REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mouloud Mammeri de Tizi-Ouzou
X•Θ Λ•Σ X [://:V •X[•Λ[•O
Faculté de génie électrique et informatique
Département d'informatique

Mémoire de fin d'études

En vue de l'Obtention Du Diplôme de Master en informatique

Option : conduite de projets informatiques et Ingénierie des systèmes d'information.

Thème

Conception et réalisation d'une approche d'indexation conceptuel le des documents semi-structurés XML

Dirigé par :
Mme. AMIROUCHE F.

Présidente
Mme. BOUARAB.F

Réalisés par :
DJELOUAH Chafik.
FENEK Aghiles.

Examinateurs
Mlle ILTACHE.S
Mr. AMIROUCHE.M.N

Remerciements

En préambule à ce mémoire, Nous tenons tout d'abord à remercier Dieu qui nous a donné la force et la patience d'accomplir ce Modeste travail.

Nous tenons à remercier sincèrement Madame AMIROUCHE.F, qui, en tant que notre encadreur, s'est toujours montrée à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu nous consacrer.

Nous tenons aussi à exprimer toute notre gratitude aux membres du jury pour avoir accepter d'évaluer et de juger notre travail.

Et en fin nous souhaitons adresser nos remerciements les plus sincères aux personnes qui de loin ou de prés nous ont apportés leur aide et qui ont contribué à l'élaboration de ce mémoire.

Table des matières

Listes des figures et des tableaux

Listes des figures et tableaux :

Liste des figures :

Liste des tableaux :

Introduction générale

Introduction générale :

Notre sujet de mémoire a d'abord été attribué dans le cadre d'un trinôme (DJELOUAH Chafik, FENEK Aghiles et FERROUK Massinissa).

La recherche bibliographique, première étape de ce travail a été réalisée dans ce contexte. Puis suite au remaniement du trinôme dicté par une note administrative le mémoire a été subdivisé comme suit :

- Thème 1 : qui concerne une conception et réalisation d'une approche d'indexation des documents XML réalisé en monôme par FERROUK Massinissa.
- Thème 2 : qui concerne une conception et réalisation d'un système de recherche d'information semi-structuré conceptuel réalisé en binôme par DJELOUAH Chafik et FENEK Aghiles.

L'état de l'art initialement réalisé à trois a été introduit dans les deux mémoire résultant du remaniement.

Contexte et problématique :

Notre travail dans ce mémoire s'intègre dans la problématique de la recherche d'information (RI) dans des documents semi-structurés (RIS) de type XML (XML est le format standard de représentation et d'échange des documents semi structurés sur le web).

Traditionnellement, la RI permet de retrouver des documents pertinents pour une requête utilisateur en comparant, dans un processus d'appariement, leurs représentations respectives construites en amont, à l'issue d'un processus dit l'indexation. La RIS combine des aspects de la RI traditionnelle, qui traite des documents « plats » ou non structurés, à des aspects plus spécifiques qui traitent des documents structurés en tenta compte de leur structure en plus de leur contenu. En RI traditionnelle, l'unité de recherche pertinente correspond à un élément (paragraphe, section, titre,…) dans le document. La granularité de la recherche XML est ainsi plus fine qu'en RI traditionnelle.

La prise en compte de la structure a amené de nouvelles problématiques et de nouveaux défis à différents niveaux de la RI :

- ✓ Au niveau de l'indexation/appariement : l'indexation des documents XML doit tenir compte de la structure et du contenu des documents XML tout en gérant le lien entre les deux.
- ✓ Au niveau de l'interrogation : les langages des requêtes permettent à l'utilisateur d'interroger les documents semi-structurés. D'une manière générale, ces langages

de requêtes doivent supporter à la fois des contraintes portant sur le contenu et la structure.

Nous nous intéressons dans le cadre de notre travail à l'indexation des documents XML. Nous nous focalisons en particulier sur l'indexation du contenu des documents XML. En effet dans la RIS, la majorité des approches de recherche d'information sont basées sur des mots clés où l'élément d'un document et la requête sont représentés par une liste de mots clés, généralement pondérés.

L'appariement document-requête est basé sur le nombre de mots clés qu'ils ont en commun. L'indexation par des mots clés est généralement imprécise. Cette imprécision est due à l'ambigüité sémantique des mots du langage naturel. En effet, un même mot peut posséder plusieurs sens et différents mots peuvent avoir une même signification. De ce fait des éléments du document bien qu'ils soient pertinents et contenant des mots sémantiquement équivalent mais lexicalement différents (synonymes) des mots de la requête, ne seront pas retrouvés. Par ailleurs, des éléments non pertinents, contenant des mots lexicalement identiques mais sémantiquement différents (homonymes) des mots de la requête seront retournés à l'utilisateur.

Une solution pour palier aux limites de l'indexation à base de mots clés est l'indexation par les sens des mots (ou concepts). C'est l'indexation sémantique(ou conceptuelle). Contrairement aux systèmes classiques à base de mots clés, dans une représentation conceptuelle l'appariement document-requête se fait via des concepts similaires mais pas nécessairement « identiques ». La RI sémantique ou conceptuelle est caractérisée par l'utilisation des ressources sémantique (thésaurus, ontologies, etc.…) dans la phase d'indexation et de recherche.

Organisation de la thèse :

Notre mémoire est articulé sur quatre chapitres :

Les deux premiers chapitres présentent respectivement les méthodes et concepts fondamentaux de la RI classique et de la RI semi-structurée. Nous y avons introduit en particulier les concepts fondamentaux de la RI sémantique et les différentes approches.

Le troisième chapitre présente l'approche de RIS conceptuelle. Nous proposons d'implémenter cette approche

Le quatrième chapitre est dédié à la représentation que nous avons réalisée en vue de l'évaluation de l'approche implémentée et des résultats obtenu à l'issue de cette évaluation.

Ce mémoire se termine par une conclusion et des perspectives

Chapitre 1 : Recherche d'information classique

I. Introduction

La recherche d'information a pour objectif de retrouver dans une collection donnée, l'ensemble des documents pertinents pour un besoin informationnel, exprimé sous forme d'une requête.

Un Système de Recherche d'Information (SRI) est un ensemble de programmes informatiques qui a pour but de retrouver dans une **collection de documents**, les **documents** qui sont pertinents pour un besoin informationnel de l'utilisateur exprimés sous forme d'une **requête**.

Cette définition fait ressortir les éléments clés suivants :

1. **Document :** On appelle un document toute unité qui peut constituer une réponse à une requête d'utilisateur et il peut être présenté sous plusieurs formes, un texte, un morceau de texte, une image, une bande vidéo, une page web…
2. **Collection de documents** : ensemble de documents.
3. **Requête :** Une requête exprime le besoin d'information d'un utilisateur.
4. **La pertinence système :** Elle représente la relation entre la requête et l'information. « Elle définit une correspondance entre un document et une requête, ou encore une mesure d'normativité du document à la requête » (Boughanem et al, 08)
5. **La pertinence utilisateur :** c'est une mesure subjective qui représente la satisfaction de l'utilisateur vis-à-vis des documents retournés.

   L'objectif d'un SRI est de faire correspondre la pertinence système à la pertinence utilisateur.

II. Le processus de recherche d'information

Le processus ou le modèle général de la recherche d'information a pour but la mise en relation des différents acteurs de la recherche d'information : collection documentaire (corpus), Utilisateur (besoin informationnel) et le Système de Recherche d'Information qui doit pouvoir traiter une grande masse d'informations de façon rapide et pertinente.

Le rôle d'un SRI est de retrouver les documents pertinents pour un besoin informationnel de l'utilisateur formellement exprimé par une requête. Pour cela, le SRI indexe les documents et la requête dans l'objectif de créer leurs représentations internes. Ces représentations internes sont ensuite appariées selon un modèle de recherche, et les documents les plus similaires à la requête sont retournés triés par ordre de pertinence.
Un SRI peut en outre intégrer un processus de reformulation de requête permettant d'améliorer les résultats de la recherche.

Le fonctionnement général d'un SRI est donné au travers du processus de recherche communément appelé processus en U [Belkin et al., 92], présenté en figure 1.1

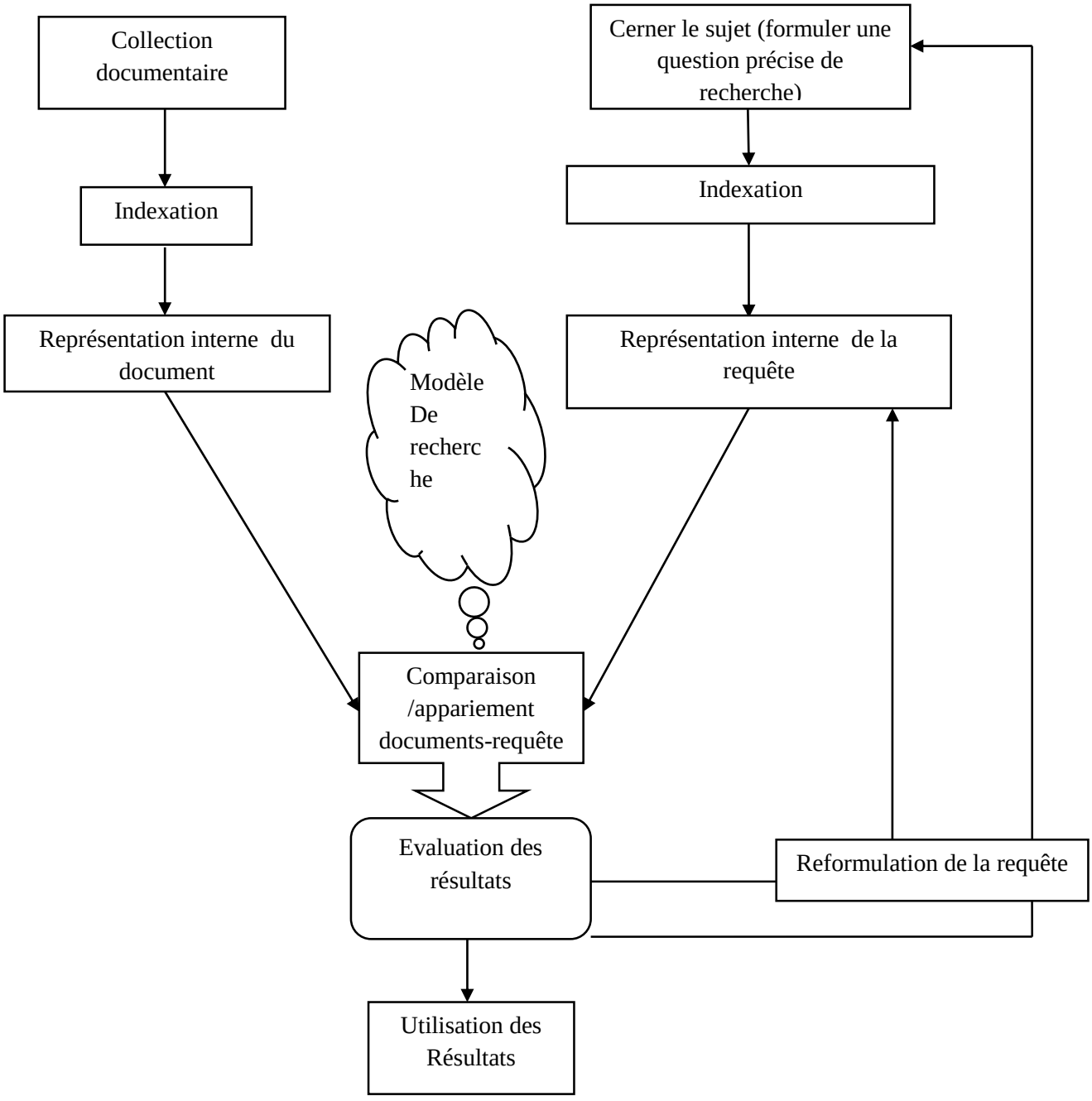


FIGURE 1.1 : Processus en U de la RI

II.1.1 L'indexation :

L'indexation est une phase fondamentale et indissociable d'un SRI. De sa qualité dépend la performance du SRI. L'indexation a pour but de construire l'ensemble des termes représentatifs du contenu de la requête et de chaque document de la collection. Ces termes, dits termes d'index, jouent un rôle très important dans la recherche d'information puisqu' ils déterminent avec quels mots on peut retrouver un document.

L'ensemble des termes d'index constituent le langage d'indexation. Ce langage peut être :

- ✓ **Un langage libre :** il est construit à partir des termes extraits du document analysé.
- ✓ **Un langage d'indexation contrôlé** : il est construit à partir d'un ensemble de termes préalablement définis et organisés dans un thésaurus ; l'indexation se fait seulement avec les termes de cette terminologie.

II.1.1.1 Approches d'indexation :

L'indexation peut être manuelle, automatique ou semi-automatique :

- ✓ **L'indexation manuelle** : c'est une opération humaine où chaque document est analysé par un spécialiste ou un documentaliste ;
- ✓ **L'indexation automatique** : C'est la technique la plus utilisée car la plus adaptée aux corpus volumineux. Le processus d'indexation est entièrement informatisé. Aucune intervention humaine n'est requise dans le choix des termes d'index.
- ✓ **L'indexation semi-automatique** : c'est une indexation mixte combinant les deux approches précédentes. Les termes d'index sont d'abord extraits automatiquement, puis le choix final des termes est réalisé par les spécialistes ou documentalistes.

II.1.1.2 Indexation automatique :

L'indexation automatique est fondée sur l'analyse des documents en vue de l'extraction des termes représentatifs de leur contenu informationnel. L'indexation automatique repose sur les étapes suivantes :

a- L'extraction des termes d'indexation : repose sur une analyse linguistique du texte du document. Elle comprend les étapes suivantes :

- ✓ Analyse lexicale : l'analyse lexicale est un processus qui permet de transformer un texte du document en un ensemble de termes.

- ✓ Elimination des mots vides : Les mots représentatifs sont gardés et les mots vides (mots outils de la langue comme les propositions, les articles, les pronoms…, mots trop fréquents, mots rares) sont éliminés.
- ✓ Normalisation des termes d'index basée sur deux approches:

  - **La lemmatisation :** La lemmatisation d'un mot consiste à transformer le mot en son lemme. Le lemme est la racine du mot telle que définie dans la langue naturelle. L'algorithme le plus populaire de lemmatisation pour la langue anglaise est l'algorithme de Porter.

Exemple : Regroupement par **lemme**

am, are, is➔be

car, cars, car's, cars'➔ car

Exemple de règles de fonctionnement de l'algorithme de Porter :

SSES ===>SS
caresses ===> caress
IES ===> I
ponies ===> poni
SS ===>SS
caress ===> caress
S===>
cats ===> cat

  - **La troncature :** Construit le radical d'un mot en le coupant à partir d'une position donnée. De cette manière, on réduit les variations morphologiques des mots.

Par exemple : le terme **Psycholog** est obtenu par troncature à 9 caractères des mots suivants qu'il représente **:**

- **psychologie ;**
- **psychologies ;**
- **psychology ;**
- **psychologue(s) ;**
- **psychological ;**
- **psychologist(s)…**

b- La pondération des termes d'index:

La pondération consiste à donner un poids d'importance $\mathbf{W_{ij}}$ à chaque terme $\mathbf{t_i}$ d'un document $\mathbf{d_j}$. Les approches de pondération se basent le plus souvent sur deux facteurs :

- un facteur **de pondération locale** qui exprime l'importance du terme $\mathbf{t_i}$ dans le document $\mathbf{d_i}$. Ce facteur est fonction de sa fréquence d'occurrences $\boldsymbol{tf_{ij}}$ dans le document.
- un facteur **de pondération globale** qui exprime son pouvoir de discrimination sur l'ensemble des documents de la collection. Ce facteur est fonction de sa fréquence documentaire inverse $\boldsymbol{idf_j}$ dans tous les documents de la collection.

Plusieurs formules existent et sont basées sur le schéma combiné suivant:

$$W_{ij} = = tf_{ij} * idf_j$$

Formule 1.1

Où :

- tf_{ij} : est la fréquence du terme t_j dans le document
- dfj_i. : est fréquence documentaire du terme (ie. Le nombre de documents de la collection contenant ce terme)
- idf_j: est la fréquence documentaire inverse du terme t_j.

II.2.2 Taxonomie des modèles de la recherche d'information :

Un modèle en recherche d'information a pour but de fournir une formalisation du processus de recherche d'information. A partir des termes issus de la phase d'indexation, le modèle remplit les rôles suivants :

- représentation interne d'un document et représentation interne de la requête
- Définir une méthode de comparaison entre ces deux représentation afin de déterminer la similarité document /requête.

On distingue principalement trois classes de modèles de recherche d'information :

- **Les modèles basés sur la théorie des ensembles** : dont **le modèle booléen** qui est le modèle le plus utilisé dans la recherche d'information.
- **Les modèles algébriques :** dont le **modèle vectoriel** qui considère que les documents et la requête font partie d'un même espace vectoriel**.**
- **Les modèles probabilistes** qui se basent sur les probabilités pour estimer la pertinence d'un document vis-à-vis d'une requête.

