REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Électrique & d'Informatique
Département d'Informatique

# *Mémoire*

*De fin d'études*

*En vue de l'obtention du Diplôme De Master II en Informatique*
*Option : Conduite de Projet Informatique*

## *Thème*

# MODÉLISATION PROBABILISTE D'UNE ACTIVITÉ DE RECHERCHE PERSONNALISÉE

***Proposé et dirigé par :***

**M^me^ ACHEMOUKH.F.**

***Réalisé par:***

***MEDDOUR Lila.***
***SAADA Wissame.***

***Promotion: 2011/2012***

# Remerciements

*Nous remercions le bon dieu pour le courage, la patience qui nous ont été utiles tout au long de notre parcours.*

*Nous tenons à témoigner notre profonde gratitude et nos remerciements les plus sincères à Mme Achmoukh Farida pour avoir dirigé notre travail pour son soutien et pour tout le temps qu'elle a consacré au bon déroulement de ce travail.*

*Nous tenons aussi à remercier les membres du jury, pour avoir accepté de juger ce modeste travail.*

*Nous remercions notre ami Medjber Amir pour son aide et son soutien, Merci chère ami.*

*Ces remerciement ne serait pas complets si on n'a pas pensé à les destiner, avec notre profonde reconnaissance, à nos parents qui nous ont offert un environnement favorable mener à terme notre travail.*

*Un grand merci pour tous ceux qui, d'une manière ou d'une autre, nous ont aidé et encouragé à la réalisation de ce modeste travail.*

# Dédicaces

*Je dédie ce modeste travail à mes chères parents qui sont ce que j'ai plus cher et qui ont toujours était la pour moi sans jamais se plaindre. A mes deux chères frères « Hacène » et « Azouaou ».*

*A celui que j'aime tant « **Mounir** ».*

*A la mémoire de ma Grand-Mère paternelle « Setti », mon oncle « Omar » et ma tente « yemma Nouara » que vos âmes reposent en paix.*

*A ma Grand-mère « yemma Tata », mes oncles, mes tentes, mes cousins et cousines.*

*A toute ma famille sans exception.*

*A notre chère promotrice Mme Achmoukh.F qui nous a beaucoup aidée, encouragée, conseillée et surtout supportée, Merci Madame.*

*A mon amie et partenaire de travail Lila.*

*Ainsi qu'à tous mes amis(es) particulièrement: Hayet, Hamida, Liza, Liza baleh, Dihya, Amir, Idir, Farid, Nourddine.*

*Lyly, Radia, Kahina, Mira, Djouhar, Yacine, Anis, Saber, Slimane, Meriou, Lala, Sabrina, Fahima, Lamia, Kamira, Katia.*

*A mes chères sœurs copines de chambre : Hayet, Kahina, Meriem, Kahina, Leila, Samira.*

*A toute personne qui me connait.*

*Du fond du cœur, je vous dis tous merci pour votre soutien et pour tout ce que vous avez fait pour moi.*

***Wissame***

# Dédicaces

*Je dédie ce modeste travail à mes chères parents qui sont ce que j'ai plus cher et qui ont toujours était la pour moi sans jamais se plaindre. A mes deux chères frères « Yacine » et « Kamel »*

*A mes chères sœurs Nassima & Karima.*

*A ma Grand-mère « yemma sadia », mes oncles, mes tentes, mes cousins et cousines.*

*A mon mari « Yacine ».*

*A mes beaux parents, mes belles sœurs et beau frère.*

*A toute ma famille sans exception.*

*A notre chère promotrice Mme Achmoukh.F qui nous a beaucoup aidée, encouragée, conseillée et surtout supportée, Merci Madame.*

*A mon amie et partenaire de travail Wissame.*

*Ainsi qu'à tous mes amis(es) particulièrement Hayet, Liza, Amir, Dyhia, Lyly, Radia, Kahina, Mira, Djouhar.*

*A toute personne qui me connait.*

*Du fond du cœur, je vous dis tous merci pour votre soutien et pour tout ce que vous avez fait pour moi.*

***Lila***

# *Liste des figures*

## Chapitre I : Généralités sur la Recherche d'Information



## Chapitre II : la recherche d'information personnalisée



## Chapitre III : Réseaux Bayésiens et recherche d'informations



## Chapitre VI : Conception et Réalisation

# *Liste des figures*

# Chapitre I : Généralités sur la Recherche d'Information

## Chapitre II : la recherche d'information personnalisée

## Chapitre III : Réseaux Bayésiens et recherche d'informations

## Chapitre VI : Conception et Réalisation

# Conclusion et perspectives

# *Introduction Générale*

# *Introduction générale*

Depuis l'essor de l'informatique, le volume d'information stockée électroniquement ne cesse de s'accroitre. Cependant les milliards de documents accessibles sur le web qui composent cette masse d'information ne sont actuellement qu'imparfaitement utilisables. Le problème qui se pose actuellement n'est plus tant la disponibilité de l'information mais la capacité d'accéder et de sélectionner l'information répondant aux besoins précis d'un utilisateur. En effet, pour un individu, rechercher une information précise dans l'amas des données en croissance exponentielle sur le net serait comme chercher une aiguille dans une botte de foin.

Le partage, la gestion et l'exploitation de ces informations nécessitent la mise en place de solutions adaptées. Le développement de modèles, méthodes et outils de recherche efficaces permettant de mettre en relation ces informations, capables d'assister un utilisateur dans sa recherche d'information pour qu'il accède juste à l'information qu'il juge pertinente est devenu un challenge important pour faire du web un réel outil de partage d'informations.

L'objet d'un système de recherche d'information est de faciliter l'accès à un ensemble de documents, afin de permettre à l'utilisateur de retrouver ceux qui sont pertinents, c'est-à-dire ceux dont le contenu correspond le mieux à son besoin en information. Les systèmes de recherche d'information classiques se basent sur une recherche par mots clés, les documents sont représentés comme des sacs de mots et la pertinence d'un document vis-à-vis d'une requête est souvent estimée en s'appuyant sur les fréquences d'apparition des mots de la requête dans ces mêmes documents sans donner d'importance à la profession de l'utilisateur par exemple deux utilisateurs de deux domaines déférents tape la même requête les résultats retournés par le système de recherche d'informations serons les même.

Pour remédier à ce problème, de nouvelles approches ont été développés dans le but d'intégrer l'utilisateur dans l'une des phases de processus de recherche afin de personnaliser le SRI.

Notre travail s’inscrit principalement dans ce contexte. Notre objectif est alors d’une part de personnaliser un SRI classique en intégrant le profil utilisateur en utilisant la plate-forme terrier-3.5et d’autre part de modéliser le système avec les réseaux bayésiens.

Ce mémoire s’articule autour de quatre chapitres comme suit :

- Dans le **chapitre I**, nous introduisons la recherche d'information et le processus de recherche d'information. On présente par la suite les différents modèles de la RI et les principaux paramètres d'évaluation d'un SRI.

- Dans Le **chapitre II,** nous détaillerons la recherche d’information personnalisée et on expliquera les déférentes approches et techniques de modélisation du profil, ainsi que son intégration dans les déférentes phases du processus de recherche, et on donnera l’architecture générale d’un SRI Personnalisée

- Dans le **chapitre III**, on va donner une description générale des réseaux bayésien ainsi que leur rapport avec le domaine de la recherche d’information et les déférents modèles existants

- Dans **le chapitre IV**, nous présentons notre modèle qui consiste en intégration du profil utilisateur dans la phase d’appariement d’un système de recherche d’information classique en se basant sur les réseaux Bayésiens.

- Et enfin une conclusion générale et quelques perspectives.

CHAPITRE I

# Généralités sur la recherche d'information classique

# Introduction

Le monde assiste depuis ces dernières décennies, à une production massive d'informations dans tous les domaines d'intérêt. De multiples directions de recherche ont tenté de mettre en œuvre des processus automatiques d'accès à l'information.

L'objectif est d'exploiter au mieux les bases volumineuses de ces informations.

Les systèmes de recherche d'information(SRI) servent d'interface entre une source (collection d'informations) contenant des quantités considérables de documents et des utilisateurs cherchant, via des requêtes, des informations susceptibles de se trouver dans cette collection.

Les SRI intègrent un ensemble de techniques permettant de sélectionner ces informations. Ces techniques peuvent être résumées en quatre fonctions, qui sont le stockage et l'organisation de l'information, la recherche d'informations en réponse à des requêtes utilisateurs et la restitution des informations pertinentes pour ces requêtes, la dernière fonction est celle qui est visible pour l'utilisateur.

Durant ce chapitre, notre intérêt se porte sur les principes de la recherche d'information(RI), en commençant par un bref historique de la RI, puis nous présenterons les concepts de base de la RI, ensuite nous décrivons les étapes d'un processus de RI. Nous passerons ensuite aux différent modèles de recherche d'informations classique et enfin nous présenterons les plus importantes mesures utilisées pour évaluer un SRI.

## I. Bref historique de la recherche d'information

Le nom Recherche d'Information RI (Information Retrieval IR) fut utilisé pour la première fois par Calvin N Mooers en 1948 dans son mémoire de maîtrise. Le premier problème de la RI a été celui de l'indexation des documents [Salton, 71]. Les techniques initiales, comme l'abstraction, l'indexation et l'utilisation des catégories de classification ont marqué la naissance de la Recherche d'Information comme discipline de recherche. La croissance du volume de données textuelles comme les livres et les articles dans les bibliothèques a imposé, durant des siècles, de définir des mécanismes efficaces pour les localiser.

D'énormes efforts ont été déployés depuis, pour développer des approches et des techniques permettant de retrouver l'information voulue effectivement et efficacement à partir de vastes collections de données textuelles. Depuis les années 1990, notamment avec l'avènement d'Internet, la recherche d'information est devenue plus d'actualité et plus exigeante que jamais [Baziz, 2005].

# II. La recherche d'information (RI)

## II.1. Définition

D'après (l'AFNOR, 79), La recherche d'information (RI) est un ensemble de méthodes et de procédures ayant pour objet extraire d'une collection de documents, les informations voulues. Dans un sens plus large, la RI est toute opération ayant pour but la collecte, la recherche et l'exploitation de l'information en réponse à une question sur un sujet précis.

## II.2. Notion de base

A partir de la définition de la RI, on peut extraire les concepts de base suivant :

### II.2.1. Collection de documents

La collection de documents (ou le fond documentaire, corpus,…) constitue l'ensemble des informations exploitables et accessibles par le SRI. Elle est constituée d'un ensemble de documents.

### II.2.2. Document

Le document constitue l'information élémentaire d'une collection de documents.

Il représente dans un SRI l'élément objectif. Dans le cadre de la RI traditionnelle, c'est du texte libre, qui peut être caractérisé selon trois vues :

- ✓ La vue présentation : c'est la mise en forme d'un document texte (entêtes, paragraphes, alignement…) ;
- ✓ La vue logique : qui présente la structure logique d'un document, elle porte des informations sur la structure ;

✓ La vue de contenu : qui se focalise sur le sens ou la sémantique d'un document le plus souvent par un ensemble de mots.

Dans les SRI traditionnels, la vue de contenu est l'unique intérêt, puisque les utilisateurs forment leurs requêtes en se fixant comme objectif le contenu textuel des documents : c'est d'ailleurs la raison même de l'utilisation de tels systèmes.

### II.2.3. Besoin en informations

Le besoin en informations est souvent assimilé au besoin de l'utilisateur. Il est exprimé par une requête.

Trois types de besoins utilisateurs ont été définis par [Ingwersen, 92] :

- ❖ Besoin vérification : l'utilisateur cherche à vérifier le texte avec les données connues. Il recherche une donnée particulière, et sait souvent comment y accéder (par exemple chercher un document ayant une adresse connue sur le web).
- ❖ Besoin thématique connu : l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connus (il est possible que le besoin s'affine au cours de la recherche).
- ❖ Besoin thématique inconnu : l'utilisateur cherche de nouveaux concepts ou de nouvelles relations hors des sujets ou domaines qui lui sont familiers.

### II.2.4. Requête

La requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur.

Divers types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique.

### II.2.5. Pertinence

La pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Elle est l'objet de tout système de recherche d'information.

Définir cette notion complexe n'est pas simple car elle est une notion vague, à voir le nombre de définitions autour d'elle.

[Saracevic, 70] les a regroupées dans un ensemble, nous citons quelques unes :

- ✓ La correspondance entre un document et une requête, une mesure d'informativité ;
- ✓ Un degré de relation (chevauchement) entre le document et la requête ;
- ✓ Un degré de la surprise qu'apporte un document, en rapport avec le besoin de l'utilisateur.
- ✓ …etc.

Analyser ces définitions nous amène à remarquer que la pertinence est difficile à automatiser, car elle est fortement subjective, c'est-à-dire dépendante de l'utilisateur.

Les travaux menés autour de la notion de pertinence ont montré qu'elle n'est pas une relation isolée entre un document et une requête, elle fait appel aussi au contexte de jugement, ainsi qu'à l'utilisateur lui même. [Denos, 97] fait remarquer qu'il existe une distance plus au moins grande entre les résultats d'un système de RI et les jugements de pertinence de l'utilisateur.

On distingue la pertinence système de la pertinence utilisateur :

- ➢ La pertinence système : c'est l'ensemble des principes qui sous-tendent la fonction de correspondance dans un système de recherche d'information. c.-à-d. c'est l'évaluation par le SRI, de l'équivalence entre des documents et une requête.

- ➢ La pertinence utilisateur : quant à elle, correspond à l'ensemble des jugements de pertinence que produit l'utilisateur du système de RI.

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

# III. Processus de recherche d'information

Un système de recherche d'information (SRI) permet de retrouver à partir d'une collection de documents les documents pertinents répondant à une requête d'utilisateur.

Pour répondre aux besoins en information de l'utilisateur, un SRI met en œuvre un processus pour réaliser la mise en correspondance des informations contenues dans un fonds documentaire d'une part, et les besoins en information des utilisateurs d'autre part. Un SRI est défini par ses modèles de représentation des documents, des besoins de l'utilisateur et sa fonction d'appariement.

Ce processus est composé de trois fonctions principales:

- **L'indexation des documents et des requêtes**

L'indexation a pour rôle d'extraire à partir d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique.

- **L'appariement document-requête ou l'interrogation**

La comparaison entre le document et la requête revient à calculer un score, supposé représenter la pertinence du document vis-à-vis de la requête.

- **La reformulation de la requête**

Qui intervient suite à une itération de recherche et consiste à modifier les requêtes en fonction des résultats présentés et le jugement de l'utilisateur.

Le processus général d'un SRI consiste à représenter le besoin en information, et en parallèle à collecter des documents et les représenter, à déterminer l'appariement entre chaque représentation interne d'un document et celle de la requête puis à décider si le document est pertinent. Les documents pertinents sont alors retournés à l'utilisateur dans une liste ordonnée par ordre décroissant de degré de pertinence. Afin d'améliorer les résultats de la recherche, le système peut être doté d'un mécanisme d'amélioration et de raffinement de la requête par reformulation.

Le fonctionnement générale d'un SRI est donné à travers le processus de recherche communément appelé processus en U, présenté en figure1.1.

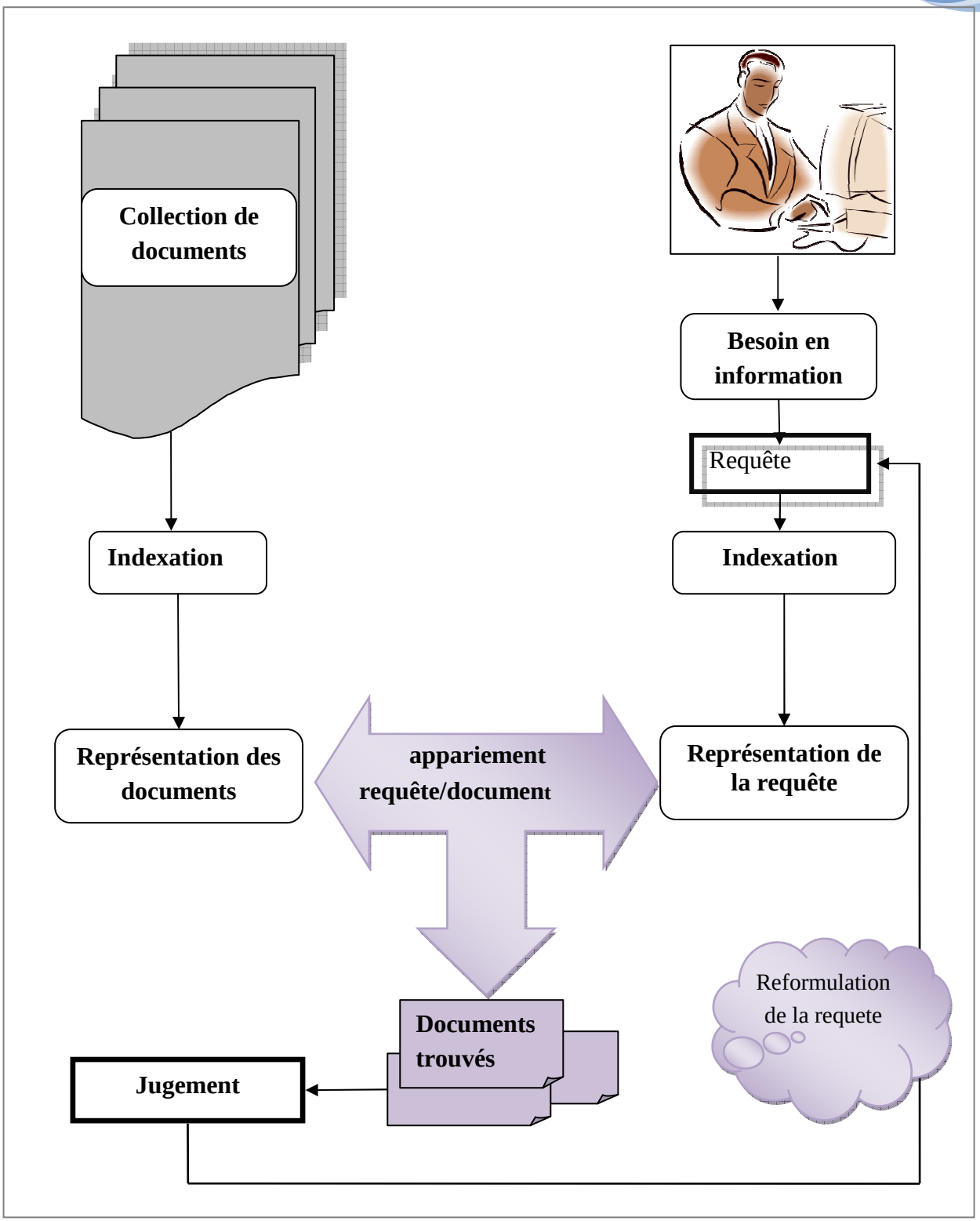



**Figure 1.1 : Processus en U de recherche d'information. [BADR, 2007].**

Ce processus fait ressortir trois mécanismes de base:

## III.1 Processus d'indexation

Un SRI gère les différentes collections de documents en les organisant sous forme d'une représentation intermédiaire permettant de refléter aussi fidèlement que possible leur contenu sémantique. L'interrogation de ce fond documentaire à l'aide d'une requête nécessite également la représentation de cette dernière sous une forme compatible avec celle des documents. Ce processus de conversion est appelé indexation (également appelé analyse pour la requête).

L'indexation donc est une étape très importante dans le processus de la recherche d'information. Elle consiste à extraire les termes représentatifs du contenu d'un document ou d'une requête. La qualité de la recherche dépend en grande partie de la qualité de l'indexation. Le résultat de l'idéation constitue, ce que l'on nomme le descripteur du document ou de la requête. Ce dernier est souvent une liste de termes ou groupe de termes significatifs pour l'unité textuelle correspondante, généralement assorti du poids représentant leur degré de représentativité du contenu sémantique de l'unité qu'ils décrivent. Les descripteurs des documents (mots, groupe de mots) sont rangé dans un catalogue appelé dictionnaire constituant le langage **d'indexation**

Techniquement, l'indexation peut être manuelle, automatique, ou semi-automatique :

- ***Manuelle***

L'indexation manuelle est réalisée par un expert documentaliste qui après analyse des documents, et utilisant son savoir faire, propose des termes représentants le contenu des documents considérés. L'indexation manuelle est fondée sur le jugement d'un être humain, et est cependant trop dépendante de l'état de connaissances des indexeurs, ce qui induit à une subjectivité des résultats. Son application est très coûteuse en temps et inadaptée à des collections de taille importante. Elle est cependant, utilisée notamment dans le monde de la documentation, comme dans les bibliothèques, et sert en général à la classification des ouvrages par thèmes. Dans ce cadre, un ensemble prédéfini de mots-clés et de catégories est défini précisément, pour faciliter l'indexation et la formulation des requêtes.

- ***Automatique***

L'indexation automatique est la technique d'indexation la plus utilisée lors d'un processus de RI. Elle offre l'avantage d'être totalement automatisée et donc plus adaptée pour les bases documentaires de grandes tailles.

Le SRI choisit les termes les plus représentatifs dans chaque document et leur associe des poids. Ce traitement est réalisé en plusieurs étapes: l'analyse lexicale, l'élimination des mots vides, la lemmatisation et la pondération.

- ✓ **L'analyse lexicale**

La première étape de l'indexation automatique consiste à extraire du document un certain nombre d'unités lexicales (termes). Une unité lexicale est définie comme étant une suite de caractères délimitée par des séparateurs (blancs, virgules …).

- ✓ **L'élimination des mots vides**

Cette étape permet de réduire la taille de l'index, et de ce fait, le temps de réponse du système. Elle consiste à éliminer les mots non significatifs, comme les articles, les prépositions et les conjonctions. Ce sont des mots qui ne sont pas discriminatoires lors d'une recherche, car ils ne traitent pas le sujet du document et sont présents en grand nombre dans les documents de la collection.

Deux techniques sont principalement utilisées pour l'élimination des mots vides:

- L'utilisation d'une liste prédéfinie de mots vides, aussi appelée anti-dictionnaire ou stoplist **[Fox, 1992]** ;
- L'utilisation de mesures statistiques sur la collection de documents pour déterminer les mots les plus fréquents ou les mots rares, qui sont « probablement » des mots vides.

- ✓ **La lemmatisation**

La lemmatisation consiste à extraire la forme canonique (racine) de chacun des termes issus des étapes précédentes. L'objectif étant d'éliminer les variations morphologiques des mots (nombre, genre …).

L'algorithme de Porter **[Porter, 1997]** est l'algorithme le plus utilisé pour la lemmatisation des mots de la langue anglaise. En français, il n'existe pas d'algorithmes fiables pour la lemmatisation vu, par exemple, la complexité des formes prises par les verbes du troisième groupe (le verbe être peut prendre les formes: est, sommes, fûmes …). De ce fait, on a souvent recours à un dictionnaire.

✓ **La pondération**

La pondération permet d'associer à chaque terme *ti* un poids représentant son importance dans un document *dj*. En général, ce poids est mesuré sur la base de considérations statistiques en utilisant les deux mesures suivantes:

- La fréquence locale (*tf*): représente la fréquence du terme *ti* dans le document *dj*. La fréquence d'un terme représente le nombre d'occurrences de ce terme dans un document. Il existe plusieurs variantes pour calculer *tf*, notamment par l'application du logarithme sur la fréquence du terme afin d'atténuer les écarts entre les fréquences (de l'ensemble des termes).

- La fréquence globale (*idf*): elle vise à donner une importance plus élevée aux termes qui apparaissent peu dans la collection de documents. Ainsi, un terme considéré comme rare dans la collection de documents aura un pouvoir discriminant plus élevé qu'un terme plus répandu **[Salton, 1975]** Pour ce faire, on associe à chaque terme une mesure égale au logarithme de l'inverse de la proportion des documents qui contiennent le terme :

$$idf(ti) = log \frac{|D|}{|\{dj: ti \in dj\}|}$$

Avec :

ti : le terme dont on cherche la fréquence;

|{dj : ti Є dj}|: le nombre de documents où le terme ti apparait;

|D| : le nombre total de documents dans la collection;

*Idf* (ti): la fréquence globale du terme ti.

