle propose également de nombreux outils de développement permettant d'automatiser la création d'applications s'interfaçant avec la base de données. Parmi ces outils on a utilisé le *SQL*PLUS* qui est une interface interactive permettant d'envoyer des requêtes

³Oracle Corporation <http://www.oracle.com>

SQL et PL/SQL à la base de données, la créer et de manipuler interactivement des objets de la base via une interface en ligne de commandes à travers le SGBD Oracle. SQL*PLUS est habituellement utilisé pour formuler une requête SQL et obtenir, sur un écran et de façon immédiate, le résultat attendu.

La figure suivante montre cette interface sur la plateforme linux :

```

oracle@ubuntu:~$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Thu May 9 20:21:33 2013
Copyright (c) 1982, 2010, Oracle. All rights reserved.
Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 167772160 bytes
Fixed Size 1218316 bytes
Variable Size 62916852 bytes
Database Buffers 180663296 bytes
Redo Buffers 2973696 bytes
Database mounted.
Database opened.
SQL> describe document;

```

Name	Null?	Type
DOC_ID	NOT NULL	NUMBER(38)
DOCUMENT		NVARCHAR2(300)
TERM_NB		NUMBER(38)
DEB		NUMBER(38)
FIN		NUMBER(38)
NBR_NIV		NUMBER(38)
KUOID		NUMBER(38)

Figure 4.4 : Illustration de l'interface SQL*PLUS sous linux.

4.3 Les outils de connexion au SGBD et à la BDD :

4.3.1 Le pilote de connexion au SGBD Oracle :

La classe permettant la connexion au SGBD est : *java.sql.DriverManager* qui se charge de la gestion, du contrôle et de la connexion au SGBD en fournissant les méthodes principales suivantes :

- *static void registerDriver(Driver driver)*: enregistre le driver pour un type de SGBD particulier ;
- *static Connection getConnection (String url, String user, String password)*: Crée une connexion permettant d'utiliser une base de données.

Avec:

driver : est un pilote dépendant du SGBD utilisé, dans notre cas il s'agit du pilote d'oracle sous linux : *jdbc:oracle:thin:@oracleserv.irit.fr:1521:test*

url : identification de la base considérée sur le SGBD à format dépendant du SGBD utilisé

user : nom de l'utilisateur qui se connecte à la base

password : mot de passe de l'utilisateur

4.3.2 L'ODBC/JDBC :

- **Le JDBC** :(Java Data Base Connectivity)

C'est un outil de connexion à la BDD conçu par Sun. JDBC est un Framework pour le langage Java permettant l'accès aux bases de données relationnelles dans un programme Java

indépendamment du type de la base utilisée (mySQL, Oracle, Postgres...) et seule la phase de connexion au SGBDR change. Il permet de faire tout type de requêtes : sélection des données dans des tables, création et insertion d'éléments dans les tables et gestion des transactions. Les packages java le configurant : *java.sql* et *javax.sql*

- **L'ODBC :**

Abrégé en ODBC Open Data base Connectivit y est un outil de connexion à une BDD, il construit un pont « *Bridge* » entre le pilote JDBC et la BDD configurée. Tel que le JDBC communique avec l'interface ODBC et non directement avec la BDD. L'intérêt réside dans le caractère standard de l'ODBC qui est utilisé dans les SGBD comme *Microsoft SQLServer* ou *Microsoft Access*. Ces différents drivers permettent d'accéder indifféremment à la plupart des SGBD.

- **Principales classes pour accéder à une BDD**

La classe *Driver Manager* charge et configure le driver de la base de données qui permet de gérer l'accès à un type particulier de SGBD.

La classe *Connection* réalise la connexion et l'authentification à la base de données.

La classe *Statement* (et *PreparedStatement*) contient la requête SQL et la transmet à la base de données.

La classe *ResultSet* permet de parcourir les informations retournées par la base de données dans le cas d'une sélection de données.

Partie 2 : Tests et évaluation des résultats

1. Protocole d'évaluation :

Pour la réalisation de nos tests, nous avons utilisé des échantillons des éléments d'évaluation de la compagnie INEX :

1.1 Les requêtes de tests :

Pour effectuer des tests de recherche dans le cadre d'INEX de notre approche et pouvoir l'évaluer, nous avons utilisé un échantillon de 20 requêtes CO (Content Only) extraites des 124 requêtes de la compagnie INEX 2006 qu'on a pris aléatoirement de quatre sous-ensembles de ces requêtes.

Le tableau suivant montre l'ensemble des requêtes utilisées :

Le numéro de la requête	Le titre de la requête
289	emperor "Napoleon I" Polish
290	"genetic algorithm"
295	software intellectual property patent license
297	"cool jazz" "West coast" musician
321	buildings designed Antoni Gaudí Barcelona architect
322	castles kasteel in the netherlands
323	founder ikea
324	composition of planet rings
325	"Cirque du Soleil" shows
327	cloning animals accepted "United States of America"
328	NBA European basketball player
362	effect nuclear power plant accident
364	+mushroom poisonous poisoning
365	economy peru international investment tourism
366	Fourier transform applications
367	true story films best director or movie award
393	Wireless devices "Health Hazards"
408	"electroconvulsive therapy" depression
409	Hybrid Vehicles -biology "fuel efficiency" "fuel sources" model engine
413	Coordinates and Population of capital cities of Europe

Tableau 4.1 : Les 20 requêtes utilisées dans les tests des formules de pondération

1.2 La collection de test :

Le tableau suivant montre les caractéristiques principales de la collection totale d'INEX 2006 :

La taille de la collection	4.6 Go
Le nombre de documents dans toute la collection	659388 répartis en 22 ensembles
Le nombre d'éléments dans toute la collection	50.000.000

Tableau 4.2 : Caractéristiques de la collection INEX 2006

Pour nos tests, nous avons utilisé un échantillon de 1000 documents différents, extraits de toute la collection :

La taille de la collection	17,1 Mo
Le nombre de documents dans la collection de test	1000
Le nombre d'éléments dans la collection de test	191549

Tableau 4.3 : Caractéristiques de la collection de test

2. Procédure d'implémentation :

Pour implémenter notre modèle, nous avons procédé en suivant certaines étapes. Pour ce faire, tel que c'est mentionné dans la partie précédente de ce chapitre, nous avons utilisé le module d'indexation « *BaseWriter* » et la Base De Données qui contient les index du système XFIRM. Concernant la recherche, ainsi que le calcul de similarité et le renvoi des résultats, nous avons développé des modules correspondant à notre méthode, comme la classe de recherche « *InexLang* » pour une recherche dans le cadre d'INEX par le modèle bayésien proposé, ainsi que les méthodes de calcul des probabilités de similarité associées à notre modèle, dans la classe « *Similarity* » et d'autres méthodes qui servent à lire l'index et récupérer des informations telles que les nœuds d'un document, dans la classe « *BaseReader* ».

Cette procédure peut être divisée en deux phases :

2.1 Phase d'indexation :

Tel que nous avons mentionné ci-dessus, notre système utilise l'indexation XFIRM utilisant un parser SAX pour analyser les documents XML. Nous avons rempli la BDD avec l'index de 1000 documents XML de la compagnie d'évaluation INEX 2006. Les index sont stockés dans une BDD Oracle 10g EX s'exécutant sous Linux (UBUNTU 10.10), pour permettre par la suite la recherche des termes d'une requête fournie, par le parcours de la

BDD qui fait correspondre à chaque document de la collection un index contenant ses éléments et ses termes, permettant ainsi la récupération flexible des données de recherche.

2.2 Phase de recherche :

C'est la phase la plus cruciale de notre implémentation, car elle consiste à trouver tous les nœuds de tous les documents de la collection pouvant répondre à la requête utilisateur, donc nous avons procédé à récupérer tous les nœuds d'un document de la collection, puis leurs calculer les scores associés par rapport à la requête.

Lancement des tests :

Pour pouvoir lancer un test dans le cadre d'INEX, notre système exécute le module « *InexLang* », pour ce faire, nous devons tout d'abord lancer un terminal pour démarrer oracle, par les commandes suivantes :

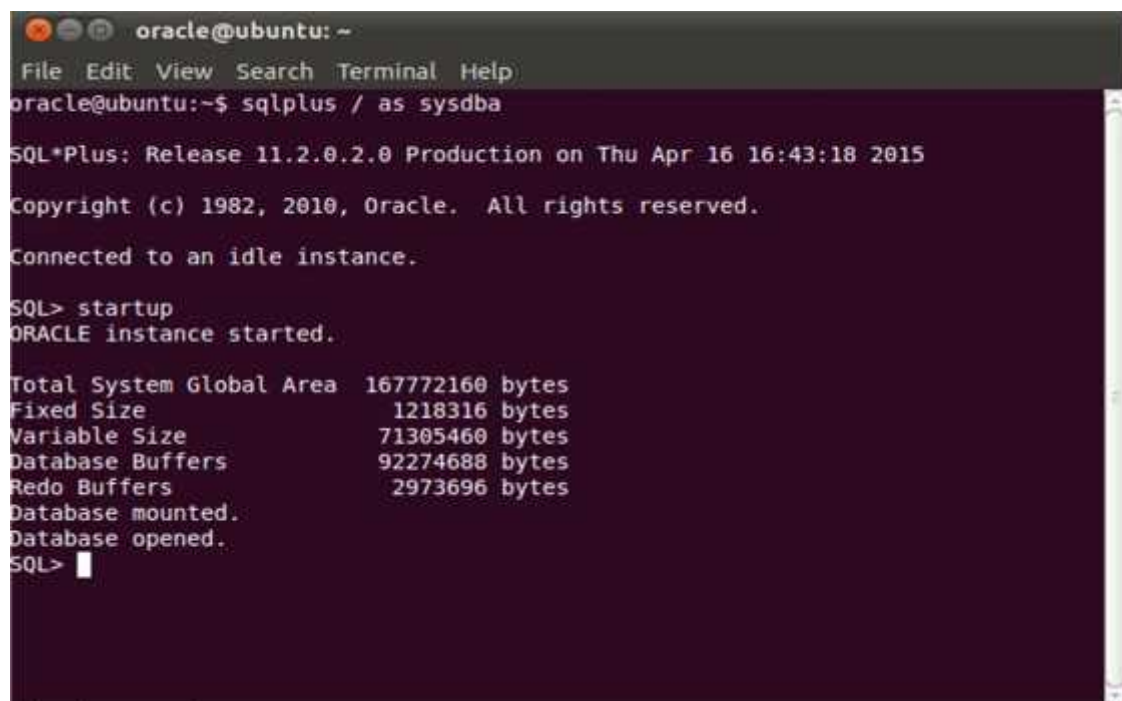
- pour lancer la console sqlplus en administrateur

```
~$ sqlplus / as sysdba
```