La figure 1.2 représente la taxonomie des modèles de la recherche d'information :

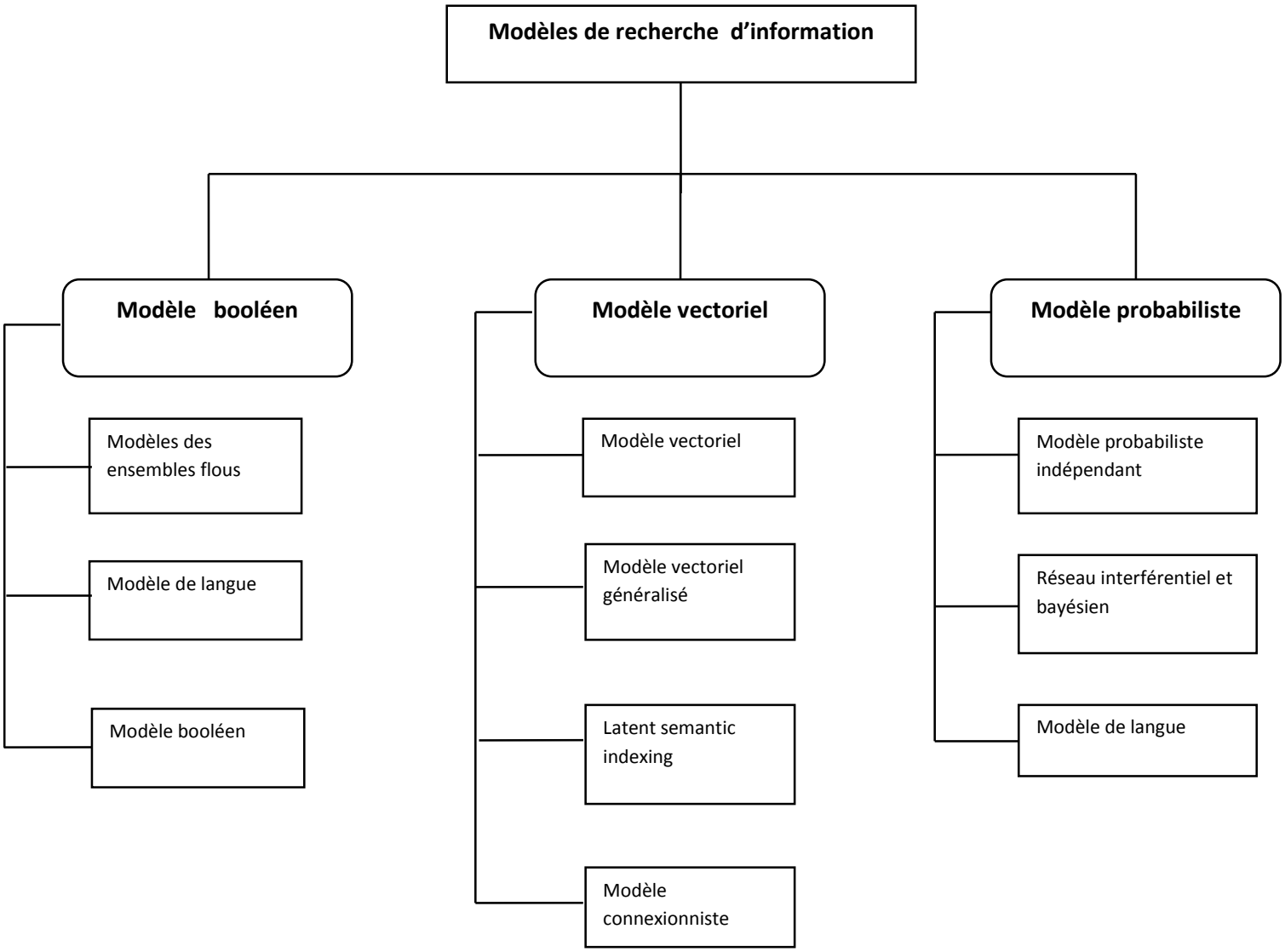


FIGURE 1.2 : Taxonomie des modèles en RI

II.2.2.1 Le modèle booléen :

Le modèle booléen est basé sur la théorie des ensembles et l'algèbre de Boole. Dans le modèle de base, le document est représenté par un ensemble de termes (non pondérés), et la requête est définie par un ensemble de mots clés reliés par des opérateurs logique (**AND, OR, NOT**). Ce modèle considère que si les termes de la requête sont présents dans

le document alors le document est pertinent pour la requête. Son score de pertinence pour la requête (**RSV(q,d)**) prend alors la valeur 1. Dans le cas contraire, le document est considéré comme non pertinent, et son score de pertinence prend la valeur 0.

$$Rsv(q,d) = \begin{cases} 1 \text{ si } d \in \text{à l'ensemble décrit par la requête} \\ 0 \text{ sinon} \end{cases}$$

L'appariement requête/document est strict et se base sur des opérations ensemblistes selon les règles suivantes :

$RSV(d,t_i) = 1$ si $t_i \in d$, 0 sinon
$RSV(d, t_i \text{ AND } t_j) = 1$ si $(t_i \in d) \wedge (t_j \in d)$, 0 sinon
$RSV(d, t_i \text{ OR } t_j) = 1$ si $(t_i \in d) \vee (t_j \in d)$, 0 sinon
$RSV(d, \text{NOT } t_i) = 1$ si $t_i \notin d$, 0 sinon

Bien que ce modèle soit simple à mettre en œuvre, il présente des inconvénients majeurs:

- ✓ les termes ne sont pas ordonnables et sont considérés de même importance tant dans le document que dans la requête ;
- ✓ On ne peut classer les documents que dans deux catégories (pertinents ou non pertinents)
- ✓ Les expressions booléennes ne sont pas accessibles à un large public.

Remarque :

Pour pallier aux inconvénients du modèle booléen de base, plusieurs extensions ont été proposées [le modèle p-norme, Salton et al., 1983], qui visent à tenir compte des poids des termes dans les documents et dans la requête, ce qui permet l'ordonnancement des documents retrouvés par ordre de pertinence (valeur approchée).

II.2.2.2 Les modèles vectoriels :

II.2.2.2.1 Le Modèle vectoriel de base :

C 'est le modèle le plus populaire et le plus utilisé dans la recherche d'information. Dans ce modèle, un document $\mathbf{d_i}$ est représenté par un vecteur de poids $\mathbf{w_{i,j}}$de dimension **n** composé de tous les termes d'index.

$d_i=(w_{i.1},w_{i.2},\ldots,w_{i.n})$ pour i = 1, 2, …, m

Où $w_{i,j}$ est le poids du terme t_i dans le document d_i ;

- **m** est le nombre de documents de la collection;
- **n** est le nombre de termes d'index dans la collection.

Une requête est aussi représentée par un vecteur de poids $wQ_{i,j}$ défini dans le même espace vectoriel que le document

$$Q = (wQ_1 , wQ_2, \ldots, wQ_n)$$

Où :

- wQ_j : est le poids du terme t_j dans la requête **Q** (ce poids peut être soit une forme de **tf*idf**, soit un poids attribué manuellement par l'utilisateur).

La pertinence du document d_i pour la requête **Q**, est définie comme le degré de corrélation des vecteurs documents/requête. Cette corrélation peut être exprimée par l'une des mesures suivantes :

- ✓ **Le produit scalaire :**

$$Sim(d_i , Q) = \left(\sum_{i=1}^{n} w_{Qj} * W_{ij} \right)$$

Formule 1.2

- ✓ **La mesure de cosinus :**

$$Sim(d_i,Q)=\frac{\sum_{i=1}^{n} WQj*Wij}{\left(\sum_{i=0}^{n} W^2Qj\right)^{1/2}*\left(\sum_{j=1}^{n} W^2_{ij}\right)^{1/2}}$$

Formule 1.3

- ✓ **La mesure de Dice :**

$$Sim(d_i , Q) = \frac{2*\sum_{i=1}^{n} WQj*Wij}{\sum_{j=1}^{n} W_{QJ}^2 +\sum_{j=1}^{n} W_{ij}^2}$$

Formule 1.4

- ✓ **La mesure de Jacard :**

$$Sim(d_i ,Q) = \frac{\sum_{j=1}^{n} W_{ij}*w_{Qj}}{\sum_{j=1}^{n} W_{QJ}^2+ \sum_{j=1}^{n} W_{IJ}^2-\sum_{j=1}^{n} W_{ij}*W_{Qj}}$$

Formule 1.5

- ✓ **Le coefficient de superposition (overlap) :**

$$Sim(d_i , Q) = \frac{2*\sum_{i=1}^{n} WQj*Wij}{\min_i\left(\sum_{j=1}^{n} W_{QJ}^2 , \sum_{j=1}^{n} W_{ij}^2\right)}$$ Formule1.6

II.2.2.2.2 Le modèle connexionniste :

L'idée de base est que la RI est un processus associatif qui peut être représenté par un processus d'activation des réseaux de neurones. La notion de réseaux est très intéressante pour la représentation des différentes relations et associations qui existent entre les termes, les documents et les (termes et documents).

Le modèle connexionniste utilise le fondement de réseaux de neurones pour la modélisation des unités textuelles et la mise en œuvre du processus de la recherche d'information. Deux modèles théoriques ont été utilisés:

- ✓ Les modèles à auto-organisation **[Lin et al., 91]** :permettent à partir de la description des documents, d'en réaliser une classification par l'apprentissage du réseau de neurones.
- ✓ Les modèles a couches : Les SRI se basent sur un modèle connexionniste à couches **[Kwok, 89; Belew, 89; Boughanem, 92; Mothe, 94]** et sont représentés par un minimum de trois couches de neurones interconnectées :
    - la couche requête (**Q**) ;
    - la couche termes (**T**) ;
    - la couche documents (**D**) ;

Le mécanisme de recherche est basé sur une activation initiale de neurones termes induite par une requête, et qui se propage vers les documents à travers les réseaux. Les documents qui ont des termes en commun avec la requête recevront une entrée **$In(d_i)$** et le modèle calculera sa sortie **$Out(d_i)$**. Les valeurs de sortie des différents documents correspondent à leurs degrés de pertinence pour la requête donnée.

II.2.2.3 Les modèles probabilistes :

I.2.2.3.1 Le modèle probabiliste de base :

Le premier modèle probabiliste a été proposé par **Maron** et **Kuhns [Maron et al., 60]** au début des années 60. Le modèle probabiliste consiste à présenter les résultats d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête.

L'idée de base dans un modèle probabiliste est de tenter de déterminer la probabilité **P(R|D)** que le document **D** sélectionné soit pertinent pour la requête **Q** et la probabilité **P(NR|D)** que le document soit non pertinent pour **Q**.

Le score de pertinence d'un document (ou **RSV(d ,Q)**) est donné, d'après le théorème de Bayes et après simplification, par :

$$\mathbf{RSV(d, Q)} = \frac{p(R/di)}{p(NR/di)} \approx \frac{p(di/R)}{p(di/NR)}$$

Formule 1.7

Où:

- **$P(d_i/R)$:** est la probabilité que **d_i** appartienne à l'ensemble des documents pertinents .
- **$P(d_i/NR)$:** est la probabilité que **d_i** appartienne à l'ensemble des documents non pertinents.

II.2.2.3.2 Modèle de langue :

Le modèle de langue considère que la pertinence d'un document vis-à-vis d'une requête est en rapport avec la probabilité que la requête puisse être générée par le document. Un document est pertinent si et seulement si la requête utilisateur est en ressemblance avec celle inférée par le document :

$$\mathbf{RSV(d,Q) = P(Q = (t_1,t_2, \ldots t_n)/Md)} = \prod_{i=1}^{n} p(t_i/d)$$

- **$P(t_i/d)$** peut être estimé en se basant sur l'estimation maximale de vraisemblance comme suit :

$$\mathbf{P\ (ti\ /d)} = \frac{\mathbf{tf(t_i/d)}}{\sum_t \mathbf{tf(t/d)}} \quad \text{Formule 1.8}$$

II.2.3 La reformulation de requête :

Vu la croissance phénoménale des bases documentaires l'utilisateur trouve des difficultés à formuler une question précise de la recherche. Afin de correspondre la pertinence utilisateur et la pertinence système, une étape de reformulation de requête est souvent utilisée.

La reformulation de requête signifie qu'à partir d'une requête initiale formulée par l'utilisateur, il puisse construire une nouvelle requête qui répond mieux à ses besoins informationnels.

Il existe deux principales méthodes de reformulation de requêtes: la méthode locale et la méthode globale.

II.2.3.1 La méthode locale :

Les méthodes locales s'appuient sur la technique de réinjection de pertinence ou *relevance feedback* **[Buckley et al., 94; Harman,92; Robertson et al., 97; Rocchio, 71]** . L'idée de base est de faire participer l'utilisateur dans le processus de reformulation.

- L'utilisateur formule sa requête.
- Le système renvoie un premier ensemble de résultat de recherche.
- L'utilisateur marque quelques documents retournés comme pertinents ou non pertinents.
- Le système calcule une meilleure représentation du besoin en l'information sur la base de rétroaction utilisateur.
- Le système visualise un ensemble révisé de résultats de la recherche

Le processus de réinjection de pertinence peut passer par une ou plusieurs itérations de ce type. La reformulation de la nouvelle requête se fait à l'aide de la pertinence utilisateur, La nouvelle requête $\mathbf{Q_m}$ est obtenue à partir de la requête initiale **Q0** en appliquant un algorithme spécifique de réinjection de pertinence.

Dans le modèle vectoriel, l'algorithme de reformulation de requête le plus utilisé est l'algorithme de **Rocchio [Salton et al., 1983; Salton, 1989]** . Dans cet algorithme, la nouvelle requête **Qm** est construite comme suit :

$$\mathbf{Qm} = \alpha \mathbf{Q0} + \beta \frac{1}{|\mathbf{R}|} \sum_{\mathbf{d_p} \in \mathbf{R}} \mathbf{d_p} - \gamma \frac{1}{|\mathbf{NR}|} \sum_{\mathbf{d_{np}} \epsilon \mathbf{NR}} \mathbf{d_{np}} \quad \text{Fomule1.9}$$

Où :

- **Qm** : la requête modifiée
- **Q0** : la requête initiale
- **dp** : (respectivement **dnp**) est un vecteur associé à un document pertinent (respectivement non pertinent)
- **R** : est l'ensemble des documents pertinents.
- **NR** : est l'ensemble des documents non pertinents.
- **α, β etγ** étant des constantes telles que **α+β+γ=1**

Les paramètres **α, β etγ** sont choisis en fonction de l'importance qu'on souhaite donner à la requête initiale.

II.2.3.2 Méthode globale :

Les méthodes globales se basent sur l'expansion de requête. La forme la plus commune d'expansion de requête est l'analyse globale en utilisant un thesaurus **[Qiu, 93]** ou une ontologie **[Mandala et al., 91; Voorhees, 94; Navigli et al., 03; Moldovan et al., 00; Bazizet al., 03a ; Baziz et al., 03b].** Pour chaque terme t, la requête peut être automatiquement étendue avec des mots du thésaurus synonymes ou liés au terme t. Le système peut ainsi apparier la requête à des documents pertinents qui ne contiennent aucun des mots de la requête originale.

II.3 Évaluation d'un SRI :

L'évaluation d'un SRI est une étape très importante pour sa validation. Elle permet de définir les caractéristiques du système en termes de qualité de service et de facilité d'utilisation selon les critères suivants:

- le temps de réponse,
- la présentation des résultats,
- l'effort fourni par l'utilisateur pour retrouver parmi les documents retournés ceux qui sont pertinents,
- le taux de rappel du système
- le taux de la précision du système.

Le temps de réponse, la représentation des résultats et l'effort fourni par l'utilisateur sont des mesures de la qualité de service rendu à l'utilisateur tandis que le rappel et la précision essentiels aux modèles de recherche couvrent quant à eux la performance du système.

L'évaluation d'un SRI consiste principalement à mesurer ces performances sur la base d'une **collection de test** contrôlée et des **métriques d'évaluation** standards définis selon des critères d'efficacité.

II.3.1 Collections de référence - Un exemple : les collections TREC :

Une collection de test (ou collection de référence) comprend un ensemble de documents à indexer sur lesquels le système sera évalué, une liste de requêtes prédéfinies et des jugements de pertinence manuellement établies par des assesseurs humains pour chaque requête.

De nombreuses collections de référence existent en RI. Elles sont principalement mises en œuvre dans le cadre de campagnes d'évaluation, dont les principales sont :

- **La campagne CLEF :** lancée en 2000, cette campagne a pour objectif de promouvoir la recherche et le développement dans le domaine de la recherche d'information multilingue.
- **La campagne Amaryllis :** est une version française de TREC de 1996 à 1999 et sous taches de CLEF en 2002
- **La campagne INEX :** Lancée en 2002, son objectif principal est de promouvoir l'évaluation de la recherche dans les documents semi-structurés en fournissant de grandes collections de test de type XML.
- **La campagne TREC** (*text retrieval conférence*) : constitue le projet le plus ambitieux d'évaluation des SRI. La campagne TREC est une campagne d'évaluation annuelle depuis 1992. Elle vise à explorer de nouveaux domaines de recherche et de démontrer la robustesse des méthodologies de recherche existantes. Chaque année, la campagne TREC lance de nouvelles tâches (ou pistes) correspondant aux centres d'intérêts actualisés des chercheurs de RI. Parmi ces tâches, on distingue :

1. La tâche *spoken document retrieval*
2. La tâche *question answering*
3. La tâche Interactive
4. **La tâche *Ad Hoc* :** c'est la tâche classique de RI. Elle vise à évaluer les performances d'un SRI sur des ensembles statiques de documents.
5. **...**

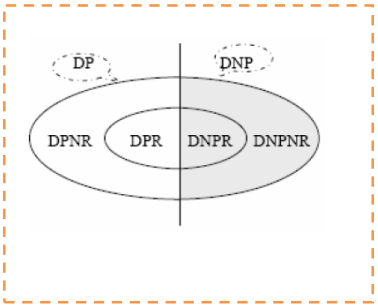
II.3.2 Les mesures d'évaluation d'un SRI :

Pour évaluer la performance d'un SRI, deux principales mesures sont utilisées

- ✓ **la précision :** détermine l'aptitude d'un SRI à rejeter les documents non pertinents vis à vis à d'une requête utilisateur
- ✓ **le rappel :** exprime la capacité d'un SRI à sélectionner tous les documents pertinents vis à vis de cette requête.