Ces deux mesures (*tf* et *idf*) sont combinées en *tf* × *idf* pour fournir le poids du terme *ti*. Ce produit représente une bonne approximation du poids du terme, particulièrement si les documents de la collection sont de taille (longueur) homogène. Cependant, si la collection contient des documents très hétérogènes en termes de longueur, les documents les plus longs se voient favorisés (car plus un document est long, plus un terme a de chances d'apparaitre). Pour pallier cet inconvénient, Singhal **[Singhal, 1996]** et Robertson **[Robertson, 1997]** proposent d'intégrer la taille des documents a la formule de pondération : on parle de facteur de normalisation.

### ➢ L'indexation semi-automatique

L'indexation semi-automatique est une alternative qui profite des deux techniques d'indexation précédentes. La liste des descripteurs est construite d'abord de façon automatique, ensuite un expert documentaliste rajoute, manuellement, les termes qu'il juge représentatifs pour chaque document.

## III.2 L'appariement document-requête

L'objectif de SRI est le calcul de la pertinence d'un document par rapport à une requête, c'est une fonction d'appariement qui détermine le degré de ressemblance d'un document par rapport à une requête, et permet éventuellement de classer les documents par ordre de pertinence. Cette fonction est notée RSV (q, d) (Retrieval Status Value), ou q est une requête et d est un document de la collection.

La fonction d'appariement est indépendante de l'indexation et de la pondération des termes, par contre elle caractérise le SRI plus que le modèle d'indexation : la plupart des approches de recherche inspirent leur noms a partir de la façon dont ils entament l'appariement.

Il existe deux types d'appariement :

- **Appariement exact**

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

- **Appariement approché**

Le résultat est une liste de documents sensés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document/requête

### III.3 Reformulation de requête

Parfois l'utilisateur est confronté à une situation difficile, il est incapable de trouver les mots précis pour exprimer son besoin en information. Alors, certains pour ne pas dire la majorité des documents qui lui sont retournés l'intéressent moins que d'autres.

Outre cela, certaines requêtes sont courtes, du coup, elles ne sont pas sémantiquement riches, pour que le SRI puisse retourner des documents pertinents [Bai & al., 2006]. Afin de pallier ces problèmes, les chercheurs en RI se sont orientés vers l'intégration d'une étape supplémentaire dans le processus de recherche : la reformulation ou l'expansion de requêtes.

Elle consiste à modifier la requête initiale de l'utilisateur par l'ajout de termes significatifs et/ou la ré-estimation de leur poids.

Si les termes rajoutés proviennent des documents de la collection, on parle de réinjection de pertinence (relevance feedback). En revanche, s'ils sont issus d'une ressource conceptuelle externe (ontologie, thésaurus ou dictionnaire) on parle, dans ce cas de reformulation de requêtes directe. **[Khan, 2000] [Baziz & al., 2003] [Voorhees, 93] [Järvelin & al., 2004] et [Bhogal & al., 2006]** donnent un état de l'art complet de l'expansion des requêtes ainsi que des différentes approches suivies dans cet axe.

## IV. Modèles de Recherche d'Information

Le modèle de RI a pour rôle de déterminer le comportement clé d'un Système de Recherche d'Information. Il fournit donc un cadre théorique pour la modélisation de la mesure de pertinence.

Dans cette partie nous citons les principaux modèles de la Recherche d'Information qui ont été décrits avant dans de nombreux travaux sur la Recherche d'Information.**[Robertson,1994].**

- **Les modèles booléens** basés sur la théorie des ensembles dont on distingue : Le modèle booléen (booleen model), le modèle booléen étendu (extended boolean model) et le modèle basé sur les ensembles flous (fuzzy set model). Dans ces modèles, les termes de la requête sont séparés par des operateurs logiques (OR, AND, NOT). On peut alors effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associes a chaque terme.

- **Les modèles algébriques (vectoriels)** comme : Le modèle vectoriel (vector model), le modèle vectoriel generalise (generalized vector model), Latent Semantic Index model (LSI) et le modèle connexionniste. Dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel.

- **Les modèles probabilistes** : Le modèle probabiliste général, le modèle de réseau de document ou d'inférence (Document Network), et le modèle de langage. Ces modèles reposent sur la théorie des probabilités : pour ces modèles, la pertinence d'un document vis-à-vis d'une requête est vue comme une probabilité de pertinence document/requête.

Dans la suite, nous détaillons les trois principaux modèles exploités en RI à savoir : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

## IV.1. Le modèle booléen

Le modèle booléen a été le premier à être utilise dans le monde de la RI. C'est le plus simple des modèles de RI, et permet de faire une recherche très restrictive et obtenir, pour un utilisateur expérimenté, une information exacte et spécifique **[Salton, 1971]**. Il est basé sur l'algèbre de Boole et considère ainsi une requête comme une expression logique. Il considère aussi que les termes de l'index sont soit présents soit absents d'un document. En conséquence, les poids des termes dans l'index sont binaires c'est-a-dire wij= {0,1}.

Dans ce modèle :

Un document *d* est représenté comme une conjonction logique des termes non pondérés.

Exemple: d =t1 $\wedge$ t2 $\wedge$ t3 $\wedge \cdots \wedge$ tn[1].

Une requête *q* est composée de termes liés par 3 connecteurs logiques ET, OU, NON. Exemple : q = (t1 $\wedge$ t2) $\vee$ (t3 $\wedge \neg$t4) ou *q* : la requête, *ti*: terme d'indexation.
La similarité entre un document et une requête est définie par :

$$RSV\ (q,d) = \begin{cases} 1\ Si\ d\ appartien\ \text{à}\ l'ensembled"crit\ par\ la\ requete \\ 0\ Sinon \end{cases}$$

Ainsi, le modèle booléen affirme que chaque document est soit pertinent soit non-pertinent, donc répond exactement a la requête qui a été formulée. Il n'y a pas de notion de réponse partielle aux conditions de la requête.

D'autre part, il est souvent très difficile pour l'utilisateur d'exprimer son besoin en information avec des expressions booléennes ce qui ne permet pas d'utiliser au mieux les caractéristiques de ce modèle.

Enfin, tous les termes dans un document ou dans une requête sont pondérés de la même façon simple (0 ou 1), donc tous les termes ont la même importance dans le document. Cela ne correspond pas a ce qu'on souhaite avoir en RI.

## IV.2. Le modèle vectoriel

Le modèle vectoriel introduit par Salton **[Salton, 1971]**, repose sur les bases mathématiques des espaces vectoriels. Dans ce modèle, les documents et les requêtes sont représentés dans un espace vectoriel engendré par l'ensemble des termes d'indexation $t_1, t_2, t_3, \ldots t_T$. Où N est le nombre total de termes issus de l'indexation de la collection des documents.
Chaque document est représenté par un vecteur : $D_j = (d_{1j}, d_{2j}, \ldots\ldots, d_{ij}, \ldots\ldots\ldots, d_{Tj})$.
Chaque requête est représentée par un vecteur : $Q = (q_1, q_2, \ldots\ldots\ldots\ldots, q_i, \ldots\ldots\ldots, q_T)$.
Avec :

**$d_{ij}$** : Poids du terme $t_i$ dans le document $D_j$.

**$q_i$** : Poids du terme $t_i$ dans la requête Q.

Les termes de poids nul représentent les termes absents dans un document alors que les poids positifs représentent les termes assignés.

La fonction de calcul du coefficient de similarité entre chaque document, représenté par le vecteur *Dj*($d_{1j}$ , $d_{2j}$ , ……,$d_{ij}$,……..,$d_{Tj}$) , et la requête, représentée par le vecteur *Q* ($q_1$,$q_2$,……,$q_i$,……...,$q_T$) est appelée (***R**etrieval **S**tatus **V**alue)* ou ***RSV***.

Ce coefficient de similarité est calculé sur la base d'une fonction qui mesure la colinéarité des vecteurs documents et requête. On peut citer notamment les fonctions suivantes :

- **Produit scalaire :**

$$RSV(Q, D_j) = \sum_{t=1}^{T} q_i * d_{ij}$$

- **Mesure de Jaccard :**

$$\frac{\sum_{t=1}^{T} q_i * d_j}{\sum_{t=1}^{T} q_i^2 + \sum_{t=1}^{T} d_{ij}^2 - \sum_{t=1}^{T} q_i * d_{ij}}$$

- **Mesure de cosinus :**

$$\frac{\sum_{t=1}^{T} q_i * d_j}{(\sum_{t=1}^{T} q_i^2)^{1/2} * (\sum_{t=1}^{T} d_{ij}^2)^{1/2}}$$

Le modèle vectoriel permet de pallier à l'un des inconvénients majeurs du modèle booléen, en permettant de trier les documents répondant à une requête. Les documents dans le modèle vectoriel, sont en effet restitués dans un ordre décroissant de leur degré de similarité avec la requête. Plus le degré de similarité d'un document est élevé, plus le document ressemble à la requête, et donc pertinent pour l'utilisateur.

Un des principaux défauts de l'approche vectorielle est, qu'elle considère tous les termes comme étant indépendants, et ne tient donc pas compte des liens entre ceux-ci (exemple : voiture, automobile)

## IV.3. Le modèle probabiliste

Le modèle probabiliste a été proposé par Robertson **[Robertson]** et Maron **[Maron, 1960]**, il utilise un modèle mathématique fondé sur la théorie de la probabilité conditionnelle (appelé aussi modèle de la théorie de pertinence). Lors du processus d'indexation deux probabilités conditionnelles sont utilisées :

- **P ( $t/_{pert}$ )** : La probabilité pour que le terme « ***t*** » apparaisse dans un document donné sachant que ce document est pertinent pour la requête.
- **P ( $t/_{NonPert}$ )** : La probabilité pour que le terme « ***t*** » apparaisse dans un document donné sachant que ce document n'est pas pertinent pour la requête.

En supposant que la distribution des termes dans les documents pertinents est la même que leur distribution par rapport à la totalité des documents, et que les variables « document pertinent » et « document non pertinent» sont indépendantes, la fonction de recherche est obtenue en calculant la probabilité de pertinence d'un document **P (Pert/D).**

- $P\ (Pert/D) = \frac{P(D|Pert)*P(Pert)}{P(D)}$

- $P\ (NonPert/D) = \frac{P(D|NonPert)*P(NonPert)}{P(D)}$

Avec P(D)= P (D/Pert)*P(Pert) +P (D/NonPert)*P(NonPert).

Où : P (D/Pert) est la probabilité d'observer D sachant qu'il est pertinent.

P (Pert) est la probabilité à priori qu'un document soit pertinent.

Le coefficient de similarité requête document (*RSV*) peut être calculé par différentes formules. Robertson et Spark-Jones proposent la formule suivante:

- $RSV = \sum log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$

Avec :

N: nombre total de documents de la base,

n: nombre de documents contenant le terme,

R: nombre de documents connus comme étant pertinents,

r: nombre de documents connus comme étant pertinents et contenants le terme.

L'ajout de 0.5 à tous les membres s'explique par la nécessité d'écarter tous les cas limites qui entraîneraient des valeurs nulles de ces membres.

Le modèle probabiliste est fonctionnel et a montré une certaine efficacité mais néanmoins il présente quelques inconvénients comme la complexité des modèles de calcul de probabilité par rapport aux autres modèles.

## V. Evaluation des systèmes de recherche d'information

L'aspect évaluation occupe une place très particulière en RI. L'évaluation des systèmes de recherche d'information **[Rijsbergen, 79], [Baeza-Yates & al., 99]** constitue une étape primordiale dans l'élaboration d'un modèle de recherche d'information. Elle permet de caractériser le modèle et de fournir des éléments de comparaison entre les modèles existants.

D'une façon générale, un système de recherche d'information idéal a deux objectifs :

- Retrouver tous les documents pertinents,

- Rejeter tous les documents non pertinents.

Ces deux objectifs sont évalués par des mesures de précision et de rappel que nous définissons ci-après.

## V.1. La mesure de Précision/Rappel

Les mesures de précision/rappel sont obtenues en partitionnant l'ensemble des documents restitués par le SRI en deux catégories : les documents pertinents et les documents non pertinents. Ces deux catégories se définissent comme suit:

- **Rappel**

Le rappel est le rapport de documents pertinents restitués par le système sur l'ensemble des documents pertinents contenus dans la base documentaire. Elle mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête.

$$\text{Rappel} = \frac{\text{nombre de documents pertinents trouvés}}{\text{nombre de documents pertinents}}$$

- **Précision**

La précision est le rapport de documents pertinents trouvés sur l'ensemble des documents restitués par le système. Elle mesure la capacité du système à rejeter tous les documents non pertinents à une requête donnée.

$$\text{Précision} = \frac{\text{nombre de documents pertinents trouvés}}{\text{nombre de documents trouvés}}$$

Le rappel et la précision permettent aussi de définir le silence documentaire et le bruit documentaire qui représentent respectivement le nombre des documents pertinents non retrouvés et le nombre de documents non pertinents retrouvés. La figure 2 schématise ces deux variables dans l'ensemble des documents.

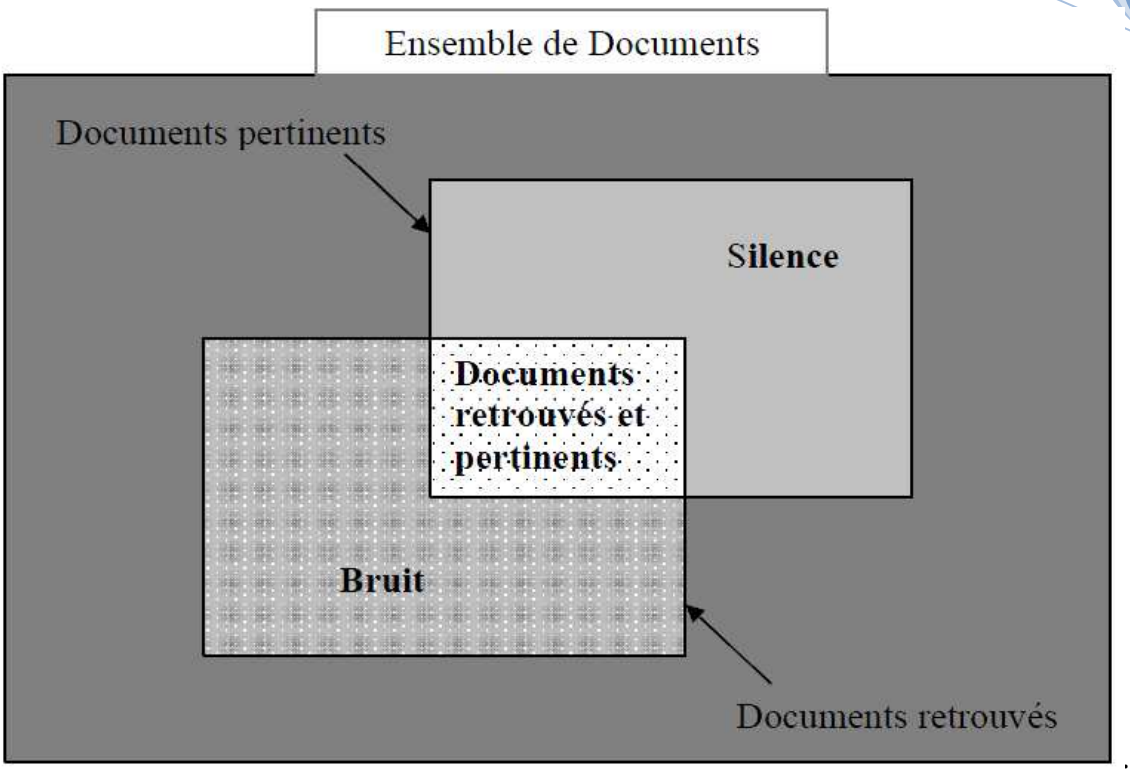



**Figure 1.2 : Bruit et Silence [Soule, 2007]**

Idéalement, on voudrait qu'un système donne de bons taux de précision et de rappel en même temps. Les deux métriques ne sont pas indépendantes. Il y a une forte relation entre elles: ***quand l'une augmente, l'autre diminue***

Le comportement d'un système peut varier en faveur de précision ou en faveur de rappel (au détriment de l'autre métrique). Ainsi, pour un système, on a une courbe de précision-rappel qui a en général l'aspect suivant:

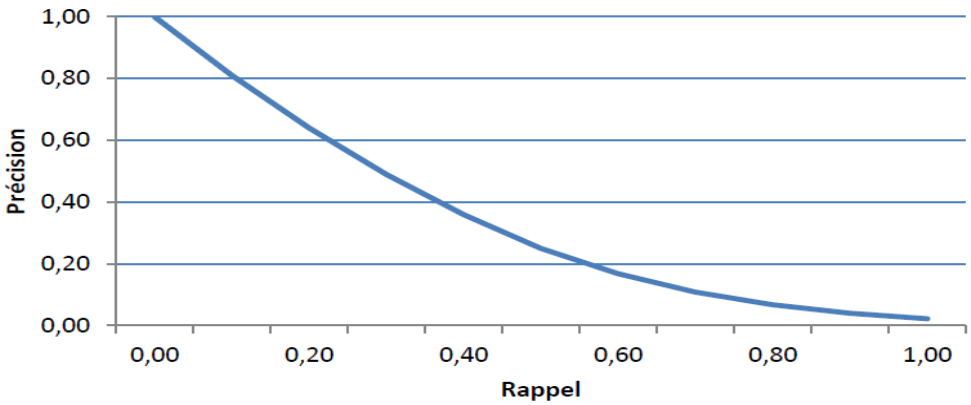



**Figure 1.3:** Courbe Rappel-Précision.

## V.2 Précision à X

Pour faciliter les calculs de précision et de rappel, on calcule en général la précision à x. Elle représente la proportion de documents pertinents présents dans les *x* premiers documents retrouvés, en supposant ces documents triés selon leur valeur de pertinence (*RSV*). Ainsi, on calcule typiquement les précisions à *5, 10, 15* pour un système, et ces mesures sont notées alors *P5, P10, P15*, etc. La *précision exacte* découle de ces mesures. C'est la précision à *x* où *x* est le nombre total de documents pertinents dans la collection pour la requête.

La *précision moyenne* est aussi calculée pour chaque requête. C'est la moyenne des précisions pour chaque document pertinent. La précision d'un document pertinent est la précision à *x, x* étant la position de ce document dans la liste triée des documents retrouvés. Si un document pertinent n'est pas retrouvé, sa précision est nulle. Enfin, les précisions moyennes pour l'ensemble des requêtes, qui sont donc les moyennes des précisions (moyennes, exactes, . . .) par requête, permettent d'obtenir une mesure de la performance globale du système.

## V.3. Autres mesures de performance

Il existe aussi d'autres mesures de performance des SRI telles que :

- **Le temps de réponse** : un SRI doit pouvoir fournir à l'utilisateur les documents correspondants à sa demande dans des temps très courts.
- **La présentation des résultats claire avec facilité d'utilisation** : capacité du système à comprendre les besoins de l'utilisateur et à mettre en valeur les documents correspondants à ceux-ci. Ceci est lié à l'interface avec l'utilisateur.
- **Le nombre total de documents pertinents retournés**, ou le rappel à 1000 documents : ces mesures permettant d'évaluer la performance globale du système au final, en fonction ou non du nombre de documents pertinents total.
- **Le rang du premier document pertinent** : cette mesure a été proposée pour prendre en compte la satisfaction de l'utilisateur qui chercherait un seul document pertinent (comme c'est éventuellement le cas pour les moteurs de recherche sur Internet).

- **La longueur de recherche** : elle est égale au nombre de documents non pertinents que doit lire l'utilisateur pour avoir un certain nombre n de documents pertinents.

## Conclusion

Dans ce premier chapitre, nous nous sommes essentiellement intéressés à l'étude des concepts de base et des principaux modèles classiques existants de la recherche d'information.

Les systèmes de recherche d'information sont conçus à l'origine pour retrouver les documents pertinents d'un sujet donné. La finalité de chaque système de recherche d'information est de satisfaire les besoins des utilisateurs. Ces derniers sont préoccupés par un seul problème : celui de pouvoir récupérer tous les documents dont ils ont besoin d'une façon rapide et efficace.

Or, le problème majeur des SRI classiques, est qu'ils se contentent de chercher les documents qui contiennent les mêmes mots que ceux de la requête. Ceci est évidement insuffisant dans la mesure où la performance de ces systèmes se voit détériorée.

Pour remédier à ce problème, plusieurs approches ont été développées dans le but de personnaliser les systèmes de recherche d'information en intégrant le profil utilisateur qui sera bien expliqué dans le chapitre suivant.

CHAPITRE

II

# Recherche d'information Personnalisée

# Introduction

Les récents progrès des technologies de l'information de manière générale, des réseaux de communication de manière particulière, ont redonné à l'information de nouveaux contours et d'avantage de valeur selon divers aspects : scientifique, technique, économique, d'usage etc.… De surcroit, les progrès techniques de numérisation et de compression de l'information, ont encouragé sa production, circulation et exploitation.

Certes, les systèmes de recherche d'information sont des outils qui ont permis, jusqu'à aujourd'hui, d'améliorer sans cesse la qualité des services d'accès à l'information, grâce à la capitalisation des théories issues de nombreux travaux de recherche ; cependant, en raison de la surabondance de l'information d'une part et de sa large accessibilité à travers le web, d'autre part, leur mise en œuvre est confrontée à de nouveaux problèmes. En effet la situation est actuellement paradoxale : la masse d'informations est telle que l'accès à une information pertinente, adaptée aux besoins d'un utilisateur donné devient à la fois difficile et nécessaire.

C'est pourquoi les travaux s'orientent actuellement vers la révision de la chaine d'accès à l'information dans la perspective d'intégrer l'utilisateur comme composante du modèle global de recherche et ce, dans le but de lui délivrer une information pertinente, adaptée à ses besoins précis, son contexte et ses préférences. Ces travaux s'inscrivent dans le cadre précis de la personnalisation de l'information qui est vue comme l'une des solutions pouvant maintenir le Web come une ressource viable.

## I. Problématique

L'accès à une information pertinente, adaptée aux besoins et profil de l'utilisateur est un enjeu capital dans le contexte actuel caractérisé par une prolifération massive de ressources d'informations hétérogènes. Malgré les développements récents dans le domaine de la recherche d'information, force est de constater que les résultats produits par un moteur de recherche lors de leurs activités quotidienne. Les raisons évoquées portent essentiellement sur la grande masse, l'incompréhensibilité et ambigüité des informations retournées à leurs requêtes.

Dans ce cadre, la personnalisation est une dimension qui permet la mise en œuvre de systèmes centrés utilisateurs, non dans le sens d'un utilisateur générique mais d'un utilisateur spécifique et ce, en vue d'adapter son fonctionnement à son contexte précis.

**Exemple :**

On considère l'exemple suivant dans lequel deux utilisateurs lancent une même requête portant le mot « virus ».

On peut supposer différents scénarios d'utilisation :

- **Scénarios 1 :** L'utilisateur est un informaticien qui cherche une documentation sur les virus informatiques.
- **Scénarios 2:** L'utilisateur est un biologiste qui cherche de l'information sur les virus dans le domaine de la biologie.

Ces deux scénarios illustrent sans doute la déférence d'interprétation d'une requête en fonction du contexte de l'utilisateur qui l'exprime.

En clair, le problème n'est pas tant la disponibilité de l'information, mais sa pertinence relativement à un contexte d'utilisation particulier.

De récentes études montrent que l'origine de ces problèmes réside dans le caractère non personnalisé du processus d'accès à l'information.

La ***RI personnalisée*** est née en vue de palier à ces problèmes.