- lancer le serveur de BDD

```
SQL > startup
```

La figure suivante montre les résultats de chacune des commandes ci-dessus :



```
oracle@ubuntu: ~  
File Edit View Search Terminal Help  
oracle@ubuntu:~$ sqlplus / as sysdba  
  
SQL*Plus: Release 11.2.0.2.0 Production on Thu Apr 16 16:43:18 2015  
Copyright (c) 1982, 2010, Oracle. All rights reserved.  
  
Connected to an idle instance.  
  
SQL> startup  
ORACLE instance started.  
  
Total System Global Area 167772160 bytes  
Fixed Size 1218316 bytes  
Variable Size 71305460 bytes  
Database Buffers 92274688 bytes  
Redo Buffers 2973696 bytes  
Database mounted.  
Database opened.  
SQL> 
```

Figure 4.5 : Illustration des commandes de lancement du SGBD du système en lignes de commande

A partir d'un deuxième terminal, se placer dans l'environnement d'exécution et configurer un *listener* pour les requêtes de la BDD, par la commande suivante :

`~$ netca`

La figure suivante montre son utilisation :

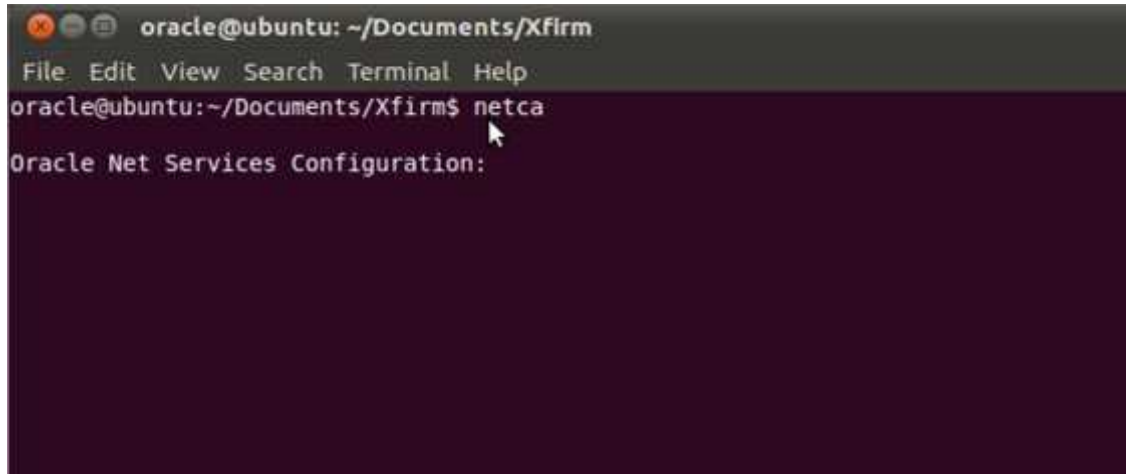


Figure 4.6 : Utilisation de la commande « netca » à partir d'un terminal

Suivre ensuite les boîtes de dialogue suivantes :



(1)



(2)



(3)



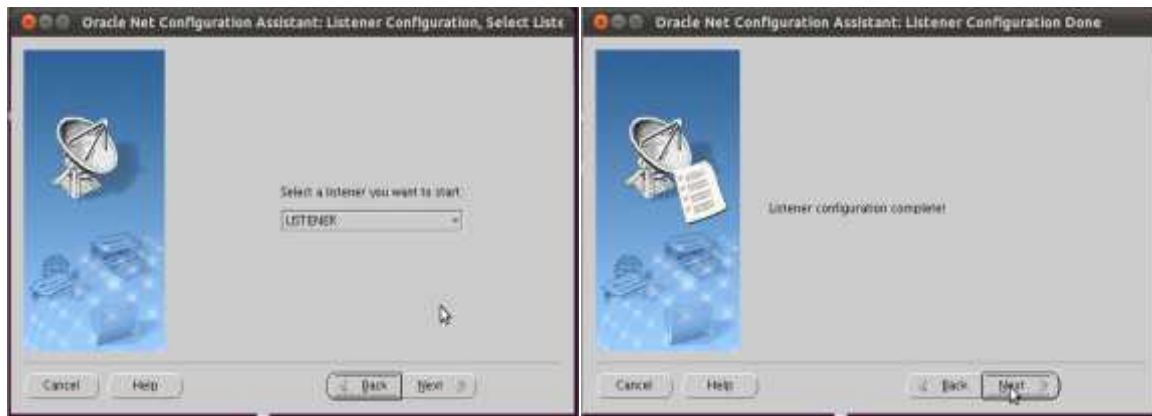
(4)



(5)



(6)



(7)

(8)



(9)

Figure 4.7 : Illustration des étapes à suivre pour la configuration d'un *listener* sous Linux

Enfin, nous pouvons lancer notre module de recherche par la commande suivante :

```
~$ java xfirm/inex/InexLang p1 p2 p3 p4
```

Avec :

p1 : chemin vers le fichier de configuration de la BDD Oracle

p2 : chemin vers le fichier contenant les requêtes de test à exécuter

p3 : chemin vers le fichier résultat qui va contenir les résultats de recherche

p4 : le nom de la collection de recherche indexée dans la BDD

La figure suivante montre un exemple de lancement d'une requête de test :

```

oracle@ubuntu: ~/Documents/Xfirm
File Edit View Search Terminal Help
oracle@ubuntu:~/Documents/Xfirm$ java xfirm/inex/InexLang xfirm/config/connect/OracleConfig.txt example/queries/req example/results/modLan/co wiki

```

Figure 4.8 : Exemple d’une requête de test par le modèle bayésien proposé

3. La mesure d’évaluation :

Afin d’évaluer notre approche, nous avons utilisé les métriques d’évaluation de INEX 2006 (Voir chapitre I).

Nous avons utilisé la classe *Resultat_final* pour calculé la valeur $nxCG$ des scores :

$$nxCG[i] = \frac{xCG[i]}{xCI[i]}$$

Le principe consiste à calculer la valeur $nxCG$ pour 5 rangs, ainsi i aura les valeurs (5, 10, 25, 50, max) qui représentent respectivement les nombre des premiers résultats considérés et la totalité des résultats.

xCG représente la somme des scores des i premiers résultats obtenu par le système considéré. xCI représente la somme des i premiers scores obtenus pour les résultats de INEX 2006 (les scores idéales)

4. Tests préliminaires :

Avant de tester notre approche, nous avons effectué au préalable des tests préliminaires afin de fixer les valeurs des paramètres α et ρ . Pour ce faire nous avons évalués 10 requêtes sur une mini collection de 100 documents de la campagne INEX 2006, et considéré le jeu de valeurs suivant des paramètres :

$\alpha = \{0.001; 0.003; 0.005; 0.007; 0.009\}$ et $\rho\{0.1; 0.3; 0.5; 0.7; 0.9\}$. Les résultats obtenus sont présentés dans les tableaux ci-après.

- Pour $\alpha = 0.001$:

Valeur de ρ	La moyenne du gain à 5	La moyenne du gain à 10	La moyenne du gain à 25	La moyenne du gain à 50
0.1	0.39780152924	0.13327899723	0.06480284547	0.03631157778
0.3	0.39780152987	0.13327899745	0.06480284561	0.03631157788
0.5	0.39780152972	0.13327899740	0.06480284556	0.03631157778
0.7	0.39780153010	0.13327899752	0.06480284565	0.03631157794
0.9	0.3978015294	0.13327899731	0.06480284551	0.03631157773

- Pour $\alpha = 0.003$:

Valeur de ρ	La moyenne du gain à 5	La moyenne du gain à 10	La moyenne du gain à 25	La moyenne du gain à 50
0.1	0.39775981943	0.133265063061	0.064847123413	0.0363386491952
0.3	0.39775981951	0.133265063087	0.064847123437	0.0363386492156
0.5	0.39775982013	0.133265063297	0.064847123577	0.0363386492911
0.7	0.39775981979	0.133265063182	0.064847123488	0.0363386493349
0.9	0.39775982010	0.133265063285	0.064847123571	0.0363386492469

- Pour $\alpha = 0.005$:

Valeur de ρ	La moyenne du gain à 5	La moyenne du gain à 10	La moyenne du gain à 25	La moyenne du gain à 50
0.1	0.39771827244	0.133251183291	0.064891288797	0.0363656521679
0.3	0.39771827193	0.133251183122	0.064891288658	0.0363656520950
0.5	0.39771827306	0.133251183500	0.064891288921	0.0363656522135
0.7	0.39771827225	0.133251183228	0.064891288759	0.0363656521637
0.9	0.39771827223	0.133251183221	0.064891288766	0.0363656520584

- Pour $\alpha = 0.007$:

Valeur de ρ	La moyenne du gain à 5	La moyenne du gain à 10	La moyenne du gain à 25	La moyenne du gain à 50
0.1	0.39767689683	0.133237360794	0.064935325527	0.0363925890826
0.3	0.39767689665	0.133237360733	0.064935325491	0.0363925890976
0.5	0.39767689730	0.133237360952	0.064935325654	0.0363925891798
0.7	0.39767689717	0.133237360907	0.064935325645	0.0363925892396
0.9	0.39767689696	0.133237360838	0.064935325572	0.0363925890839

- Pour $\alpha = 0.009$:

Valeur de ρ	La moyenne du gain à 5	La moyenne du gain à 10	La moyenne du gain à 25	La moyenne du gain à 50
0.1	0.39763568280	0.133223592286	0.064979239041	0.0364194572359
0.3	0.39763568196	0.133223592002	0.064979238901	0.0364194570984
0.5	0.39763568203	0.133223592026	0.064979238880	0.0364194570169
0.7	0.39763568294	0.133223592331	0.064979239091	0.0364194572838
0.9	0.39763568260	0.133223592218	0.064979239032	0.0364194571554

Après avoir examiné les résultats nous avons conclu que $\alpha = 0.001$ et $\rho = 0.7$ est la configuration qui nous donne les meilleures valeurs de gains.

5. Tests et résultats :

Après avoir fixé les valeurs des paramètres α et ρ , nous avons effectués nos expérimentations sur la collection de documents présentés dans le tableau 4.3 avec un jeu de requêtes présentés dans le tableau 4.1. Nous avons également effectuées les mêmes tests sur un système différent du notre à savoir le système XFIRM afin de vérifier la cohérence des résultats obtenus. Nous avons par la suite comparé notre modèle à un autre modèle probabiliste, proposé et évalué par Melle Belkacem Thiziri[] une étudiante de master dans le cadre de son projet de fin d'étude.

En effet le modèle proposé par Belkacem est sur modèle de recherche dans les documents XML basé sur un modèle de langue qui prend en compte la taille d'un élément, la taille d'un document en nombre d'éléments le constituant et la taille de la collection en nombre d'éléments.