Etant donnée une requête Q, les documents de la collection peuvent être globalement classifiés en fonction de leur rapport à la requête en :

- ensemble de documents pertinents DP
- ensemble de documents pertinents retrouvés DPR
- ensemble de documents pertinents non retrouvés DPNR
- ensemble de documents non pertinents DNP
- ensemble de documents non pertinents retrouvés DNPR
- ensemble de documents non pertinents non retrouvés DNPNR



Formellement :

$$\textbf{Rappel} = \frac{|DPR|}{|DP|} \qquad \textbf{Précision} = \frac{|DPR|}{|DPR \cup DPNR|}$$

Formule 1.10

Les mesures de rappel et de précision utilisées seules ne sont pas de bons indicateurs de la performance d'un SRI. Par ailleurs, dans certaines applications, le rappel peut être plus intéressant que la précision.

Comment trouver un compromis entre le rappel et la précision ?
Plusieurs approches proposées dont :
Agrégation du rappel et de la précision dans une seule mesure : **le F-score [VanRijbergen, 79]**.

La F-mesure permet d'agréger le rappel et la précision dans une mesure unique :

$$\boldsymbol{F = \frac{1}{a*\frac{1}{P}+(1-a)*\frac{1}{R}}}$$

Formule1.11

Où $\boldsymbol{\alpha} \in [\mathbf{0}, \mathbf{1}]$

Apres développement, on obtient la forme communément utilisée suivante :

$$\boldsymbol{F_\beta = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R}}$$

Fromule1.12

Où $\beta=\frac{1-a}{a}$

La F-mesure par défaut est donnée comme suit :

$$\boldsymbol{F_1 = \frac{2 * P * R}{P + R}}$$

Fromule1.13

- Les mesures de rappel, précision et ***F-score*** sont des mesures basées-ensembles :
- Elles permettent d'évaluer des ensembles non ordonnés de résultats.
- On parle de mesures d'évaluation non ordonnées.

D'autres mesures d'évaluation ordonnée existent : De nombreuses mesures d'évaluation permettent d'agréger le rappel et la précision en une valeur unique, parmi lesquelles:

- La précision moyenne ***Pmoy***,
- La ***MAP***,
- La ***R-précision***(ou précision exacte).

La précision Moyenne :

L'idée est de générer une valeur unique de *ranking* en moyennant les valeurs de précision obtenues après chaque document pertinent observé.

MAP :

La ***MAP*** est la moyenne des précisions moyennes (***Pmoy***) obtenues sur l'ensemble des requêtes à chaque fois qu'un document pertinent est retrouvé :

$$\boldsymbol{MAP} = \frac{\sum_{q \in Q} \boldsymbol{P_{moy}(a)}}{|\boldsymbol{Q}|}$$

Fromule1.12

Q étant l'ensemble des requêtes.

R-précision :

La ***R-précision :*** est un bon paramètre pour observer le comportement d'un système pour chaque requête individuellement.

La R-précision moyenne calculée sur toutes les requêtes n'a pas d'intérêt.

L'idée est de générer une valeur de *ranking* unique en calculant la précision au rang **R**, où **R** est le nombre de documents pertinents pour la requête courante.

$$\boldsymbol{R - Prec = P@R = \frac{|DPR|}{R}}$$

Fromule1.13

II.3.3 Protocole d'évaluation TREC :

Pratiquement, pour évaluer un SRI, les participants à la campagne TREC doivent suivre le protocole suivant : Pour chaque requête de la collection de test fourni, les 1000 premiers documents restitués par le système sont examinées et les précisions à x points (notées P@ x), sont calculées à différents points (à 5, 10, 15, 30, 100 et 1000 premiers documents restitués). La précision exacte découle de ces précisions. Puis, une précision moyenne ***MAP*** est calculée pour chaque requête. Il s'agit de la moyenne des précisions de chaque document pertinent pour cette requête. La précision d'un document est la précision à x, tel que x est le rang de ce document dans l'ensemble des documents pertinents retrouvés.

Finalement, les précisions moyennes pour l'ensemble des requêtes sont calculées permettant d'obtenir une mesure de la performance globale du système.

II.4 La recherche conceptuelle en RI classique :

L'indexation conceptuelle se réfère à la construction de structures conceptuelles à partir des textes pour décrire le contenu des documents et requêtes. C e type d'indexation est caractérisé par l'utilisation des ressources sémantiques et des mesures de similarité sémantique entre les concepts. Les ressources sémantiques permettent d'identifier les concepts à partir du texte d'un document ou d'une requête. Les mesures de similarité sont utilisées dans la désambiguïsation des termes, dans la pondération des concepts et dans l'évaluation de la pertinence.

II.4.1 Problèmes liés à la RI basés sur des mots clés :

Les problèmes liés à la SRI classique (ou SRI semi-structurée) basés sur des mots-clés se présentent dans l'ambiguïté des mots et leur disparité **[Boubekeur, 2008]**.

II.4.1.1 L'ambiguïté syntaxique :

Elle se rapporte à des différences dans la catégorie syntaxique. Par exemple, dans la phrase « Nous avions des avions », le mot « avions » peut apparaître en tant que nom ou verbe.

II.4.1.2 L'ambiguïté sémantique :

Elle se rapporte à des différences dans la signification et est décomposée en homonymie et polysémie **[Krovetz, 1997]**.

II.4.1.3 disparité des mots :

Elle se réfère à des mots lexicalement différents mais portant un même sens. Ceci implique qu'il est impossible de trouver des parties des documents représentés par un mot M1 synonyme d'un mot M2, où M2représente une requête.

II.4.2 Préliminaires : définitions et notations

Mot, terme et concept

Définition 1 : Mot

C'est la plus petite unité signifiante qui peut exister de façon autonome dans une phrase, dans un texte écrit il est délimité par des blancs ou par des signes de ponctuation.

Définition 2 : concept

Un concept est une unité de pensée qui désigne un sens, une idée, de façon non ambiguë.

Définition 3 : Terme

Un terme est un mot (terme simple) ou une séquence de mots (terme complexe) qui symbolise un concept. Un terme peut dénoter de façon alternative plusieurs concepts. De même un concept peut être dénoté par plusieurs termes.

Définition 4 : Objet

Élément de la réalité qui peut être perçu ou conçu, les objets peuvent être matériels (par exemple: la Tour Eiffel,) ou immatériels (par exemple: la liberté).

Similarité sémantique :

La similarité sémantique est une notion définie entre deux concepts, Dans cette section une mesure de similarité entre concept est utilisée.

La désambigüisation :

Un terme peut dénoter différents concepts, la désambigüisation consiste à déterminer celui qui correspond au mieux à ce terme dans son contexte d'énonciation. Un contexte est défini comme un ensemble de termes qui apparaissent ensemble dans une phrase d'un texte.

La pondération des concepts :

Dans la recherche des documents semi-structurés, le poids d'un terme est exprimé de manière locale au seing du document (ou l'élément) ou de manière globale au seing de la collection.

Le poids d'un terme est généralement évalué selon trois dimensions :

- ***tf :*** la fréquence du terme dans l'élément ;
- ***idf :*** la fréquence inverse du document pour le terme ;
- ***ief*** *: la fréquence inverse de l'élément pour le terme.*

Appariement Nœuds / Requête :

L'appariement nœuds/requêtes a pour but d'attribuer des scores de pertinence aux éléments d'un document (les nœuds de type texte et nœuds de type élément ans l'arbre XML).

Pour l'utilisation des ressources sémantique dans la RI classique plusieurs approches ont été proposées. **[Baziz et al, 2007], [Khan et al., 2004],** système **OntoSeek [Guarin.o et al., 1999].**

II.5 Conclusion :

Dans ce premier chapitre, nous avons présenté les méthodes, modèles et concepts fondamentaux de la RI « classique ».

Les modèles décrits dans ce chapitre fonctionnent généralement sur tous les formats (structurés, non structurés et balisés, non balisés), mais ils ne permettent d'exploiter que le contenu textuel du document. Or, avec l'avènement du web, de plus en plus de documents sont structurés ou semi-structurés de format XML. La structure apportant des informations supplémentaires qu'il est utile d'exploiter en vue d'affiner la recherche. Il devient donc impératif de réfléchir à des SRI capables de tenir compte de la structure en plus du contenu pour une recherche efficace. Dans le chapitre suivant, nous présentons des modèles de RI traitant les documents comportant du contenu et de la structure.

Chapitre 2 : Recherche d'information Semi-Structurée

I. Introduction :

Le type des documents mis à disposition des utilisateurs évolue. Du simple document texte " plat ", on assiste aujourd'hui à la généralisation des documents structurés ou semi-structurés, du SGML au HTML et autre XML qui a vu son importance augmenter grâce à l'expansion d'Internet.

La communauté de la 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. Mais la coexistence de l'information structurelle et de l'information de contenu dans ce type de documents soulève de nouvelles problématiques parmi lesquelles :

- L'indexation des documents et la pondération des termes qui doit prendre en considération l'information structurelle.
- L'interrogation des documents qui doit permettre d'exprimer des besoins diversifiés portant sur le contenu et/ou sur la structure
- Le modèle de recherche où lors de l'appariement on doit aussi prendre en compte les conditions structurelles…

Pour répondre à ces nouvelles problématique, la communauté en RI a du parfois adapter les techniques de la RI Classique (vectoriel, probabiliste, de langue, etc.), et d'autres fois proposer de nouvelles.

Dans ce chapitre, nous présentons la notion documents structurés et documents semi-structurés. Nous nous intéressons en particulier aux documents XML objets de notre étude. Nous présentons ensuite les problématiques rencontrées en recherche d'information semi-structurée, ainsi que les techniques et modèles proposés par la communauté de la RI pour solutionner ces dernières.

II. Document et Structure :

Les documents structurés et semi-structurés sont des documents dans lesquels le contenu est organisé par une suite de balises hiérarchiquement imbriquées définissant sa structure logique.

Une balise encadre une zone documentaire appelée élément qui peut contenir elle aussi d'autres éléments. Au niveau le plus bas, un élément encadre une zone de texte élémentaire dite atome. Un atome peut être du contenu textuel ou multimédia. Il est non structuré. La balise est identifiée par un nom ou label qui n'est pas spécifique à un document donné mais peut être attribué à plusieurs autres documents.

Il y a une différence entre documents semi-structurés et documents structurés. Les premiers sont hétérogènes et peuvent contenir des 'contenus mixtes' (un élément contenant du contenu textuel et un ou plusieurs autres éléments), alors que les seconds ne contiennent pas de contenu mixte et ont une structure assez régulière. Les documents XML, auxquels nous nous intéressons dans le cadre de notre travail, sont des documents semi-structurés.

III- Les documents XML :

III.1- Le langage XML :

XML (*Extensible Markup Langage*) ou (langage à balises extensible) est un standard mis en place par le W3C, qui comporte les fonctionnalités du SGML mais d'une façon réduite afin de le rendre utilisable sur le web. Le langage XML est un format de description de données, permettant de structurer le contenu des documents en les marquant par des balises. Le langage XML est dit générique car le choix des noms des balises et des attributs ainsi que leur organisation est laissé au libre choix du créateur.

III.2- Structure des documents XML :

La structure du document XML décrit les différentes parties qui le composent. Tout document XML est composé de 3 parties :

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

Un exemple de document XML est présenté en figure 2.1suivante.

```
<? Xml version= «1.0 » encoding = » ISO-8859-1» ? >
<!−−Exemple de fichier XML d´écrivant un article scientifique −−>
<article annee=”2003”>
<en-tete>
<titre>Recherche d’information sur le web : la grande révolution</titre>
<auteur>André Dupont</auteur>
</en-tete>
<corps>
<section>
<sous-titre>Histoire de l’hypertexte : des pères fondateurs au World Wide Web</sous-titre>
        <par>Afin de maîtriser les enjeux des systèmes hypertexte, il convient, même...</par>
</section>
<sous-titre>Moteurs de recherche</sous-titre>
<par>On distingue plusieurs types de moteurs de recherche...</par>
</section>
        …
</corps>
</article>
```

Figure 2.1- Exemple d’un document XML.

III. 2.1 Le Prologue :

Il peut contenir une déclaration XML, des instructions de traitement et une déclaration de type document.

a)-Déclaration XML : permet donc d’indiquer la version XML utilisée, le jeu de codage de caractères utilisé et l’autonomie du document comme suit :

? xml version : Version XML utilisée dans le document (version 1.0)

Encoding : Jeu de codage de caractères utilisé ici est : ISO-8859-1 (contient le code ASCII en ajoutant de nombreux caractères accentués occidentaux)

Standalone : Existence ou non de déclarations extérieures au document qui doivent être prises en compte pour le traitement de celui-ci :

- S’il n’y a pas de **DTD** ou si elle est interne, le document est autonome et la valeur de l’attribut ***standalone*** peut être définie à ‘**yes**’.
- Si la **DTD** référencée est externe la valeur de cet attribut doit être définie à ‘**no**’.
- Si l’attribut ***standalone*** est omis, c’est la valeur **‘no’** qui est prise par défaut.

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

b)- 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. Une instruction de traitement (figure 2.2) commence par **< ?** et se termine par **?>**, Les instructions de traitement qui servent le plus souvent, sont la déclaration XML ainsi que la déclaration de feuille de style.

```
< ? xml version = «''1.0''» ?>
```

Figure 2.2 – Instruction de traitement XML.

c)- Déclaration de Type de Document (ou DTD):

La DTD permet d'établir les règles de définition de la structure 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. Bien que facultative, il est souvent intéressant pour un document XML de se conformer à une DTD. Lorsqu'une DTD est associée à un document XML on dit qu'il est valide. Un exemple de DTD est donné en figure 2.3 suivante :

```
< ? xml version="1.0" ?>
< !--DTD pour personne.xml-->
< !ELEMENT personne(nom, prenom,fonction)>
< !ELEMENT nom (# PCDATA)>
< !ELEMENT prenom (# PCDATA)>
< !ELEMENT fonction (# PCDATA)>
```

Figure 2.3 - déclaration de type DTD.

III.2.2 Les Commentaires :

Des commentaires peuvent être insérés dans les documents. Ils sont encadrés par les marques de début **< !--** et de fin **-- !>** de commentaire (exemple en figure 2.4).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.

```
< !-- ceci est un commentaire -- !>
```

Figure 2.4 – un commentaire XML.

III.2.3 L'arbre d'éléments :

La partie essentielle d'un document XML sera toujours formée d'une hiérarchie d'éléments qui dénote la sémantique de son contenu: c'est l'arbre d'éléments. Dans un document XML il n'existe qu'un et un seul élément racine, qui contient tous les autres.

Chaque élément d'un document XML se compose d'une balise d'ouverture, d'un contenu d'éléments et d'une balise de fermeture.

Syntaxe :

```
<nom_balise>contenu-élément</nom_balise>
```

Figure 2.5 – un élément XML.

Un élément peut contenir des données, des références à des entités, des sections littérales et des instructions de traitement. Un élément peut avoir un contenu récursif, c'est à dire qu'il peut contenir une instance du même type d'élément que lui-même. Un élément vide est représenté par <nom/>

Remarque: La balise d'ouverture d'un élément peut inclure des attributs sous la forme de paires nom='valeur' (exemple en figure 2.6). La valeur d'un attribut est une chaîne de caractères encadrés par des apostrophes (' ') ou par des guillemets (« »).

```
<employer id = « chf-677 »>Abd Ennour </employer>
```

Figure 2.6 – attribut XML.

III.3 Les parseurs XML:

Un parseur XML est un analyseur, qui permet d'accéder aux documents XML pour pouvoir en extraire les données et les traiter, ainsi éventuellement de vérifier la validité d'un document. On distingue deux principaux parseurs XML : DOM et SAX.

- DOM : représente le document XML sous forme d'arbre de nœuds typés (de types élément, attribut, ou texte) et liés par des relations de structure (père-fils). Le contenu textuel est situé dans les nœuds feuilles. La **figure 2.7** montre un exemple de document XML et sa structure d'arbre DOM.

```
<doc année="2009">
<titre> Approche de RI conceptuelle </titre>
<sect1 >
 <par1> L'indexation conceptuelle se réfère à la construction...< par1>
<par2> Une ressource sémantique est utilisée..... </par2>
</sect1>
<sect2 >Un document est décrit par un arbre de nœuds...
<par3> DOM est une spécification du W3C dont l'objectif.... </par3>
 </sect2>
</doc>
```

Figure 2.7 – Document XML

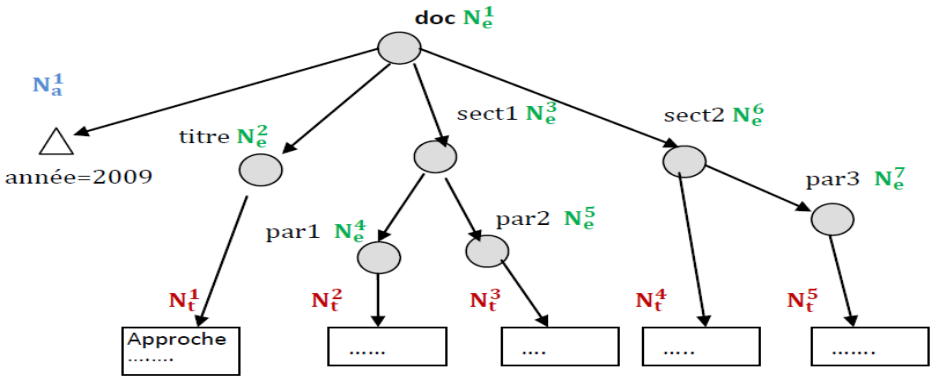


Figure 2.8 – représentation DOM

- SAX : est un parseur orienté évènements. Il traite le document XML au fur et à mesure de la rencontre des élèments de documents, et pour chaque élèmnet rencontré un évènement est envoyé (ie. une fonction est appellée). Les évènements envoyés peuvent être des balises ouvrantes, des balises fermantes, des élèments texte, etc…

Remarque : Le traitement d'un document XML avec DOM est plus facile car l'information est facilement exploitable, mais l'inconvénient est qu'elle occupe de l'espace mémoire surtout si le document est volumineux. Par contre, le parseur SAX utilise moins de mémoire car il n'accumule aucune donnée dans une structure, mais il nécessite des traitements supplémentaires pour traiter un document XML (par exemple : sauvegarder des données au moment de parcourir un nœud d'un document pour lés traiter plus tard).