# II. Le profil utilisateur et la personnalisation de l'information

## II.1 Définition de la recherche d'information personnalisée

La recherche d'information personnalisée (RIP) est une direction de recherche qui permet la mise en œuvre des systèmes d'accès à l'information qui intègrent les caractéristiques de l'utilisateur, tant dans le processus de recherche que dans la visualisation de l'information et ce afin d'adapter le fonctionnement de système aux buts, préférences et capacités de l'utilisateur **[Bel & Dje ,09].**

D'une façon plus générale la recherche d'information personnalisée combine un ensemble de technologies et de connaissances portant sur la requête et le profil utilisateur, dans une infrastructure et ce, dans le but de délivrer les réponses les mieux appropriées à son besoin en information **[NFUHR ,2000]**. La recherche d'information personnalisée est une activité qui fait intervenir en grande partie l'utilisateur, assimilé en pratique, non plus à la requête, mais à un ensemble de facettes qui le décrivent. Ces facettes décrivent une structure qui est appelée communément ***profil*** et qui couvre, de manière non exhaustive, ses caractéristiques socioprofessionnelles, ses centres d'intérêt et ses préférences.

Indépendamment de l'objectif applicatif visé, on identifie trois aspects à promouvoir dans les systèmes de recherche d'information personnalisée :

- Capacité à identifier l'intention conceptuelle de l'utilisateur,
- Possibilité d'exprimer des informations liées au contexte d'utilisation courant,
- Convivialité des interactions utilisateurs-système.

## II.2 Profil utilisateur

Le profil de l'utilisateur est une structure d'informations hétérogènes qui couvre des aspects larges tels que l'environnement cognitif, social et professionnel de l'utilisateur ; ces informations sont généralement exploitées dans le but de préciser ses intentions au cours d'une session de recherche [COO, 02].

On doit également prendre en compte ce que l'utilisateur préfère et l'historique des interactions avec le système, les centres d'intérêts et les besoins en informations de l'utilisateur :

### ➢ Les préférences

Une préférence utilisateur est un ensemble de descriptions englobant ce qu'un utilisateur envisage d'accomplir dans le système (les activités) et la manière de le faire (séquentielle, concurrente, conditionnelle), le type et l'ordre des résultats de ces activités (le contenu), et la manière dont il aimerait que l'information soit affiché. De cette définition nous distinguons trois types de préférences :

- **Les préférences d'Activité :** Concernent les activités qu'un utilisateur souhaite et peu accomplir dans le système.
- **Les préférences de résultat :** Concernant le contenu et le format préféré des résultats des fonctionnalités (par exemple de la vidéo, du texte, ou des images).
- **Les préférences d'Affichage :** Concernent la manière dont l'utilisateur souhaite que l'information soit affichée.

### ➢ Le centre d'intérêt

Les centres d'intérêt traduisent des domaines de connaissances ciblés régulièrement par l'utilisateur au cours de ses sessions de recherche. IL peut être défini par un ensemble de mots clés (concepts) ou un ensemble d'expressions logiques (requête)

### ➢ Le besoin en information

Un besoin en information de l'utilisateur est exprimé à travers une requête en utilisant le langage d'interrogation qui dépond du système de recherche d'informations.

## III. Modélisation de l'utilisateur

- **Définition**

La modélisation de l'utilisateur est une discipline de recherche datant des années 70 et évoquant en premier lieu les travaux *d'Allen, Cohen et Perrault*. La préoccupation majeure de cette discipline est d'améliorer la qualité des interactions homme-machine par inférence et prédiction des buts, préférences et contexte des utilisateurs à partir de faits observés. Les premières applications étaient les systèmes de reconnaissances de plans.

Par la suite, les méthodes issues de la modélisation de l'utilisateur ont investi et continuent à investir de nombreux domaines portant sur la mise en œuvre de systèmes intelligents tels que les systèmes ayant recours à l'analyse de langage naturel, système d'aide à l'apprentissage, système hypermédia adaptatifs et tous les systèmes personnalisées de manière générale.

Indépendamment du domaine d'application, tout système mettant en œuvre des méthodes de modélisation de l'utilisateur inclut en partie les paquets d'informations suivant :

- Des informations personnelles associées à l'utilisateur telles que l'âge, le pays, la langue,
- Les préférences : peuvent être de différents nivaux tels que préférences de forme (style de la page, longueur d'un document) et préférences de domaine permettant de cibler le centre d'intérêt de l'utilisateur,
- Historique de l'utilisateur : les interactions passées de l'utilisateur représentent une source pour prédire ses intentions et lui recommander des objets.

Les approches et techniques de la modélisation utilisateur peuvent être basé sur des modèles simples ou complexes dépendant de l'objectif final ou domaine d'application du système ; un effort de standardisation pour la généralisation de tels systèmes afin de produire des *Shell* a toutefois été mené et semble donner une meilleure portée au devenir des méthodes de modélisation de l'utilisateur [Kob 2001].

En outre, ces méthodes peuvent être interactives ou implicites, peuvent avoir une portée d'adaptation sur une session d'utilisation du système avec des informations très générales sur l'utilisateur ou sur plusieurs sessions d'utilisations du système avec un modèle plus élaboré.

## III.1. Approches et techniques de modélisation de l'utilisateur

On distingue trois principales approches de modélisation de l'utilisateur **[Gow ,2003]** approche canonique, approche explicite et approche automatique.

### ➢ Approche canonique

Cette approche préconise l'intégration de modèles d'utilisateurs typiques lors de la conception du système. Les interactions permettent de cataloguer l'utilisateur courant par rapport à un modèle prédéfini dans le système. Cette approche a été peu performante, notamment en raison de l'inadéquation des langages des concepteurs et des utilisateurs pour décrire les situations permettant d'apparier l'utilisateur à un modèle canonique [Gre ,84].

Renseignées par sélection d'un profil préexistant créé par des experts du domaine (***profil expert***) : cette méthode nécessite que l'on adresse la liste des profils et que l'on décrive chacun d'entre-deux. Cette approche est statique, donc une fois que le système à démarré, il est difficile de mettre à jour les profils utilisateurs.

### ➢ Approche explicite

Dans le cas de cette approche, le système maintient un panel de modèles canoniques caractérisés par une partie flexible qui est contrôlée par l'utilisateur lors de ses interactions avec le système.

Cette approche remédie aux inconvénients de l'approche canonique en réduisant l'erreur de catalogage due à une description incertaine des situations. Cependant, elle induit une surcharge cognitive pour l'utilisateur et une complexité dans la conception du système.

### ➢ Approche automatique (Implicite)

Dans le but de pallier au problème de surcharge cognitive et d'incertitude engendré par les approches précédentes, l'approche automatique préconise d'inférer le modèle de l'utilisateur, non pas à partir de ses interactions explicites avec le système, mais à partir des informations collectées implicitement lors de ses sessions d'utilisation du système. En clair, le comportement de l'utilisateur est la source permettant de prédire implicitement son modèle.

C'est le modèle le plus répandu actuellement. Deux principales classes de techniques sont issues de cette approche et peuvent être combinées : les techniques collaboratives et techniques statistiques, développées dans se qui suit **:**

- **Techniques collaboratives**

Les techniques collaboratives sont basées sur l'idée de prédire le modèle individuel d'un utilisateur courant sur la base d'un comportement assimilable à celui d'un groupe d'utilisateurs. Les utilisateurs du système participent ainsi collectivement à alimenter des stéréotypes qui sont affectés à des groupes d'intérêts communs puis utilisés pour prédire les préférences inconnues de nouveaux utilisateurs.

Les systèmes de recommandation sont généralement basés sur de telles techniques. Cependant leur efficacité dépend fortement du degré de corrélation du groupe **[Pcg ,03]**. De

plus, des problèmes liés à la taille et composition des groupes sont posés. En effet, l'approche reste peu performante pour un nouvel utilisateur (avec peu d'informations collectées à partir du groupe) et de nouveaux centres d'intérêt **[Zuc , 01].**

- **Techniques statiques**

Ces techniques sont basées essentiellement sur des modèles théoriques issus de la statistique, soutenus par des heuristiques et algorithmes appropriés. Les principaux modèles sont : le modèle linéaire, le modèle markovien, les réseaux de neurones, la classification, les règles d'induction et les réseaux bayésiens :

✓ **Modèle linéaire**

Le modèle linéaire a une structure simple. L'hypothèse de base est que la valeur de prédiction présumée est inconnue, un objet cible du système est une combinaison linéaire des valeurs calculées à partir d'un comportement passé de l'utilisateur. Le modèle linéaire peut être aisément combiné avec des techniques collaboratives où les valeurs connues sont issues de l'appréciation des membres du groupe associé à l'utilisateur courant [Riw, 94].

✓ **Modèle Markovien**

Ce modèle est essentiellement basé sur l'hypothèse markovienne qui permet de représenter une séquence d'événements passés, la théorie markovienne offre alors des éléments pour calculer la probabilité d'occurrences des événements futurs**.**

✓ **Réseaux de neurones**

Les réseaux de neurones sont des structures basées sur l'interconnexion de nœuds et un principe d'activation par propagation de signaux à travers les connexions depuis les entrées jusqu'aux sorties. La signification effective des nœuds, des connexions et valeurs d'activation, dépond du problème pour lequel est dédié le réseau. De manière générale, les réseaux de neurones sont destinés à résoudre des problèmes de décision non linéaires. Dans le cas précis du vaste domaine d'application de la modélisation utilisateur, l'entrée représente

une situation ou faits observables à partir de l'utilisateur, les sorties représentent des objets cibles du système avec des valeurs d'activation qui traduisent le degré de prédiction.

- ✓ **Classification**

Les méthodes de classification permettent de partitionner un espace d'objets en classes de manière à réduire sa dimension.

Les objets d'une même classe ont des propriétés partageables, dont le degré de corrélation est calculé à l'aide métriques basées sur la similarité. De nombreuses stratégies de classification sont proposées dans la littérature. Du point de vue de la modélisation utilisateur, les méthodes de classification permettent généralement d'identifier la classe de caractéristiques de l'utilisateur courant à partir d'informations dérivées de son comportement.

- ✓ **Induction de règles**

L'induction de règles consiste en l'apprentissage de règles de prédiction à partir d'un ensemble de faits observés.

Contrairement aux méthodes de classification, les techniques d'induction de règles requièrent, durant l'apprentissage du système, l'identité de la classe associée à chaque observation.

Ces techniques produisent un ensemble de règles proprement dit ou des arbres de décision.

- ✓ **Réseaux Bayésiens**

Les réseaux bayésiens [Per ,88] ont largement investi, ces dernières années, les travaux sur la modélisation utilisateur [Jam ,96]. Les réseaux bayésiens sont des graphes acycliques orientés où les nœuds correspondent à des variables aléatoires. Les nœuds sont interconnectés à l'aide de liens orientés qui représentent des liens de causalité entre nœuds parents et nœuds fils. A chaque nœud est associée une distribution de probabilités conditionnelles qui permet d'assigner au nœud, une valeur de probabilité dépendante de la combinaison des valeurs de probabilités possibles des nœuds parents. Les réseaux bayésiens sont plus flexibles que les techniques précédentes, en ce sens qu'ils permettent de représenter explicitement les relations de causalité entre faits et d'émettre des prédictions sur de nombreux paramètres du système [Zuc, 01].

La modélisation d'un profil utilisateur se fait en deux grandes étapes : Représentation du profil et construction du profil.

## III.2. Représentation du profil utilisateur

Le profil de l'utilisateur n'a pas forcément de structure explicite qui le représente, Il peut être constitué de paquets divers d'informations qui traduisent une connaissance éparse sur l'utilisateur. Dans ce sens, la représentation des profils rejoint en grande partie la représentation de l'information dans le contexte de la recherche d'information. Il n'ya pas à notre connaissance de modèle spécifique, dédié à la représentation du profil de l'utilisateur. Les modèles de représentation proposés puisent largement de ceux proposés en recherche d'information. On en cite principalement quatre type de représentation : vectorielle, sémantique, connexionniste et multidimensionnelle.

- **Représentation vectorielle**

Ce type de représentation s'appuie généralement sur le modèle vectoriel **[Sal 1971]**. Le profil est représenté par un ou plusieurs vecteurs défini dans un espace de termes obtenus implicitement ou explicitement à partir de plusieurs sources d'informations. Les coordonnées des vecteurs correspondent aux poids des termes dans le profil, l'utilisation de plusieurs vecteurs permet de prendre en compte la diversité des domaines d'intérêts ou évolution dans le temps.

Ce type de représentation offre l'avantage indéniable de la simplicité de mise en œuvre. Cependant les modèles proposé ne mettent pas en évidence ni la dimension liée au temps marquant l'évolution des profils, ni à l'organisation des informations pour hiérarchiser les centres d'intérêt.

- **Représentation sémantique**

Cette représentation est essentiellement basée sur l'utilisation d'ontologie ou réseaux sémantiques probabilistes. Elle met en évidence, dans ce cas les relations sémantiques informations de contenant. Elle permet d'exprimer des relations sémantiques entre les unités d'information décrivant le profil **[PG 1999 ; SKW 2000].**

- **Représentation connexionniste**

C'est un type de représentation basé sur l'interconnexion de nœuds représentant les termes, préférences **[JH 1999]** ou documents. Il offre le double avantage de la structuration et de la représentation associative permettant de considérer l'ensemble des aspects représentatifs du profil. **[Jen, 93]**.

- **Représentation multidimensionnelle**

C'est un type de représentation qui se veut globale dans le sens ou il permet de capturer puis catégoriser l'ensemble des informations caractérisant le profil de l'utilisateur. Dans cette direction, les propositions de standards P3P pour la sécurisation des profils ont défini des classes distinguant :

- ✓ Les attributs démographiques des utilisateurs (identité, données personnelles).
- ✓ Les attributs professionnels (employeur, adresse, type).
- ✓ Les attributs de comportement (trace de navigation).

Ces tentatives de structurations sont louables mais insuffisantes pour couvrir le champ de la personnalisation. Par ailleurs, elles se contentent de catégoriser les informations de profil sans expliciter les corrélations qui existent entre elles. En effet, les attributs de comportement peuvent être corrélés aux attributs personnels ou professionnels. De même, les données de sécurité peuvent caractériser aussi bien les données personnelles que les données collectées.

On distingue principalement sept dimensions (ou catégories), capables d'accueillir la plupart des informations caractérisant un profil. **[Kostadinove, 03]**

- Les données personnelles,
- Le centre d'intérêt,
- L'ontologie de domaine,
- La qualité attendue des résultats délivrés,
- La customisation,
- La sécurité et la confidentialité,
- Le retour de préférences (feedback).

- **Les données personnelles**

Les données personnelles sont la partie statique du profil. Elles comprennent l'identité de l'utilisateur (nom, prénom, numéro de sécurité sociale, etc.), des données démographiques (âge, genre, adresse, situation familiale, nombre d'enfants, etc.), les contacts personnels et professionnels de l'utilisateur et d'autres informations comme le numéro de la carte bancaire ou de la carte Vitale.

Les données personnelles sont relativement stables dans le temps et ne demandent pas de mise à jour automatique par le gestionnaire de profils. Dans plusieurs approches ces données ne jouent pas un rôle dans le processus de recherche d'informations, mais servent souvent comme monnaie d'échange contre les services de personnalisation ou les services d'accès. C'est le cas des systèmes de vente sur Internet qui demandent aux utilisateurs de fournir des formulaires de données personnelles qui sont ensuite utilisées pour faire des statistiques pour mieux cibler les clients.

- **Le centre d'intérêt**

Le centre d'intérêts exprime le domaine d'expertise de l'utilisateur ou son périmètre d'exploration. Il peut être défini par un ensemble de mots clés ou un ensemble d'expressions logiques (requêtes). On peut rapprocher le centre d'intérêt de la notion de vue en bases de données, à ceci prés qu'en recherche d'information, on n'a pas la notion de schéma du réel auquel on s'intéresse et sur lequel une vue est généralement définie. Le centre d'intérêt peut vu comme une présélection virtuelle qui réduit la masse d'informations à prendre en compte. Par conséquent toute requête émise par l'utilisateur sera enrichi avec les mots clés ou les prédicats des requêtes définissant le centre d'intérêt.

- **L'ontologie du domaine**

L'ontologie du domaine complète la définition du centre d'intérêts en explicitant la sémantique de certains termes ou de certains opérateurs employés par l'utilisateur dans son profil ou dans ses requêtes. Par exemple, on peut explicitement définir que « BD » signifie « bases de données » dans le profil et non « bande dessinée », que dans le contexte du profil « client » et « consommateur » sont synonymes, et que l'opérateur « Proche de Y » signifie à

moins de 5 km de distance. Il est ainsi possible d'avoir plusieurs interprétations pour un même opérateur en fonction du profil dans lequel est utilisé.

- **La qualité attendue des résultats délivrés**

La qualité est l'un des facteurs clés de la personnalisation ; elle permet d'exprimer des préférences extrinsèques comme l'origine de l'information, sa précision, sa fraicheur, sa durée de validité, le temps nécessaire pour la produire ou la crédibilité de sa source. Les attributs de cette dimension expriment la qualité attendue ou espérée par l'utilisateur. Le système de recherche d'information aura en charge de confronter ces exigences de qualité à la qualité effective des données qu'il aura explorées ou sélectionnées.

Il faut noter que la qualité d'un produit informationnel ne se mesure pas toujours sur le produit lui-même, mais quelquefois sur sa source de production ou son processus de production. La qualité d'une information factuelle (précision de l'adresse d'une personne) ne se mesure pas comme la qualité d'un agrégat statistique (moyenne des revenus des français).

Le mode de production et les opérandes de la production peuvent êtres déterminants dans ce dernier cas. La qualité attendue peut donc être raffinée en plusieurs sous-dimensions caractérisant le contenu, le contenant, le processus de production, etc.

- **La customisation**

La customisation concerne d'abord tout ce qui est lié aux modalités de présentation des résultats en fonction de la plateforme, de la nature et du volume des informations délivrés, des préférences esthétiques ou visuelles de l'utilisateur. A ces modalités de présentation, on peut ajouter les modalités d'exécution, décrivant le moment d'exécution d'une requête (mode pull ou push), la manière de notifier les résultats (différé, immédiat par exemple) et la qualité de résultats que l'on souhaite recevoir (les Top k, les premiers calculés, etc.).

- **La sécurité et la confidentialité**

La sécurité est une dimension fondamentale du profil. Elle peut concerner les données que l'on interroge ou modifie, les informations que l'on calcule, les requêtes utilisateurs elles-mêmes ou les autres dimensions du profil. La sécurité des données peut être exprimée par des niveaux de sécurité prédéfinis qui dépendent de la hiérarchie des vues autorisées, par des

clauses d'octroi ou de révocation de droits à SQL, par le support d'identification ou de stockage (carte à puce, certificat web etc.) et par des moyens de transmission utilisés (protocoles, cryptage). La sécurité du processus exprime la volonté de l'utilisateur de cacher un traitement qu'il effectue. Ceci peut être fait en définissant le degré de visibilité de certaines opérations.

- **Le retour des préférences**

On désigne par ces termes ce qu'on appelle communément le « feedback » de l'utilisateur. Cette dimension regroupe l'ensemble des informations collectées sur le comportement de l'utilisateur, que ces informations soient directement fournies par lui ou qu'elles soient récupérées ou dérivées à son insu. Il peut s'agir par exemple de l'exclusion des résultats qu'il n'aime pas, du nombre de clicks qu'il a effectué sur le lien d'une page ou du nombre et de la nature des requêtes qu'il a émises.

## III.3 Construction du profil

La construction du profil traduit un processus qui permet d'instancier sa représentation. Ce processus peut être explicite ou implicite. La construction explicite est basée sur une collecte d'information directement fournies par l'utilisateur via l'interface du système. La construction implicite, largement motivée par les travaux actuels dans le domaine, repose sur un procédé d'inférence du contexte et préférences de l'utilisateur via son comportement lors de l'utilisation du système ou d'autres applications quotidiennes. Les informations exploitées pour la construction sont généralement issues : [**Tam ; Bough.05]**

- Directement de l'utilisateur (explicite) :
  - ✓ Jugement explicite sur la pertinence des termes, documents,
  - ✓ Définition de différents attributs : domaine d'intérêts, niveau, langue, etc.
  - ✓ Sélection des thèmes, sites favoris.
- Indirectement de l'utilisateur (implicite) :
  - ✓ Contenu des documents crées, consultés,
  - ✓ liens explorés,
  - ✓ durée de lecture des documents,
  - ✓ dernières pages visitées,
  - ✓ type d'application.

# IV. Les systèmes de recherche d'information personnalisée (SRIP)

**IV.1 Définition [ZEM, 03/04]** Un SRIP est un SRI qui intègre totalement l'utilisateur tout au long du processus de recherche. Il répond ainsi de manière personnelle aux besoins en informations de chaque utilisateur.

Un SRI Personnalisé inclut :

- Des modèles et algorithmes pour capturer et modéliser le but, les préférences et les centres d'intérêts de l'utilisateur ou un groupe d'utilisateur. Un modèle de profil est alors décrit et instancié.
- Une procédure de mise à jour du profil qui traduit son évolution dans le temps.
- Des mécanismes pour extraire les caractéristiques descriptives des documents à l'aide de métadonnées.

## IV.2 Architecture d'un SRI Personnalisé

Un système de recherche d'information personnalisée (SRIP) est un système qui intègre l'utilisateur, en tant que structure informationnelle, tout au long de la chaine d'accès à l'information. Le SRIP ne se limite pas seulement à modéliser les caractéristiques des utilisateurs en des profils. Il doit être capable de déduire à partir de ces profils, l'intention de l'utilisateur lorsqu'il effectue sa recherche, en d'autres termes son contexte de recherche, et de détecter l'évolution des profils de manière dynamique. Le système doit donc inclure :

- Des techniques et algorithmes pour capturer et modéliser le but, les préférences et les centres d'intérêt de l'utilisateur ou un groupe d'utilisateurs. Un modèle de profil utilisateur est alors décrit et instancié.
- Une procédure de mise à jour du profil qui traduit son évolution dans le temps.
- Des mécanismes et algorithmes pour intégrer le profil de l'utilisateur dans le processus d'accès et retourner l'information pertinente en fonction de ce profil.