Les résultats obtenus sont présentés ci-après:

- Les résultats obtenus pour le système XFIRM :

Requête \ Gain	Gain	Gain à 5	Gain à 10	Gain à 25	Gain à 50
289		0.36797	0.12397	0.05778	0.03023
290		0.60411	0.23964	0.12739	0.07775
295		0.40824	0.15318	0.07372	0.03898
297		0.50477	0.21346	0.11045	0.05830
321		0.63571	0.23271	0.12725	0.069530
322		0.67522	0.25722	0.12906	0.06941
323		0.33505	0.11225	0.05208	0.02722
324		0.36292	0.12378	0.05831	0.03055
325		0.70369	0.26034	0.12965	0.06833
327		0.60822	0.23544	0.12568	0.06730
328		0.38126	0.12929	0.06096	0.03205
362		0.54916	0.19030	0.09532	0.05227
364		0.64507	0.31998	0.34118	0.29901

365	0.43273	0.17188	0.10611	0.05774
366	0.58469	0.21195	0.10621	0.05745
367	0.44465	0.15308	0.07429	0.03926
393	0.47278	0.17107	0.08973	0.04872
408	0.60835	0.23324	0.14181	0.08681
410	0.45752	0.17556	0.09701	0.05409
413	0.64888	0.25607	0.14534	0.08611
La moyenne	0.52155	0.19822	0.11247	0.06755

Tableau 4.4 : Les résultats de test obtenus pour le système XFIRM

Voici l'histogramme associé :

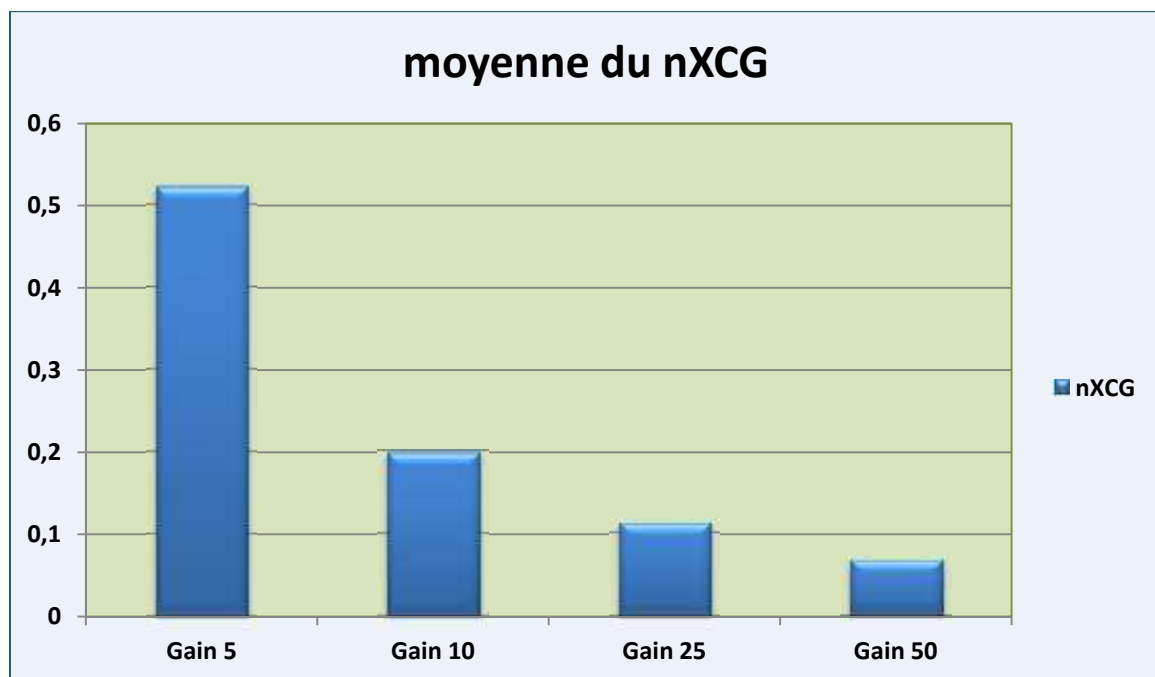


Figure 4.9 : L'histogramme associé aux valeurs du gain pour les différents niveaux pour le système XFIRM

- Les résultats obtenus par le modèle de BELKACEM Thiziri :

Requête \ Gain	Gain à 5	Gain à 10	Gain à 25	Gain à 50
289	0.88885	0.42375	0.36727	0.31054

290	0.88203	0.41243	0.27368	0.16782
295	0.98779	0.54145	0.54619	0.44362
297	0.92249	0.47197	0.49806	0.45314
321	0.88545	0.45121	0.47225	0.40077
322	0.87796	0.42614	0.41614	0.39219
323	0.79078	0.31713	0.18170	0.10812
324	0.97754	0.51608	0.48031	0.40376
325	0.95183	0.49412	0.37576	0.21860
327	0.99155	0.53353	0.45880	0.36502
328	0.85660	0.38572	0.28185	0.18329
362	0.94555	0.50263	0.51589	0.45183
364	0.93064	0.48351	0.50430	0.44836
365	0.95067	0.49943	0.46518	0.38192
366	0.67806	0.29945	0.18137	0.11526
367	1	0.60024	0.56116	0.55533
393	0.93967	0.48133	0.43653	0.37080
408	0.99145	0.52034	0.45976	0.32450
410	0.98779	0.57840	0.53628	0.51469
413	0.92344	0.49047	0.42967	0.27896
La moyenne	0.91826	0.46741	0.42617	0.34443

Tableau 4.5 : Les résultats de test obtenus pour le modèle de BELKACEM Thiziri

Et voici l'histogramme associé :

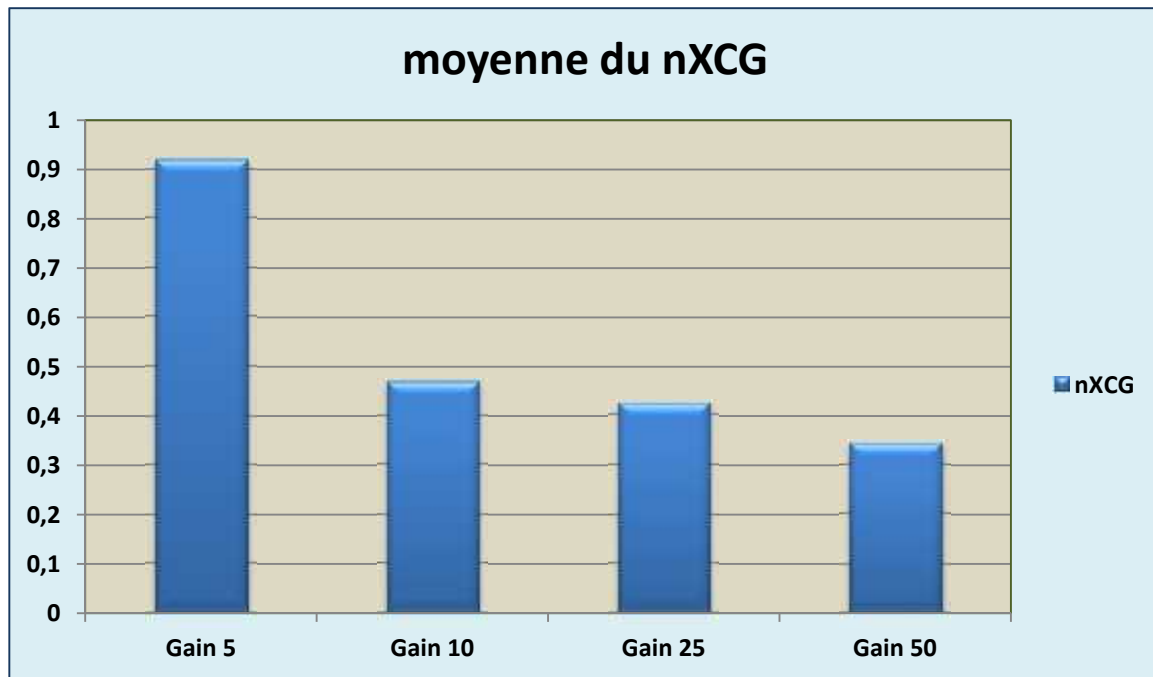


Figure 4.10 : L'histogramme associé aux valeurs du gain pour les différents niveaux pour le modèle de BELKACEM Thiziri

- Les résultats obtenus par notre approche :

Requête \ Gain	Gain à 5	Gain à 10	Gain à 25	Gain à 50
289	0.89660	0.52483	0.54729	0.39758
290	0.96540	0.50845	0.48466	0.44017
295	0.99464	0.52100	0.50684	0.39571
297	0.99830	0.55662	0.67041	0.70042
321	0.96540	0.50428	0.50378	0.45027
322	0.97411	0.49255	0.38951	0.28849
323	0.88677	0.35283	0.21882	0.14783
324	0.93036	0.46651	0.32099	0.21207
325	0.99658	0.55470	0.64610	0.64777
327	0.85642	0.40594	0.37331	0.30814
328	0.96540	0.50387	0.47020	0.43261

362	0.87907	0.47677	0.45232	0.36465
364	0.98563	0.53537	0.60128	0.59792
365	0.85369	0.40502	0.33958	0.28280
366	0.65650	0.31749	0.32811	0.32504
367	1	0.56115	0.64764	0.50614
393	0.96540	0.51732	0.48671	0.44124
408	0.96733	0.52221	0.50051	0.44846
410	0.97238	0.52811	0.61997	0.64251
413	0.96222	0.51814	0.60364	0.53771
La moyenne	0.93386	0.48866	0.48558	0.42838

Tableau 4.6 : Les résultats de test obtenus par le modèle bayésien

L'histogramme associé :

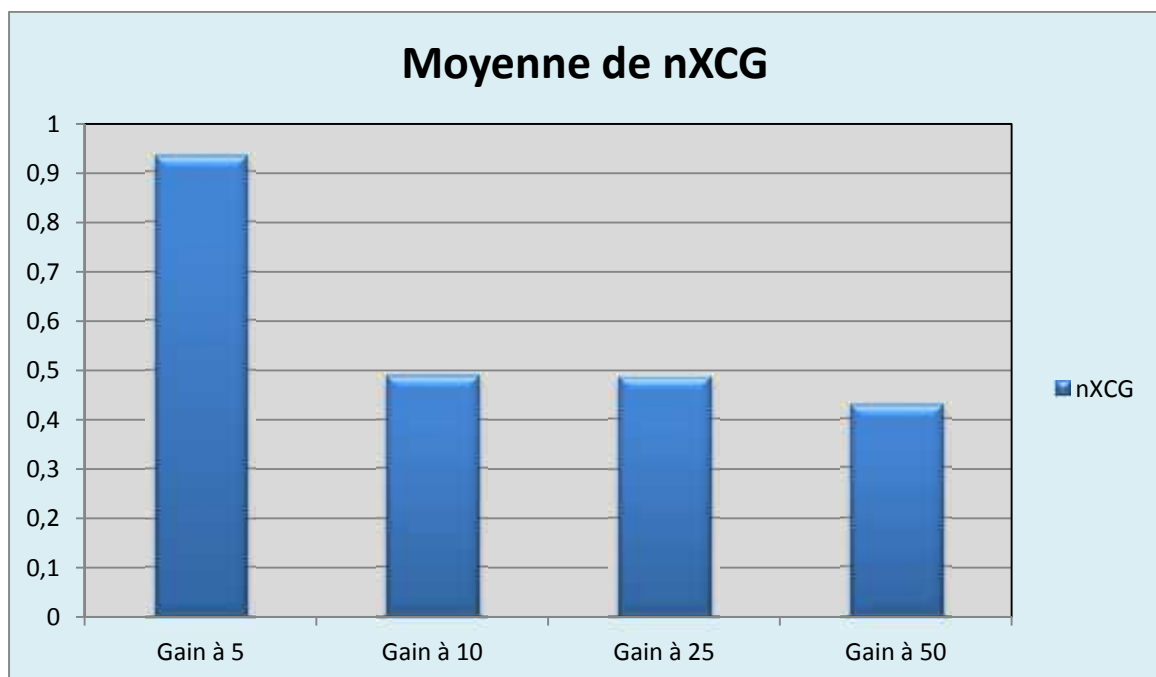


Figure 4.11 : L'histogramme associé aux valeurs du gain pour les différents niveaux pour notre modèle