IV. Problématiques de la Recherche d'Information semi-structuré

La Recherche d'Information semi-structurée (RIS) est différente de la RI Classique en plusieurs aspects :

- L'organisation de l'information en unités (éléments) plus fines qu'un document, réactualise la granularité de l'information à retourner à l'utilisateur, et c'est plus judicieux de retourner seulement ces éléments, voir un ensemble d'éléments issus de documents différents, et cela pour éviter que l'information utile ne soit trop dispersée.
- L'expression des besoins utilisateur qui peut se porter sur le contenu ou/et la structure.

La figure suivante (**tableau. 2.1**) résume les principaux critères de comparaison entre la RI classique et la RI semi-structurée.

	RI Classique	RI Semi-Structurée
Type de document	Document non structure : document plat	Document semi-structuré
Contenu	Texte seulement	Texte + structure
Unité de recherche	Document	Elément
Type de requête	Contenu	Contenu et/ structure
Langage de requête	Langage libre	Langage structuré

Tableau 2.1 – comparaison entre la RI Classique et RI Semi-structurée.

Pour prendre en considération ces différents aspects, les techniques de la RI Classique doivent être adaptées et pourquoi pas proposer de nouvelles techniques de recherche d'information semi-structurée qui puissent répondre aux problématiques particulières suivantes:

1. La première problématique concerne l'indexation : la structure du document entrant en jeu dans le processus d'indexation, les questions suivantes se posent **[sauvagnat, 04]:**
   * Comment indexer la structure des documents ?
   * Que doit-on indexer de la structure des documents ?
   * Comment relier cette structure au contenu du document ?
   * Quels sont les paramètres à considérer pour la pondération des termes d'indexation ?
   * Quelles structures d'index utiliser ?

2. La seconde problématique concerne les langages d'interrogations qui doivent permettre d'utiliser des requêtes orientées contenu et/ou structure et d'une manière simple. Avec l'absence de la structure dans les requêtes, les systèmes

doivent décider de la granularité idéale de l'information à renvoyer à l'utilisateur, les conditions de la structure donnent des indications sur le type d'élément à renvoyer.

3. La dernière problématique concerne les modèles de recherche et de tri des unités d'information. Cette problématique est liée au score de pertinence attribué à l'unité d'information (un nœud de document XML) vis-à-vis d'une requête, la pertinence est évaluée selon les deux notions : l'exhaustivité (l'élément doit contenir toutes les informations requises par la requête) et la spécificité (le contenu de l'élément concerne la requête), en plus l'élément renvoyé se doit d'être auto-explicatif (l'élément peut être compris indépendamment du reste du document). Cependant en prenant compte ces trois notions (l'exhaustivité, la spécifié et l'auto-explicativité) l'élément à renvoyer doit être *la plus petite unité d'information cohérente pertinente pour la requete* **[Piwowarski, 03].**

V. Indexation des documents semi-structurés

Dans ce qui suit, nous proposons de décrire les approches d'indexation des documents XML, ainsi que les structures d'index utilisées.

V.1. Approches d'indexation des documents XML

Les approches d'indexation des documents sont classifiées en approches d'indexation basées sur le contenu et approches d'indexation basées sur la structure.

V.1.1. Approches basées sur le contenu ou indexation de l'information textuelle :

L'unité d'information pertinente dans les documents semi-structurés correspond à l'élément XML. Ainsi, l'indexation du contenu textuel d'un document semi-structuré consiste à indexer les éléments du document. Cependant, si beaucoup considèrent les éléments comme des unités disjointes sans connexion entre elles, d'autres propagent certains termes ou tous les termes d'un élément donné vers ses ancêtres (technique dite des sous arbres imbriqués). On distingue ainsi deux classes d'approches d'indexation du contenu :

1. **Les approche par sous arbres imbriqués** : elles indexent le contenu textuel en attribuant un poids, puis en propageant chaque terme de l'index vers ses ancêtres en diminuant son poids selon la distance entre le nœud qui le contient et celui vers lequel le terme est propagé. L'index contient des informations redondantes car les nœuds de l'index sont imbriqués les uns dans les autres.

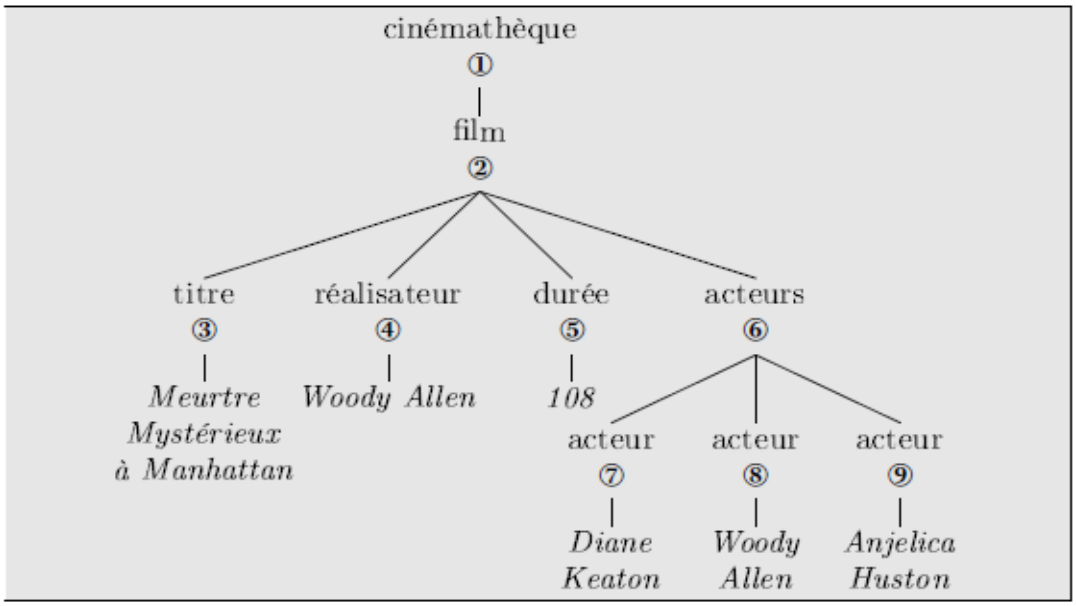
2. **Les approches par unités disjointes** : elles décomposent le document en unités disjointes, c'est-à-dire elles n'indexent que les éléments contenant du texte sans propager le contenu aux ancêtres.

V.1.2 Approches basées sur la structure ou indexation structurelle :

L'indexation de la structure passe par le choix du mode de représentation des éléments structurant les documents. L'information structurelle peut être indexée selon des granularités variées, c'est-à-dire que toute l'information structurelle n'est pas forcement utilisée dans le processus d'indexation. On trouve :

Indexation basé sur les champs :

On restreint la structure de l'arbre au nœud feuille qui la compose en associant à chaque balise les termes indexés qu'elle contient avec leur fréquence, l'inconvénient est qu'elle ne sauvegarde pas les relations hiérarchique entre les nœuds.



Terme	Sans propagation	Avec propagation
Meurtre	Titre	cinémathéque, film, titre
Allen,	réalisateur, acteur	cinémathèque, réalisateur, film, titre, acteurs, acteur

Tableau 2.2 – exemple d'indexation basé sur les champs.

Indexation basée sur les chemins :

Les noms de balises sont remplacés par les chemins d'accès où leur notation est basé sur XPath, l'avantage de cette méthode est d'accélérer le processus de recherche pour une information se situant dans plusieurs balises qui portent le même nom .L'inconvénient est qu'elle ne représente pas les relations de descendance des différents éléments du document XML.

Terme	Sans propagation	Avec propagation
Meurtre	(/cinémathèque/film/titre)	(/cinémathèque) (/cinémathèque/film) (/cinémathèque/film/titre)
Allen	(/cinémathèque /film/réalisateur)	(/cinémathèque) (/cinémathèque/film) (/cinémathèque/film/réalisateur) (/cinémathèque/film/acteurs/acteur) (/cinémathèque/film/acteurs)

Tableau 2.3 – indexation basés sur les chemins.

Indexation basés sur les Arbres :

La spécificité de cette méthode est d'attribuer un identifiant à chaque nœud d'arbre, pour permettre de le localiser d'une façon précise et trouver les relations hiérarchiques entre les éléments, L'identifiant unique peut également être, tout simplement, le chemin d'accès (XPath absolu, avec le numéro des éléments) de l'élément.

Terme	Sans propagation	Avec propagation
Meurtre	(/cinémathèque [1]/film [1]/titre [1]) (3)	(/cinémathèque [1]) (1) (/cinémathèque [1]/film [1]) (2) (/cinémathèque [1]/film [1]/titre [1]) (3)
Allen	(/cinémathèque [1]/film [1]/réalisateur [1]) (4)	(/cinémathèque [1]) (1) (/cinémathèque [1]/film [1]) (2) (/cinémathèque [1]/film [1] /réalisateur [1]) (4) (/cinémathèque [1]/film [1]/acteurs [1]/acteur [2]) (7) (/cinémathèque [1]/film [1]/acteurs [1]) (8)

Tableau 2.4- indexation basés sur les arbres.

Nous présentons si dessous quelques méthodes d'identification des nœuds :

1. **Codage Préordre / postordre :**

Le codage pré/post ordre est une paire (**pré, post**) attribuée à un nœud, où '**pré**' est la valeur du préordre et '**post**' est la valeur du postordre du nœud.

- **La valeur préordre** (parcours préfixe) : consiste à parcourir l'arbre de haut en bas, de gauche à droite et d'attribuer un numéro à chaque nœud rencontré et ensuite l'incrémenter.

- **La valeur postordre** (parcours postfixe): est de parcourir l'arbre de bas en haut, de gauche à droite et d'attribuer un numéro à chaque nœud rencontré puis l'incrémenter.

Exemple :

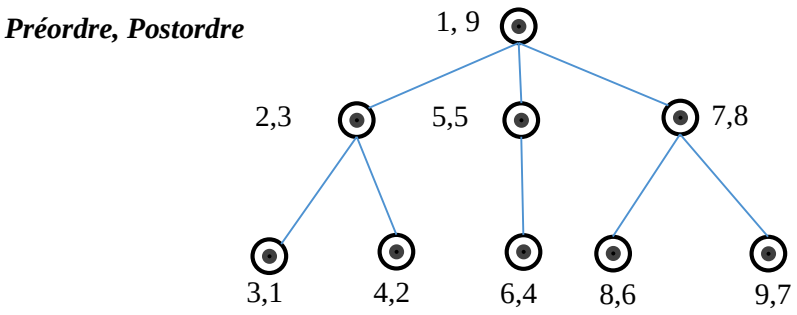


Figure 2.9 : numérotation post/pré ordre

Avec cette annotation on peut identifier les rapports ancêtres descendants avec les relations suivantes :

n_1 est ancêtre de n_2 : pré (n_1) <pré (n_2) et post (n_1) > post (n_2).

L'inconvénient de cette méthode et de recalculer tous les identifiants des nœuds pour chaque nouvelle insertion d'un nouveau nœud.

2. Nombre de frères (*Sibling numbers*)

Pour identifier un nœud de document sans équivoquement, le chemin de la racine menant à ce nœud est représenté par [Sacks,94] comme suit: Pour chaque nœud sur le chemin, le nombre de ses frères (fils du même parent) précédents du même type (avec la même étiquette) est calculé. C'est son nombre de '*Siblings'*. Les étiquettes entrantes de tous les nœuds sur le chemin de la racine forment ensemble un chemin d'étiquette. Chaque étiquette est annotée avec le nombre de siblings du nœud qu'il mène. Le chemin étiquette/sibling résultant est une sorte de chemin de position. La Figure 2.10 montre un exemple d'identification 'sibling numbers'.

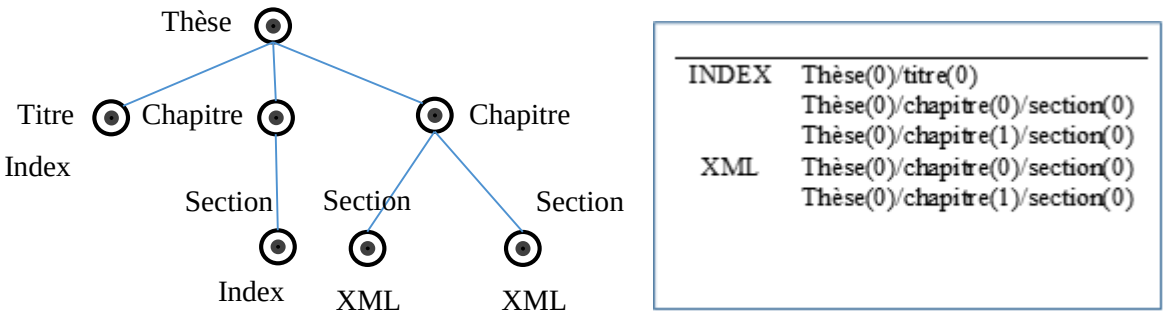


Figure 2.10 exemple d'identification ' ***Sibling numbers***'

Comme on peut le voir dans La **Figure 2.10**, les enfants du même nœud avec des étiquettes identiques ne peuvent pas être confondus. Par exemple, les deux nœuds chapitre directement au-dessous de la racine du document sont mentionnés comme /Thèse/chapitre [0]'' et /Thèse/chapitre [1], respectivement.

V.4 Pondération des termes d'indexation :

Dans le cadre des documents XML, l'importance du terme au niveau de l'élément vient s'ajouter à celui de son importance au niveau local au sein du document et celui globale au sein de la collection, donc les formules **tf*idf** sont adaptées pour prendre en considération la force discriminatoire d'un terme **t** pour une balise **b** relative à un document **d** qui est présenté sous forme **tf.itdf (Term Frequency-Inverse Tag and Document Frequency)**.

Les paramètres pris en compte pour calculer l'importance du terme au niveau de l'élément sont:

- Le nombre de nœuds qui contiennent le terme.
- Le nombre de nœuds dans la collection.
- La longueur du nœud qui contient le terme.
- La longueur moyenne des nœuds feuilles dans la collection.
- La fréquence du terme dans le nœud

Quelque formule de pondération :

$$w_{T,E}^{t} = tf_{T,E} * idf_{T}^{t} = \frac{freq_{T}(E)}{\max(freq(E))} * (\log\left(\frac{|E^{T}|}{n_{T}}\right) + 1)$$

Formule 2.1

Avec :

- **$Freq_T$ (E) :** est le nombre d'occurrences de **T** dans **E**.
- **maxfreq(E)** : est le nombre maximal d'éléments de la collection possédant la même étiquette que **E**.
- **$|E^T|$:** est le nombre d'éléments de type **t** et **n_T** le nombre d'éléments contenant **T**. [*Schileder,02*]

Azevedo et al. [2004]

$$W_D(t_i, e) = log(tf(t_i, e)) * log(N * idef(t_i, e))$$

Formule 2.2

Où :

- *tf (ti; e) :* calcule le nombre d'occurrences du terme *ti* dans **e**.
- *idf(ti; e) :* est l'inverse du nombre d'éléments contenant *ti*.
- ***N***: est le nombre total d'éléments dans la collection.

Hatano & al *[Hatano & al, 2002]*

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

Formule 2.3

- **w(t,e)** : est le poids du terme **t** dans l'élément **e**.
- **tft(e)** : est la fréquence du terme **t** dans l'élément **e**.
- **T** : est l'ensemble du vocabulaire.
- **E** : est l'ensemble des éléments du document considéré.
- **N** : est le Nombre total de documents dans le corpus (collection).
- **nt** : est le Nombre total de documents contenant le terme **t**.

Sauvagnat [Sauvagnat, 2005]

- w_{inf} = poids du terme ***i*** dans le nœud feuille n_f est calculé par la formule suivante

$$W_i = tf_i^{nf} * Ief_i$$

Formule 2.4

Ou :

- tf_i^{nf} : est la fréquence du terme ***i*** dans le nœud feuille ***nf***
- ***ief*** ***(*** *Inverse Element Frequency*) se calcule comme suit :

$$\mathbf{ief_i = \log\left(\frac{N}{n+1}\right)+1}$$

Formule 2.5

Où

- ***N :*** mesure le nombre total de nœuds feuilles
- ***n*** : est le nombre des nœuds feuillent contenant le terme ***i***.