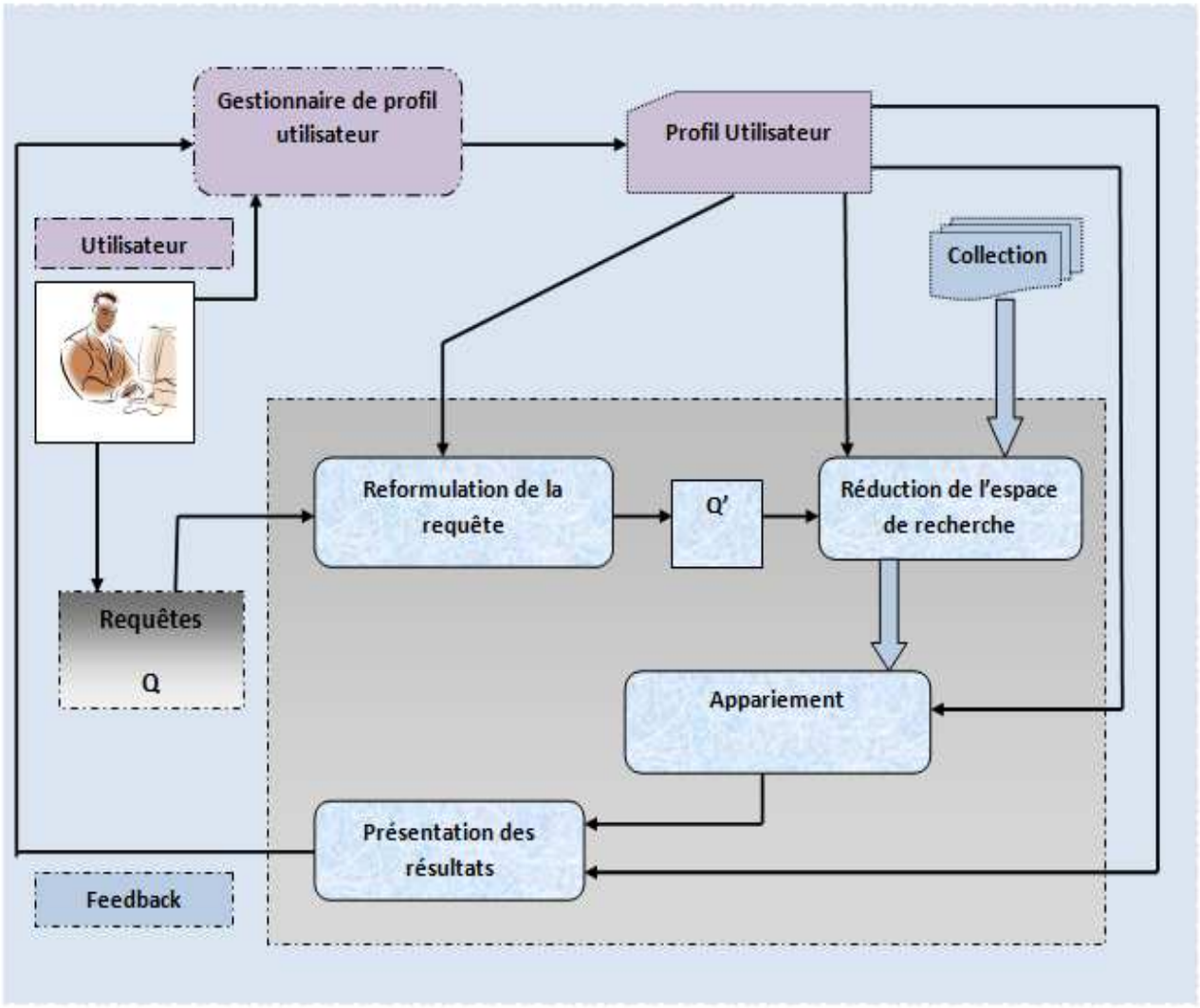



**Figure2.1: Processus globale d'accès personnalisé à l'information.**

Le processus d'accès personnalisé à l'information s'effectue en intégrant le profil utilisateur dans l'une des phases suivantes :

1. **La phase de reformulation de la requête :** afin de mieux cibler le contexte de la recherche de l'utilisateur,
2. **La phase de réduction de l'espace de recherche** : pour restreindre l'espace de recherche aux documents qui ciblent les besoins de l'utilisateur,
3. **La phase d'appariement** : pour calculer la pertinence des documents en fonction des caractéristiques spécifiques de l'utilisateur,

4. **La phase de présentation des résultats** : pour restituer les informations selon le contexte et les préférences de l'utilisateur.

### IV.2.1 Intégration du profil utilisateur dans la phase d'évaluation de requête

D'après ZEM, Ils ont posé que la requête initiale de l'utilisateur contient des valeurs de paramètres du profil utilisateur fournies directement par l'utilisateur. Le système peut déduire ainsi les données nécessaires pour instancier son profil. En choisissant une classe d'intérêts, l'utilisateur exprime son but de recherche et le système va extraire les documents correspondants et effectue la reformulation de la requête. La requête ainsi améliorée par les données du profil est nommée « ***requête profilée*** ».

### IV.2.2 Intégration du profil utilisateur dans la phase réduction de l'espace

La phase de réduction de l'espace de recherche consiste à cibler dans l'ensemble du corpus géré par le SRIP, le sous-ensemble de documents susceptibles d'être pertinents en fonction du profil utilisateur. Le système présélectionne les documents ayant un degré de similarité élevé pour fermer un corpus personnalisé de documents qui sera exploité lors de la phase d'évaluation de la requête.

### IV.2.3 Intégration du profil utilisateur dans la phase d'appariement document-requête

L'intégration du profil utilisateur dans cette phase consiste à modifier le calcul de la fonction d'appariement classique entre document-requête en fonction des centres d'intérêt. Cette fonction correspond à une mesure de probabilité de pertinence d'un document sélectionné dans l'espace de recherche réduit, la requête profilée et le profil utilisateur.

### IV.2.4 Intégration du profil utilisateur dans la phase de présentation des résultats

Dans cette phase l'interaction entre le profil et le module de représentation du résultat se fait à travers des informations fournies par la catégorie de « customisation », le SRIP va déployer des mécanismes d'adaptation.

## IV.3 Objectifs

L'objectif d'un SRIP est de délivrer une information pertinente en fonction des caractères spécifiques de l'utilisateur. La personnalisation de l'information peut être alors vue comme un processus de définition, construction, exploitation et évolutions des profils d'un utilisateurs ou un groupes d'utilisateurs en vue de répondre de façon adaptée à un besoin en informations exprimé par un type de requête.

Ces systèmes peuvent être abordés selon deux points de vue orthogonaux [Bou, 2004] les domaines d'application et les technologies de base.

✓ Les domaines d'application qui ont recours à la personnalisation de l'information sont nombreux : le commerce électronique, l'accès aux bibliothèques électroniques, les systèmes d'information mobiles, la configuration de logiciels…etc.

Selon les domaines, la personnalisation consistera en l'une ou plusieurs des taches suivantes :

- filtrer un flux d'informations entrant pour éliminer le bruit,
- guider la navigation dans un espace d'informations trop vaste,
- recommander un ensemble d'informations à l'utilisateur de manière plus intrusive (nouvelles offres par exemple),
- ajuster le résultat d'une requête selon une interface (ordre de présentation des résultats par exemple),
- adapter l'interaction à la situation de l'utilisateur (matérielle, géographique), etc.

✓ Les technologies qui permettent de supporter ces applications. On distingue entre autre Les systèmes de bases de données, les moteurs de recherche d'informations, les interfaces homme-machine.

Chacune de ces technologies a une offre différente en personnalisation :

- introduction de préférences dans les langages de requête,
- utilisation de reformulation des requêtes,
- pondération et ordonnancement des résultats des recherches, exploitation d'ontologie, etc.

## IV.4. L'évaluation du SRIP

Les méthodes d'évaluation largement adoptées en recherche d'information, sont empiriques (évaluation par observation expérimentale). En effet, les résultats obtenus sont issus de la comparaison de mesures et de métriques en termes de rappel et précision, sur les réponses fournies par le système relativement à celle issues des réponses attendues. Mais ces méthodes sont contestées en raison de la non prise en considération du contexte dans lequel s'effectue la recherche, et de la perception (estimation) de pertinence des utilisateurs dans ce même contexte.

## Conclusion

Ce chapitre introduit la thématique de recherche nommée « personnalisation de l'information » à travers les éléments clés liés à l'entité utilisateur, et qui a pour but de fournir une information pertinente, qui correspond exactement aux besoins de l'utilisateur.

Nous avons abordé la notion de personnalisation de l'information dans un contexte globale incluant la notion du profil le SRIP et son objectif, puis à travers l'architecture du processus globale d'accès personnalisé à l'information, nous avons définit les principales phases d'un tel système.

De plus, nous avons abordé le concept de modélisation de l'utilisateur incluant les approches de représentation du profil, sa construction.

La finalité de tout SRIP est de satisfaire les besoins des utilisateurs, pour qui la préoccupation majeure est la restitution des informations pertinentes relatives à leurs contextes d'utilisation spécifiques d'une manière rapide et efficace.

Répondre aux besoins en information des utilisateurs d'une manière personnelle, ne peut se faire sans inclure l'utilisateur dans le processus de RI. Inclure l'utilisateur dans le processus de la recherche d'information implique la représentation de ce dernier dans un modèle ou par une structure qui permet son exploitation par le SRI. Le prochain chapitre traite les réseaux Bayésien en recherche d'information.

CHAPITRE

III

# Les réseaux Bayésiens et la recherche d'information

# Introduction

L'approche Bayésienne est connue comme une bonne solution aux problèmes contenant de l'incertitude. Le concept de chance et d'incertitude est présent dans la vie quotidienne. Comme la RI est aussi un processus incertain, l'application de l'approche Bayésienne dans la RI a été étudié. Une des premières études est celle de Turtle et Croft, **[Turtl, 91]**, **[HTBC, 91]**. Elle montre qu'un modèle de RI basé sur un réseau Bayésien est plus général et peut englober d'autres modèles comme le modèle probabiliste, booléen ainsi que la pondération tf-idf du modèle vectoriel. Ainsi, Ricardo **[Ricard, 99]** indique que le Réseau Bayésien fournit un formalisme clair pour combiner les différentes sources d'évidences (requêtes passées, cycle de rétroaction, formulation de requêtes) pour le calcul de la correspondance entre la requête et les documents.

Les Réseaux Bayésiens (RB) sont la combinaison des approches probabilistes et de la théorie de graphes. Autrement dit, ce sont des modèles qui permettent de représenter des situations de raisonnement probabiliste à partir de connaissances incertaines. Ils sont une représentation efficace pour les calculs d'une distribution de probabilités.

# I. Historique

Le premier article publié sur les réseaux Bayésien été en 1764. C'est un essaie écrit par Mr Bayes qui est la base de toute sa théorie. Dans cet essai Bayes ne cherche pas à obtenir une probabilité exacte et précise sur la réalisation d'un événement, mais sur la confiance que l'on a dans la réalisation de cet événement par l'expression que l'on a pu acquérir.

L'objet centrale de cet essai est la fameuse règle de Bayes, qui dit que, si l'on a deux événements A et B, alors :

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

# II. Définition d'un réseau Bayésien

Un réseau Bayésien est un modèle probabiliste graphique permettant d'acquérir de capitaliser et d'exploiter des connaissances. Il allié la rigueur d'un formalisme mathématique puissant et stable a l'efficacité d'une représentation **"distribuée"** de connaissances et à la lisibilité des modèles à base de règle.

Un réseau Bayésien est un graphe orienté acyclique avec un ensemble N de nœuds et un ensemble A d'arcs orientés.

Un nœud contient le nom d'une variable aléatoire, une table de probabilités indiquant comment la probabilité de cette variable dépend des combinaisons de valeurs parentales.

Pour construire un Réseau Bayésien:

1. Choisir un ensemble de variables pertinentes ordonnées X1, X2, ..., Xm.
2. Pour i=1 à m
3. Ajouter Xi au graphe
4. Parents(Xi) = sous-ensemble minimal de {X1, ..., Xi-1} tel qu'il y ait indépendance conditionnelle de Xi et de tous les autres éléments de {X1, ..., Xi-1} étant donné Parents(Xi)
5. Définir la table de probabilités P (Xi=k|valeurs affectées à Parents(Xi).

## III. Construction des réseaux Bayésiens

La construction d'un réseau Bayésien s'effectue en trois étapes essentielles, qui sont présentées sur la figure 3.1 ci-dessous.

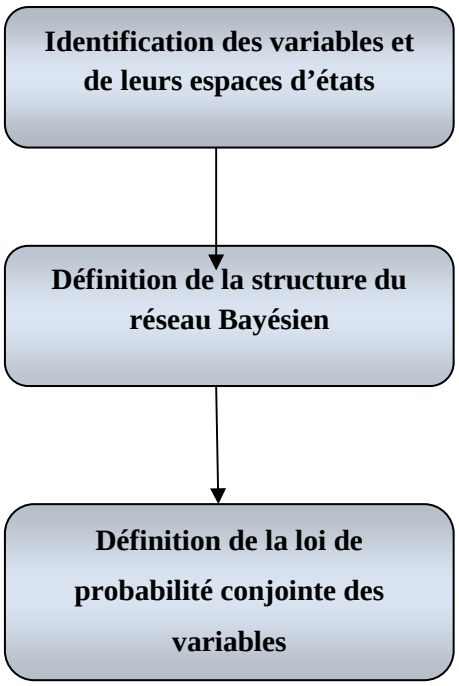



**Figure3.1 :** Étapes de construction d'un réseau Bayésien.

## III.1. Identification des variables et de leurs espaces d'états

La première étape de construction du réseau Bayésien est la seule pour laquelle l'intervention humaine est absolument indispensable. Il s'agit de déterminer l'ensemble des variables Xi, catégorielles ou numériques, qui caractérisent le système.

Lorsque les variables sont identifiées, il est ensuite nécessaire de préciser l'*espace d'états* de chaque variable Xi, c'est-à-dire l'ensemble de ses valeurs possibles.

## III.2. Définition de la structure du réseau Bayésien

La deuxième étape consiste à identifier les liens entre variables Xi, un réseau Bayésien ne doit pas comporter de circuit orienté ou boucle.

## III.3. Loi de probabilité conjointe des variables

La dernière étape de construction du réseau Bayésien consiste à renseigner les tables de probabilités associées aux différentes variables.
Dans un premier temps, les lois de probabilité des variables sont intégrées au modèle.

Concrètement, deux cas se présentent selon la position d'une variable Xi dans le réseau Bayésien :

• La variable Xi n'a pas de variable parente d'où l'obligation de préciser la loi de probabilité marginale de Xi.

• La variable Xi possède des variables parentes d'où la nécessité d'exprimer la dépendance de Xi en fonction des variables parentes, soit au moyen de probabilités conditionnelles, soit par une équation déterministe qu'on convertira ensuite en probabilités.

# IV. Principe d'un réseau Bayésien

Les Réseaux Bayésiens (RB) sont des modèles probabilistes qui s'appuient sur des graphes traduisant par des nœuds les variables du système et par des arcs l'existence de liaisons directes entre ces variables.

L'étude d'un modèle de Réseau Bayésien nécessite une base de données et cherche à fournir à cette base une modélisation sous forme de graphe caractérisant les dépendances conditionnelles des différentes variables. Elle se déroule en deux phases **[Hallouli, 2004]** :

- **Apprentissage ou constitution du réseau :** Il s'agit de trouver la structure et les probabilités associées du réseau, à partir des données de la base et de traitements principalement statistiques.

- **Inférence Bayésien :** A partir des résultats de la première phase, le réseau permet la propagation d'information à l'intérieur de la structure, permettant toute interrogation sur la base et peut fournir pour chaque état partiel ou complet de la base (instanciation partielle ou complète des variables de celle-ci) des probabilités d'occurrence de toutes les valeurs possibles de toutes les variables.

# V. L'utilisation des réseaux Bayésiens

L'utilisation pratique d'un réseau Bayésien peut être envisagée au même titre que celle d'autres modèles : réseau de neurones, système expert, arbre de décision, modèle d'analyse de données (régression linéaire), arbre de défaillances, modèle logique. Naturellement, le choix de la méthode fait intervenir différents critères, comme la facilité, le coût et le délai de mise en œuvre d'une solution. En dehors de toute considération théorique, les aspects suivants des réseaux bayésiens les rendent, dans de nombreux cas, préférables à d'autres modèles :

- **Acquisition des connaissances**. La possibilité de rassembler et de fusionner des connaissances de diverses natures dans un même modèle.

- **Représentation des connaissances**. La représentation graphique d'un réseau Bayésien est explicite, intuitive et compréhensible par un non spécialiste, ce qui facilite à la fois la validation du modèle, ses évolutions éventuelles et surtout son utilisation.

- **Utilisation des connaissances**. Un réseau Bayésien est polyvalent, on peut se servir du même modèle pour évaluer, prévoir, diagnostiquer, ou optimiser des décisions, ce qui contribue à rentabiliser l'effort de construction du réseau Bayésien.

# VI. Modèle de RI basés sur les réseaux Bayésiens

L'avantage apporté par l'utilisation de ces réseaux a été principalement de pouvoir combiner des informations provenant de différentes sources pour restituer les documents qui seraient les plus pertinents étant donnée une requête. Il y'a deux grands modèles de la représentation de recherche d'information.

## VI.1. Le modèle d'inférence

La naissance du modèle d'inférence **[SR, 75] [CCH, 92]** est le résultat de l'extension de deux idées :

1. La proposition d'utiliser des logiques non classiques pour déterminer le degré auquel un document implique ou correspond à une requête **[RT, 89]** ;
2. la notion d'inférence plausible et la possibilité de combiner plusieurs sources pour inférer la probabilité de pertinence d'un document étant donné une requête **[CWB, 87]**.

### VI.1.1. Architecture générale

Turtle [112], [111] a proposé un modèle de RI basé sur les réseaux d'inférence Bayésiens pour calculer la probabilité de pertinence d'un document étant donnée une requête. Ce modèle considère différentes représentations possibles des documents comme source d'information du contenu des documents et différentes représentations de la requête comme source d'information du besoin utilisateur.

Deux composantes de réseau ont été définies:

- La première définit le réseau document ainsi que ses termes d'indexation;
- la seconde représente le besoin utilisateur et déférentes représentations de la requête**.**

Comme l'illustre la figure3.2 suivante :

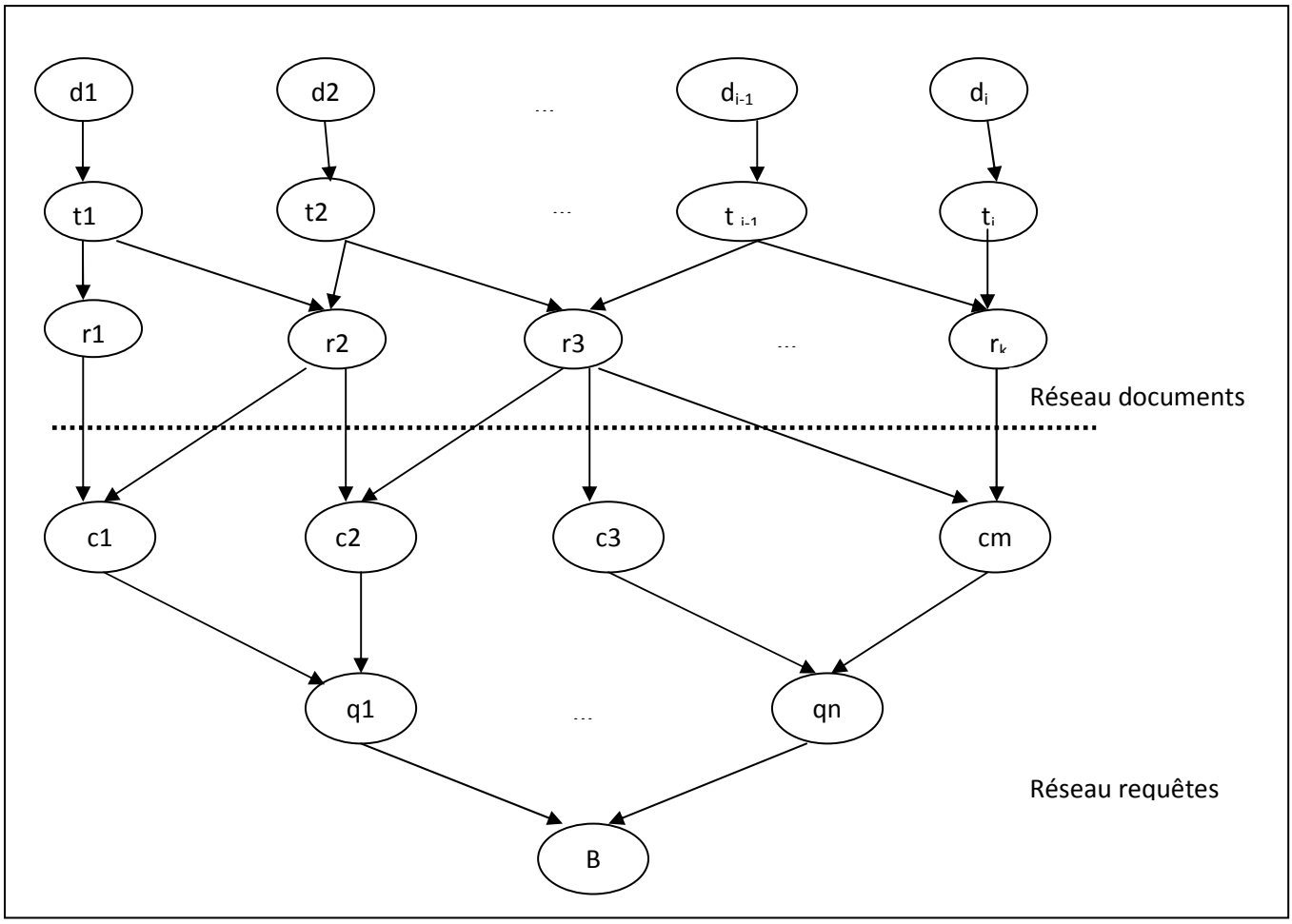



**Figure3.2 :** Exemple de réseau d'inférence pour la RI **[BRI, 05].**

Le réseau document représente les nœuds documents (dj) de la collection, les nœuds de représentation des textes (ti), les nœuds de représentation des concepts (rk). Un noeud document correspond à l'événement qu'un document donné de la collection est observé. Un nœud de représentation de texte correspond au texte indexant le document et correspond a l''´evénement qu'un texte d'un document est rencontré.

La dépendance entre un document et un texte est symbolisée par un arc entre les nœuds document et texte. Les nœuds de représentation de concepts correspondent à différentes techniques par exemple une indexation automatique et une indexation manuelle. Un même concept peut ainsi être généré par les deux techniques d'indexation, et l'arc reliant dans ce contexte le concept au document aura deux sens différents.

Les domaines de tous les nœuds sont binaires {vrai, faux} signifiant que le nœud est instancié ou non. Par exemple, un nœud représentant le texte aura pour instanciation vrai uniquement lorsque son nœud parent, document, est aussi instancié à vrai.

Le réseau requête est un graphe acyclique oriente représentant le besoin utilisateur (B) et des nœuds racines qui représentent les concepts de ce besoin (ci). Plusieurs expressions de la requête peuvent être utilisées et représentées dans ce réseau (qk). Les auteurs suggèrent de simplifier cette représentation en supprimant ces nœuds et de répercuter leur signification sur le nœud global

La valeur d'instanciation du nœud B est vraie lorsqu'elle désigne qu'un besoin utilisateur est rencontré dans un document. Le domaine des nœuds qk est vrai pour désigner que la représentation de la requête est satisfaite. L'apport le plus important de ce modèle a été de pouvoir combiner l'information provenant de représentations déférentes de documents ainsi que de combiner déférentes formulations de la requête.

Une simplification du réseau a été proposée **[SR, 75]** et exposée dans la figure3. Dans cette topologie, les nœuds documents sont des nœuds racine et il existe une relation entre les termes d'indexation et les documents ou la requête pour désigner que l'objet (document ou requête) contient le terme.

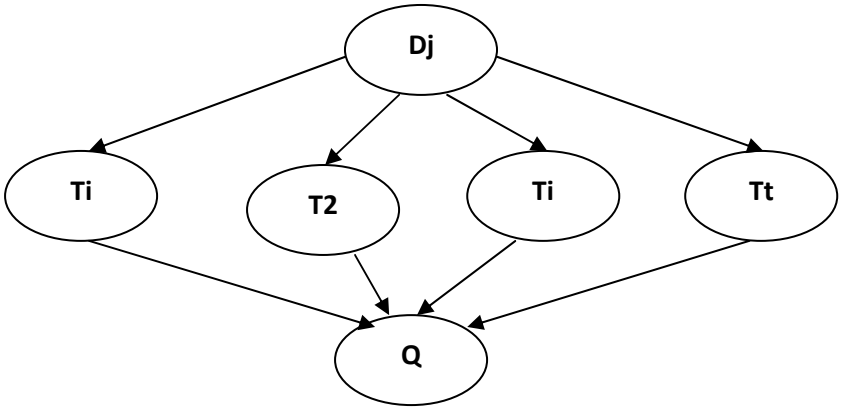



**Figure3.3 :** Architecture simplifiée.

Soit Dj un document dont le domaine est dom(Dj) = {dj, $\overline{dj}$}.

Un terme d'indexation est référencé par Ti dans la figure3.3, et le domaine des termes, noté dom(Ti), est dom(Ti) = {ti, $\overline{ti}$}.