Tableau comparatif :

Niveau Système	Gain à 5	Gain à 10	Gain à 25	Gain à 50
XFIRM	0.52155	0.19822	0.11247	0.06755
Modèle de langue	0.91826	0.46741	0.42617	0.34443
Modèle bayésiens	0.93386	0.48866	0.48558	0.42838
Taux d'amélioration de notre modèle par rapport à XFIRM	+41.231%	+29.044%	+37.311%	+36.083%
Taux d'amélioration de notre modèle par rapport au modèle de langue	+1.56%	+2.125%	+5.941%	+8.395%

Tableau 4.7 : tableau comparatif des résultats obtenu

Voici l'histogramme associé :

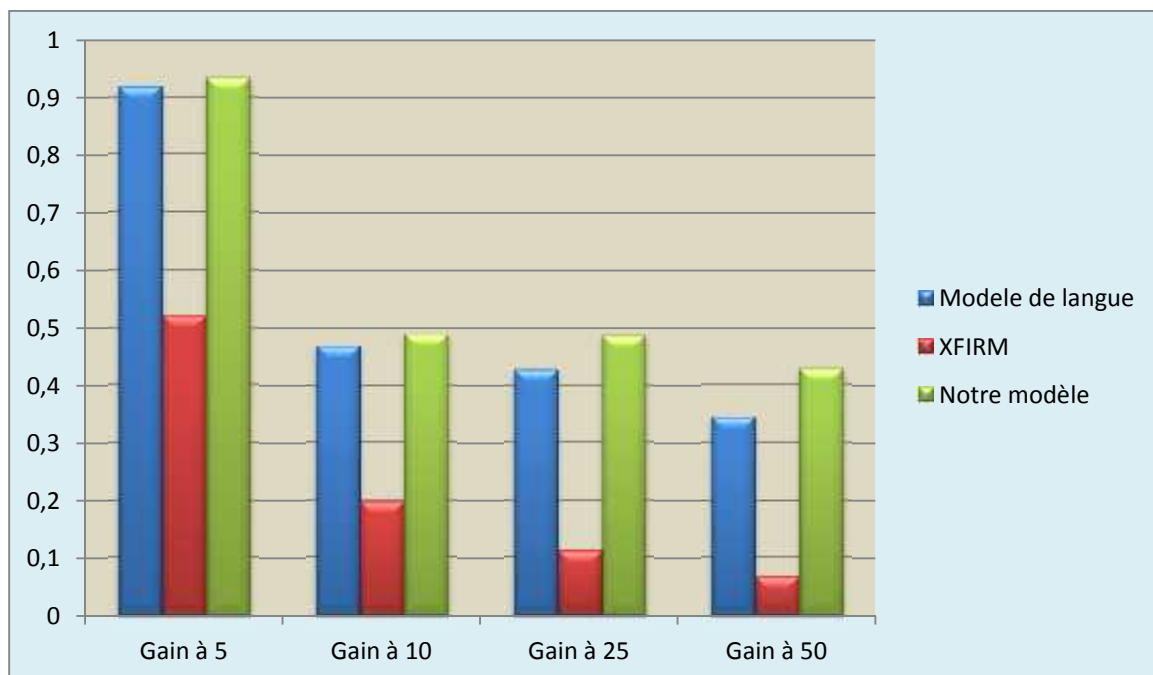


Figure 4.12 : L'histogramme associé au tableau comparatif

Description :

Nous avons présenté ci-dessous les résultats obtenus par notre approche basée sur les réseaux bayésiens avec ceux obtenus par le système XFIRM, ainsi que ceux obtenus par le modèle de langue de BELKACEM. L'objectif de la comparaison par rapport à XFIRM est de pouvoir juger des qualités des résultats obtenus par rapport à un modèle qui a prouvé son efficacité

dans la RIS. Tant dis que la comparaison avec le modèle de langue étant donné que ce sont deux modèles probabilistes, nous avons jugé intéressant d'établir une comparaison.

Le tableau et l'histogramme associé ci-dessus, montrent les différences entre les gains obtenues sur cinq (05) niveaux (les 5, les 10, les 25 et les 50 premiers résultats).

Par rapport au système XFIRM, on remarque que sur les 5 premiers résultats, notre modèle atteint une amélioration de **41.2%** ainsi que plus de **29%** d'amélioration par rapport aux 10 premiers résultats et jusqu'à plus de **36%** sur les 50

Par rapport au modèle de langue, nous avons aussi remarqué une petite amélioration. **1.56%** sur les 5 premiers résultats, **2.125%** sur les 10 premiers résultats et jusqu'à **8.395%** sur les 50.

Nous concluons que notre modèle présente une nette amélioration par rapport au système XFIRM. Une légère amélioration par rapport au modèle proposé.

Conclusion :

Dans ce chapitre nous avons présenté notre démarche d'implémentation ainsi que les résultats de nos tests, ensuite nous les avons comparés avec le modèle XFIRM et un modèle de langue. Nous avons constaté que notre modèle a présenté une amélioration du gain sur les différents niveaux considérés, et par rapport aux deux modèles.

Conclusion

Conclusion générale :

Le travail présenté dans ce mémoire nous a donné l'opportunité de nous familiariser avec le domaine de la recherche d'information. Nous nous sommes principalement intéressés à la recherche d'information structurée basée sur les réseaux bayésiens.

L'objectif de notre contribution était d'implémenter et de tester une approche bayésienne proposée par [Djebbar et lamrous, 2013] qui s'adapte à la nouvelle granularité d'information présentée en recherche d'information dans les documents XML. Dans ce contexte, le résultat peut être soit un élément ou un ensemble d'éléments ou encore un document tout entier.

Cette approche a pour objectif de remédier à l'un des inconvénients des réseaux bayésiens, qui est la réduction du temps d'inférence dans le réseau, en diminuant le nombre de configuration à estimer avant d'effectuer le calcul de pertinence.

Après avoir implémenté l'approche, nous avons réalisé des expérimentations préliminaires sur une mini collection documentaire afin de fixer les valeurs des paramètres utilisés dans les formules avant de procéder aux tests finaux. Les tests finaux ont été menés sur un ensemble de requêtes et de documents de la campagne d'évaluation INEX 2006. Nous avons voulu suite à cela vérifier la cohérence de nos résultats, en les comparant à ceux obtenus par le système XFIRM pour le même protocole de test. Le système XFIRM est certes un système basé sur une propagation des scores mais il reste un système qui est à notre portée d'utilisation et c'est un système qui a fait ses preuves dans le cadre de la RI dans les documents XML. Rassurés par la cohérence de nos résultats, nous les avons comparés aux résultats d'un autre modèle probabiliste, à savoir le modèle de langue proposé, implémenté et testé par notre camarade Belkacem Thiziri dans le cadre de son projet de fin d'études en master pour la recherche dans les documents XML.

Les résultats ont révélés que l'approche proposée par [Djebbar et lamrous, 2013] présente une nette amélioration de près de **41%** sur le gain à 5, de plus de **29%** sur le gain à 10 et de **36%** sur le gain à 50 par rapport au modèle XFIRM. Ainsi qu'une intéressante amélioration par rapport au modèle de langue de Belkacem de **1.56%** sur le gain à 5 de **2.125%** sur le gain à 10 de **8.395%** sur le gain à 50.

Nous pouvons en définitif conclure que le modèle bayésien proposé par [Djebbar et lamrous, 2013] a abouti à des résultats satisfaisants.

Perspectives :

Les perspectives à entreprendre pour améliorer ce travail se résument comme suit :

- Tester l'approche proposée en utilisant toute la collection INEX 2006 sur toutes les requêtes et évaluer les résultats.
- Etablissement des comparaisons de résultats avec d'autres modèles basés sur les réseaux bayésiens
- Tester l'approche sur des requêtes orientées CAS (content and structure)

- Exploiter la probabilité à priori $P(D_i)$ en proposant une probabilité utile qui permettra d'ajouter une plus grande valeur à la pertinence

Annexe A

1.1. Méthodes d'indexation et formules de pondération :

1.1.1. Indexation :

Définition :

La phase d'indexation est une étape importante dans la RI, en effet elle permet de créer une représentation facile et manipulable des documents et des requêtes. Dans ce cas extraire d'un document ou d'une requête les termes représentatifs afin d'en produire des mots clé (appelé aussi descripteur) et cela d'une manière à pouvoir distinguer les documents pertinents à une requête des documents non pertinents.

L'indexation peut être :

Manuelle : l'extraction des termes et leurs descriptions est réalisé par un spécialiste (documentaliste).

Avantage :

- Elle permet d'assurer une meilleur pertinence dans les réponses apporté par le système de RI.

Inconvénients :

- Très couteuse en temps.
- Très difficile à appliquer sur un gros volume de données.
- deux indexeurs différents peuvent présenter des termes différents pour caractériser un même document, et un indexeur à deux moments différents peut présenter deux termes distincts pour représenter le même concept [Sauvagnat,2005].

Semi-automatique : l'indexation se fait d'une manière automatique, cependant le documentaliste se charge du choix final de ceux qu'il juge nécessaire.

Automatique : le processus d'indexation est complètement informatisé ie automatique c'est l'indexation la plus utilisé et la plus rapide.

L'indexation automatique regroupe un ensemble de traitements automatisés sur un document. On distingue :

- Extraction des termes.
- Elimination des mots vides.
- La lemmatisation (radicalisation ou normalisation).
- Le repérage de groupe de mots.
- La pondération des mots.
- La création de l'index.

. **Extraction des termes** : L'analyse lexicale constitue la première étape du processus d'indexation. Elle consiste à extraire d'un document un ensemble de termes ou de mots simples. L'identification de ces termes se fait en reconnaissant les espaces de séparation des mots, les caractères spéciaux, les chiffres, les ponctuations etc.

.**Elimination des mots vide** : les mots vides sont les mots les moins significatifs qui sont éliminés lors du processus d'indexation. On distingue deux techniques pour le filtrage :

- L'utilisation d'une liste prédéfinie de mots vides (aussi appelée anti-dictionnaire ou stop-List),

- Le comptage du nombre d'apparition d'un mot dans un document de la collection. (si le mot contient une fréquence qui dépasse un certain seuil on le supprime et il devient alors un mot vide)

L'élimination de ces mots permet de :

- réduire l'index, on gagne alors en espace mémoire, et en réduction de temps d'exécution.