Le poids de terme : dans les nœuds feuille est calculé par la formule suivante :

$$Wi,j = tf_{ij} * ief_j = \frac{nbocc(t_i,f_j)}{nbterm(f_i)} * log(\frac{|F_c|}{|nf_j|})/log(|F_c|).$$

Formule 2.6

- $\boldsymbol{nbocc(t_i,f_j)}$: est le nombre d'occurrences de terme t_i dans le nœud feuille f_j.
- $\boldsymbol{nbterm(f_i)}$ **:** est le nombre de termes dans le nœud feuille.
- $|F_c|$: est le nombre de nœuds feuilles de la collection.
- nf_j : est le nombre de nœuds feuilles contenant le terme t_i.

Sauvagnat [Sauvagnat, 2005]

$$Wi,j = tf_{ij} * ief_j = \frac{nbocc(t_i,f_j)}{nbterm(f_i)} * log(\frac{N}{n+1})+1.$$

Formule 2.7

- ***N***: est le nombre de nœuds feuilles de la collection.
- ***n*** : est le nombre de nœuds feuilles contenant le terme.

[Sauvagnant ;2006]

$$W_{i,j} = tf_{ij} * ief_j = nbocc(t_i, f_j) * log\left(\frac{|F_c|}{|nf_i|}\right)$$

Formule 2.8

- $nbocc(t_i, f_j)$: est le nombre d'occurrences du terme t_i dans le nœud feuille f_j.
- $|F_c|$: est le nombre de nœuds feuilles de la collection.
- nf_j : est le nombre de nœuds feuilles contenant le terme t_i.

VI- Interrogation :

Le processus d'interrogation est composé de trois étapes :

- La formulation du besoin par l'utilisateur peut être effectuée de plusieurs manières (que nous citerons plus bas), puis cette requête sera représentée avec une structure semblable à celle des documents pour pouvoir les apparier.
- L'appariement requête/élément : consiste à attribuer des scores de pertinence à chaque élément du document par apport à la requête grâce à une fonction d'appariement.
- Le tri par ordre de pertinence et la présentation des résultats.

Les requêtes utilisateur peuvent être exprimées de deux manières :

- Requêtes orientées contenu : consiste à introduire de simples mots clés où le système fera une recherche complète sur la collection en parcourant tous les éléments.
- Des requêtes orientées contenu et structure : où l'utilisateur peut poser des conditions sur la structure soit d'une manière précise ou vague en ayant déjà des connaissances préalables sur la structure des documents.

VII. Les modèles de recherche :

Dans cette section, nous présentons le modèle vectoriel étendu adapté aux documents semi-structurés.

Les mesures de distance dans un espace vectoriel sont utilisées pour calculer la similarité entre l'élément et la requête qui sont représentés Par des vecteurs de termes pondérés.
En générale les approches du modèle vectoriel propagent les termes des nœuds feuilles dans l'arbre du document.

La similarité d'un nœud ***n*** à une requête **q = {t1, t2, ...tT}** est exprimée selon la formule suivante :

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

Formule 2.9 : fonction de similarité du modèle vectoriel étendu

Où :

- **α(T) :** est un facteur permettant de prendre en compte le type du nœud,
- **S :** est le nombre de nœuds enfants $\mathbf{n_k}$de **n**,
- **B :** est un paramètre permettant d'assurer que le nombre d'enfants n'introduit pas un biais dans la formule.

La fonction **cosm** est définie de la façon suivante :

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

Formule 2.10

Où :

- w_i^q et $\mathbf{w_i^q}$ respectivement le poids du terme $\mathbf{t_i}$ dans la requête **q** et dans le nœud **n**,
- **|n|** le nombre de termes dans le nœud **n**.

(Hlaoua et al., 2006), (Sauvagnat, 2005) :

Le modèle est basé sur une méthode de propagation de pertinence. Des valeurs de pertinence sont d'abord calculées 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 en prenant compte de la distance séparant les nœuds ancêtre et descendant. La pertinence d'un nœud ***nf*** pour une requête composée de ***m*** termes ***q*= {*t*1, *t*2,..., *tm*}** est exprimée selon l'équation suivante :

$$RSV(Q,nf) = \sum_{j=1}^{j=m} w_j^Q * w_j^{nf}$$

Formule 2.12

Où :

- $\mathbf{w_j^Q = tf_j^Q * ief_i}$ est le poids du terme t_j dans Q
- $\mathbf{w_j^{nf} = tf_j^{nf} * idf_j * ief_j}$ est le poids du terme tj dans nf.

Avec : $ief_i = \log(\frac{N}{n})$

Où :

- **N** : est le nombre total des nœuds feuilles
- **n** : est le nombre des nœuds feuilles contenant le terme **i**.

La valeur de pertinence d'un nœud interne **n** est calculée comme suit :

$$RSV(Q,n) = \left|F_n^p\right| * \sum_{k=1..F_n} \alpha^{dist(n,nf_k)-1} * RSV(q,nf)$$

Formule 2.13

Où :

- $\mathbf{nf_k}$: sont les nœuds feuilles descendants du nœud n,
- **dist(n, $\mathbf{nf_k}$)** : est la distance entre le nœud **n** et le nœud feuille $\mathbf{nf_k}$ dans l'arbre ,
- $\mathbf{|F_n^p|}$: est le nombre de nœuds feuilles descendants du nœud **n** ayant un score différent de zéro,
- $\mathbf{F_n}$: est le nombre total de nœuds feuilles descendants de **n**,
- $\mathbf{\alpha \in [0,1]}$: est un paramètre qui quantifie l'importance de la structure dans l'évaluation du score de pertinence.

VIII. La recherche conceptuelle en RI semi-structurée :

La recherche d'information sémantique dans les documents semi-structurés se base essentiellement sur la représentation de documents et de la requête par une liste de concepts.

Les systèmes d'indexation et de recherche conceptuelle nécessitent une ressource sémantique (ontologie, thesaurus,...) afin d'extraire les concepts des termes des textes et de la requête et un modèle de mesure de similarité entre les concepts.

Les documents semi-structurés sont caractérisés par la présence d'une structure organisant le contenu textuel de ces documents. Ainsi, les systèmes conceptuels

d'indexation et de recherche de documents semi-structurés se divisent en trois approches correspondant à trois manières de tenir compte de la structure et du contenu textuel dans l'indexation **(Harrathi et al, 2010).**

Approche orientées contenu :

Dans les approches orientées contenu, la structure n'est pas prise en considération. Ainsi elles proposent d'indexer uniquement le contenu textuel des nœuds feuilles puis à propager les poids des termes d'index vers les nœuds internes ou agréger le contenu vers les nœuds internes.

- ✓ Dans **(Zargayouna, 2005)**, l'auteur propose une approche d'indexation sémantique des documents XML. Dans cette approche, les documents sont représentés par des vecteurs de termes.

Pour chaque balise dans le document XML un vecteur lui est associé. Les poids des termes sont calculés en fonction de leur distribution dans les balises. Ce poids noté **sem W(t,b,d)** auquel est ajouté les similarités conceptuelles des co-occurrents dans la même balise est de la manière suivante :

$$\text{Sem W(t,b,d)}=TF\_IDTF(t,b,d)+\frac{\sum_{i=1}^{j=n} sim(t,t_i)*TF\_ITDF(t_i,b,d)}{n}$$

Formule 2.14

Où :

- t_i:est un terme qui appartient à l'ensemble des n termes dans la balise b
- *TF-ITDF* (*Term Frequency–Inverse Tag and Document Frequency*) : est le poids initial attribué aux termes.
  TF-ITDF permet de calculer la force discriminatoire d'un terme t pour une balise b relative à un document d
- (t, t_i) : est la somme des similarités sémantique des concepts dénotant les termes t et t_i.

Dans cette approche l'auteur permet de représenter les documents par des vecteurs de concepts et cela en utilisant la ressource sémantique wordnet.

Une mesure de similarité sémantique est définie en se basant sur la mesure wu-palmer **[Zargayouna, 2005],**

$$\mathbf{sim_{wPalmer}}(\mathbf{c_1},\mathbf{c_2})=\frac{\mathbf{2}*\mathbf{prof}(\mathbf{c})}{\mathbf{dist}(\mathbf{c_1},\mathbf{c})+\mathbf{dist}(\mathbf{c_2},\mathbf{c})+\mathbf{2}*\mathbf{prof}(\mathbf{c})}$$

Formule 2.15

Où :

- ***c*** : est le concept le plus spécifique qui subsume les deux concepts c_1 et c_2 .
- ***prof(c)*** : est le nombre d'arcs qui sépare ***c*** de la racine et ***dist(c_i, c)*** le nombre d'arcs qui séparent c_i de ***c***.
- Cette mesure est utilisée pour la désambiguïsation des sens des termes.

✓ **[Harrathi et al, 2010]**, dans cette approche les nœuds (élements XML) sont représentés par des vecteurs de concepts pondérés, ces concepts sont extraits d'une ressource sémantique.
Pour définir les concepts adéquats pour chaque nœud la désambiguïsation est utilisée.

Cette désambiguïsation se base sur la similarité sémantique entre les différents concepts présents dans le même nœud

Dans cette approche l'auteur permet aussi l'appariement entre nœud/requête qui utilise la méthode « cosinus » adaptée pour prendre en considération le contexte sémantique entre les nœuds et la requête.

Approches orientées structure :

Les approches orientées structure proposent d'indexer uniquement la structure **[Kim et al, 2005], [Taha et al, 2008]**. Les noms des éléments (les noms des balises) dans les documents XML sont indexés par des Concepts. Par exemple les mots « car » et « automobile » sans indexés par le même concept. L'intérêt de cette approche est de retourner les éléments présents dans les documents portant le même sens que ceux présents dans la requête.

Par exemple : pour la requête chercher un élément dont le nom est « automobile » le système de recherche peut retourner tous les éléments dont le nom est « automobile » et « car ».

- **CXLEngine [Taha et al, 2008]**. Ce système utilise une ontologie de noms (ontology label) pour faire correspondre les noms des balises aux concepts de l'ontologie.

Approches orientées structure et contenu :

Consistent à indexer la structure et le contenu textuel par des concepts en utilisant une ontologie. Où un document XML est considère comme un ensemble de paires (concept élément, valeur) où « valeur » désigne l'index du contenu textuel qui est représenté par un ensemble de concepts pondérés. Le score de pertinence attribué a un élément est

calculé en utilisant une fonction de similarité enter l'ensemble de concepts de la requête et la valeur de l'élément.**[harrathi .al 2010]**

- Le système de recherche XXL [**Schenkel et al, 2005], [Schenkel et al, 2003]**, permet l'interrogation de documents XML. Le moteur de recherche XXL présente une architecture s'appuyant sur 3 structures d'index **[Schenkel et al, 2005]**.

    - **Index de la structure :** les noms des éléments sont indexés par des concepts. Cet index permet la navigation dans la hiérarchie d'arbre d'un document, ce qui permet aussi de calculer la distance entre deux nœuds.
    - **Index du contenu :** indexe le contenu, son objectif est de retrouver le terme là où il apparait. Le poids des termes est calculé par le **TF-IEF** (Term Frequency -Inverse Element Frequency).
    - **Index ontologie :** permet de retrouver des termes reliés sémantiquement à un terme donné. Il calcule pour cela une similarité qui peut être restreinte à un certain type de liens. A partir de cette valeur une mesure de similarité peut être calculée entre deux concepts **[harrathi ,.al ;2010].**

La mesure de similarité $\mathbf{sim_p}$ entre deux concepts est calculée suivant un chemin, $\boldsymbol{p = (n_0, \dots, n_k)}$ comme suit **[Schenkel et al, 2003]**:

$$\mathbf{sim_p(c_1, c_2) = \prod_{i=1}^{|p|-1} poids(n_i - n_{i+i})}$$

Formule 2.16

Où :

- $\mathbf{|p|}$ est la longueur du chemin $\mathbf{p}$, poids $\mathbf{(ni, -n_{i+i})}$ dénote le poids de l'arc $\boldsymbol{e = (ni, -n_{i+i})}$.

Le calcul de la similarité entre deux concepts est calculé par cette formule:

$$\mathbf{sim}\ (c1, c2) = \max \{\ \mathbf{sim_p}(c1, c2\ )\}$$

Formule 2.17

IX. La compagne INEX

IX.1. INEX :

INEX *(Initiative for the Evalution of the XML Retrieval)* est actuellement la seule compagnie d'évaluation des différents systèmes de recherche d'information pour les documents XML. Le but principal d'INEX est de promouvoir l'évaluation de la recherche sur les documents XML en fournissant des collections de test (par exemple, une collection d'articles IEEE en 2002 et 2005, des documents Wikipédia en 2006-2010, et une collection de livres numérisés sous licence de Microsoft à partir de 2007) des mesures d'évaluation uniformes et un forum pour toutes les organisations qui y participent à comparer leurs résultats et à améliorer leurs stratégies. La collection de test est constituée d'un ensemble de documents XML, un ensemble de besoin en information (requête) et des réponses à ces besoins informationnels.

Nous utilisons la collecte, les requêtes fournies par INEX et notre logiciel pour produire les éléments XML pertinents vis-à-vis de notre système, les résultats retournés sont évalués en utilisant les mesures d'évaluation de pertinence fournies par INEX.

IX.2. La structure du document INEX :

A partir de 2009, INEX utilise une nouvelle collection de documents basée sur Wikipedia. La collection INEX 2009 est d'environ 50,7 Go en taille repartie sur 995 parties. Cette collecte est de 8,6 fois la taille de la collecte prévue en 2007 et en 2008.

Cette collection contient un certain degré de structure uniforme comme celle des documents XML, mais ne suivant pas strictement la DTD (Document Type Définition) d'où la collection est considérée comme semi-structurée .La collection actuelle a plus de 30.000 balises dont la plus part sont éliminées lors de l'analyse. La structure d'un document INEX 2009 est comme suit :

```
<? Xml version = "1.0" encoding = "UTF-8"?>
<! - Généré par CLiX / Wiki2XML [MPI-INF, MMCI @ UdS] $ LastChangedRevision: 92
$ Sur 17.04.2009 3:27:08 [mciao0828] ->
<! DOCTYPE article SYSTEM "../ article.dtd">
<Article xmlns: xlink = "http://www.w3.org/1999/xlink">
<Header>
<Title> Portail: football anglais image / Sélectionné / 21 </ title>
<Id> 16183995 </ id>
<Révision>
<Id> 196856544 </ id>
<timestamp> 2008-03-08T21: 32: 02Z </ timestamp>
<Contributor>
<username>Jardin</ username>
<Id> 5019622 </ id>
</ Contributor>
</ Révision>
<Catégories>Portail de football<category> Anglais photos sélectionnées </ category>
</ Catégories>
</ Header>
<bdy>
```

Figure 2.11: exemple d'un document de la collection d'INEX 2009.

IX.3. La structure de la requête (topic) INEX :

```
<topic id="2009001" ct_no="186">
<title>Nobel prize</title>
<castitle>//article[about(., Nobel prize)]</castitle>
<phrasetitle>"Nobel prize"</phrasetitle>
<description>information about Nobel prize</description>
<narrative>
        I need to prepare a presentation about the Nobel prize. Therefore, I want to collect
        information about it as much as possible. Information, the history of the Nobel prize or the
        stories of the award-winners for example, is in demand.
</narrative>
</topic>
```

Figure 2.12 – exemple d'une requête INEX 2009

Les requêtes INEX se divisent en deux catégories principales :

- Les CO (content only) sont des requêtes courtes composées de simples mots clés.
- Les CAS (content and structure) ces requêtes contiennent des contraintes sur la structure des documents

Chaque requête est constituée d'un ensemble de champs qui explique le besoin informationnel de l'utilisateur

- Le champ <Castitle> : est une courte explication du besoin informationnel en précisant tout changement structurel. Aussi connu comme le contenu de la structure (CAS), Il donne la forme structurée de la requête ;

- Le champ <Title> : donne une courte explication du besoin d'information en utilisant des mots clés simples. Il sert comme résumé au contenu informationnel.

- Champs <Description><Narrative> : donne une explication détaillée de la nécessité de l'information et de ce qui fait un élément pertinent. Le <Narrative> ne devrait pas seulement expliquer l'information recherchée mais aussi le contexte et la motivation de la nécessité de l'information, il est important que cette description soit claire est précise. La présence de ces deux champs est optionnelle.

IX.4. La tache ad-hoc [sauvagnat, 06]:

La tâche principale d'INEX est la tâche de recherche *ad hoc*. Comme en recherche d'information traditionnelle, la recherche ad hoc est considérée dans INEX comme une simulation de l'utilisation d'une bibliothèque, où un ensemble statique de documents est interrogé avec des besoins utilisateurs, c'est-à-dire des requêtes. Les requêtes peuvent

contenir à la fois des conditions structurelles ou du contenu et en réponse à une requête, des éléments (et non forcément des documents) peuvent être retrouvés à partir de la bibliothèque. La tâche *ad hoc* se divise en quatre sous-tâches :

IX.4.1. la tâche de recherche exhaustive (*thorough task*) : elle consiste à retourner tous les éléments (éventuellement imbriqués les uns dans les autres) répondant au sujet de la requête et triés par degré de pertinence.

IX.4.2. la tâche de recherche focalisée (*focused task*) elle retourne les éléments les plus précis possibles satisfaisant le besoin en information de l'utilisateur (éléments imbriqués interdits).

IX.4.3. la tâche de recherche de pertinence en contexte (*relevant in context task*) : sert à retourner des éléments dans le contexte d'articles entiers (les unités d'informations sont triées par document).