Le domaine de la requête est dom(Q) = {q, $\overline{q}$}. Nous ne sommes intéressés que par le cas ou la requête introduit de l'information à travers le réseau, c'est-à-dire lorsque le nœud est instancié positivement, et nous noterons Q indifféremment lorsque cela ne prête pas à confusion.

## VI.1.2. Calcul de la pertinence

Le calcul de la pertinence revient dans ce modèle à instancier chaque document de la collection et à calculer la croyance de satisfaire la requête étant donné le document instancié. Le réseau pris dans sa globalité représente les dépendances qui existent entre une requête et les documents de la collection.

Un seul document est instancié positivement (Dj = dj) à la fois. La propagation de l'information est déclenchée par cette instanciation.

La propagation tente de calculer la probabilité que l'information a été rencontrée étant donné un document instancié à Dj = dj. Ce processus est réitère pour tous les documents de la collection. Une liste des documents ordonnés par ordre décroissant de pertinence est restituée.

La probabilité conditionnelle d'un nœud est fonction de toutes les configurations possibles de ses nœuds parents. Soit θ l'ensemble des configurations possibles des parents de Q, et $\theta_i^j$ une instance d'un nœud particulier $T_i$ telle que dans la configuration de $\theta^j$ et θ. Par exemple, soit la requête Q composée des deux termes $T_1$ et $T_2$. Q= {$T_1$, $T_2$} ; alors l'ensemble des configurations des parents de la requête, tel que leurs domaines est binaire θ = {{$t_1$, $t_2$}, {$t_1$, $t_2$}, {$t_1$, $t_2$}, {$t_1$, $t_2$}}. L'instance $\theta^1{}_1$ du terme $T_1$ dans la première configuration de θ, $\theta^1$ = {$t_1$, $t_2$}, est $\theta^1{}_1$= $t_1$.

La propagation dans le réseau dont la topologie est donnée dans la figure est :

$$P(Q\backslash dj) = \sum_{\forall \theta k \in \theta} (P(Q\backslash \theta k). \prod_{Ti \in Q \wedge Dj} P(\theta ki|dj).P(dj))$$

La quantification totale de la pertinence revient à quantifier chaque membre de la formule précédente. Des probabilités a priori sont affectées aux documents de la collections, égale à P (Dj= dj) = $\frac{1}{N}$ mais elles sont supprimées du calcul de la propagation globale parce que ce terme est considère comme un coefficient uniforme applique à tous les documents de la collection.

### VI.1.3. Agrégation de la requête

Turtle [112], [111] a proposé cinq formes canoniques pouvant répondre à tout type de recherche. La requête peut être agrégée par les opérateurs booléens (ET, OU, et NON). D'autre part, l'utilisation des réseaux permet d'agréger la requête par la somme probabiliste ou une de ses variations la somme pondérée. Pour évaluer les probabilités conditionnelles P (Q | θ) d'un nœud Q ayant n parents, $\{\theta_1,..., \theta_N\}$, et, P $(\theta_1= t_1)=p_1,...,$
P $(\theta_n= t_n)=p_n$ les agrégations suivantes sont définies :

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

$P_{Et}(Q|\theta) = p1 \times ... \times p_n.$

$P_{Non}(Q|\theta_1) = 1-P_1.$

$P_{Somme}(Q \mid \theta) = \frac{p1+\cdots+pn}{n}$

$P_{Somme\ Ponderee}(Q|\theta) = \frac{w1p1+\cdots+wnpn)wq}{w1+\cdots+wn}$

Lorsque la négation d'un terme est spécifiée dans la requête, la quantification de sa présence dans le document est obtenue par 1 − pi telle que décrite par la formule précédente. Ici, la négation du terme n'est pas son absence de la représentation du document lorsqu'il est spécifie dans la requête. Les termes de la requête absents des représentations des documents ne sont pas considères dans le calcul de la pertinence d'un document en réponse à une requête. La somme probabiliste tient compte du nombre de parents instanciés positivement dans la configuration des parents (|θj = tj |) et la somme pondérée mesure la configuration positive en fonction du poids de chaque parent instancié positivement, ainsi que du poids de la requête $w_q$. Le poids utilisé peut être le facteur de discrimination idf ou une de ses variantes

ou un poids attribué par l'utilisateur. Ces deux dernières techniques d'agrégation permettent un gain de temps lors des calculs de la pertinence puisque uniquement les termes présents dans une configuration (documents) sont considères.

## VI.2. Le modèle de croyance

La seconde principale utilisation des réseaux concerne le modèle basé sur les réseaux dits « de croyance » **[RNM, 96] [CCH, 92]**. Ce modèle est basé sur la définition préalable d'un espace d'échantillonnage qui permet de séparer clairement les portions de documents des portions de requêtes et donc de calculer d'une manière « efficace » **[CCH, 92]** les degrés de croyance. Il permet de généraliser tous les modèles classiques de RI (booléens, probabilistes et vectoriels) ainsi que de générer les résultats trouvés par le modèle différentiel **[CCH, 92]**. L'avantage de la généralisation est qu'elle permet de combiner les caractéristiques des modèles classiques les considérant comme sources d'information.
Les relations de dépendance définies dans ce modèle différent de celles de Turtle. Dans ce modèle le processus de recherche est déclenché par la réception de la requête.

### VI.2.1. Architecture générale

L'architecture générale du modèle de croyance est présentée dans la figure ci-après. L'univers de discours est donné par l'ensemble des termes d'indexation utilisés dans le système, noté U, et U = {T1, .., TT} ou T est le nombre de termes manipulés dans le système (pour représenter les documents ou la requête).θ est l'ensemble des configurations possibles sur U.

La définition des domaines d'un nœud terme donné est similaire à celle du modèle de Turtle. Un terme appartient ou non à un concept. Un concept peut être une requête ou un document. Les termes d'indexation pointent vers les documents et la requête qu'ils indexent.

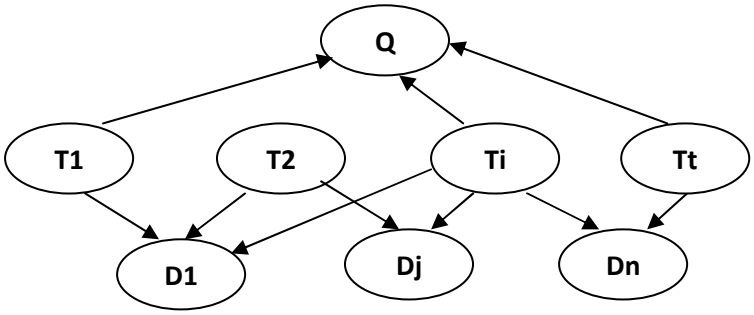



**Figure3.4 :** architecture générale de modèle de croyance **[BRI, 05].**

Un document Dj, est une variable aléatoire de domaine binaire, dom(Dj) = {dj, dj}. Un document Dj de la collection est instancié à Dj = dj pour indiquer que le document couvre complètement U.

Une variable aléatoire binaire est associée à une requête Q de domaine dom(Q) = {q, q}. Q = q signifie que la requête couvre complètement l'espace des termes. La couverture de l'espace U par un concept (document ou requête) est la conformité du concept avec chaque élément de l'espace U.

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

## VI.2.2. Calcul de la pertinence

Selon la topologie donnée par la figure4, l'instanciation de la requête permet de calculer la probabilité de pertinence d'un document étant donnée une requête, P (Dj | Q), donnée par la formule

$$P\ (Dj\ |Q) = \frac{P(Dj\ \wedge Q)}{P(Q)}$$

La probabilité P(Q) est calculée pour tous les documents de la collection, et est considérée comme une constante. Ainsi, une approximation possible de la probabilité d'un document étant donnée une requête peut être

$$P\ (Dj \mid Q) \propto P\ (Dj \wedge Q)$$

D'âpres la topologie du réseau, l'instanciation des termes d'indexation (ici, cas de d-séparation) rend les nœuds documents et requête indépendants. Ainsi :

$$P(Dj\ |Q) \propto \sum_{\theta} P\ (Dj\ |\theta) \times P(\ Q\ |\theta)\ \times P(\theta)$$

θ représente l'ensemble des configurations possibles des termes de l'univers U.

Ce modèle généralise les modèles classiques de la RI **[CCH, 92]**. Nous donnons ici le traitement opère sur le calcul de la propagation pour reproduire le modèle vectoriel. →d = {t1, .., tT} et →q = {t′1, .., t′T} désignent respectivement le vecteur document et le vecteur requête. Pour chaque document, une similarité par cosinus **[Sal, 88]** entre un document et une requête est calculée. La probabilité d'un document étant donnée une configuration d'un concept est approximée par le produit entre les poids des termes du document et de la requête. Ainsi :

$$sim\ (\overrightarrow{dj}, \vec{q}) = \frac{\overrightarrow{dj}\ .\ \vec{q}}{|\overrightarrow{dj}| \times |\vec{q}|}$$

Ou | · | désigne la cardinalité. Nous présentons dans ce qui suit, les mesures Proposées pour calculer les probabilités conditionnelles d'un document donné (Dj) et de la requête (Q) étant donnés leurs parents respectifs, not´es ParDj et ParQ.

### VI.2.2.1. Probabilités des documents P ($D_j$, Par $_{Dj}$)

L'univers de discours U définit l'ensemble des termes d'indexation du système. La probabilité P (dj) donne le degré auquel le document Dj couvre complètement l'espace des termes U. Cette couverture est calculée en contrastant chaque élément de U avec le document Dj, `a travers P (Dj | θ) et en additionnant les contributions de chacun. Cette somme est pondérée par la probabilité P(θ) avec laquelle θ apparat dans U. Cette probabilité répondrait à la croyance associée à la proposition Est-il vrai que dj couvre complètement U ?et est donnée par :

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

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

De plus, toujours dans le contexte vectoriel décrit dans la section ci-dessus, nous avons :

$$P(dj\ |\theta^Q) = \frac{\sum_{\theta_{i=1}^Q}^{T} w_{ij} \times w_{iq}}{\sqrt{\sum_{\theta_{i=1}^Q}^{T} w_{ij}^2 \times \sum_{\theta_{i=1}^Q}^{T} w_{iq}^2}}$$

$$P(dj\ |\theta^Q\ ) = 1 - P(dj\ |\theta^Q)$$

Ou $\theta_i^Q$ est la configuration des termes telle que donnée dans la requête Q, et $w_{ij}$ , $w_{iq}$ les poids du terme $t_i$ dans le document $d_j$ et la requête Q respectivement.

Les poids $w_{ij}$ sont des variantes de la pondération par tf * idf **[SalB, 88]**.

### VI.2.2.2. Probabilité de la requête P (Q | $Par_Q$)

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

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

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

Le calcul de cette équation nécessiterait 2T calculs, où T est le nombre de termes manipulés par le système, mais en réalité uniquement les termes indexant la requête sont considères. La valeur 1 est attribuée aux arcs reliant les termes d'indexation à la requête lorsque tous les termes présents dans la requête sont instanciés positivement dans une configuration donnée des parents. Ainsi :

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

$$P(Q|\theta) = 1 - P(Q|\theta)$$

Où $\theta_i^Q$, $\theta_i$ l'instanciation du terme $T_i$ dans la requête et dans θ respectivement.

## Conclusion

Dans ce chapitre, nous avons présenté les différents modèles de RI basés sur les Réseaux Bayésiens

Les modèles basés sur un RB combinent la théorie des probabilités avec la théorie des graphes. Ces modèle a non seulement la capacité de modéliser les variables et leurs dépendances, mais aussi la capacité de modéliser les évidences et leurs influences sur les variables via un processus d'inférence. Ce dernier permet de mettre à jour les probabilités des variables dans tout le réseau.

Pour cette tache, l'utilisation de la matrice d'association est une méthode simple appliquée dans des réseaux simples. Ce calcule peut être simplifié pour certain type de combinaison spéciale des parents.

Le chapitre suivant présente notre approche d'intégration de profil utilisateur dans a phase d'appariement d'un système de cherche d'information, en se basant sur les réseaux Bayésiens.

CHAPITRE

IV

# Conception et réalisation

# Introduction

Le travail présenté dans ce mémoire se situe dans le contexte de la recherche d'information personnalisée en se basant sur les réseaux bayésiens.

Dans les chapitres précédents, nous avons présentés les différents concepts de la recherche d'information personnalisée et des réseaux bayésiens.

Dans ce chapitre nous allons expliquer notre travail qui consiste en la personnalisation d'un SRI classique par la définition de centre d'intérêts utilisateur. Ainsi, que ces derniers sont exploités lors de l'évaluation de la requête.

Dans ce qui suit nous allons donner l'architecture de notre modèle, puis un exemple illustratif ainsi que les outils de développement utilisés.

## I. Modèle de Recherche d'Information Personnalisée Basé sur l'Inférence Bayésienne

L'objectif de notre travail est de personnaliser un SRI classique et cela par la définition de centre d'intérêt de l'utilisateur, comme composante dans la phase d'appariement Requête-Document du processus de recherche on se basant sur l'inférence Bayésienne. Le score final de chaque document à retourner pour une requête utilisateur se calcule par combinaison de score de similarité initial du document avec la requête et celui de la requête avec le profil utilisateur.

Le modèle est formalisé à l'aide d'un réseau Bayésien qui est un outil puissant pour la représentation des différents types d'informations. La figure suivante montre son architecture:

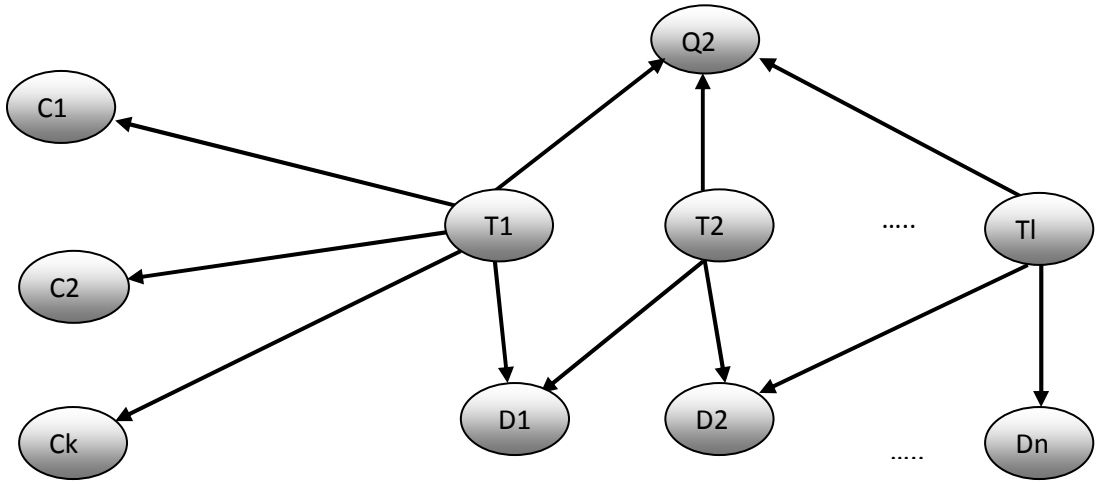



**Figure 4.1 :** modélisation probabiliste du modèle de recherche d'information personnalisée proposé.

Le modèle consiste en un graphe acyclique orienté G= (V, E), il se compose de nœuds V qui englobent : requête, termes, documents et centres d'intérêt.

**Une requête utilisateur** Q représente une variable aléatoire binaire dont le domaine est Dom(Q)= $\{q, \vec{q}\}$, ou q et $\vec{q}$ désignent que la requête Q est instanciée respectivement. N'est pas instanciée. On ne s'intéresse ici qu'à l'instanciation positive de Q.

**L'ensemble des termes** indexant les documents de la collection T= {t1, t2, t3, …, tk}, chaque nœuds terme ti représente une variable aléatoire binaire dont le domaine est Dom (ti)=$\{ti, \overrightarrow{ti}\}$, ou ti et $\overrightarrow{ti}$ désignent que le terme ti est présent respectivement ; n'est pas présent dans un document dj,

Une requête Q ou un centre d'intérêt ck.

On a **la collection de documents** D= {d1, d2, d3, …, dn}, chaque nœuds documents dj représente une variable aléatoire binaire dont le domaine est Dom (dj)=$\{dj, \overrightarrow{dj}\}$, ou dj et $\overrightarrow{dj}$ désignent que le document dj est instancié respectivement ; n'est pas instancié. Un seul document est instancié positivement à la fois.

**L'ensemble des centres d'intérêt** C= {c1, c2, c3, …, ck}, chaque nœuds centres d'intérêt ck représente une variable aléatoire binaire dont le domaine est Dom (ck)={$ck, \overrightarrow{ck}$}, ou ck et $\overrightarrow{ck}$, désignent que le centre d'intérêt ck est instancié respectivement, n'est pas instancié.

Un seul centre d'intérêt est instancié à la fois.

Les relations de dépendance entre nœuds sont traduites par des arcs orientés des nœuds termes vers les nœuds documents, centres d'intérêt ou requête pour désigner qu'un terme appartient respectivement, à un document, à un centre d'intérêt ou à une requête.

## I.1. Librairie de centres d'intérêt

Nous considérons dans ce travail, qu'un utilisateur est représenté par un centre d'intérêt, ce qui est dit profil utilisateur.

Pour la définition de la librairie de centres d'intérêt on prend *M* différentes requêtes utilisateurs de la collection AP88. Soumises au SRI terrier.

La figure suivante montre la création de la librairie de centre d'intérêt d'une manière implicite dans un SRI.

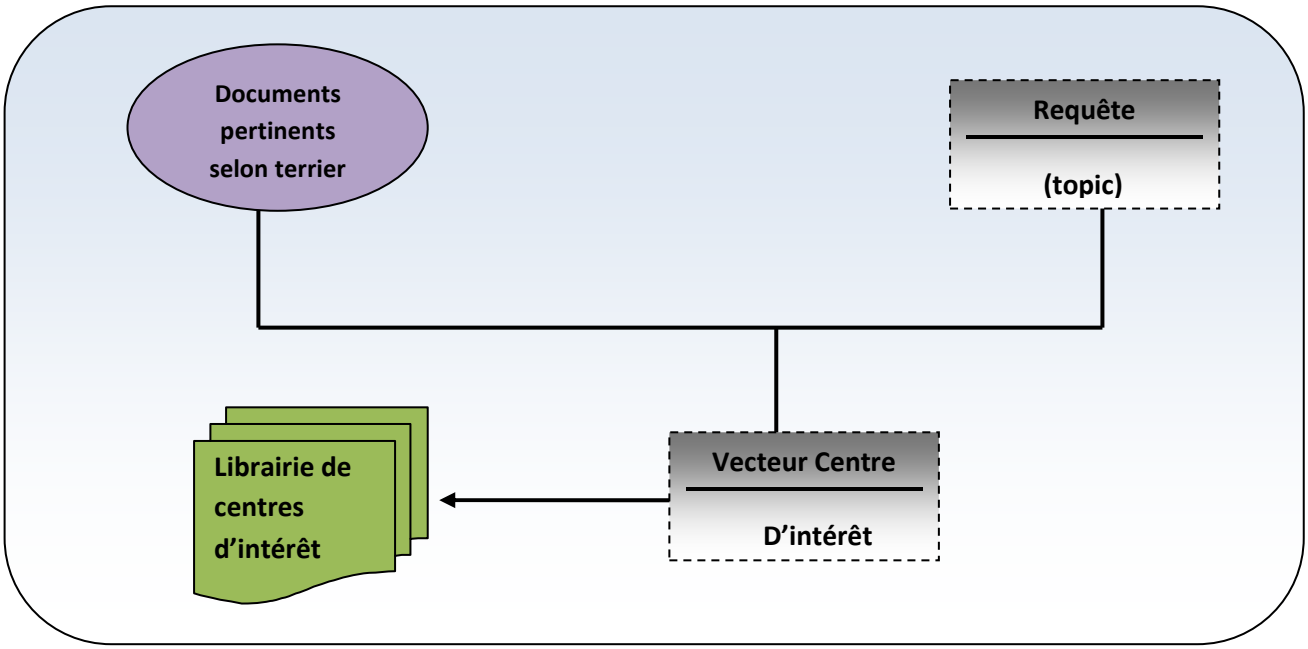



**Figure 4.2 :** Architecture générale de la construction d'un profil utilisateur.

Pour le choix des documents pertinents face à toute requête utilisateur, un centre d'intérêt *ck* est défini comme un vecteur de termes pondérés en appliquant la formule *BM*25 suivante :

$$\mathbf{w(ti, Ck) = log} \frac{\mathbf{(r + 0.5)/(R - r + 0.5)}}{\mathbf{(n - r + 0.5)/(N - n - (R - r) + 0.5)}}$$

Avec :

*N :* le nombre total de documents de la collection ;

*n* : le nombre de documents de la collection contenant le terme *ti* ;

*R* : le nombre de documents pertinents face à une requête utilisateur ;

*r* : le nombre de documents pertinents contenant le terme *ti*.

## I.2. Architecture de système Personnalisé

L'objectif de la modélisation de l'utilisateur est donc de pouvoir personnaliser les réponses du système de RI et cela en intégrant son profil dans au moins l'une des phases du processus de recherche, dans notre cas c'est dans la phase d'appariement.

La figure suivante montre notre modèle de RI en intégrant le profil utilisateur.

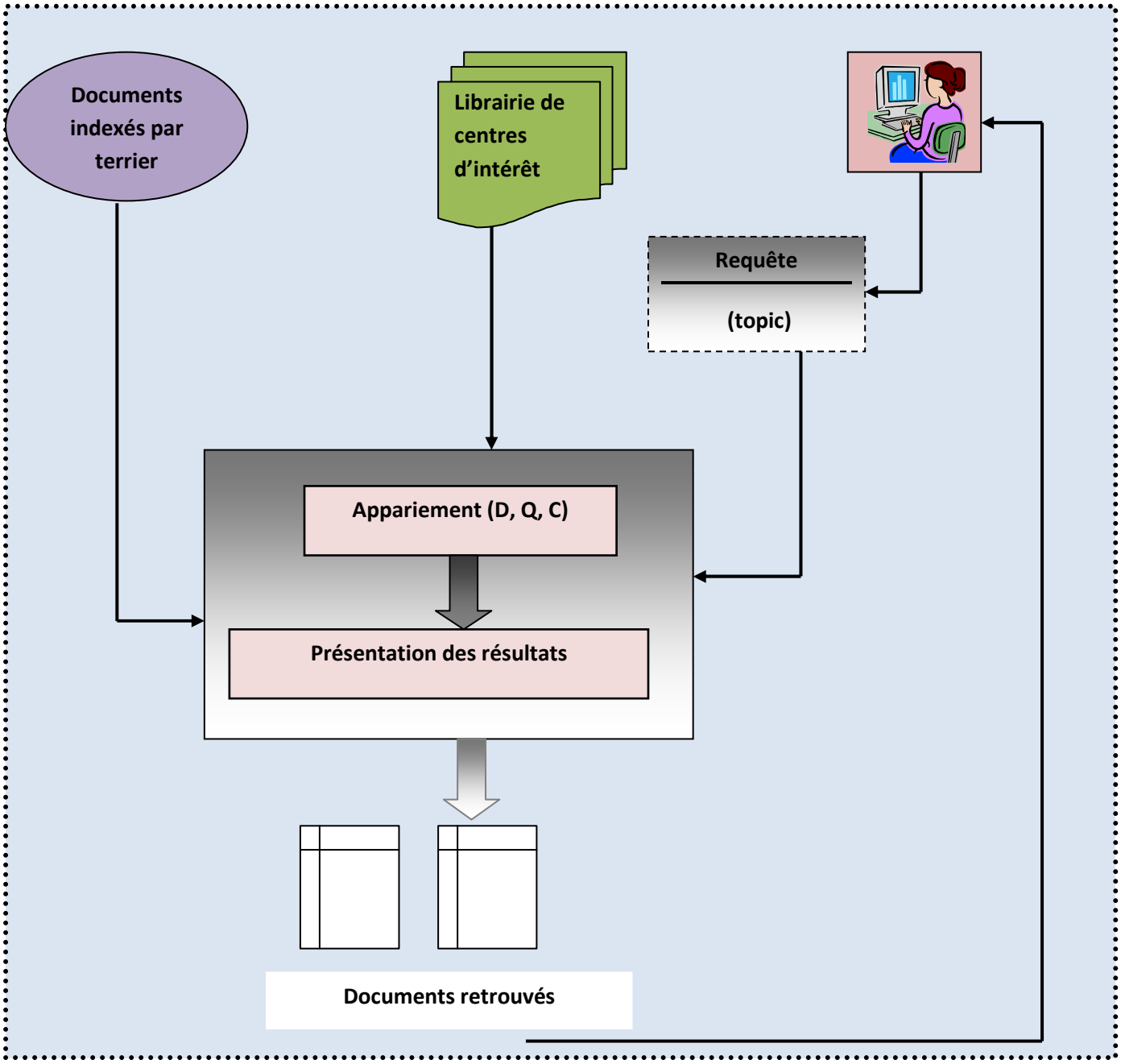



**Figure 4.3 :** Architecture de Système de recherche d'information personnalisé incluant le profil utilisateur dan la phase d'appariement.