.Radicalisation : ou lemmatisation consiste à ne garder des mots que leurs racines. Un mot peut se présenter sous différentes formes comme les mots de la même famille. Cette étape consiste à éviter à l'utilisateur d'entrer le même mot sous différentes formes lors d'une recherche. Exemple : jardin, jardinier, jardinage... dans ce cas il n'est pas utile d'indexer tous les mots un seul suffira pour représenter le concept voulu.

.pondération des termes : permet de mesurer par le calcul d'un poids le degré d'importance d'un terme dans un document.

D'après [Salton and Buckley,1997] Il existe plusieurs techniques pour la pondération des termes. Les plus courantes sont basées sur les notions de fréquence des termes dans un document (*tf* pour *term frequency* mesure l'importance local du terme) et de fréquence des termes sur l'ensemble des documents de la collection étudiée (*idf* pour *inverse document frequency* mesure l'importance globale du terme).

[Zipf, 1949] et [Luhn, 1957] établissent une relation entre la fréquence des termes et leurs pouvoirs discriminant ie leurs importances dans la participation au sens du texte $tf \times idf$.

Approches basées sur la fréquence locale (tf) : est souvent exprimé selon l'une des formules suivantes :

tf = Occurrence d'un terme dans un document

$$tf = \log(1 + tf)$$

Approches basées sur la fréquence globale (idf) : est souvent exprimée selon l'une des formules suivante : $idf = \log \frac{|D|}{|d_j: t_i \in d_j|}$ ou $idf = \log \frac{|D - d_j|}{|d_j: t_i \in d_j|}$

Avec :

$|D|$: Le nombre total de documents dans le corpus.

$|d_j: t_i \in d_j|$: Le nombre de documents où le terme t_i apparaît.

Approches basées sur $tf \times idf$: consiste à multiplier les deux mesures *tf* et *idf* comme suit :

$$tf \times idf = \log 1 + tf * \log \frac{|D|}{|d_j: t_i \in d_j|}$$

-la création de l'index : est le processus de stockage des informations recueillis lors de l'indexation. Plusieurs solutions de stockage ont été proposées, nous citons :

- les fichiers séquentiels indexés
- les fichiers inverses (inverted files)
- les fichiers de signatures (signature files)
- les tableaux de suffixes (suffix arrays)

Pour plus de détails sur ces solutions voir Annexe.

1.2. L'appariement document requête :

L'appariement document requête est la comparaison entre les termes de la requête et ceux d'un document. Pour cela elle s'appuie sur un formalisme précis défini par un modèle de RI.

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 censés 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.

Modèle de RI :

Il existe différents types de modèles de système de recherche et ils diffèrent par rapport à la façon dont les informations sont représentées et aussi de la façon dont le besoin en information de l'utilisateur a été présenté. Un modèle de la RI modélise la fonction d'appariement. On peut distinguer trois grandes classes de modèle :

1. **Ensemblistes** : sont basés sur la théorie des ensembles, ce sont les plus simples et les premiers à avoir été mis en place.
2. **Algébriques** : ils mesurent les distances entre la représentation de la pertinence d'un document vis-à-vis d'une requête dans un espace vectoriel.
3. **Probabilistes** : qui sont basées sur la théorie des probabilités, et estiment des probabilités de pertinence d'un document en fonction de la requête.

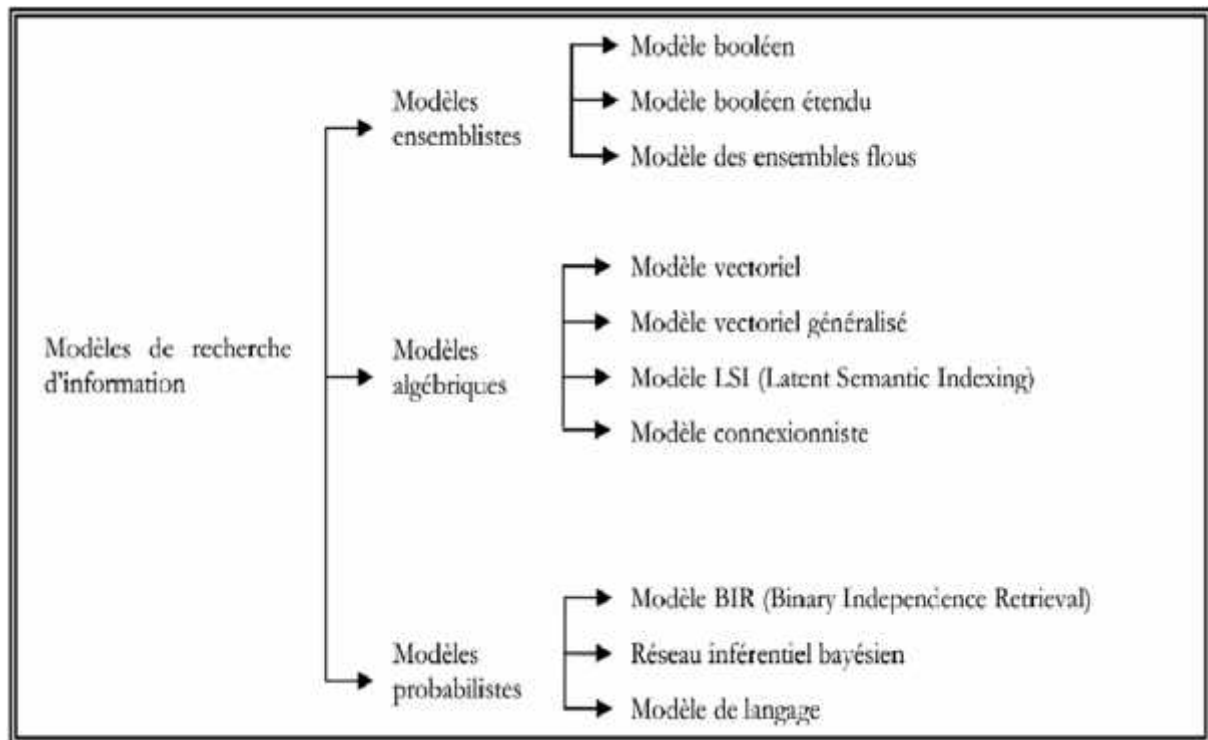


Figure 1.7: Taxonomie des modèles de recherche d'information [Mataoui, 2007]

Dans ce qui suit nous avons présentés quelques modèles de recherche d'information parmi ceux représentés dans la figure 1.7.

1.2.1. Le modèle booléen :

1.4.1.1. Le modèle booléen de base:

Dans ce modèle un document est représenté par un ensemble de termes. Une requête est une expression logique composée de termes assemblés par les opérateurs : ET ($\wedge$), OU ($\vee$) et NON ($\neg$).

La pertinence entre une requête et un document (noté $RSV(Q,D)$ (Retrieval Status Value)) est déterminée de la façon suivante:

- $RSV(t_i, D) = 1$ si $t_i \in d$; 0 sinon. (la requête contient un seul terme t_i) ;
- $RSV(Q_1 \wedge Q_2, D) = 1$ si $RSV(Q_1, D) = 1$ et $RSV(Q_2, D) = 1$; 0 sinon. (la requête contient deux termes reliés par l'opérateur ET);
- $RSV(Q_1 \vee Q_2, D) = 1$ si $RSV(Q_1, D) = 1$ ou $RSV(Q_2, D) = 1$; 0 sinon. (la requête contient deux termes reliés par l'opérateur OU);
- $RSV(\neg Q_1, D) = 1$ si $RSV(Q_1, D) = 0$; 0 sinon. (la requête est composée de la négation d'un seul terme) ;

Avantage :

- Ce modèle est très simple à mettre en œuvre.

Inconvénients :

- Le système retourne un ensemble de documents non ordonnés comme réponse à une requête. Cependant on ne peut pas déterminer quel document est plus pertinent qu'un autre.
- Ce modèle repose sur un critère exclusivement binaire : un document est soit pertinent soit non pertinent. Il ne prend pas aussi en considération la pondération des termes : un mot a un poids a 1 si il apparait dans le document et 0 sinon.
- La formulation des requêtes est complexe, non accessible à un large public.

Le modèle booléen étendu :

Le modèle booléen étendu appelé aussi modèle PNorm, a été introduit en 1983 par Salton et al [Salton & al, 1983]. Ce modèle peut être vu comme une combinaison entre le modèle vectoriel (qui sera présenté dans la section suivante) et le modèle booléen. Ce modèle consiste à calculer la distance entre les coordonnées d'un document et les coordonnées d'une requête.

Considérons un ensemble de termes $t_1, \dots, t_n$, et soit d_{ij} le poids du terme t_i dans le document $D_j = (d_{1j}, \dots, d_{nj})$, avec $1 \leq i \leq N$ et $0 \leq d_{ij} \leq 1$. La similarité entre le document D_j et une requête Q_k décrite sous une forme conjonctive ou disjonctive est donnée comme suit :

$$\text{Opérateur OR : } \text{RSV}(D_j, Q_k) = \left(\frac{\sum_{i=1}^N q_{ik}^p d_{ij}^p}{\sum_{i=1}^N q_{ik}^p} \right)^{1/p}$$

$$\text{Opérateur AND : } \text{RSV}(D_j, Q_k) = 1 - \left(\frac{\sum_{i=1}^N q_{ik}^p (1 - d_{ij}^p)}{\sum_{i=1}^N q_{ik}^p} \right)^{1/p}$$

Où

- P une constante $0 \leq P \leq \infty$,
- q_{ik} le poids du terme t_i dans la requête Q_k .

Remarque :

Lorsque $p=1$, il n'y a aucune distinction entre les deux connecteurs ET et OU. En effet on aura :

$$\left| \frac{\sum_{i=1}^N q_{ik}^p d_{ij}^p}{\sum_{i=1}^N q_{ik}^p} \right|^{1/p} = \left| \frac{\sum_{i=1}^N q_{ik}^p (1 - d_{ij}^p)}{\sum_{i=1}^N q_{ik}^p} \right|^{1/p}$$

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

1.2.2. Modèle Vectoriel :

i. Le Modèle Vectoriel :

Dans ce modèle la pertinence d'un document vis-à-vis d'une requête est défini par des mesures de distance dans un espace vectoriel. Les documents et les requêtes sont représenté dans un espace vectoriel à N dimensions (N : nombre de termes($t_1, t_2, t_3, \dots, t_n$)) par des vecteurs et sont représenté comme suit :

$$D_j = \begin{bmatrix} d_{1j} \\ d_{2j} \\ \vdots \\ d_{Nj} \end{bmatrix} \quad Q_k = \begin{bmatrix} q_{1k} \\ q_{2k} \\ \vdots \\ q_{Nk} \end{bmatrix}$$

Où

d_{ij} : est le poids du terme t_i dans le document D_j .

q_{ik} : est le poids du terme t_i dans la requête Q_k .