VI.4.4. la tâche de recherche du meilleur en contexte (*best in context task*) : son rôle est de renvoyer un seul élément par article, à savoir le meilleur point d'entrée dans l'article.

IX.5. Les jugements de pertinence :

Les jugements concernant la pertinence des éléments renvoyés par les systèmes sont effectués par les participants eux-mêmes à l'aide d'une interface mise à disposition en ligne, [**Lalmas et al.,2005**], **[Piwowarski et al.,2008]**.

Pour faire les jugements de pertinence, les dimension d'exhaustivité et de spécificité ont été utilisées **[Malik et al., 2005]** . L'exhaustivité est jugée par l'utilisateur sur une échelle à quatre niveaux **[Lalmas et al., 2005]** :

- **Non exhaustif** : aucune correspondance entre la requête et l'élément ;
- **Partiellement exhaustif** : l'élément ne traite que certains aspects de la requête ;
- **Très exhaustif** : traite tous ou presque tous les aspects de la requête ;
- **Trop petit** : l'élément peut être considéré comme pertinent à certains égards, mais la partie pertinente est trop petite pour être considérée.

Ainsi pour la spécificité une échelle à quatre niveaux est proposée :

- **Pas spécifique** : le sujet de la requête n'est pas un thème de l'élément ;
- **Marginalement spécifique** : le sujet de la requête est un thème mineur de l'élément;
- **Assez spécifique** : le sujet de la requête est un thème majeur de l'élément ;
- **Très spécifique** : le sujet de la requête est le thème de l'élément.

IX.6. Mesures d'évaluation :

Les mesures traditionnelles utilisée dans la RI classique ont été adapté pour prendre en considération le besoin supplémentaire induit par la recherche d'information semi-structurée, nous présentons dans ce qui suit les mesures utilisé dans INEX 2009: **[NAFFAKHI et al, 13]**

– **Précision interpolée selon quatre niveaux de rappel sélectionné :**

$_iP[_jR], j \in [0, 00; 0, 01; 0, 05; 0, 1]$ La précision à un rang **r** est définie Comme suit :

$$\mathbf{P[r]} \frac{\sum_{t=1}^{r} \mathbf{rsize(p_i)}}{\sum_{t=1}^{r} \mathbf{size(p_i)}}$$

Formule 2.18

Avec :

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

Le rappel à un rang **r** est défini comme suit :

$$\mathbf{R[r]} \frac{\sum_{t=1}^{r} \mathbf{rsize(p_i)}}{\mathbf{Trel(q)}}$$

Formule 2.19

Où :

- **Trel(q)** : est la quantité totale du texte pertinent pour une requête **q**.

– Moyenne des précisions moyennes interpolées selon 101 niveaux de rappel (**MAiP**), Elle est utilisée pour la mesure du degré de pertinence dans une recherche ciblée [**LAITANG et al, 13**], Pour **n** requêtes, **MAiP** est calculer comme suit :

$$\mathbf{MAiP} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{t}} \mathbf{AiP}\,(\mathbf{t})$$

Formule 2.20

Où :

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

$$\mathbf{AiP} = \frac{1}{100} \sum_{x=0.0;0.01;\ldots;1.0} \mathrm{iP}[x]$$

Formule 2.21

Où :

- **iP[x]** : est la précision interpolée pour le rappel **x**

Précision interpolée à 1% de rappel, aussi connu comme **iP[0,01]**, est la métrique utilisée pour l'évaluation de la tache ciblée.

Recall : est le terme utilisé pour indiquer la mise en évidences du texte récupéré.

La précision : est le terme utilisé pour indiquer la fraction des textes récupérés.

La formule pour trouver la précision interpolée**iP**[**x**] :

$$\mathbf{iP}[\mathbf{x}] = \begin{cases} \max\limits_{\mathbf{l} \le \mathbf{r} \le |\mathbf{L_q}|} (\mathbf{P}[\mathbf{r}] \cap \mathbf{R}[\mathbf{r}]) & \textbf{if } \mathbf{x} \le \mathbf{R}[|\mathbf{L_q}|] \\ \mathbf{0} & \textbf{sinon} \end{cases}$$

Formule 2.22

Ou :

- $\mathbf{L_q}$: est la liste ordonnée d'éléments
- $\mathbf{|L_q|}$: est la taille de la liste ordonnée d'éléments
- **P[r]** : est la précision au rang **r**
- **R[r]** : est le rappel au rang **r**
- **R[|Lq|]** : est le rappel sur tous les documents récupéré

Pour mesurer l'exhaustivité des résultats obtenus, les formules suivant sont utilisé :

Précision généralisé GP [r]:est la somme des scores de documents jusqu'au document du rang **r**, divisé par le rang **r**.

$$\mathbf{gP[r]}\frac{\sum_{t=1}^{r}\mathbf{S(d_i)}}{\mathbf{r}}$$

Formule 2.23

Où :

- **S (d)** : est le score de document individuel.

Rappel généralisé GR [r]: est le Nombre de documents pertinents récupérés jusqu'au document du rang **r**, divisé par le nombre total de documents pertinents

$$\mathbf{gR[r]} = \frac{\sum_{t=1}^{r}\mathbf{IsRel(d_i)}}{\mathbf{Nrel}}$$

Formule 2.24

Où :

- **Nrel** : est le nombre des documents pertinents pour une requête donnée ***q***,
- **IsR(*di*) :** est un réel qui vaut **1** si le document **d** de position ***i*** est pertinent pour la requête ***q***, sinon **0**.

AgP(AverageGeneralizedPrecision): est La moyenne des précisions généralisée ***AgP*** mesure la performance globale.

Cette mesure est calculée comme suit :

$$\mathbf{AgR[r]} = \frac{\sum_{r=1}^{|L|}\mathbf{IsRel(d_r).gP[r]}}{\mathbf{Nrel}}$$

Formule 2.25

La précision généralisée moyenne arithmétique (MAgP) : est simplement la moyenne de la moyenne des scores de précision généralisée sur tous les sujets, elle permet de mesurer l'exhaustivité des résultats obtenus.

La MAgP : est une généralisation de la précision et du rappel. Cette mesure est utilisée dans le cadre d'une recherche pertinente dans le contexte.

En supposant qu'il y'a **n** sujets :

$$\mathbf{MAgP} = (\mathbf{1/2}) * \sum_{\mathbf{1..n}} \mathbf{AgP(t)}$$

Formule 2.26

X. Conclusion :

Dans ce second chapitre nous avons présenté les modèles et algorithmes fondamentaux utilisés en recherche d'information semi-structurée.

L'information est considérée avec une autre gradualité que le document tout entier, cela vu la dimension structurelle apportée au contenu textuel.

L'adaptation des modèles de la RI traditionnelle à la RI classique voit naitre quelques problèmes tels que : (la pondération des termes, l'attribution des scores de pertinences aux nœuds des documents XML, le traitement des conditions de structure…).

Dans le chapitre suivant nous présenterons une approche qui essayera de résoudre ces problèmes ou du moins les minimiser.

Chapitre 3 : Approche de recherche d'information sémantique dans les documents semi-structurés

I. Introduction :

La recherche d'information sémantique (ou conceptuelle) dans les documents semi-structurés se base essentiellement sur l'indexation des documents et requête par des concepts (on parle alors d'indexation conceptuelle) plutôt que par les simples mots-clés d'index. Pour extraire ces concepts, les systèmes d'indexation et de recherche conceptuelle nécessitent une ressource sémantique (ontologie, thesaurus,...).

Dans le cas des documents semi-structurés, l'indexation conceptuelle consiste à indexer les nœuds de l'arbre d'un document par des concepts. Une approche intéressante serait de représenter ces indexes par un vecteur de concepts **[woods, 1997], [Berry et al., 1999]**. Les dimensions du vecteur sont les concepts d'une ressource sémantique (thesaurus, ontologie…).

L'approche de RI conceptuelle **[Harrathi et al. , 2010]** qui nous intéresse ici est basée sur ce principe. Nous la présentons dans ce qui suit.

II. Approche de RI conceptuelle [Harrathi et al. , 2010]

II.1. Aperçu général :

Etant donnée une collection de documents semi-structurés de type XML, il s'agit de construire un système de RI conceptuelle. Un tel système intègre :

1. un module d'indexation conceptuelle des documents et requête semi-structurés
2. un module d'appariement conceptuel

Nous explicitons ces modules dans les sections suivantes.

III.2 L'indexation conceptuelle :

Le processus d'indexation conceptuelle comporte les étapes suivantes :

1. **Identification des concepts** : cette étape consiste à identifier et sélectionner les concepts représentatifs des nœuds texte et de la requête à l'aide d'une ressource sémantique.
2. **Pondération des concepts :** cette étape consiste à associer des poids aux concepts représentatifs des termes des documents.
3. **Construction de l'index conceptuel d'un nœud** : l'objectif de cette étape, est de construire les vecteurs de concepts indexant les nœuds des documents.

II.2.1 Identification des concepts :

Cette étape nous permet d'extraire les concepts qui structurent l'unité textuelle d'un nœud texte ou d'une requête. Une unité textuelle à l'intérieur d'un document peut être représentée par une phrase, un paragraphe ou un nœud texte dans un document XML.
Le processus d'identification des concepts comporte trois étapes principales **(Figure 1.3):**

- **Etape 1 : Analyse syntaxique et morphosyntaxique.** L'objectif de cette étape est d'analyser le texte afin d'extraire les mots qu'il contient. Ces mots sont ensuite étiquetés syntaxiquement (ie. on associe à chaque mot sa catégorie grammaticale) et lemmatisés.

- **Etape 2 : Enumération des termes candidats.** Cette étape vise à repérer les séquences de mots susceptibles d'être des termes. Un terme est considéré comme le label d'un concept. L'idée est alors de ne retenir que les termes qui ont une correspondance dans la ressource sémantique. Si un terme ne correspond à aucun concept dans la ressource sémantique, il est ignoré.

- **Etape 3 : Recherche des concepts.** Cette étape consiste à identifier les concepts à partir des termes. Si un terme correspond à plusieurs entrées de la ressource sémantique (ie. à plusieurs concepts), il est ambigu. Il faut alors le désambiguïser. La désambiguïsation consiste à retrouver le concept qui définit le mieux le terme d'index considéré dans son contexte d'énonciation (soit la phrase du texte où il apparaît, ou une fenêtre de *n* termes autour du terme considéré).

Approche de désambiguïsation utilisée:

Etant donné un terme t_k apparaissant dans un contexte (phrase) donné d'un texte. On définit le contexte de t_k comme étant l'ensemble des termes qui apparaissent dans la même phrase. Soit donc :
Contexte $= \{t_1, \ldots, t_k, \ldots, t_n\}$

Où :

- t_k est un terme et **K** son ordre d'apparition dans la phrase.

Et soit Ω une ressource sémantique. On dénote par $concept_\Omega(t_k)$ l'ensemble des concepts de la ressource sémantique Ω dénotés par le terme t_k.

$$concept_\Omega(t_k) = \{ C_k^1, \ldots, C_k^i, \ldots, C_k^n \}$$

où C_k^i est le i-ième concept dénoté par t_k et $m = |concept_\Omega(t_k)|$ la cardinalité de l'ensemble des concepts dénoté par t_k. Chaque concept C_k^i de t_k pouvant être combiné avec chacun des concepts C_l^j associé aux termes t_j **(j=1 ..n)** de son contexte. Plusieurs combinaisons de concepts sont donc possibles dans un contexte donné.

Le problème de la désambiguïsation consiste à sélectionner une seule combinaison parmi cet ensemble des combinaisons.

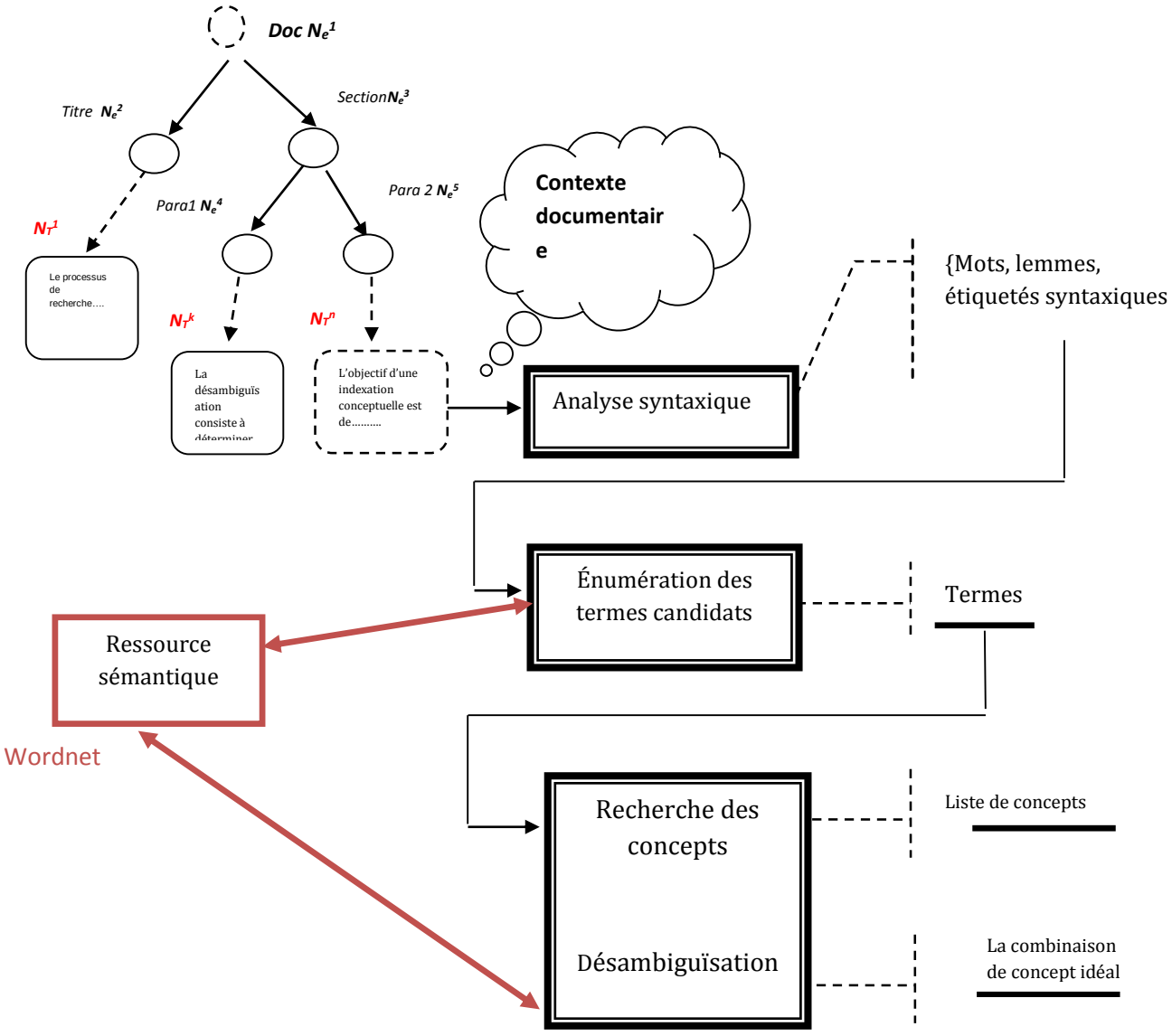


Figure3.1 : Processus d'identification des concepts

Pour sélectionner une combinaison de concepts, on doit calculer la similarité globale entre les concepts de cette combinaison.

- La similarité entre deux concepts est mesurée par un score de similarité définissant leur proximité sémantique dans la ressource utilisée. Différentes mesures de

similarité entre concepts existent dans la littérature (**La mesure de Zargayouna, La mesure de Leacock et Chorodow, La mesure de Resnik**).

La mesure de Leacock et Chorodow (Leacock et al., 1998). Cette mesure est basée sur le plus court chemin entre deux concepts.

$$\mathbf{sim_{leacock\_chorodow(C_1,C_2)}} = -\log\left(\frac{\min \mathrm{len}(C_1,C_2)}{2*D}\right)$$

Formule 3.1

- $\mathbf{min\ len(C_1, C_2)}$: est la longueur du plus court chemin entre C_1 et C_2
- **D :** est la profondeur maximale de l'ontologie

- La similarité globale d'une combinaison de concepts $\mathbf{CB_{CA} = \{C_1, \ldots, C_k, \ldots, C_n\}}$ d'un contexte d'apparition **CA** , est définie comme la moyenne des similarités MS entre ses concepts.

$$\mathbf{MS(CB_{CA})} = \frac{\mathbf{2.\ \sum_{i=1}^{i=n} \sum_{j=i+1}^{n} sim(C_i, C_j)}}{\mathbf{n.\ (n-1)}}$$

Formule 3.2

La désambiguïsation consiste à sélectionner la combinaison des concepts $\mathbf{CB_{CA}^{Max}}$ dont la moyenne de similarité entre les concepts est maximale

$$\mathbf{max} = \mathbf{argMax}\ (\ \mathbf{MS}\left(\mathbf{CB_{CA}^{i}}\right)\ )$$

Formule 3.3

Ou :

- $\mathbf{CB_{CA}^{i}}$ est la ième combinaison de concepts du contexte d'apparition **CA**

A cause de la nature combinatoire du problème de la désambigüisation, il faut déterminer la taille du contexte optimale à utiliser. Il s'agit en fait de déterminer le nombre minimal des termes les plus proches du terme cible. Ces termes forment une fenêtre du contexte. Une fenêtre peut être définie par *m* termes avant et après le terme cible.