- ***Appariement***

Le profil utilisateur est intégré dans la mesure d'appariement requête-document et cela en utilisant des diagrammes d'influence qui permettent de formaliser l'utilité des décisions associées à la pertinence des documents compte tenu de la requête et du profil de l'utilisateur.

Le calcul de la pertinence revient à instancier chaque document de la collection et un à un chacun des centres d'intérêts utilisateur. La propagation de l'information dans le réseau est déclenchée par ces instanciations. L'estimation de la pertinence de la requête revient à estimer la probabilité de satisfaire la requête étant donnés le document et le centre d'intérêt instanciés.

D'après la loi de Bayes la probabilité :

*P (Q/D, C)* peut être exprimée par :

$$\frac{\boldsymbol{P(Q \wedge D \wedge C)}}{\boldsymbol{P(C \wedge D)}}$$

Les concepts requête, document et centre d'intérêt sont indépendants (d-séparés par les nœuds termes). Donc :

$$\boldsymbol{P(q \wedge dj \wedge ck) = \sum_{u \in U} P(q/u) \times P(dj/u) \times P(ck/u) \times P(u)} \quad \boldsymbol{(1)}$$

L'espace de termes est réduit à la configuration *u* couverte par la requête *q*. Dans ce cas *P (q/u)=1 si q=u et 0 sinon* (*q* et *u* sont des sous-ensembles qui contiennent exactement les même termes).

On fait référence au sous-ensemble de documents et centres d'intérêt comme *uq,* ce qui réduit l'équation (1) en :

$$\boldsymbol{P(q \wedge dj \wedge ck) = \sum_{u \in U} P(dj/uq) \times P(ck/uq) \times P(uq)} \quad (1)$$

Puisque le modèle « réseau bayésien » généralise « le modèle vectoriel » l'équation (2) peut donc être :

$$\mathrm{P(q \wedge dj \wedge ck)\ correspond\ donc\ à :\ P(dj, q) \times P(ck, q)}$$

$$\boldsymbol{P(q \wedge dj \wedge ck) = P(dj/q) \times P(ck/q)}$$

Tels que :

$$\boldsymbol{P(\mathbf{dj},\mathbf{q}) = \frac{\sum_{i=1}^{l} w(ti,dj) \times w(ti,q)}{\sqrt{\sum_{i=1}^{l} w(ti,dj)^2} \times \sqrt{\sum_{i=1}^{l} w(ti,q)^2}}}$$

$$\boldsymbol{P(\mathbf{Ck},\mathbf{q}) = \frac{\sum_{i=1}^{l} w(ti,ck) \times w(ti,q)}{\sqrt{\sum_{i=1}^{l} w(ti,ck)^2} \times \sqrt{\sum_{i=1}^{l} w(ti,q)^2}}}$$

Ou $w(ti,q)$=1 si $ti$ appartient à la requête 0 sinon

$w(ti,dj)$= tf*idf

$w(ti,ck)$ est déja calculé , dans la phase de définition de centre d'intérêt.

L'appariement entre un document, une requête et un centre d'intérêt est effectué en appliquant l'algorithme suivant :

**Pour** chaque document de la collection des documents **faire**

**Pour** chaque centre de la librairie **faire**

$$\boldsymbol{P(\mathbf{dj},\mathbf{q}) = \frac{\sum_{i=1}^{l} w(ti,dj) \times w(ti,q)}{\sqrt{\sum_{i=1}^{l} w(ti,dj)^2} \times \sqrt{\sum_{i=1}^{l} w(ti,q)^2}}}$$

$$\boldsymbol{P(\mathbf{Ck},\mathbf{q}) = \frac{\sum_{i=1}^{l} w(ti,ck) \times w(ti,q)}{\sqrt{\sum_{i=1}^{l} w(ti,ck)^2} \times \sqrt{\sum_{i=1}^{l} w(ti,q)^2}}}$$

**Fait**

**Fait**

- ***Résultats retournés***

Des documents qui correspondront plus aux besoins de l'utilisateur.

Notre objectif à travers cette proposition est de récupérer les résultats de terrier et les utiliser pour la représentation du profil utilisateur par $\boldsymbol{w(ti, Ck)}$ et d'autre part, le calcule de similarité entre documents, une requête et un centre d'intérêt déjà défini, en utilisant la mesure de similarité :

$$\boldsymbol{P(Dj, q) * P(Ck, q)}$$

La figure suivante donne un aperçu de nos classes :

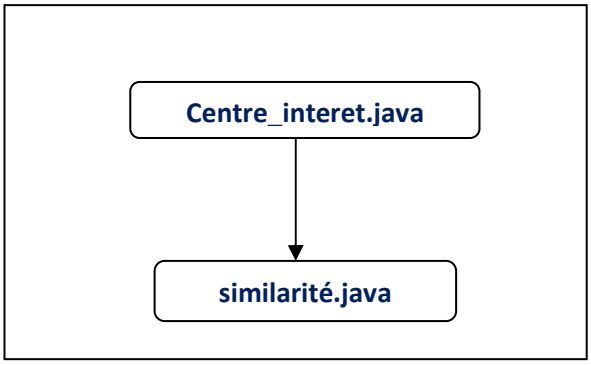



**Figure 4.5:** les classes programmées pour personnalisé les résultats retournés par la RI classique (terrier).

### ❖ Exemple d'Illustration

Supposons une collection de cinq documents *Collection= {d1, d2, d3, d4, d5}* dans laquelle la distribution des termes est la suivante :

*d1*= {$10t_1$, $2t_2$, $4t_3$, $5t_5$} *d2*= {$4t_1$, $9t_2$, 7t6} *d3*= {$5t_2$, $7t_3$, $12t_4$, $9t_6$}

*d4*= {$2t_1$, $11t_4$, $3t_5$, $7t_6$} *d5*= {$10t_2$, $15t_3$, $5t_4$, $8t_5$, $14t_6$}

Chaque document *dj* est représenté par un vecteur de termes pondérés selon une variante de pondération qui repose sur la loi de Zipf. La table 1 présente les poids *w(ti, dj)* des termes *ti* indexant les documents *dj*.

| | t1 | t2 | t3 | t4 | t5 | t6 |
|---|---|---|---|---|---|---|
| **d1** | 0.31739 | 0.02772 | 0.12695 | 0 | 0.15869 | 0 |
| **d2** | 0.14106 | 0.13864 | 0 | 0 | 0 | 0.10783 |
| **d3** | 0 | 0.05776 | 0.18514 | 0.31739 | 0 | 0.10398 |
| **d4** | 0.05770 | 0 | 0 | 0.31739 | 0.08656 | 0.08822 |
| **d5** | 0 | 0.09242 | 0.31739 | 0.10579 | 0.16927 | 0.12939 |

**Table 1.** Poids *w(ti, dj)* des termes *ti* indexant les documents *dj*.

Pour la définition des centres d'intérêts on suppose un ensemble de cinq requêtes *Q={q1, q2, q3, q4, q5}* contenant respectivement les termes suivants : *q1*={$t_1$}, *q2*={$t_2$}, *q3*={$t_3$}, *q4*={$t_4$}, *q5*={$t_5$}. Les probabilités de pertinence des documents face aux différentes requêtes utilisateurs sont mesurées selon la formule :

$P(D = dj/Q = q) = sim(\vec{dj}, \vec{q}) \times P(uq)$, avec $P(uq) = 1/2^6$ . La table 2 donne les pertinences de divers documents face à différentes requêtes.

| | d1 | d2 | d3 | d4 | d5 |
|---|---|---|---|---|---|
| **q1** | 0.83996 | 0.62629 | 0 | 0.16704 | 0 |
| **q2** | 0.07336 | 0.61554 | 0.08538 | 0 | 0.22694 |
| **q3** | 0.33597 | 0 | 0.27370 | 0 | 0.77936 |
| **q4** | 0 | 0 | 0.46921 | 0.91885 | 0.25977 |
| **q5** | 0.41997 | 0 | 0 | 0.25059 | 0.41565 |

**Table 2.** Valeurs de pertinences de divers documents face à différentes requêtes.

On estime la valeur du seuil minimal de pertinence à 0.40. Donc les documents pertinents pour chaque requête sont ceux qui ont une valeur de probabilité ≥ 0.40. Les différents centres d'intérêts sont alors définis dans la table 3.

| | ***t1*** | ***t2*** | ***t3*** | ***t4*** | ***t5*** | ***t6*** |
|---|---|---|---|---|---|---|
| ***c1*** | 0.9208 | 0.4772 | 0 | 0 | 0 | 0 |
| ***c2*** | 0.4771 | 0.1092 | 0 | 0 | 0 | 0.1097 |
| ***c3*** | 0 | 0.1092 | 0.4771 | 0.4771 | 0.4771 | 0.1097 |
| ***c4*** | 0 | 0 | 0 | 0.9208 | 0 | **0.4772** |
| ***c5*** | 0 | 0.4772 | 0.9208 | 0 | 0.9208 | 0 |

**Table 3.** Définitions des centres d'intérêts.

Supposons maintenant une nouvelle requête utilisateur $q = \{t_6\}$ pour laquelle on cherche le centre d'intérêt adéquat et les documents pertinents. La table 4 suivante présente la matrice *R* de dimension $5 \times 5$ *(nombre de documents* $\times$ *nombres de centres d'intérêts)* dont les valeurs représentent les probabilités de similitude entre chaque instanciation d'un document *dj* et d'un centre d'intérêt *ck* avec la requête *q :*

| | ***c1*** | ***c2*** | ***c3*** | ***c4*** | ***c5*** |
|---|---|---|---|---|---|
| ***d1*** | 0 | 0 | 0 | 0 | 0 |
| ***d2*** | 0 | 0.10470 | 0.06246 | **0.22028** | 0 |
| ***d3*** | 0 | 0.03361 | 0.02005 | **0.07072** | 0 |
| ***d4*** | 0 | 0.05585 | 0.03332 | **0.11751** | 0 |
| ***d5*** | 0 | 0.06948 | 0.04145 | **0.14618** | 0 |

**Table 4.** Probabilités de similitude entre chaque instanciation d'un document *dj* et d'un centre d'intérêt *ck* avec la requête *q.*

D'après ces résultats on peut noter que le centre d'intérêt correspondant à la requête $q = \{t_6\}$ est *c4,* combiné avec les documents présentés selon un ordre décroissant de pertinence : *d2, d5, d4, d3.* Effectivement d'après la distribution des termes dans les centres d'intérêts (cf. table 3), le terme $t_6$ figure dans le centre d'intérêt *c4* avec la plus grande valeur de pondération et il figure aussi dans les quatre documents (cf. table 1). La fréquence d'apparition du terme $t_6$ est plus élevée dans le document *d5* mais le document le plus pertinent est le document *d2* (cela est du à la prise en compte de la distribution des termes dans toute la collection de documents).

# II. Outils de développements

## II.1. La collection de test AP88

Pour évaluer les différentes formules proposées [référence], nous nous appuyons sur une collection de test TREC AP88 (Associated Press 1988) qui est de taille moyenne et qui contient des documents plats, et un ensemble de requêtes qui se trouve dans le fichier Topics251-300.

Le tableau ci-dessous montre quelques statistiques sur la collection AP88

| Collection | Taille | Documents | Topics |
|---|---|---|---|
| AP88 | 237 Mo | 79 919 | 251-300 |

Le format d'un document dans la collection AP88 est le suivant :

<DOC>
<DOCNO> AP880212-0001 </DOCNO>
<FILEID>AP-NR-02-12-88 2344EST</FILEID>
<FIRST>u i AM-Vietnam-Amnesty 02-12 0398</FIRST>
<SECOND>AM-Vietnam-Amnesty,0411</SECOND>
<HEAD>Reports Former Saigon Officials Released from Re-education Camp</HEAD>
<DATELINE>BANGKOK, Thailand (AP) </DATELINE>
<TEXT>
More than 150 former officers of the overthrown South Vietnamese government have been released from a re-education camp after 13 years of detention, the official Vietnam News Agency reported Saturday...
</TEXT>
</DOC>

## II.2. Requête (topics251-300)

L'ensemble des requêtes se trouvent dans le fichier Topics251-300 qui est un document plat.

Le format d'une requête est le suivant:

<top>

<num> Number: 251

<title> Exportation of Industry

<desc> Description:

<narr> Narrative:

</top>

## II.3. Terrier

### II.3.1. Architecture de terrier

La figure suivante montre l'architecture générale de terrier.

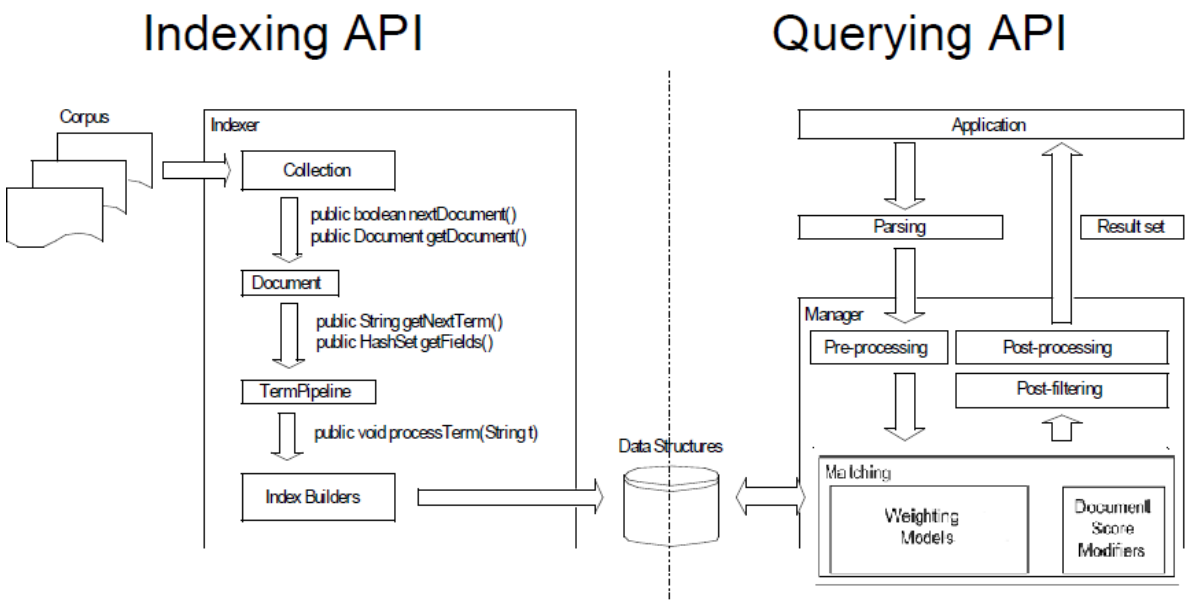



**Figure4.6 :** Vue de l'architecture Terrier.

Indexing er Quering APIs (Application Programming Interface) est facile à étendre, adapte de nouvelle applications, dispose d'une architecture modulaire, facile pour travailler avec, elle offre plusieurs options de configuration.

#### II.3.1.1. API d'indexation

La figure suivante donne une vue d'ensemble d'interaction des composants principaux impliqués dans le processus d'indexation.

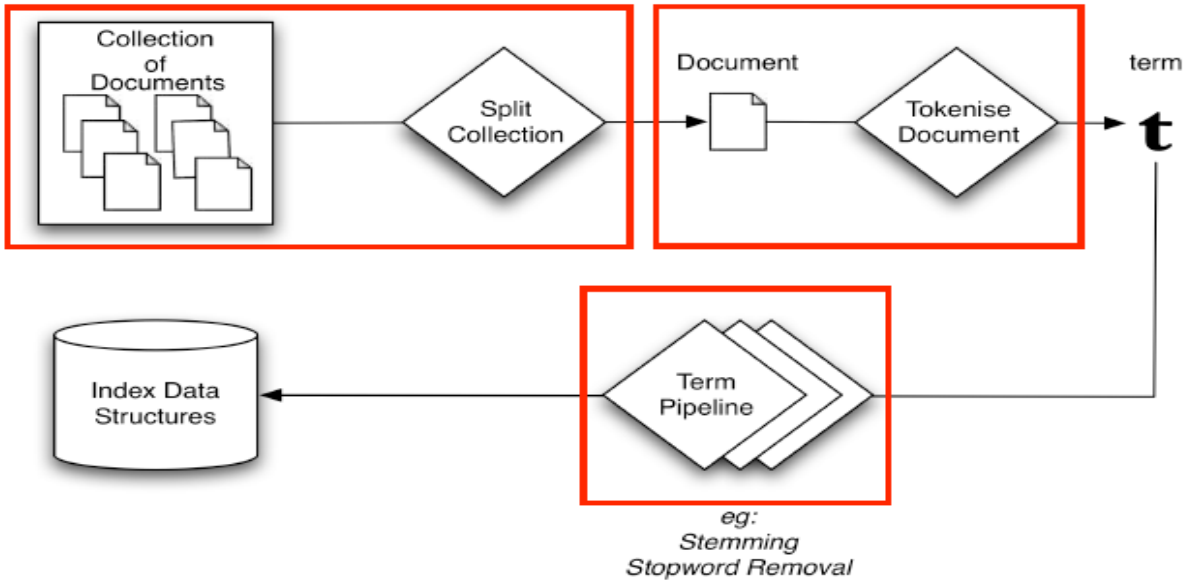



**Figure 4.7:** Le processus d'indexation dans terrier.

- **Collection :** Interface dans le pacage…\src\uk\gla\terrier\indexing, permet de splitter une collection (ou corpus) en documents.
- **Document :** Interface dans le package …\src\uk\ac\gla\terrier\indexing, permet de parcourir les documents et extraire les termes (en utilisant Tokeniser).
- **Terme Pipeline (Interface) :**effectue les traitements des termes extraits :

  - Eliminer les mots vide (Stopwords)

  - Lemmatisation des termes selon la langue.

### II.3.1.2. Les structures de l'indexe

Après L'indexation, les termes sont stockés en structures de données suivantes :

- Lexicon : contient les informations sur chaque terme de la collection (Terme, Id terme, nombre docs qui contiennent le terme, la fréquence de terme dans la collection, Offset dans le fichier inverse).
- Inverted index : Fihier inverse (Id Terme, Id document, Fréquence terme dans le document, #Filds).

- Direct index : index (Id Terme, Id document, Fréquence terme da ns le document, #filds).
- Document Index (Id Terme, fréquence Terme, #filds).

C. API de recherche

La figure suivante donne une vue d'ensemble d'interaction des composants de Terrier dans la phase de recherche.

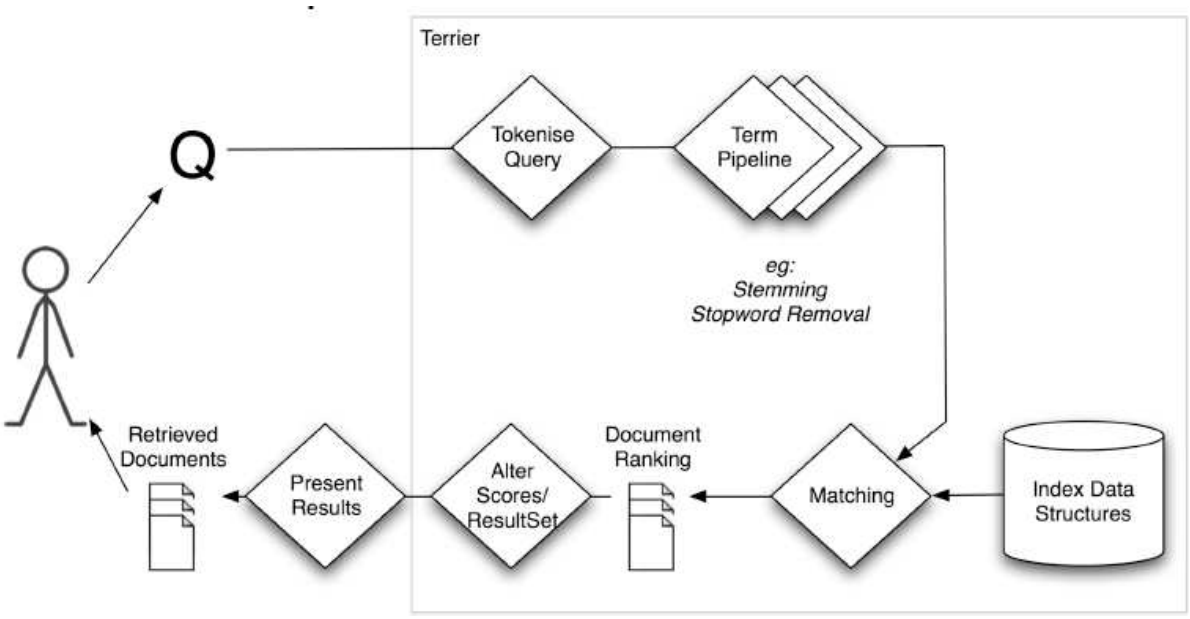



**Figure 4.8 :** Le processus de recherche dans Terrier.

L'API de recherche contient les classes suivantes :

- **Query** : classe abstraite qui représente la requete.
  Terrier supporte trois modèles de requetes :
  SingleTermeQuery : Désigne la requete qui contient un seul terme.
  MultiTermeQuery : Désigne la requete qui contient plusieurs termes.
  FieldQuery : Terme qualifié par un champ (Exemple : dans le titre du document).
- **Manager** : Chargé de la gestion de la recherche, il contient les étapes suivantes :
  Pre-processing : Appliquer l'élimination des mots vides et troncature.
  Matching : Déterminer les documents qui répondent à la requête en initialisant :
  - WeightingModels : Assigner un score pour chaque terme de la requete dans le document (Pondération), plusieurs modeles de pondération sont implémentés : TF_IDF, BM25,…

- DocumentScoreModifiers : Il permet de modifier le score d'un document en fonction du langage de la requete.

Post-filtring : Filtrer les documents pertinents selon un critère bien défini (exemple : type de document).

Poste-Processing : Reclasser les documents pertinents s'il y a une expantion de la recherche

- **Set-Results :** contient la liste des documents terournés classées selon leur degré de pertinence.

**Une description très détaillée est donnée dans l'annexeB.**

## II.4. Le langage JAVA

Notre choix s'est porte sur ce langage pour les raisons suivantes:

- Java est un langage multiplateforme qui permet aux concepteurs, selon la célèbre maxime « Write once, run everywhe », d'écrire un code capable de fonctionner dans tous les environnements. Pour cela, il suffit que l'environnement possède une JVM (Java Virtual Machine);

- Java est un langage oriente objet, simple, qui réduit le risque d'erreurs d'incohérence;

- Java est dote d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, boites de dialogue, contrôles, menus, graphisme) et la gestion des exceptions;

- Le JDK (Java Développement Kit), fournit gratuitement par Sun, regroupe l'ensemble des éléments permettant le développement, la mise au point et l'exécution des programmes Java.