Le modèle vectoriel estime le degré de pertinence entre un document et une requête par un degré de corrélation entre leurs vecteurs associés. Cette corrélation peut être spécifiée par le calcul de similarité entre vecteurs.

Les types de mesures les plus répandus sont :

- Le produit scalaire :

$$RSV(D_j, Q_k) = \sum_{i=1}^N d_{ij} q_{ik}$$

- Coefficient de Dice :

$$RSV(D_j, Q_k) = \frac{2 \sum_{i=1}^N d_{ij} q_{ik}}{\sum_{i=1}^N (d_{ij}^2 + q_{ik}^2)}$$

- Mesure de cosinus :

$$RSV(D_j, Q_k) = \frac{\sum_{i=1}^N d_{ij} q_{ik}}{\sqrt{\sum_{i=1}^N d_{ij}^2 * \sum_{i=1}^N q_{ik}^2}}$$

- Mesure de Jaccard :

$$RSV(D_j, Q_k) = \frac{\sum_{i=1}^N d_{ij} q_{ik}}{\sum_{i=1}^N d_{ij}^2 + \sum_{i=1}^N q_{ik}^2 - \sum_{i=1}^N d_{ij} q_{ik}}$$

Avantage :

- Le modèle vectoriel permet de pallier à l'un des inconvénients majeurs du modèle booléen, en permettant de trier les documents répondant à une requête (les plus pertinents).
- Les documents dans le modèle vectoriel, sont en effet restitués dans un ordre décroissant de leur degré de similarité avec la requête. Plus le degré de similarité d'un document est élevé, plus le document ressemble à la requête, et donc pertinent pour l'utilisateur.

Inconvénients :

- L'approche vectorielle considère tous les termes comme étant indépendants, et ne tient donc pas compte des liens entre eux-ci.
 - Indépendance mutuelle des termes : chaque terme est représenté par un vecteur dans un espace vectoriel, dont les axes sont orthogonaux par construction : les axes sont les produits logiques des termes d'indexation. Un document est représenté, par la moyenne des vecteurs représentant les termes qu'il contient.
- **Le modèle connexionniste (basé sur les réseaux de neurones) :**

L'objectif du modèle connexionniste (réseaux de neurones), est d'imiter les fonctionnalités du cerveau humain.

Un réseau de neurones en recherche d'information est généralement composé de plusieurs couches dont :

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

Le mécanisme d'appariement des documents et des requêtes se compose de deux phases d'activation qui sont :

- Phase une d'activation : la requête se propage vers les nœuds (termes) de la collection ceci fait l'activation initial des neurones, ensuite les nœuds de la collection envoient des signaux aux nœuds document à leur tour à travers les différentes connexions du réseau et enfin les documents qui ont reçus le plus de signaux sont considérés comme les plus pertinents.
- Phase deux d'activation : c'est un traitement itératif de propagation du signal qui peut compléter la phase une d'activation : les nœuds des documents considérés comme étant pertinents génèrent de nouveaux signaux en direction des nœuds (termes : qui correspondent à des mots considérés comme très représentatifs du contenu des documents en question) de la collection et ces nœuds envoient à leur tour de nouveaux signaux vers les nœuds documents.

Cette seconde phase permet de retrouver les documents dont les termes ne sont pas nécessairement contenus dans la requête.

La figure 1.8 suivante représente un modèle de réseaux de neurones pour la recherche d'information.

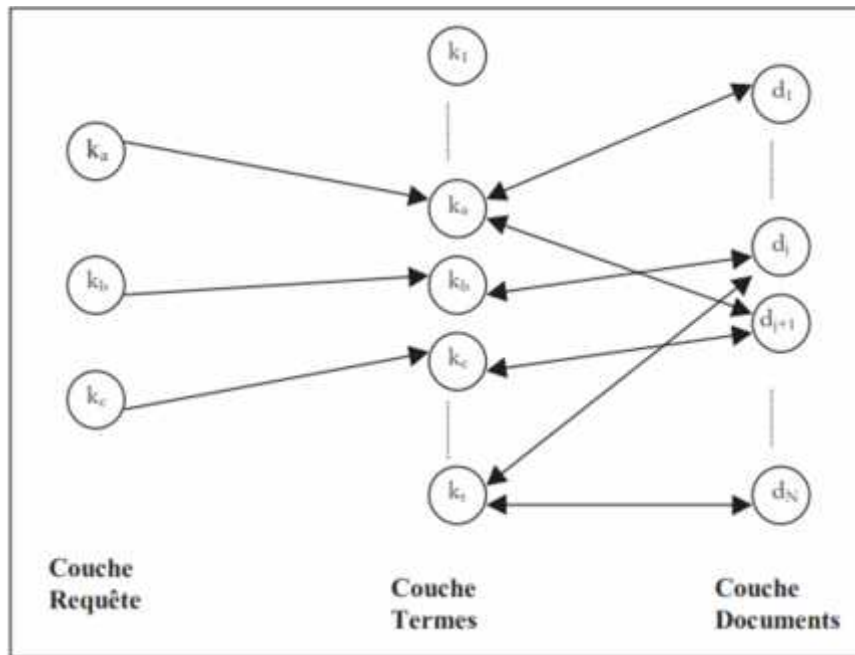


Figure 1.8 : un modèle de réseau de neurones pour la RI [Sauvagnat, 2006]

Avantage :

- Il prend en considération l'indépendance entre les éléments d'informations.

Inconvénients :

- Interprétation difficile des résultats par l'utilisateur, en effet celui-ci ne comprend pas pourquoi tel document lui a été retourné.

1.2.3. Modèle probabiliste :

Le premier modèle a été proposé par Maron et Kuhn [Maron & Kuhns, 1960]. Ce modèle utilise un modèle mathématique fondé sur la théorie de la probabilité. Le principe de base consiste à retrouver les documents qui ont au même temps une forte probabilité d'être pertinent, et une faible probabilité d'être non pertinent.

Prenons une requête utilisateur q_i et un document d_j , il s'agit donc de calculer la probabilité de pertinence du document pour cette requête.

Deux événements seront présentés :

- R : d_j est pertinent pour q_i
- NR : d_j n'est pas pertinent pour q_i

Le score noté $RSV(D, Q)$ d'appariement entre le document D et la requête Q est donné par :

$$RSV(q_i, d_j) = \frac{P(R|d_j)}{P(NR|d_j)}$$

En utilisant la règle de Bayes :

$$RSV_{q_i, d_j} = \frac{P_{d_j} R}{P_{d_j} NR}$$

Cela vient à ordonner les documents.

Un document sera décomposé en un ensemble d'événement $d_j(t_1, t_2, \dots, t_N)$ afin d'estimer les probabilités $P_{R d_j}$ et $P_{NR d_j}$. Chaque événement dénotera la présence ou l'absence d'un terme t_i dans n document d_j .

Supposons l'indépendance des termes des documents, la formule de Bayes devient :

$$RSV_{q_i, d_j} = \frac{N}{i=1} \log \frac{P_{w_{ij}} R}{P_{w_{ij}} NR}$$

Où :

- w_{ij} : indique la présence ou l'absence d'un terme t_i dans le document d_j .

Avantage :

- Les documents jugés pertinents seront triée par ordre de leurs pertinences.

Inconvénients :

- Difficulté de calcul des probabilités conditionnelles
- Le modèle ne prend pas en compte des dépendances entre les termes d'indexation.

- Les modèles de langage :

Le modèle de langage utilise une méthode différente des autres modèles, en effet les autres modèles cherche a comparé une représentation de la requête de l'utilisateur avec une représentation du document recherché.

Le but est de générer des requêtes à partir des documents et les comparer à celle posé par l'utilisateur.

La pertinence d'un document est estimée en calculant la probabilité que la requête utilisateur soit inférée par celui-ci. La probabilité (notée

$P(Q_k/D)$) que la requête soit inférée par le document, en supposant que les termes de la requête sont indépendants, est : [Ponte, 1998]

Etant donnée la requête Q_k exprimée par une suite de termes $q_{1k}, q_{2k}, \dots, q_{nk}$ et que les termes $q_{n+1k}, \dots, q_{1k}$ sont absents dans la requête. La probabilité peut être exprimée comme suit :

$$P_{Q_k D_j} = \frac{P_{q_{ik} D_j}}{q_{ik} \square \bar{Q}_k} \times \frac{1 - P_{q_{ik} D_j}}{q_{ik} \square \bar{Q}_k}$$

- Le modèle bayésien :

Les réseaux bayésiens ont vu le jour dans les années 90. Les réseaux bayésiens se divisent en deux catégories :

- Les réseaux d'inférence : développé par [Turtle et Croft, 1990], qui utilise un graphe de dépendance orienté et acyclique.
- Les réseaux de croyance : développé par [Ribeiro-Neto et al., 1996]

Ce modèle sera détaillé dans le chapitre III

1.3. Paramètres d'évaluation d'un SRI :

D'après [C.Cleverdon et al.,1997] et [D.Hull, 1993] l'évaluation des SRI est un problème majeur sur lequel la communauté de la RI a investi beaucoup d'efforts. L'intérêt est de pouvoir comparer des SRI entre eux à l'aide des critères objectifs (les réponses idéales que l'utilisateur souhaite recevoir).

Plusieurs quantités mesurables ont été proposées pour l'évaluation d'un SRI : le temps de réponse, la pertinence, la qualité et la présentation des résultats, la facilité d'utilisation. . .

Le plus important parmi ces qualités est la pertinence des résultats retournés, en effet la mesure de la capacité du système à satisfaire le besoin en information de l'utilisateur.

Quatre facteurs permettent d'évaluer ce critère :

- Le rappel
- La précision
- Le bruit
- Le silence

Le rappel est une mesure du pourcentage des documents pertinents ayant été retrouvée parmi tous les documents pertinents dans la base (collection).

$$\text{Rappel} = \frac{DPR}{DPC}$$

DPR : désigne le nombre de documents pertinent retrouvés.

DPC : désigne le nombre de documents pertinents dans la collection

Parallèlement la précision calcule le pourcentage de documents pertinents parmi les documents retrouvés, ou le bruit du résultat, et mesure donc la spécificité du système. Elle est calculée comme suit :

$$\text{Précision} = \frac{DPR}{DR}$$

ND : désigne le nombre total de documents retrouvés.

Une des façons d'évaluer un système est de tracer une courbe de précision/rappel.

Ainsi, si le résultat de recherche dépend d'un certain paramètre, par exemple le rang d'un document restitué, alors pour chaque valeur du paramètre les valeurs de rappel et précision peuvent être calculées.

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

$$\text{bruit} = \frac{DNPR}{DR}$$

DNPR : désigne le nombre de documents non pertinents retrouvés

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

$$\text{silence} = \frac{DPNR}{DR}$$

DPNR : désigne le nombre de documents pertinents non retrouvés

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 sont 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 :

$$\textbf{Rappel} + \textbf{silence} = \textbf{1}$$

$$\textbf{Précision} + \textbf{bruit} = \textbf{1}$$

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.

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 (Text Retrieval Conference) [HAR, 92] qui constituent la référence en ce qui concerne l'évaluation des SRI.