L'utilisation de ces fenêtres nous permet de réduire la complexité algorithmique du problème de la désambigüisation dans le cas où les termes présentent une ambigüité élevée et la taille du contexte est très grande.

L'utilisation des fenêtres dans la désambiguïsation doit tenir compte du fait que les termes de même contexte sont sémantiquement proches.

Cette méthode consiste à utiliser une fenêtre de taille fixe et à la translater. Nous illustrons un exemple pour désambigüiser les termes du contexte {t1, t2, t3, t4, t5} et en prenant une fenêtre de taille 3, l'identification des concepts se déroule de la façon suivante :

- 1. identification des concepts {c1, c2} à partir des combinaisons de la fenêtre {$concept(t1)$, $conceptΩ(t2)$, $conceptΩ(t3)$}.
- 2. identification du concept {c3} à partir des combinaisons de la fenêtre {c2, $concept(t3)$, $conceptΩ(t4)$}.
- 3. identification des concepts {c4, c5} à partir des combinaisons de la fenêtre {c3, $conceptΩ(t4)$, $(t5)$}.

Certaines études indiquent que désambiguïser en utilisant une fenêtre de ± 2 termes donne les mêmes résultats que de le faire en utilisant toute la phrase

II.2.2 La pondération des concepts :

Dans la recherche des documents semi-structurés, le poids d'un terme est exprimé de manière locale au sein du document (ou l'élément) ou de manière globale au sein de la collection.

Le poids d'un terme est généralement évalué selon trois dimensions :

- ***tf :*** *est* la fréquence du terme dans l'élément ;
- ***idf :*** *est* la fréquence inverse du document pour le terme ;
- ***ief*** *:est* la fréquence inverse de l'élément pour le terme.

Une étude sur la pondération des termes a montré que la combinaison de ***tf et ief*** donne la meilleure performance, ainsi nous adoptons ces mesures pour calculer la pondération des concepts.

Le poids d'un concept s'exprime en fonction des facteurs suivants :

- **le facteur de pondération locale** : la pondération locale permet de mesurer la représentativité locale d'un concept. Elle reflète l'importance du nœud de type texte. La fonction de pondération locale est exprimée par la fréquence d'un concept (CF_i^j) dans un nœud texte comme le montre la formule suivante :

CF_i^j=le nombre d'occurrences d'un concept C_j dans un nœud texte N_T^j

- **le facteur de pondération globale *:*** la pondération globale reflète la représentativité globale du concept dans l'ensemble des nœuds texte de la collection. Un poids plus important doit être attribué aux concepts qui apparaissent moins fréquemment dans la collection. En effet les concepts qui apparaissent dans de nombreux nœuds sont moins utiles pour la discrimination que ceux qui apparaissent dans peu de nœuds. La fonction de pondération globale est exprimée par la fréquence inverse de l'élément pour le concept, elle est donnée par la formule suivante :

$$IECF_j = log\left(\frac{|N_T|}{|N_T^{cj}|}\right)$$

Formule 3.4

Ou

- N_T *:* est le nombre total de nœuds texte de la collection *;*
- N_T^{cj} *:* est le nombre total de nœuds texte de la collection contenant le concept C_j

Le poids d'un concept C_j dans un nœud texte N_T^i est exprimé par la formule suivante :

$$W_{ij} = CF_i^j * IECF_j$$

Formule 3.5

II.2.3 Construction des index des nœuds :

Dans cette approche, le modèle vectoriel sémantique est utilisé pour la représentation des index des nœuds texte, un nœud de type texte N_T^j est représenté par un vecteur des poids des concepts $\overrightarrow{N_T^J} = (W_{1j}, \dots, W_{kj}, \dots, W_{nj})$

Comme les documents semi-structurés possèdent une structure arborescente, les index des nœuds sont imbriqués les un dans les autres. Par conséquent, l'index d'un nœud de type élément contient les index de ces nœuds descendants de type texte. Ainsi, les concepts des nœuds de type texte sont propagés dans l'arbre des documents (voir la **figure 3.2**)

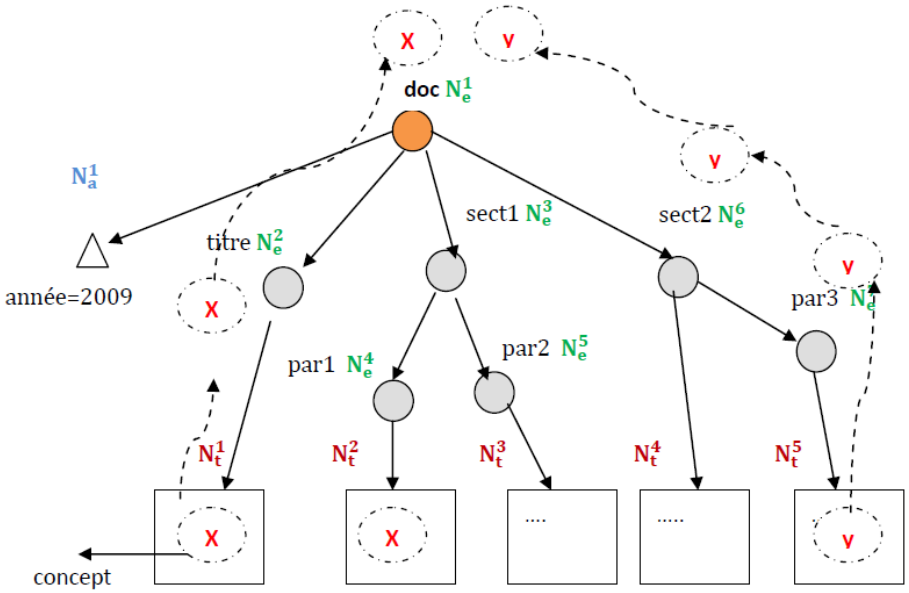


Figure3.2 : propagation des concepts dans l'arbre d'un document.

La construction des index se base sur deux hypothèses [Harrathi et al, 2010] :

- **Hypothèse 1 :** les concepts qui apparaissent prés de la racine dans un arbre paraissent plus porteurs d'information pour le nœud associé que ceux situés plus bas dans l'arbre. Autrement dit, plus la distance entre un nœud texte et son ancêtre est importante, moins il contribue à sa représentation.

- **Hypothèse 2** : les concepts qui apparaissent plusieurs fois dans les nœuds descendants sont plus porteurs d'information pour les nœuds ancêtres. Plus un concept apparaît dans tous les nœuds descendants plus il contribue à sa représentation même si sa fréquence est faible dans chaque nœud.

Nous modélisons **l'hypothèse 1** par l'utilisation de la fonction de propagation du paramètre (N_E, N_T^k), qui permet de représenter la distance entre le nœud élément N_E et ses nœuds descendants de type texte N_T^k dans l'arbre du document, c'est-à-dire le nombre d'arcs séparant les deux nœuds. En tenant compte de **l'hypothèse ,** le vecteur sémantique représentant les nœuds élément N_E est calculé de la façon suivante :

$$\overrightarrow{N_E} = \sum_{k=1}^{m} \lambda^{dist(N_E, N_T^k)-1} * \overrightarrow{N_T^k}$$

Formule 3.6

Ou :

- $\overrightarrow{N_T^k}$ est le vecteur sémantique représentant le **k**-ième nœud descendant du nœud élément N_E;
- $\lambda \in [0, 1]$ ***est*** un paramètre qui permet de quantifier l'importance de la distance séparant les nœuds dans la formule de propagation.

Ainsi, le poids W_j du concept C_j dans le vecteur $\overrightarrow{N_E}$:

$$W_j = \sum_{k=1}^{m} \lambda^{dist(N_E, N_T^k)-1} * W_{kj} \ , pour \ 1 \leq j \leq n$$

Formule 3.7

Ou :

W_{kj} : est la poids du concept C_j dans le vecteur $\overrightarrow{N_T^k}$.

Nous modélisons l'**hypothèse2** par l'utilisation de la fonction de propagation du paramètre $|C_E^j|$, qui représente le nombre de nœuds texte descendant de N_E contenant le concept C_j.

Plus le nombre $|C_E^j|$ est grand, plus le concept C_j contribue dans la représentation du nœud N_E. La formule de calcul de poids en tenant compte de l'**hypothèse 1** et **l'hypothèse 2** est :

$$W_j = \sum_{k=1}^{m} |C_E^j|^{\beta} * \lambda^{dist(N_E, N_T^k)-1} * W_{kj} \quad , pour \ 1 \leq j \leq n$$

Formule 3.8

Ou :

- $\beta \in [0,1]$, est le paramètre permettant de quantifier l'importance du nombre des nœuds texte descendants contenant le concept C_j dans la formule de propagation.

Remarque :

Afin identifier les éléments des documents XML et leurs associer l'index du contenue, nous avons utilisé l'indexation par arbre déjà présentée et expliquée dans le deuxième chapitre.

II.3. Appariement Nœuds / Requête :

L'appariement nœuds/requêtes a pour but d'attribuer des scores de pertinence aux éléments d'un document (les nœuds de type texte et nœuds de type élément ans l'arbre XML).

Dans cette approche nous avons apporté une modification au niveau de l'appariement où nous avons modifié la formule de calcul du score de pertinence pour que la similarité sémantique entre les concepts de nœuds et ceux de la requête soit prise en considération, alors que dans l'appariement de **[harrathi et al, 10]** la similarité sémantique des concepts de nœuds est faite séparément de celle de la requête.

Nous présentons d'abord l'appariement de [**harrathi et al,10]** puis nous détaillerons l'appariement que nous avons proposé.

Score de pertinence d'un nœud :

Dans cette approche, on utilise le modèle vectoriel sémantique où dans un espace conceptuel d'indexation, un nœud texte et une requête sont indexés par deux vecteurs de poids des concepts. La requête et le nœud sont chacun représenté par un graphe pondéré dont les nœuds représentent les concepts et les arrêtes les poids entre les concepts (voire la figure 3). Dans cette approche, la proximité sémantique entre la représentation sémantique de la requête et la représentation sémantique du nœud est considérable, si et seulement si les concepts de la requête sont proches des concepts du nœud.

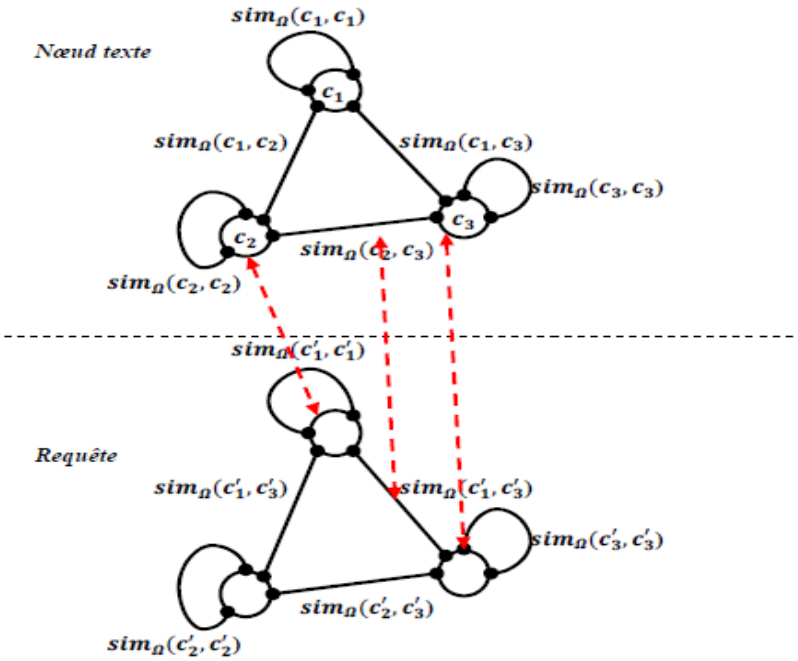


Figure 3.3 **Graphe sémantique d'une requête et un nœud**

En pratique, mesuré la similarité entre un nœud texte et une requête revient à comparer le graphe sémantique de la requête avec celui du nœud, chaque graphe étant représenté sous forme d'une matrice. Ce qui revient à calculer la similarité entre les matrices correspondantes.

Etant donné un vecteur sémantique d'un nœud $\overrightarrow{N_T^j} = (W_{1j}, \dots, W_{kj}, \dots, W_{nj})$ la matrice **(notéeM_T^j)** représentant le graphe sémantique du nœud est une matrice carrée d'ordre n qui est définie de la façon suivante :

$$M_T^j[a,b] = W_{aj} * w_{bj} * sim_\Omega(C_a, C_b) \; pour \; 1 \leq a,b \leq n$$

Formule 3.9

Où :

- W_{aj} , w_{bj} sont les poids respectifs des concepts a et b dans le nœud N_T^j.
- $sim_\Omega(C_a, C_b)$ est la mesure de similarité sémantique définie sur l'ontologie Ω entre les concepts C_a, C_b.

Remarque : Nous avons tenu compte des poids des concepts dans la matrice. En effet si le poids d'un concept est nul, cela veut dire que ce nœud n'a pas de similarité avec les autres concepts.

De la même façon on définit le graphe sémantique de la requête :

$\vec{Q} = (W_1, \dots, W_k, \dots, W_n)$: est représenté par une matrice carrée d'ordre n qui est définie de la façon suivante :

$$M_Q[a,b] = W_a * w_b * sim_\Omega(C_a, C_b) \; pour \; 1 \leq a,b \leq n$$

Formule 3.10

L'évaluation de la pertinence entre le graphe sémantique du nœud et le graphe sémantique de la requête revient à mesurer la similarité entre les deux matrices en utilisant la mesure de cosinus.

Le cosinus entre deux matrices carrées A et B de même ordre n, est définit comme suit :

$$cosinus\ (A,B)\ \frac{< A,B >_F}{\|A\|_F \|B\|_F}$$

Formule 3.11

Ou :

- $\|A\|_F \; et \|B\|_F$ Sont des normes de Frobenius de A et B.

La norme de *Frobenius* d'une matrice A d'ordre n est définie de la façon suivante :

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_i, j|^2}$$

Formule 3.12

$< A,B >_F$ Est le produit interne de Forbenius de A et B

$$< A, B >_F = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} * b_{i,j}$$

Formule 3.13

Le score de pertinence d'un nœud vis-à-vis d'une requête Q est obtenu en utilisant la mesure de cosinus entre les deux matrices M_T^j M_Q.

$$cosinus\ (M_T^j, M_Q)\ \frac{< M_T^j, M_Q >_F}{\|M_T^j\|_F \|M_Q\|_F}$$

Formule 3.14

Notre appariement proposé :

Dans cette approche, on utilise le modèle vectoriel sémantique où dans un espace conceptuel d'indexation, un nœud et une requête sont indexés par deux vecteurs de poids de concepts. Dans cette approche, la proximité sémantique entre la représentation sémantique de la requête et la représentation sémantique du nœud texte est considérable, si et seulement si les concepts de la requête sont proches des concepts du nœud.

Remarque : L'appariement nœud/requête est calculé de la même manière pour les nœuds feuilles et les nœuds élément (interne).

Les nœuds sont représentés par un vecteur de concept pondéré:

Le vecteur de concept :

$$Vc\_n = \{wcn_1, wcn_2, \dots, wcn_n\}$$

- wcn_i: représente le poids du $i^{éme}$ concept dans le nœud.
- n : est le nombre de concepts dans le nœud.

La requête est représentée de la même forme que celle des noeuds par un vecteur de concept pondéré.

$$Vc\_q = \{wcq_1, wcq_2, \dots, wcq_m\}$$

- wq_i est le poids de de $i^{éme}$ concept dans la requête.
- m est le nombre de concept de la requête.

Remarque : Les concepts de la requête sont construits de la même manière que ceux des nœuds feuilles.

Nous avons utilisé une matrice à laquelle nous avons associé le vecteur de la requête et celui des concepts de nœuds

Le score de pertinence nœud/requête est calculé selon la formule suivante :

$$\text{RSV(n, q)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} wc_{ni} * wcq_j * sim(cn_i, cq_j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} wc_{ni} * wcq_j}$$

Formule 3.15

- La formule permet de calculer la similarité sémantique entre le contexte de nœud et celui de la requête.
- La division sur $m * n$ permet de normaliser les scores de pertinence. Les nœuds sont retournés par ordre de pertinence à l'utilisateur

III. Conclusion :

Nous avons présenté dans ce chapitre, une approche conceptuelle pour la recherche des documents semi-structurés. Où les nœuds de l'arbre d'un document ainsi que la requête sont représentés par des vecteurs sémantiques de concepts.

Dans cette approche nous avons utilisé la désambigüisation des termes et le calcul des scores de pertinence des nœuds de l'arbre d'un document.

L'évaluation de notre approche s'est basée sur l'utilisation de la collection de la campagne d'évaluation INEX et la ressource sémantique Wordnet. A bout des expérimentations nous avons montré que notre approche de recherche conceptuelle permet d'améliorer la performance du processus de recherche selon la mesure officielle d'INEX 2009.

Chapitre 4 : Evaluation et résultats

I. Outils et développement :

I.1. Java :

Java est un langage de programmation moderne développé par **Sun Microsystems** (aujourd'hui racheté par **Oracle** C'est un langage de programmation orienté objet. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).

I.2. MySQL :

MySQL (*Structured Query Language*) Système de gestion de bases de données relationnelles (SGBDR) sous licence GNU très utilisé pour mettre en ligne des bases de données. Il stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table, cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. MySQL fonctionne sur beaucoup de plates-formes différentes, incluant AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000 et XP.

I.3. Api SAX xerces:

L'api SAX xerces est utilisée pour parser les documents XML, elle nous permet de construire l'arbre du document XML et d'extraire le contenue.