- Java est caractérise aussi pour sa sécurité, un programme Java plante ne menace pas le système d'exploitation. Il ne peut pas y avoir d'accès direct à la mémoire, ainsi que sa simplicité de mise en œuvre.

Pour le développement de notre application, nous avons choisi Eclipse Galileo qu'est un environnement de développement intègre (IDE) dont le but est de fournir une plateforme modulaire pour permettre de réaliser des développements informatiques.il à était développer par I.B.M.

L'image suivante présente l'interface de travail sous Eclipse :

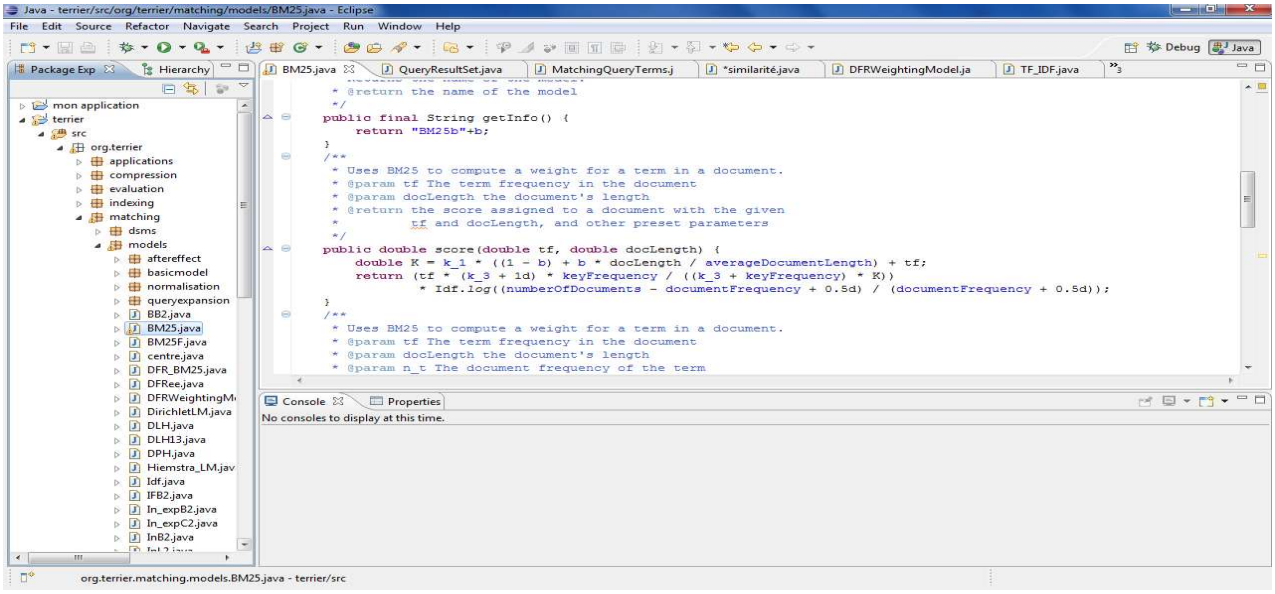



**Figure 4.9:** capture d'écran présentant l'interface de développement d'Eclipse.

## III. Test et évaluation

1. On définit la librairie de centre d'intérêts C= {c1, c2, c3, c4, c5, c6, c7, c8, c9, c10}, pour la définition des 10 centres on' a pris 10 requêtes Q= {q1, q2, q3, q4, q5, q6, q7, q8, q9, q10}.

| | Numéro de la requête | Titre de la requête |
|---|---|---|
| Q1 | 251 | Exportation of Industry |
| Q2 | 252 | Combating Alien Smuggling |
| Q3 | 253 | Cryonic suspension services |
| Q4 | 254 | Non-invasive procedures for persons with heart ailments |
| Q5 | 255 | Environmental Protection |
| Q6 | 256 | Negative Reactions to Reduced Requirements |
| Q7 | 257 | Cigarette Consumption |
| Q8 | 258 | Computer Security |
| Q9 | 259 | New Kennedy Assassination Theories |
| Q10 | 260 | Evidence of human life |

2. Le poids de chaque terme ti dans chaque centre ck est calculé on appliquant la formule de BM25 suivante :

$$\mathbf{w(ti, Ck)} = \log \frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - (R - r) + 0.5)}$$

Le tableau ci-dessus présente les valeurs des poids des termes ti dans un centre ck :

| termes | W (ti, ck) |
|---|---|
| american | 0.32872486 |
| build | 0.92458168 |
| car | 1 |
| close | 0.68330876 |
| compani | 0.97510711 |
| contract | 1 |
| end | 0.36440821 |
| export | 1 |
| gener | 0.67676067 |
| includ | 0.24146062 |
| increas | 0.76110123 |
| industri | 0.95612314 |
| last | 0.38287485 |
| mexico | 1 |
| million | 0.35109856 |
| move | 0.52513625 |
| onli | 0.35820794 |
| part | 0.50693292 |
| peopl | 0.01935619 |
| presid | 0.291869 |
| sai | 0.02935923 |
| state | 0.23819253 |
| time | 0.2035115 |
| ……. | …… |

**Tableau5 :** les valeurs des poids ***W (ti, ck).***

***3.*** On prend deux requêtes Q1 et Q2 tel que :

| | |
|---|---|
| Q1 | Exportation of American Industry. |
| Q2 | Environmental agricultur Protection. |
| Q3 | Threat posed by Fissionable Material |

Cherchant le centre d'intérêt adéquat et les documents pertinents pour les deux requêtes.

On Prend les documents pertinents pour Q1 retourné par la RI classique, on les indexe avec terrier et on récupère les poids des termes dans chaque document.

Le tableau suivant présente les valeurs des poids des termes ti existant dans la requête Q1 indexant dj.

| | t1 | t2 | t3 |
|---|---|---|---|
| **d1** | 0 | 0 | 0 |
| **d2** | 0 | 0 | 0.25803421 |
| **d3** | 0 | 0 | 0 |
| **d4** | 1 | 0 | 0 |
| **d5** | 0.44569546 | 0.44569546 | 0.89139092 |
| **d6** | 0 | 0.23345953 | 0 |
| **d7** | 0 | 0.23345953 | 0 |
| **d8** | 0 | 0.61283126 | 0 |
| **d9** | 1 | 0.18856346 | 0.18856346 |
| **d10** | 1 | 1 | 0 |
| **d11** | 0 | 0 | 0.98053001 |
| **d12** | 1 | 0 | 0 |
| **d13** | 0.89139092 | 0 | 0 |
| **d14** | 0.91924688 | 0.30641563 | 1 |
| **d15** | 1 | 0 | 1 |
| **d16** | 0.75425385 | 0 | 0.37712693 |
| **d17** | 0 | 0.37712693 | 0.75425385 |
| **d18** | 0 | 0.23345953 | 0 |
| **d19** | 0 | 0.23345953 | 0 |
| **d20** | 2.10113573 | 0 | 0 |
| **d21** | 0.44569546 | 0.44569546 | 0.89139092 |
| **d22** | 0 | 0 | 0.25803421 |

**Tableau6 :** les valeurs des poids ***W (ti, dj).***

4. Les probabilités de pertinence des documents face à la requête Q1 sont mesurées selon la formule suivante :

$$P(\mathrm{dj}, \mathrm{q}) = \frac{\sum_{i=1}^{l} w(ti, dj) \times w(ti, q)}{\sqrt{\sum_{i=1}^{l} w(ti, dj)^2} \times \sqrt{\sum_{i=1}^{l} w(ti, q)^2}}$$

Le tableau ci-après montre les valeurs de pertinence des différents documents dj face à la requête Q1.

| | d1 | d2 | d3 | d4 | d5 | d6 | d7 |
|---|---|---|---|---|---|---|---|
| **Q1** | 0 | 0.02384334 | 0 | 0 | 0.08237361 | 0.02280397 | 0.02441204 |

| | d8 | d9 | d10 | d11 | d12 | d13 | d14 |
|---|---|---|---|---|---|---|---|
| **Q1** | 0.04203314 | 0.08879378 | 0.08382117 | 0.05439283 | 0.07779161 | 0.0919601 | 0.13430292 |

| | d15 | d16 | d17 | d18 | d19 | d20 | d21 | d22 |
|---|---|---|---|---|---|---|---|---|
| **Q1** | 0.07899775 | 0.04256283 | 0.05865464 | 0.02441204 | 0.02280397 | 0.08237361 | 0.18973666 | 0.02384334 |

**Tableau7 :** les valeurs de probabilités ***P (dj, q1).***

***b.*** Les probabilités de pertinence d'un centre ck face à la requête Q1 sont présentées dans le tableau ci-dessus on appliquant la formule suivante :

$$P(\mathrm{ck},\mathrm{q}) = \frac{\sum_{i=1}^{l} w(ti,ck) \times w(ti,q)}{\sqrt{\sum_{i=1}^{l} w(ti,ck)^2} \times \sqrt{\sum_{i=1}^{l} w(ti,q)^2}}$$

| | c1 | c2 | c3 | c4 | c5 |
|---|---|---|---|---|---|
| **Q1** | 0.18448214 | 0.10130889 | 0 | 0.08499806 | 0.20783195 |

| c6 | c7 | c8 | c9 | c10 |
|---|---|---|---|---|
| 0.10458608 | 0.32421475 | 0.02627161 | 0.13617909 | 0 |

**Tableau8 :** les valeurs des probabilités ***(ck, q1).***

***b.*** les probabilités de similitude entre chaque instanciation d'un document *dj* et d'un centre d'intérêt *ck* avec la requête *Q* sont calculés par :

$$\boldsymbol{P\,(dj,q) \times P\,(ck,q)}$$

Et sont représentées dans le tableau suivant :

| | **c1** | **c2** | **c3** | **c4** | **c5** | **c6** | **c7** | **c8** | **c9** | **c10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **d1** | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 |
| **d2** | 0.00439867 | 0.00241554 | 0 | 0 | 0.00495541 | 0.00249368 | **0.00773036** | 0.0006264 | 0.00324696 | 0 |
| **d3** | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 |
| **d4** | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 |
| **d5** | 0.01519646 | 0.00834518 | 0 | 0.0070016 | 0.01711987 | 0.00861513 | **0.02670674** | 0.00216409 | 0.01121756 | 0 |
| **d6** | 0.00420692 | 0.00231024 | 0 | 0.00193829 | 0.00473939 | 0.00238498 | **0.00739338** | 0.0005991 | 0.00310542 | 0 |
| **d7** | 0.00450358 | 0.00247316 | 0 | 0.00207498 | 0.0050736 | 0.00255316 | **0.00791474** | 0.00064134 | 0.00332441 | 0 |
| **d8** | 0.00775436 | 0.00425833 | 0 | 0.00357274 | 0.00873583 | 0.00439608 | **0.01362776** | 0.00110428 | 0.00572403 | 0 |
| **d9** | 0.01638087 | 0.0089956 | 0 | 0.0075473 | 0.01845418 | 0.00238498 | **0.02878825** | 0.00233276 | 0.01209186 | 0 |
| **d10** | 0.01546351 | 0.00849183 | 0 | 0.00712464 | 0.01742072 | 0.00876653 | **0.02717606** | 0.00220212 | 0.01141469 | 0 |
| **d11** | 0.01003451 | 0.00551048 | 0 | 0.00462329 | 0.01130457 | 0.00568873 | **0.01763496** | 0.00142899 | 0.00740717 | 0 |
| **d12** | 0.01435116 | 0.00788098 | 0 | 0.00661214 | 0.01616758 | 0.00813592 | **0.02522119** | 0.00204371 | 0.01059359 | 0 |
| **d13** | 0.016965 | 0.00931638 | 0 | 0.00781643 | 0.01911225 | 0.00961775 | **0.02981482** | 0.00241594 | 0.01252304 | 0 |
| **d14** | 0.02477649 | 0.01360608 | 0 | 0.01141549 | 0.02791244 | 0.01404622 | **0.04354299** | 0.00352835 | 0.01828925 | 0 |
| **d15** | 0.01457367 | 0.00800317 | 0 | 0.00671466 | 0.01641826 | 0.00826207 | **0.02561224** | 0.00216409 | 0.01075784 | 0 |
| **d16** | 0.00785208 | 0.00431199 | 0 | 0.00361776 | 0.00884592 | 0.00445148 | **0.0137995** | 0.00111819 | 0.00579617 | 0 |
| **d17** | 0.01082073 | 0.00594224 | 0 | 0.00498553 | 0.01219031 | 0.00613446 | **0.0190167** | 0.00154095 | 0.00798754 | 0 |
| **d18** | 0.00450358 | 0.00247316 | 0 | 0.00207498 | 0.0050736 | 0.00255316 | **0.00791474** | 0.00064134 | 0.00332441 | 0 |
| **d19** | 0.00420692 | 0.00231024 | 0 | 0.00193829 | 0.00473939 | 0.00238498 | **0.00739338** | 0.0005991 | 0.00310542 | 0 |
| **d20** | 0.01519646 | 0.00834518 | 0 | 0.0070016 | 0.01711987 | 0.00861513 | **0.02670674** | 0.00216409 | 0.01121756 | 0 |
| **d21** | 0.03500302 | 0.01922201 | 0 | 0.01612725 | 0.03943334 | 0.01984381 | **0.06151542** | 0.00498469 | 0.02583817 | 0 |
| **d22** | 0.00439867 | 0.00241554 | 0 | 0.00202664 | 0.00495541 | 0.00249368 | **0.00773036** | 0.0006264 | 0.00324696 | 0 |

**Tableau9 :** les valeurs des probabilités ***(d, q1, c).***

D'après les résultats obtenus la meilleure combinaison correspondante à la requête Q1 est le couple (C7, d21) effectivement les deux contiennent les termes de la requête.

Le centre d'intérêt le plus adéquat a la requête Q1 est le C7 et l'ordonnancement des documents pertinents obtenus est le même avec l'ordonnancement obtenu avec la recherche classique et si nous fixons un seuil de pertinence on trouve que le nombre de documents retournés est élevé par rapport aux résultats obtenus dans la recherche classique.

Voici une figure illustrant les résultats d'évaluation des deux requêtes Q1 et Q2:

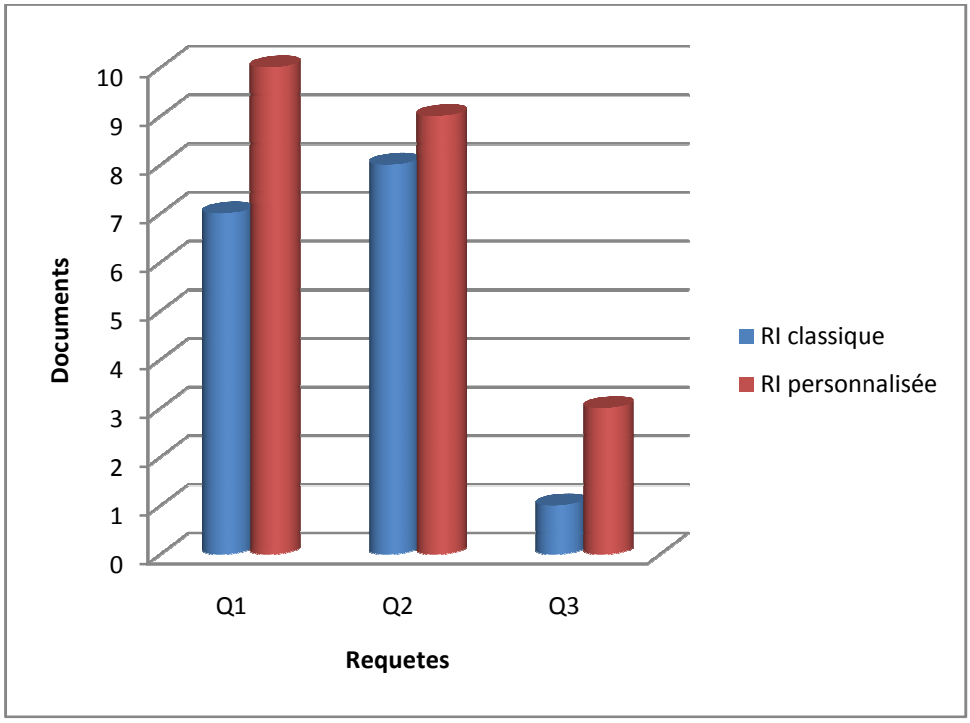



**Figure4.9:** Comparaison de nombre de documents retournés par la RI classique et la RI personnalisée.

D'après les résultats obtenus l'intégration de profil utilisateur dans la phase d'appariement à apporter une amélioration en terme de nombre de documents pertinents retournées.

# Conclusion

Ce chapitre a été consacré à la représentation de notre système en spécifiant son architecture générale, la description de modèle proposé ainsi que les outils des développements pour réaliser notre modèle.

A la suite de ce chapitre, nous avons abordé les différentes testes que nous avons réalisées. Selon les résultats obtenus, il résulte une amélioration en termes de nombre de documents pertinents retournés.

Pour pouvoir conclure d'une façon certaine sur l'efficacité de notre modèle, il faut la tester sur toutes les requêtes.

# *Conclusion générale*

# *Et perspectives*

# *Conclusion générale*

Dans ce mémoire de mastère, nous avons abordé le problème lié à l'intégration du profil utilisateur dans la phase d'appariement basant sur les réseaux bayésiens.

En premier lieu nous avons donné des généralités sur la recherche d'information classique ensuite on s'est focalisé sur la recherche d'information personnalisée.

Nous avons détaillé par la suite les concepts généraux relatifs aux réseaux bayésiens, puis nous avons donné l'algorithme permettant de créer la librairie de centres d'intérêt et puis les formules utilisés pour calculer la similarité.

Et comme l'objectif de notre projet est de faire une étude comparative entre un modèle de recherche classique et un modèle de recherche personnalisée nous avons fait une évaluation du modèle avec la collection de test TREC AP88.

Le modèle implémenté permet en premier lieu la création la librairie de centres d'intérêt en utilisant la formule BM25 et puis le calcul du score finale qui la combinaison du score initiale et le score personnalisé.

Cette approche a montré une certaine amélioration par rapport à l'augmentation du nombre de documents retournés par le SRI classique sans l'intégration des centres d'intérêt.

## Perspectives :

Suite à notre travail quelques perspectives sont envisagées :

- L'intégration des centres d'intérêt dans Terrier pour le personnaliser;
- La Validation du modèle sur la collection personnalisée;
- Ainsi que la prise en compte de l'évolution du profil utilisateur dans le modèle proposé.

# ANNEXES

# 1. Rappels de probabilités

## 1.1. Probabilité conditionnelle

- A et M deux événements
- Information a priori sur A : P(A)
- M s'est produit : P(M) ≠0
- S'il existe un lien entre A et M, cet événement va modifier notre connaissance sur A
- Information a posteriori : $$\boldsymbol{P(A|M) = \frac{P(A,M)}{P(M)}}$$

## 1.2. Indépendance

- A et B sont indépendants si :
  P (A, B) = P (A) × P (B)
  P (A|B) = P (A)
  P (B|A) = P (B)

## 1.3. Indépendance conditionnelle

- A et B sont indépendants conditionnellement à C si :

$$\boldsymbol{P\,(A|B,\,C)=P(A|C)}$$

## 1.4. Théorème des probabilités totales

Un événement A peut résulter de plusieurs causes Mi. Quelle est la probabilité de A connaissant :

- Les probabilités élémentaires P(Mi) (a priori)
- Les probabilités conditionnelles de A pour chaque Mi

$$\boldsymbol{P(A) = \sum_{i} P(A|Mi)P(Mi)}$$

# 2. La formule de bayes

Un événement A s'est produit. La probabilité que ce soit la cause Bn qui l'ait produit :

$$\boldsymbol{p\,(B_k/\,A)\; = \frac{P(A\,/B_k)\;p(B_k)}{P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + \cdots + P(A|B_n)P(B_n)}}$$

- P ($B_k$| A) : probabilité a posteriori.

La formule de Bayes dans un contexte C défini par le fait qu'on possède certaines connaissances sur l'état des variables.

**Exemple**

Il y a 4% d'absentéisme chez les employés travaillant de jour, 8% chez ceux qui travaillent le soir et 22% chez ceux qui travaillent de nuit. Sachant qu'il y a 80% des employés qui travaillent de jour, 10% qui travaillent de soir et 10% qui travaillent de nuit, déterminez la probabilité qu'un employé donné travaillent de jour sachant qu'il était absent du travail.

**Solution**

B1 = l'employé travaille de jour,
B2 = l'employé travaille de soir,
B3 = l'employé travaille de nuit,
A = l'employé est absent.

Nous savons que

P (A |B1) = 4%, P (A |B2) = 8%, P (A |B3) = 22%,
P (B1) = 80%, P (B2) = 10%, P (B3) = 10%.
Nous cherchons

$$\boldsymbol{P(B1|A)} = \frac{\boldsymbol{P(A\,|B1)\,p(B1)}}{\boldsymbol{P(A|B1)P(B1) + P(A|B2)P(B2) + P(A|B3)P(B3)}}$$

$$= \frac{0.04 \times 0.80}{0.04 \times 0.80 + 0.80 \times 0.10 + 0.22 \times 0.10}$$

$$= 0.51613.$$

## 3. Règle du produit en chaine pour le calcul des probabilités jointes

$$\boldsymbol{P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | Parent\ X_i)}$$

Où *P*(*Xi*) est l'ensemble des parents (causes) de *Xi* dans le graphe.

Un Réseau Bayésien est alors constitué de deux composantes :

- Un graphe causal orienté acyclique : il est la représentation qualitative de la Connaissance. S'il y a un arc du nœud *X* vers le nœud *Y*, c'est que la variable *X* a une influence directe sur la variable *Y* (*X* cause *Y*).

- Un ensemble de distributions locales de probabilités : il est la représentation quantitative de la connaissance (paramètres du réseau). A chaque nœud est associé une Table de Probabilités Conditionnelles (TPC) qui quantifie les effets des parents.

**Exemple**

Système de sécurité doté d'une alarme
- fonctionne correctement pour les cambriolages
- se déclenche parfois à tort pour des tremblements de terre mineurs

John et Mary se proposent d'appeler quand ils entendent l'alarme mais
- John confond parfois la sonnerie du téléphone avec l'alarme
- Mary n'entend pas toujours l'alarme

C = cambriolage
T = Tremblement de terre
A = alarme
J = Appel de John
M = Appel de Mar

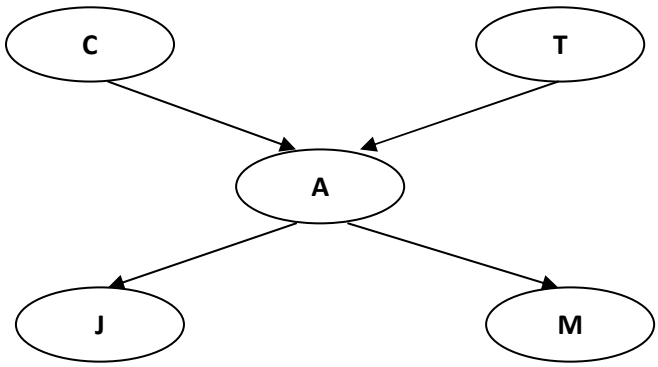



**Figure1 :** exemple1.

P (C) = 0,001 P (T) = 0,002 P (J | A) = 0.9 P (J | ¬A) = 0.05
P (M | A) = 0.7 P (M | ¬A) = 0.03
P (A |C, T) = 0.95 P (A |C, ¬T) = 0.94
P (A |¬C, T) = 0.29 P (A |¬C, ¬T) = 0.001.