Conclusion :

Dans ce chapitre, nous avons présenté les concepts de base de 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.

Annexe B

1.1. Les liens dans les documents XML :

XLINK :

Xlink permet la construction de liens XML semblables aux liens HTML mais avec davantage de puissance et d'extensibilité.

Les liens HTML possèdent les caractéristiques suivantes [Tittel, 2003] :

- ils sont unidirectionnels.
- les liens sont intégrés dans le document
- les liens peuvent cibler une portion du document mais le document entier est accédé et non simplement le fragment requis.
- Les liens HTML ne peuvent se référer qu'à deux ressources. La première est le document appelant (qui contient l'hyperlien) et la seconde le document cible.

Xlink a résolu chacune de ces limitations de la manière suivante :

- des directions de traversée multiples peuvent être spécifiées entre ressource.
- Un hyperlien peut être spécifié comme partie d'un document externe.
- En utilisant Xpointer avec Xlink, un fragment de document peut être lié
- Il est possible de définir des liens qui incluent de multiples ressources.

Pour remplir les fonctions mentionnées ci-dessus, Xlink utilise un certain nombre d'attributs qui sont :

- type indique le type d'élément Xlink crée, ce qui est obligatoire pour les liens ;
- href appelé attribut localisateur, indique l'URI utilisé pour extraire une ressource ;
- role spécifie la ressource dans un format lisible par une machine ;
- title spécifie la ressource dans un format lisible par l'homme, utilisé pour la création d'info bulle;
- show précise la façon dont la ressource doit être affichée une fois extraite, elle peut remplacer (replace) celle existante, s'ajouter à l'existante (embed) ou simplement être vue ailleurs, dans une nouvelle fenêtre (new)
- actuate : permet de définir si le lien doit être activé au moment du chargement (onLoad) ou sur demande (onRequest).
- from et to indiquent les directions des liens.

Deux types de liens Xlink sont définis : les liens simples (simple link) et les liens étendus (extended link). Comme pour un lien hypertexte HTML, le lien simple permet de mettre en relation, de façon unidirectionnelle, une source et une destination.

Le lien étendu permet de mettre en relation (arc) un nombre arbitraire de ressources qui peuvent être locales (resource) (à l'endroit où est défini le lien) ou distantes (locator).

Le type arc définit la direction que doit suivre le lien, le type resource utilisé dans un élément de type étendu pour créer des ressources locales et le type locator utilisé dans un élément de type étendu pour indiquer les ressources distantes.

Exemple de lien XLINK étendu :

```

<film xlink:href="affiches.xml" xlink:type="extended">
  <titre xlink:type="arc"
    xlink:from="film:titre"
    xlink:to="film:affiche"
    xlink:show="replace"
    xlink:actuate="onRequest"
    xlink:role="film:obtenirImageAffiche"
    xlink:title="Obtenez une image de l'affiche du film">
    " Le magicien d'Oz "
  </titre>
</film>

```

XPOINTER :

XPointer est utilisé pour adresser des fragments de documents XML. Xpointer est construit sur le langage Xpath, de qui il a adopté les fonctionnalités et la syntaxe. Cependant Xpointer offre des extensions à Xpath comme:

- localiser des informations par correspondances de chaînes;
- Viser non seulement des nœuds entiers mais également des parties de nœuds dans un document XML.
- Compléter les URI (URI#XPOINTER), pour localiser des adresses et étendre ainsi la puissance et la flexibilité des liens.
- Utiliser deux nouveaux types de localisation (tout type de nœud autorisé sous Xpath). le point et l'étendue.
- Utiliser les fonctions : string-range(), range() et range-to() ; here() et origin() ; startpoint() et end-point().

Le point et l'étendue : Le point est une localisation dans un document XML qui peut être défini comme étant d'un seul caractère de longueur [Tittel, 2003]

L'étendue est définie par deux points (un point de début et un point de fin) et elle comprend toute la structure et le contenu XML se trouvant entre ces deux points [Hunter & al, 2001].

Description des fonctions présentées ci dessus :

Start-point : considère un ensemble de localisation comme paramètre et renvoie et un ensemble de localisation contenant les points de départ de tous les emplacements trouvés dans l'ensemble de localisation d'entrée. Par analogie la fonction end-point fournit les points de terminaison utilisés de la même manière avec la même structure d'arguments.

Range() et Range-to() renvoient des étendues, elle prennent un ensemble de localisation comme argument et renvoient un ensemble de localisation comme résultat.

La fonction here() et origin() sont utilisées pour retourner l'élément situé aux points ou le pointeur Xpointer est localisé(here()) ou à l'endroit où la traversée a commencé(origin()).

Exemple : Soit le document Xml suivant et les expressions Xpointer s'y referant :

```
<bibliothequefilms>
  <film>
    <titre>le magicien d'Oz </titre>
    <annee>1939</annee>
    <genre categorie= « comédie musicale » </genre>
  </film>
  <film>
    <titre>la soupe au canard </titre>
    <annee>1933</annee>
    <genre categorie= »comédie » </genre>
  </film>
  <film>
    <titre>Autant en emporte le vent </titre>
    <annee>1939</annee>
    <genre categorie= » drame » </genre>
  </film>
  .....
</bibliothequefilms>
```

- Xpointer (string-range(//genre , « drame »)) : retourne les ensembles de nœuds dans l'enfant de genre contenant la sous chaîne drame, dans ce cas categorie
- Xpointer (/film to /genre) ce code retourne l'étendue entre film et genre

Bibliographie

Bibliographie

[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.

[Alimazighi et al, 2005] : Utilisation des réseaux bayésien pour la recherche d’information structuré Z. Alimazighi, B. Bessaïb, R. Djiroun, 2005.

[Allorge, 2000] : Stéphane Allorge, " XML ", département ASI, 2000.

[Amann, 2003] : Bernd Amann, Cours "XML et les bases de données : Introduction à la gestion de contenus Web et XML", Module Données et services sur le Web 2003/2004.

[Benaouicha, 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.

[Belkacem, 2015] : Belkacem Thiziri. Proposition et expérimentation d’un modèle de langue pour la recherche d’information dans les documents XML. Thèse de master en informatique, Université mouloud mammeri tizi ousou, algérie, juin 2015.

[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

[Braga & al, 2002] : D. Braga, A. Campi, E. Damiani, P. Lanzi, and G. Pasi. FXpath: flexible querying 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.

[Buntine, 1994] : Buntine W., "Representing Learning with graphical Models". Technical Report, FIA 94-14, Artificial Intelligence Research Branch, NASA Ames Research Center, USA, 1994.

[Carmel & al, 2003] : D. Carmel, D. Efraty, G. Landau, Y. Ma'aker, and Y. Mass, "An Extension of the Vectorspace Model for querying XML documents via XML fragments", In R. Baeza Yates, N. Fuhr and Y. S. Ma'arek, editors, XML and Information Retrieval workshop of SIGIR 2002, Tampere, Finland, Aug 2002.

[Crestani et al, 2003] :Crestani, F., de Campos, L. M., Fernandez-Luna, J. M., andHuete, J. F. Rankingstructured documents using utilitytheory 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-layeredBayesian network model for structured document retrieval. In: Symbolic and Quantitative Approaches to ReasoningwithUncertainty, 7th European Conference, ECSQARU 2003, 2003-07-02 - 2003-07-05, Aalborg, Denmark.

[Cui & al, 2003] :H.cui, J-R.Wen, J-R.Chua, "Hierarchicalindexing and flexi elementretrieval for structured document", april 2003.

[De Campos et al, 2003] : de Campos, L., Fernandez-Luna, J., and Huete, J. Improvingthe efficiency of the bayesian network retrieval model by reducingrelationshipsbetweeneterms. International Journal of Uncertainty, Fuzziness and Knowledge-BasedSystems 11, Supplement (2003), 101–116

[Denoyer et al, 2004] :Denoyer, L., Wisniewski, G., and Gallinari, P. Document structure matching for heterogeneouscorpora, 2004. Proc. Of the International ACM-SIGIR Conference : Workshop on XML and Information Retrieval.

[Djebbar et Lamrous, 2013]: Djebbar Sonia et Lamrous fadhila. Proposition d’une approche base sur les réseaux bayésiens pour la recherche d’information dans les documents XML. These de master en informatique. Univeristé mouloud mammeri tizi ouzou, algérie. 2012/2013.

[Fellag, 2006] :Mme Samia FELLAG. Recherche d’information dans les documents semi structuré XML. Thèse de magistère en informatique, Université Mouloud Mammeri, Tiziouzou, septembre 2006.

[Fuhr& al, 2001] : N. Fuhr, K. Grossjohann, "XIRQL, a querylanguage for information retrieval in XML documents", In proceedings of SIGIR 2001, Toronto Canada 2001.

[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. Pittsburgh, 1993.

[Geva, 2005] : S.Geva. GPX-gardenspointXMLIRatINEX2005. In INEX’05, pages 240–253, 2005.

[Gövert& al, 2002] :N. Gövert, M. Abolhassanni, N. Fuhr, K. Grossjohann, “Content-Oriented XML RetrievalwithHyreX”. In INEX 2002 Workshop Proceedings, p. 26-32,Germany, 2002.

[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

[Hatano & al, 2002] : K.Hatano, H.Kinutani, M.Yoshikawa, and S.Uemura, "Information retrieval system for XML document", 2002

[INEX 2007] : « INEX 2007 Evaluation Measures (Draft) » Jovan Pehcevski¹, Jaap Kamps², Gabriella Kazai³, Mounia Lalmas⁴, Paul Ogilvie⁵, Benjamin Piwowarski⁶, and Stephen Robertson³

1 INRIA Rocquencourt, France

2 University of Amsterdam, The Netherlands

3 Microsoft Research Cambridge, United Kingdom

4 Queen Mary, University of London, United Kingdom

5 Carnegie Mellon University, USA

6 Yahoo! Research Latin America, Chile

[INEX, 2012]: "Report on INEX 2012" P.Bellot, T.Chappell, A.Doucet, S.Geva, S.Gurajada, J.Kamps, G.Kazai, M.Koolen, M.Landoni, M.Marx, A.Mishra, V.Moriceau, J.Mothe, M.Preminger, G.Ramírez, M.Sanderson, E.Sanjuan, F.Scholer, A.Schuh, X.Tannier, M.Theobald, M.Trappett A. Trotman Q.Wang

[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.

[Jensen, 2000] : Jensen F. V., "Bayesian Networks and Decision Graphs", Wiley, 2000.

[L.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.

[L.M. de camps and 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

[Laitang & al, 2013] : Cyril Laitang, « *Impact de la structure des documents XML sur le processus d'appariement dans le contexte de la recherche d'information Semi-Structurée* », Université Toulouse.

[Lalmas, 1997] : M. Lalmas. Dempster-shafer's theory of evidence applied to structured documents: modeling uncertainty. In *Proceedings of SIGIR'97, Philadelphia, USA*, pages 110–118, 1997.