I.4. JWI

JWI (the MIT Java Wordnet Interface) est une bibliothèque Java pour l'interfaçage avec Wordnet. JWI permet l'accès aux versions wordnet 1.6 jusqu'à 3.0 pour récupérer des données pour les applications java afin de les traiter.

I.5. TreeTagger :

L'analyse syntaxique du contenu textuel des documents XML est effectuée par l'API TreeTagger. Cette API nous permet d'extraire les catégories grammaticales et les lemmes des mots du texte.

I.6. Wordnet :

Wordnet une base de données lexicographique pour l'anglais qui couvre la grande majorité des noms, verbes, adjectifs et adverbes de la langue anglaise. Développée par le

Cognitive Science Laboratory (Princeton University), son développement a commencé manuellement en 1985.Wordnet est une ressource libre est bien documentée.

I.6.1. Eléments de bases de wordnet :

- ✓ **Les synsets (synonym set)** : un ensemble de synonymes qui sont reliés entre eux par des relations sémantiques et conceptuelles.
  Les mots ayant plusieurs sens appartiennent à plusieurs synsets ou chaque synset est accompagné :
  - D'un descripteur du sens qu'il représente
  - D'un Exemples d'usage
- ✓ **Les relations entre synsets** : hyperonymie- hyponymie (is-a), méronymie, implication, dérivation morphologique…

Méronymie : relation sémantique orientée entre signification de deux mots tel que le premier mot constitue une partie pour le deuxième. Par exemple : ''corps'' a pour Méronymes ''bras'', ''jambes'', ''tête''…

Hyperonymie : Relation d'inclusion établie entre un terme général et un ou plusieurs termes spécifiques. Si on pose Y est un hyperonyme de X alors X est un type de Y. Exemple : '' Meuble '' est un hyperonyme de ''chaise'' et ''table''.

Holonymie : L'holonymie est une action sémantique (Type de relation sémantique) entre deux mots. C'est le nom de la classe globale dont les noms méronyme font partie.

Hyponymie : Relation d'inclusion entre deux mots dont l'un est l'hyponyme de l'autre. Elle est la relation inverse de l'hyperonymie. Par exemple ''maison'' est un hyponyme de ''bâtiment''.

Le terme (représentant du synset) :c'est le terme pour lequel le concept (synset) est identifié.

I.7. API JWS :

Afin de calculée la similarité sémantique entre deux concepts de la base de WordNet l'API JWS (Java Wordnet Similarty) est utilisé, elle met en œuvre certain nombre d'algorithme de similarité sémantique.

I.8. Eclipse IDE :

Eclipse *IDE (Integrated Development Environment*) c'est un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aides à la programmation. Il est développé par IBM, il est gratuit et disponible pour la plupart des systèmes d'exploitation.

I.9. JDBC :

JDBC (Java DataBase Connectivity) désigne une API pour permettre un accès aux bases de données avec Java.

II. Résultat de l'évaluation de notre application :

Nos expérimentations ont été portées sur une collection de 633 documents d'INEX 2009 avec une taille de 26 MO répondant à 9 requêtes.

L'évaluation c'est porté sur la tache Thorough Task de INEX 2009.

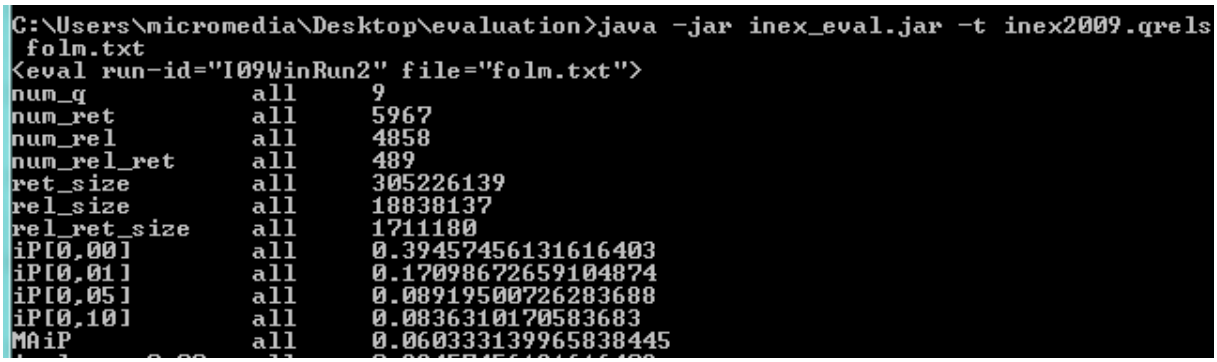


Figure 4.1 – résultat d'évaluation de notre approche dans la tache throught task de INEX 2009.

```
C:\Users\micromedia\Desktop\evaluation>java -jar inex_eval.jar -t inex2009.qrels
 folcc.txt
<eval run-id="I09WinRun1" file="folcc.txt">
num_q           all     9
num_ret         all     4458
num_rel         all     4858
num_rel_ret     all     393
ret_size        all     255919441
rel_size        all     18838137
rel_ret_size    all     1365447
iP[0,00]        all     0.3650536097721601
iP[0,01]        all     0.3650536097721601
iP[0,05]        all     0.3650536097721601
iP[0,10]        all     0.33947713835977944
MAiP            all     0.23096000729038446
```

Figure 4.2 – résultat d'évaluation d'une approche à base de mot clés dans la tache [sauvagnat, 2005] '*thorough task*' de INEX 2009.

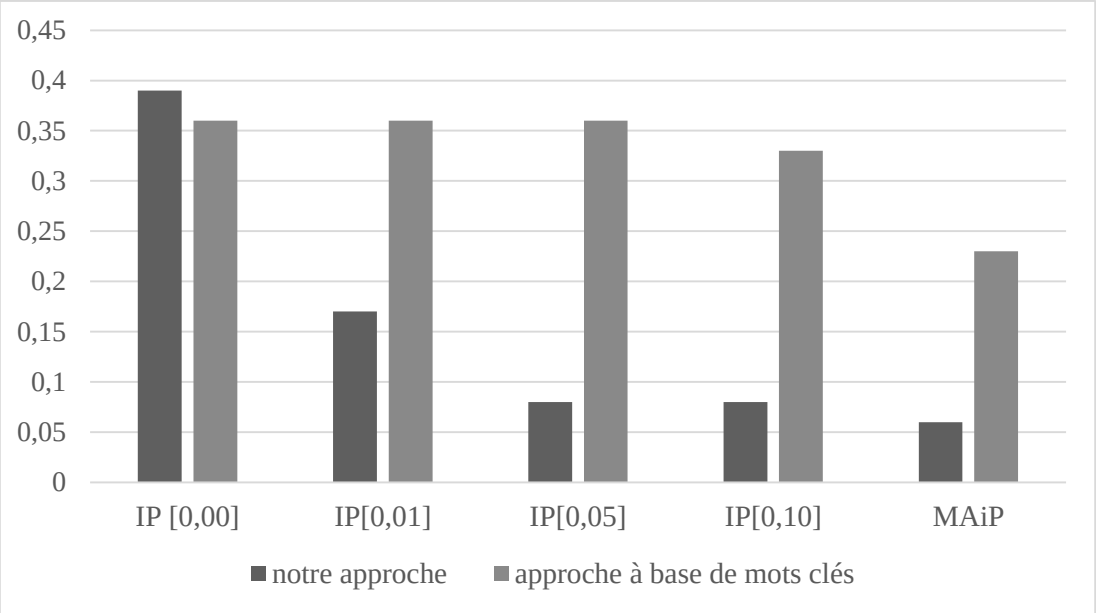


Figure 4.3 – histogramme de comparaison entre les deux approches.

Nous constatons que les résultats retournés par notre approche sont inférieurs par apport à l'approche à base de mots clés. Cela dû aux :

- La fenêtre de désambiguïsation utilisée est de nombre 3, cette fenêtre est trop petite pour pouvoir déterminer le contexte d'un terme.

On perspective on aimera évaluer l'approche avec une plus grande fenêtre de désambigüisation.

Conclusion générale :

Dans le cadre de ce mémoire, nous nous sommes intéressés à la RI semi structurées. Nous avons cerné les principales différences séparant la RI classique de la RI semi-structurées.

Nous avons évoqué la nécessité de développer des techniques permettant d'identifier les éléments les plus pertinents dans un document par rapport à une requête donnée Ces techniques doivent permettre à l'utilisateur d'interroger des documents structurés en spécifiant des conditions sur le contenu textuel et aussi sur la structure selon ses besoins.

Par la suite nous avons essayé de cerner les principales problématiques de la RIS au niveau de l'indexation/appariement et au niveau de l'interrogation et ce qu'elles peuvent présenter comme insuffisances. Puis, nous nous sommes intéressés spécifiquement à la problématique de l'indexation/appariement à base de mots-clés et avons introduit l'indexation sémantique comme solution à ce problème. C e type d'indexation est caractérisé par l'utilisation des ressources sémantiques et des mesures de similarité sémantique entre les concepts. Nous avons introduit une approche de RIS conceptuelle. Cette approche est fondée sur trois étapes :

- Une première étape qui permet l'identification des concepts à partir du texte grâce à une désambiguïsation des termes qui se base sur la notion de mesure sémantique et qui tient compte du contexte des termes. L'approche proposée utilise la notion des fenêtres afin de réduire la complexité algorithmique du problème de la désambigüisation.
- La seconde étape quand à elle permet de construire des index des nœuds de l'arbre d'un document ce qui consiste à construire les vecteurs de concepts représentant les nœuds à partir de ses nœuds descendants en tenant compte de la structure dans la pondération des concepts.
- Une troisième étape d'appariement nœuds/requête qui consiste à évaluer des scores de pertinence des nœuds en tenant compte de la proximité sémantique entre les concepts.

Nous avons enfin évalué notre approche en utilisant la collection INEX 2009. Dans cette approche nous avons effectué une comparaison entre les résultats obtenus et les résultats officiels obtenus dans INEX 2009.

Bibliographie :

[Azevedo, 2004] INEX'04 Proceedings of the Third international conference on Initiative for the Evaluation of XML Retrieval Pages 311-321

[Baziz. 2005]. Baziz M. indexation conceptuelle/sémantique guidée par ontologie pour la recherche d'information, Thèse de Doctorat en informatique effectuée à l'Institut de Recherche en Informatique de Toulouse (IRIT) .

[Belew, 1989] Belew, R. K. Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. In ACM SIGIR Proceedings. p. 11-20, 1989.

[Berry et al. 1999]. Berry, M. W., Z. Drmac, et E. R. Jessup : . Matrices, vector spaces, and information retrieval. SIAM Rev. 41(2), 335–362(1999).

[Boubekeur. 2008]. Boubekeur-Amirouche F. Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets, thèse en informatique ,Université Toulouse III - Paul Sabatier .

[Boughanem et al. 2004]. Boughanem M., Kraaij W., and Nie J-Y. Modèles de langue pour la recherche d'information. In Les systμemes de recherche d'informations, pages 163-182. Hermes-Lavoisier, 2004.

[Harrathi et al. 2010]. Harrathi R., CalabrettoS. Une approche de recherche sémantique dans les documents semi- structurés . Dans RISE (Recherche d'Information SEmantique)dans le cadre de la conférence INFORSID'2010, Marseille 2010 .

[Hlaoua. 2007]. Hlaoua L. Reformulation de Requêtes par Réinjection de Pertinence dans les Documents Semi-Structurées, thèse en informatique , l'Université Toulouse III -Paul Sabatier .

ISO 1087. (1990). ISO. International Organization for Standardization (ISO). Terminology-Vocabulary = Terminologie-Vocabulaire, Genève : Organisation internationale denormalisation (ISO 1087 – 1990.) 1990.

ISO. (1986). ISO. Standard Generalized Markup Language (SGML). International Organization for Standardization (ISO), Information Processing – Text and Office Systems –ISO8879-1986 .

[Khan et al. 2004]. Khan L., McLeod D., Hovy E.,. Retrieval effectiveness of an ontology-based model for information selection. TheVLDB Journal (2004) 13:71–85.

[Kamps et al. 2007]. Kamps J., Pehcevski J., Kazai G., Lalmas M., Robertson S. INEX 2007 Evaluation Measures », in Proceeding of INEX, 2007 pages 24-33 , 2007.

[Kim et al. 2005]. Kim M. S. and. Kong Y.-H:. Ontology-DTD Matching Algorithm for Efficient XML Query, in FSKD (2), ser. Lecture Notes in Computer Science, L. Wang and Y.Jin, Eds., vol. 3614. Springer, 2005, pp.1093-1102.

[Krovetz. 1997]. Krovetz R. Homonymy and polysemy in information retrieval. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (A CL-97}, pages 72-79.

[kwak et al. 1989]. K.L. Kwok, a neural for probabilistic information retrivel. 12th international ACM SIGIR Conference on research and developpement in information retrival, pp 21-30, 1989.

[Leacock et al. 1998]. Leacock C., and Chodorow M. Combining local context andWordNet similarity for word sense identification. In Fellbaum, C., editor, WordNet : an electronic lexical database, volume 11 of Language, Speech and Communication,pages 265–283. The MIT Pr, , Cambridge, Massachusetts.

[Malik et al. 2005]. Malik S., Kazai G., Lalmas M. , and Fuhr N. Overview of INEX 2005. In Pre-Proceedings of the International Workshop of the Initiative for the Evaluation ofXML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005 pages 1–15.

[Maron et al. 1960]. Maron, M., and Kuhns, J. On relevance, probabilistic indexing and information retrieval. Journal of the Association for Computing Machinery7 (1960), pages 216–244.

[Mass et al. 2002]. Mass Y., Mandelbrod M., Amitay E. , Maarek Y. and Soffer A. JuruXML-an XML retrieval system at INEX'02. In Proceedings of the First Workshop of the INiative for the Evaluation of XML REtrieval(INEX), pages 73-80. Dagsthul, Germany, Decemder 2002 .

[Mass et al. 2004]. Mass Y. and Mandelbrod M. Component ranking and automatic query reffinement for XML retrieval. In INEX 2004 Workshop Proceedings, pages 73,84.Dagsthul, Germany, December 2004.

[Mihajlovic et al. 2005]. Mihajlovic V. , Blok H.E., Hiemstra D. , and Apers P.M.G. . Score Region Algebra: Building a Transparent XML-IR Database. In Proceedings of the 14th International Conference on Information and Knowledge Management (CIKM),2005 .

[Piwowarski et al. 2002]. Piwowarski B., Faure G., and . Gallinari P. Bayesian networks and inex. In Proceedings of the First Workshop of the INiative for the Evaluation of XMLREtrieval(INEX), pages 149,154. Dagsthul, Germany, December 2002.

[Piwowarski et al. 2008]. Piwowarski B., Trotman A., Lalmas M. Sound and complete relevance assessment for XML retrieval. ACM Trans. Inf. Syst. 27(1): (2008) .

[Rocchio, 1971] J .Rocchio .Realevance feedback in information retrieval.

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

[Salton et al. 1983]. Salton G., Fox E. A., Wu H. Extended Boolean information retrieval system. CACM 26(11), pp. 1022-1036, 1983.

[Salton. 1970]. Salton G. Automatic processing of foreign language document – Journal of the American Society for Information Science, 21(3):187-194, May.

[Sauvagnat. 2005]. Sauvagnat k. Modèle fexible pour la recherche d'information dans des corpus de documents semi-structurés . Thèse de doctorat en informatique ,Université PaulSabatier, Toulouse, France, juin 2005 .

[Schenkel et al. 2005]. Schenkel R., Theobald A., and Weikum G, . Semantic similarity search on Semi structured data with the XXL search engine, Information Retrieval, 8(4): pp.521–545.

[Schlieder et al. 2002]. Schlieder T. and Meuss H. Querying and Ranking XML Documents. Journal of American Society for Information Science and Tecnology (JASIST), Special Topic Issue on XML and Information Retrieval, 53(6):489–503, Apr. 2002.

[Taha et al. 2008]. Taha K. and Elmasri R. CXLEngine: A Comprehensive XML Loosely Structured Search Engine, In Proceedings of the EDBT workshop, Nantes, France 2008. ACMInternational Conference Proceeding Series, New York, USA; Vol. 261, ISBN:978-1-59593-966-1, pp 37-42 .

[Walker et al. 1997]. Walker S., Robertson S., Boughanem M., Jones G., and Jones K. S.. Okapi at TREC-6 automatic and ad hoc, VLC, routing, filtering and QSDR. In Proceedings of TREC-6, pages 125–136, 1997.

[Lin . 1998]. Lin D. An information-theoretic definition of similarity. In Proc. 15th International Conf. on Learning, Morgan Kaufmann, San Francisco, CA, pp. 296–304 .
[Wu Z. and Palmer M].: Verb semantics and lexical selection. In 32nd. Annual Meeting of the Association for Computational Linguistics, pp. 133 –138, (1994).

[Woods. 1997]. Woods, W. A.,. Conceptual indexing : A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, Mountain View,CA, April. ww.sun.com/research/techrep/1997/abstract-61.html.

[XPath 2.0. 2007]. XML Path Language (XPath) 2.0. XML Path Language (XPath) 2.0: W3C Recommendation , 2007. http://www.w3.org/TR/xpath20/ .

[Zargayouna et al. 2004]. Zargayouna, H., Salotti, S. Mesure de similarité dans une ontologie pour l'indexation sémantique de documents XML. Actes de la conférence IC'2004 .

[Zargayouna. 2005]. Zargayouna H. Indexation sémantique de documents XML. Thèse en informatique, Université Paris XI Orsay.