[Larson, 2002] : Larson R.R. Cheshire ii atinex : Usingahybridlogisticregression and Boolean model for xmlretrieval. In Proceedings of the First Workshop of the INiative for the Evaluation of XML REtrieval(INEX), pages 18 25. Dagsthul, Germany, December 2002.

[Mass & al, 2003] : 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.

[Moffat& al, 2002] :A.Moffat, R.Sacks-Davis, R.Wilkinson and J.Zobel, "*Retrieval of partial documents*",In TREC-2, 1993

[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 Developmentin Information Retrieval, pages 138{140, Melbourne, Australia, Aug. 1998. ACM Press, New York.

[Najehnaffakhi et al, 2009] :Réseau bayésien pour un modèle de Recherche d'Information agrégée dans desdocuments structurés Najeh NAFFAKHI, Mohand BOUGHANEM, Rim FAIZ, 2009

[Pearl, 1988] : Pearl, J. (1988). ProbabilisticReasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, second editionin 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

[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.onResearch and Development in Information Retrieval, 253260, 1996.

[Rob & al, 1976] :S.Robertson&K.Sparck Jones, *Relevance Weighting for SearchTerms*. Journal of The American Society for Information Science, Vol 27, N°3, 1976.

[Salton et al, 1984] : G. Salton and M. McGill. Introduction to modern information retrieval. McGraw-Hill Int. Book Co, 1984.

[Sauvagnat, 2005] :Karen Sauvagnat, "*Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés*", Thèse Doctorat, Université Paul Sabatier de Toulouse, 2005

[Sauvagnat,2005] :Sauvagnat K. le processus de recherche d'information. Thèse de doctorat en informatique, Université Paul Sabatier, Toulouse, France, juin 2005

[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.

[Savoy et al., 1991] :Savoy J., Dubois D., et al., "Information Retrieval in HypertextSystems an ApproachUsingBayesian Networks",Electronic Pub., 4(2) : 87-108, 1991.

[Shafer, 1976] :G. Shafer, " *A mathematicaltheory of evidence*", Princeton, NJ: Princeton UniversityPress, 1976.

[Sigurbjaornsson& al, 2003] :Sigurbjaornsson B., Kamps J., and de Rijke M. An element-basedapproach 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-basedevidential information in abelief network model. In Proc. of the International ACM-SIGIR Conference(2000), pp. 96–103.

[Tabia, 2005]: Développement de mécanisme de coopération entre algorithmes d'apprentissage automatique/ classification dans un environnement incertain. Présenté par TABIA Karim Avril 2005.

[Trotman, 2005] : Trotman A. and O'Keefe R. A. Identifying and ranking relevant document element. In Proceedings of INEX 2003 Workshop, pages 149,154. Dagsthul, Germany, December 2003.

[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 inferencenetwork-basedretrieval model", In ACM Transaction on Information System, 9(3) : 187–222, 1991.

[Yang & al, 2007] : Yang J., Zhang F. XML Document Classification Using Extended VSM.
INEX 2007: 234-244.

Bibliographie

[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.

[Alimazighi et al, 2005] : Utilisation des réseaux bayésien pour la recherche d’information structuré Z. Alimazighi,B.bessaibessai,R. Djiroun ,2005.

[Allorge, 2000] :Stéphane Allorge, " *XML* ", département ASI, 2000.

[Amann, 2003] : Bernd Amann, Cours "*XML et les bases de données : Introduction à la gestion de contenus Web et XML*", Module Données et services sur le Web 2003/2004.

[Benaouicha, 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.

[Belkacem, 2015] : Belkacem Thiziri. Proposition et expérimentation d’un modèle de langue pour la recherche d’information dans les documents XML. Thèse de master en informatique, Université mouloud mammeri tizi ouzou, algérie, juin 2015.

[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

[Braga & 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.

[Buntine, 1994] : untine W., “Representing Learning with graphical Models”. Technical Report, FIA 94-14, Artificial Intelligence Research Branch, NASA Armes Research Center, USA, 1994.

[Carmel & al, 2003] :D.Carmel, D.Efraty, G.Landau, Y.Maaker, and Y.Mass, "An Extension of the Vectorspace Model forquerying 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.

[Crestani et al, 2003] :Crestani, F., de Campos, L. M., Fernandez-Luna, J. M., andHuete, J. F. Rankingstructured documents using utilitytheory 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-layeredBayesian network model for structured document retrieval. In: Symbolic and Quantitative Approaches to ReasoningwithUncertainty, 7th European Conference, ECSQARU 2003, 2003-07-02 - 2003-07-05, Aalborg, Denmark.

[Cui & al, 2003] :H.cui, J-R.Wen, J-R.Chua, "Hierarchicalindexing and flexi elementretrieval for structured document", april 2003.

[De Campos et al, 2003] : de Campos, L., Fernandez-Luna, J., and Huete, J. Improvingthe efficiency of the bayesian network retrieval model by reducingrelationshipsbetweeneterms. International Journal of Uncertainty, Fuzziness and Knowledge-BasedSystems 11, Supplement (2003), 101–116

[Denoyer et al, 2004] :Denoyer, L., Wisniewski, G., and Gallinari, P. Document structure matching for heterogeneouscorpora, 2004. Proc. Of the International ACM-SIGIR Conference : Workshop on XML and Information Retrieval.

[Djebbar et Lamrous, 2013]: Djebbar Sonia et Lamrous fadhila. Proposition d’une approche base sur les réseaux bayésiens pour la recherche d’information dans les documents XML. These de master en informatique. Univeristé mouloud mammeri tizi ouzou, algérie. 2012/2013.

[Fellag, 2006] :Mme Samia FELLAG. Recherche d’information dans les documents semi structuré XML. Thèse de magistère en informatique, Université Mouloud Mammeri, Tiziouzou, septembre 2006.

[Fuhr& al, 2001] : N. Fuhr, K. Grossjohann, "XIRQL, a querylanguage for information retrieval in XML documents", In proceedings of SIGIR 2001, Toronto Canada 2001.

[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. Pittsburgh, 1993.

[Geva, 2005] : S.Geva. GPX-gardenspointXMLIRatINEX2005. In INEX’05, pages 240–253, 2005.

[Gövert& al, 2002] :N. Gövert, M. Abolhassanni, N. Fuhr, K. Grossjohann, “Content-Oriented XML RetrievalwithHyreX”. In INEX 2002 Workshop Proceedings, p. 26-32,Germany, 2002.

[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

[Hatano& al, 2002] : K.Hatano, H.Kinutani, M.Yoshikawa, and S.Uemra, "Information retrieval system for XML document", 2002

[INEX 2007] : « INEX 2007 Evaluation Measures (Draft) » Jovan Pehcevski¹, Jaap Kamps², Gabriella Kazai³, Mounia Lalmas⁴, Paul Ogilvie⁵, Benjamin Piwowarski⁶, and Stephen Robertson³

1 INRIA Rocquencourt, France

2 University of Amsterdam, The Netherlands

3 Microsoft Research Cambridge, United Kingdom

4 Queen Mary, University of London, United Kingdom

5 Carnegie Mellon University, USA

6 Yahoo! Research Latin America, Chile

[INEX, 2012]: "Report on INEX 2012" P.Bellot, T.Chappell, A.Doucet, S.Geva, S.Gurajada, J.Kamps, G.Kazai, M.Koolen, M.Landoni, M.Marx, A.Mishra, V.Moriceau, J.Mothe, M.Preminger, G.Ramírez, M.Sanderson, E.Sanjuan, F.Scholer, A.Schuh, X.Tannier, M.Theobald, M.Trappett A. Trotman Q.Wang

[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.

[Jensen, 2000] : Jensen F. V., "Bayesian Networks and Decision Graphs", Wiley, 2000.

[L.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.

[L.M. de camps and 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

[Laitang& al, 2013] : Cyril Laitang, « *Impact de la structure des documents XML sur le processus d'appariement dans le contexte de la recherche d'information Semi-Structurée* », Université Toulouse.

[Lalmas, 1997] : M. Lalmas. Dempster-shafer's theory of evidence applied to structured documents: modeling uncertainty. In *Proceedings of SIGIR'97, Philadelphia, USA*, pages 110–118, 1997.

[Larson, 2002] : Larson R.R. Cheshire ii atinex : Usingahybridlogisticregression and Boolean model for xmlretrieval. In Proceedings of the First Workshop of the INiative for the Evaluation of XML REtrieval(INEX), pages 18 25. Dagsthul, Germany, December 2002.

[Mass & al, 2003] : 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.

[Moffat& al, 2002] :A.Moffat, R.Sacks-Davis, R.Wilkinson and J.Zobel, "*Retrieval of partial documents*",In TREC-2, 1993

[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 Developmentin Information Retrieval, pages 138{140, Melbourne, Australia, Aug. 1998. ACM Press, New York.

[Najehnaffakhi et al, 2009] :Réseau bayésien pour un modèle de Recherche d'Information agrégée dans desdocuments structurés Najeh NAFFAKHI, Mohand BOUGHANEM, Rim FAIZ, 2009

[Pearl, 1988] : Pearl, J. (1988). ProbabilisticReasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, second editionin 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

[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.onResearch and Development in Information Retrieval, 253260, 1996.

[Rob & al, 1976] :S.Robertson&K.Sparck Jones, *Relevance Weighting for SearchTerms*. Journal of The American Society for Information Science, Vol 27, N°3, 1976.

[Salton et al, 1984] : G. Salton and M. McGill. Introduction to modern information retrieval. McGraw-Hill Int. Book Co, 1984.

[Sauvagnat, 2005] :Karen Sauvagnat, "*Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés*", Thèse Doctorat, Université Paul Sabatier de Toulouse, 2005

[Sauvagnat,2005] :Sauvagnat K. le processus de recherche d'information. Thèse de doctorat en informatique, Université Paul Sabatier, Toulouse, France, juin 2005

[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.

[Savoy et al., 1991] :Savoy J., Dubois D., et al., "Information Retrieval in HypertextSystems an ApproachUsingBayesian Networks",Electronic Pub., 4(2) : 87-108, 1991.

[Shafer, 1976] :G. Shafer, " *A mathematicaltheory of evidence*", Princeton, NJ: Princeton UniversityPress, 1976.

[Sigurbjaornsson& al, 2003] :Sigurbjaornsson B., Kamps J., and de Rijke M. An element-basedapproach 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-basedevidential information in abelief network model. In Proc. of the International ACM-SIGIR Conference(2000), pp. 96–103.

[Tabia, 2005]: Développement de mécanisme de coopération entre algorithmes d'apprentissage automatique/ classification dans un environnement incertain. Présenté par TABIA Karim Avril 2005.

[Trotman, 2005] : Trotman A. and O'Keefe R. A. Identifying and ranking relevant document element. In Proceedings of INEX 2003 Workshop, pages 149,154. Dagsthul, Germany, December 2003.

[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 inferencenetwork-basedretrieval model", In ACM Transaction on Information System, 9(3) : 187–222, 1991.

[Yang & al, 2007] : Yang J., Zhang F. XML Document Classification Using Extended VSM.
INEX 2007: 234-244.