## 4. Circulation de l'information et D-séparation

Pour étudier la circulation de l'information dans un réseau causale, nous allons utiliser l'exemple suivant :

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

La représentation graphique du modèle causal utilisé par M. Holmes est la suivante :

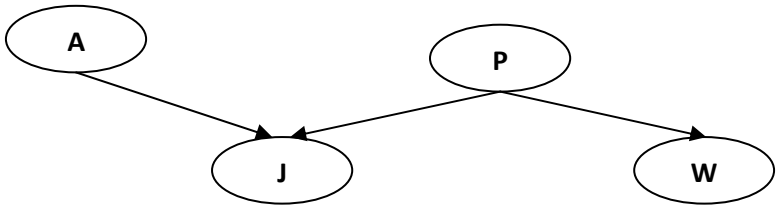



**Figure2** : exemple2.

- Le nœud A représente l'événement que M. Holmes a oublié de débrancher son arroseur automatique.
- Le nœud P représente l'événement qu'il a plu cette nuit.
- Le nœud J représente l'événement que l'herbe de son jardin est humide.
- Le nœud W représente l'événement que l'herbe du jardin de M. Watson est humide.

Comment l'information peut circuler dans ce graphe ? Évidemment l'information peut circuler depuis les causes vers les effets, d'ailleurs on remarque *M. Holmes a décidé d'aller au bureau sans vérifier son arroseur après avoir su que l'herbe du jardin de M. Watson est humide aussi*. C'est-à-dire la connaissance de W peut modifier la connaissance de P, ou autrement dit l'information peut circuler dans la direction inverse.

### 4.1. Les chemins de circulation d'information

Dans l'exemple précédent, nous avons vu qu'une information certaine se propage dans un graphe en modifiant les croyances que nous avons des autres faits.

Nous allons étudier quels chemins cette information peut prendre à l'intérieur d'un graphe.

Nous allons considérer les trois cas suivants, qui décrivent l'ensemble des situations possibles faisant intervenir trois événements.

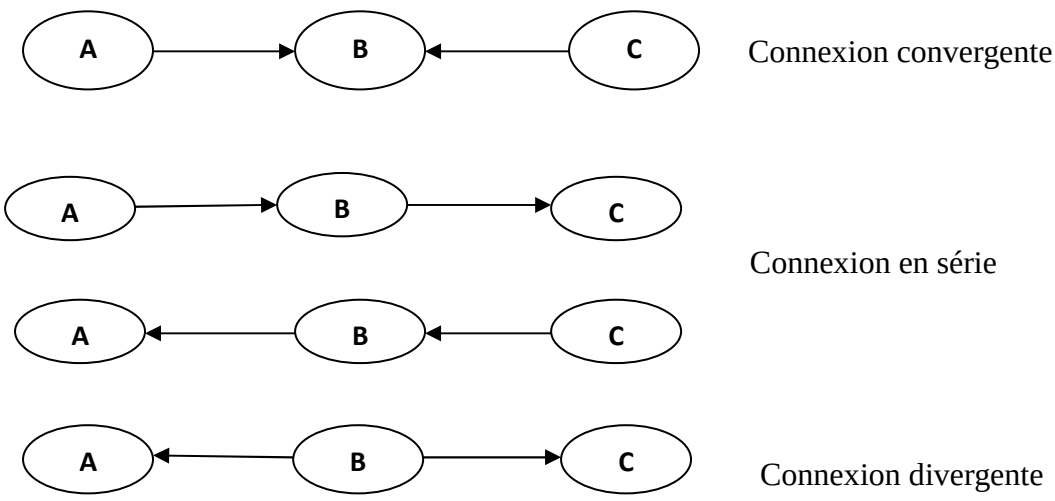



**Figure3** : circulation des informations.

## 4.2. La D-séparation

La d-séparation est une propriété graphique qui renseigne sur la circulation de l'information dans un réseau causal. Cette propriété, purement graphique, est très utile notamment pour l'inférence dans la mesure où elle permet de traiter une information localement sans perturber le reste du graphe.

Elle explicite les conditions dans lesquelles l'information peut circuler entre deux sous-ensembles de variables de manière complète, les indépendances entre ces variables, ainsi, le calcul d'une probabilité d'intérêt se trouve énormément simplifié.

### 4.2.1. Le principe

Déterminer si deux variables quelconques sont indépendantes conditionnellement à un ensemble de variables instanciées.

### 4.2.2. Définition

Deux variables A et B sont d-séparées si pour tous les chemins entre A et B, il existe une variable intermédiaire C différente de A et B telle que la connexion est en série ou divergente et C est instancié la connexion est convergente et ni C ni ses descendants ne sont instanciés. Si A et B ne sont pas d-séparés, ils sont d-connectés.

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

**Exemple**
**Ordre topologique : C, S, R, T (non unique)**

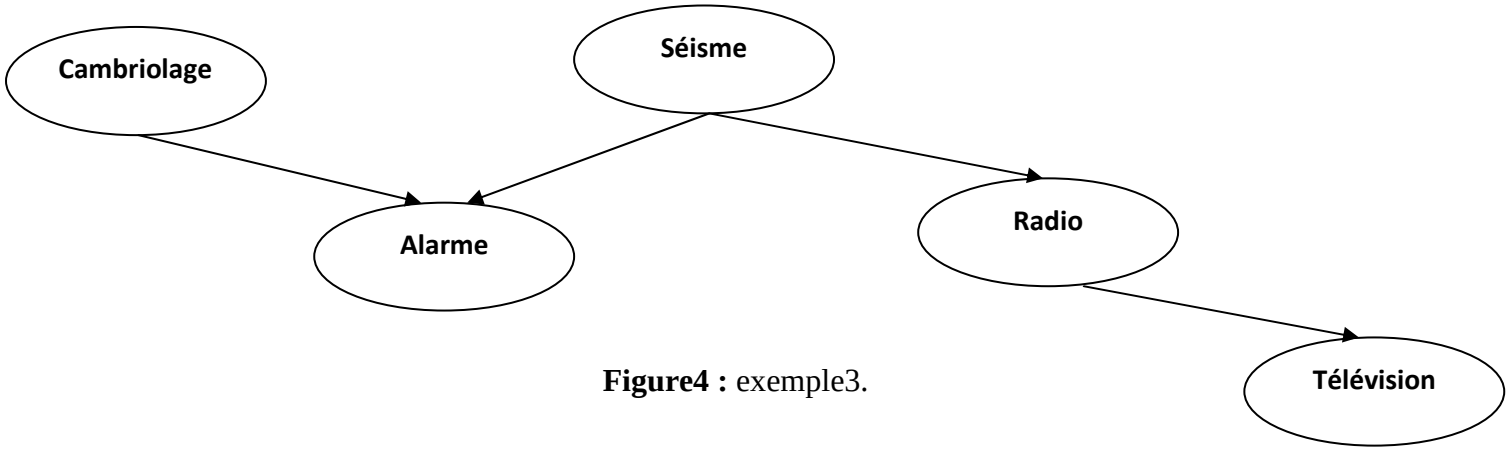



**Figure4 :** exemple3.

**P (cambriolage) = [0.000 0.999]**
**P (séisme) = [0.0001 0.9999]**
**P (radio | séisme)**

| | **Séisme=** | |
|---|---|---|
| | **O** | **N** |
| **Radio =O** | **0.99** | **0.01** |
| **Radio =N** | **0.01** | **0.99** |

**P (télévision | radio)**

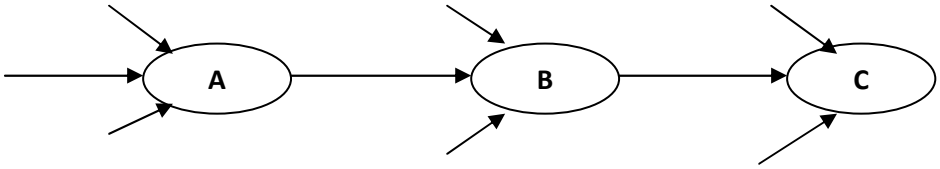
| | **Radio =** | |
|---|---|---|
| | **O** | **N** |
| **Télé =O** | **0.99** | **0.50** |
| **Télé =N** | **0.01** | **0.50** |

**P (alarme | cambriolage| séisme)**

| | **Cambriolage, Séisme=** | | | |
|---|---|---|---|---|
| | **O,O** | **O,N** | **N,O** | **N,N** |
| **Alarme =O** | **0.75** | **0.10** | **0.99** | **0.10** |
| **Alarme =N** | **0.25** | **0.90** | **0.01** | **0.90** |

### 4.2.3. La relation en série

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




P(A) dépend d'autres valeurs.

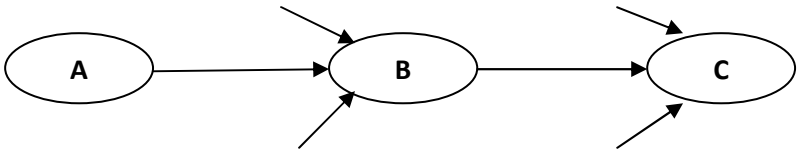


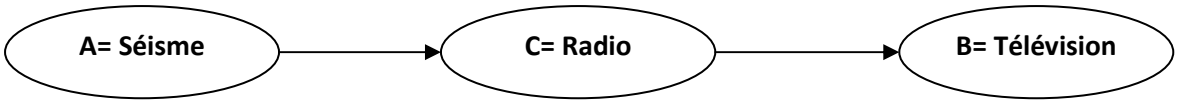

**Figure5 :** D-séparation cas des variables en série.

P(A) est fixée.

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

- Si on ne sait rien sur P(B), P(C) dépend toujours de P(A).
- Par contre, si P(B) est connu alors P(C) ne dépend plus de P(A).

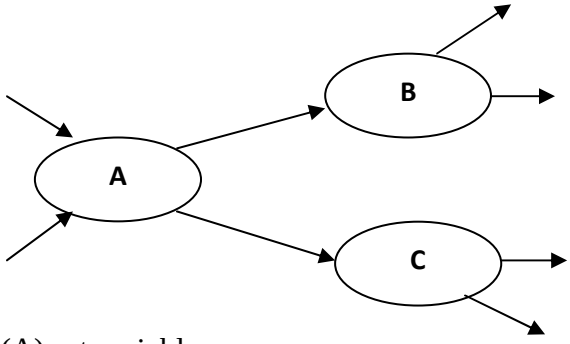
**Exemple**



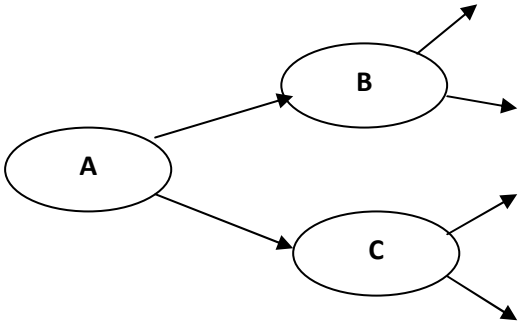

- A et B sont dépendants
- A et B sont indépendants conditionnellement à C
    - Si C est connue, A n'apporte aucune information sur B
    - P (B | C, A) = P (B|C)= P (B| parents(B))

### 4.2.4. Relation divergente

Soit le cas ou A, B et C sont en relation divergente




P(A) est variable




P(A) est fixé

**Figure6 :** D-séparation cas des variables divergentes.

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

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

**Exemple**

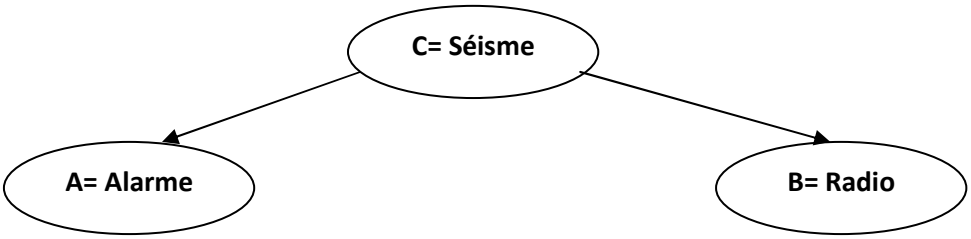
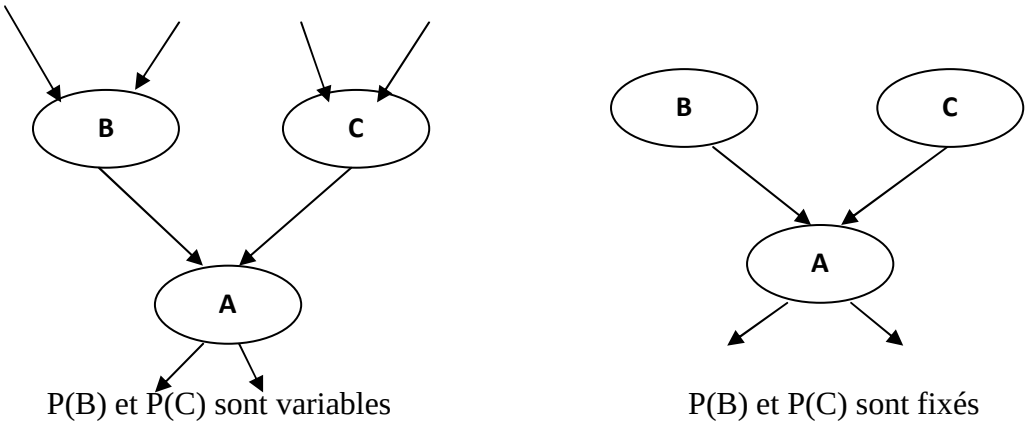



**Figure7 : exemple4.**

- A et B sont dépendants
- A et B sont indépendants conditionnellement à C
  - Si C est connue, A n'apporte aucune information sur B
  - P(B| C, A)= P (B|C) = P (B| parents(B)

### 4.2.5. Relation convergente

Soit le cas ou A, B et C sont en relation convergente




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

**Figure8 :** D-séparation cas des variables convergentes.

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

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

**Exemple**

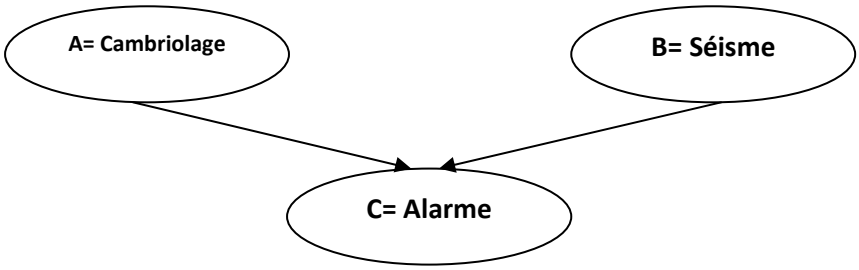



**Figure9 :** exemple5.

- A et B sont indépendants
- A et B sont dépendants conditionnellement à c
  - Sis C est connu, A apporte une information sur B
  - P (C| A, B)= P(C | parents(C))

## Introduction

Terrier, Terabyte Retriever, est un projet initialisé par l'université de Glasgow en 2000, dans le but de fournir une plate forme flexible pour le développement rapide des applications de recherche d'information. Terrier est un produit Open Source écrit en Java, qu'il a été mis à disposition du général public depuis Novembre 2004 sous la licence de MPL.

Le projet de terrier a exploré de nouvelles, efficaces et effectives méthodes de recherche pour les collections de documents, nouvelles combinaisons et idées de découpage-bord (cutting-edge) de théorie probabiliste, analyse statistique, et techniques de compression de données.

Ceci a mené au développement de déverses approches de recherche en utilisant un nouveau et un fort cadre probabiliste pour RI, dans le but pratique de la combinaison efficace et effective de diverses sources pour augmenter la performance de recherche.

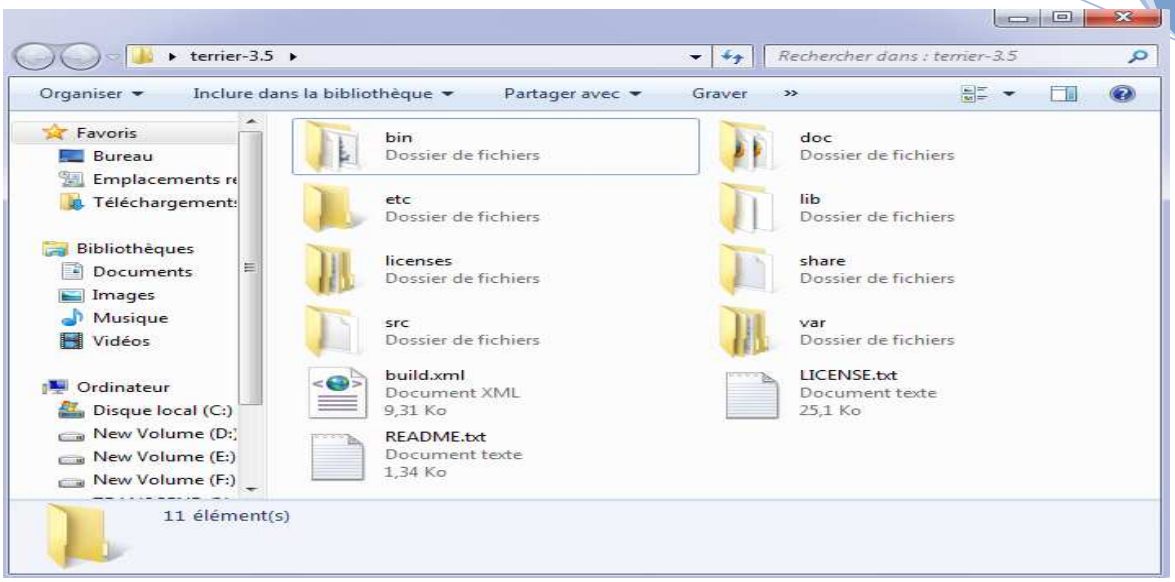
Terrier offre pliusieurs modèles de pondération de documents et expansion de requêtes basées sur le Framework DFR (Divergence From Randomness), Comme touts les moteurs de recherche, Terrier possède les principales facettes suivantes :

**Indexation :** Permet l'extraction des termes des termes des différents documents du corpus (basic indexed unit).

**Recherche :** Permet de générer des résultats aux requêtes formulées par les utilisateurs.

## 1. Installation de terrier 3.5 sous Windows

**Terrier version 3.5 sous Windows** : Extraire (décompresser) le contenu du fichier **terrier-3.5.zip** téléchargé à partir du site de la plate forme Terrier : **http://www.terrier.org**




## 2. Structure des répertoires de Terrier

Les répertoires de terrier sont structurés comme suit :

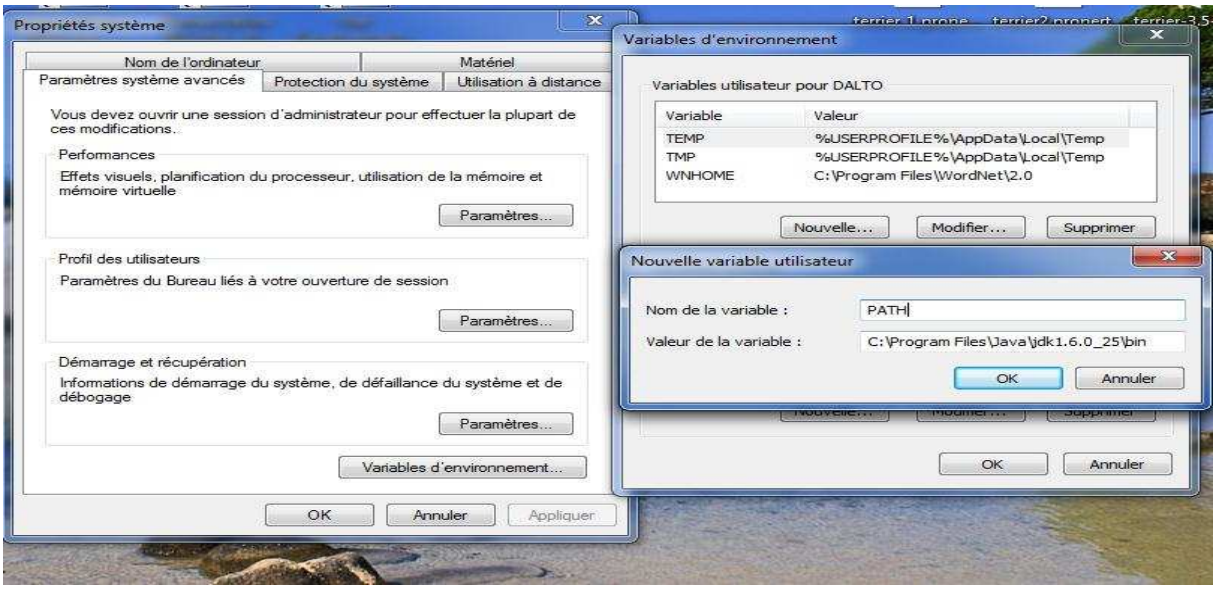
- **bin**/ pour exécuter terrier
- **doc**/ documentation relative à terrier
- **etc**/ Fichiers de configuration de terrier,le fichier **terrier.properties.sample** contient la plupart des propriétés de configuration de terrier.
- **lib**/ Classes compilés de terrier et les différentes librairies externes utilisées par terrier
- **share**/ Liste des mots vides (stopword-list.txt) et des exemples de documents à tester sur terrier
- **scr**/ Code source Java de terrier
- Var/

- Index/ structures de données après indexation (fichier inverse, fichier lexicon, index direct, index documents).
- Result/ Résultats de la recherche et l'évaluation.

## 3. Les applications de Terrier 3.5

❖ Terrier requiert la verssion Java 1.6 ou supérieure

- Vérifier si l'exécution des commandes java et javac dans l'invité de commandes sont reconnus (vérifier l'exécution du java.exe et le compilateur javac.exe)
- Sinon installer Java et ajouter une nouvelle variable d'environnement utilisateur PATH qui prend comme valeur : chemin vers le dossier bin du Java installé.



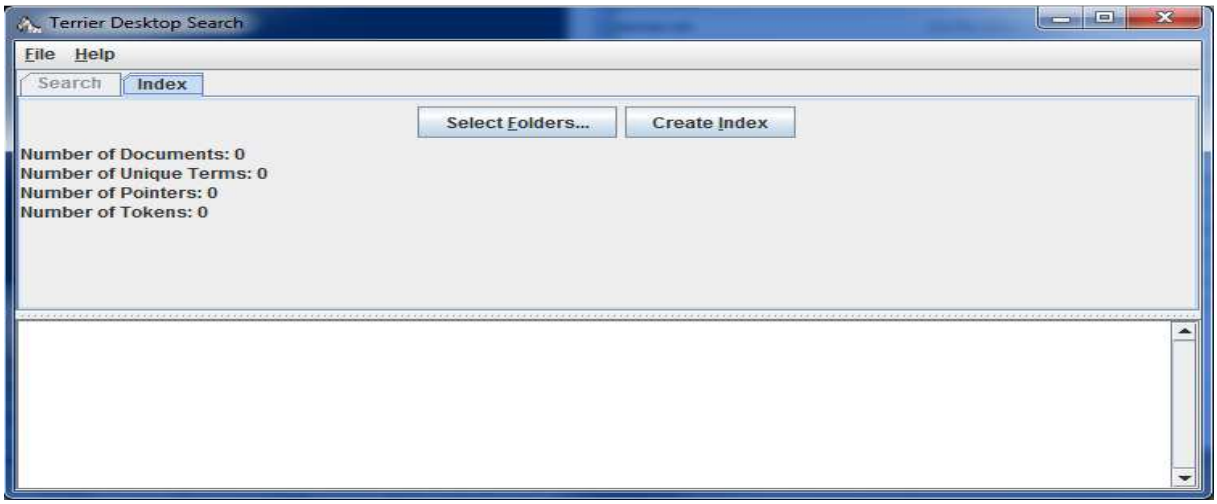

**Configuration Java**

Terrier est livré avec des applications :

1. Desktop Terrier
2. Trec (Batch) Terrier
3. Interactive Terrier

## ✓ Desktop Terrier

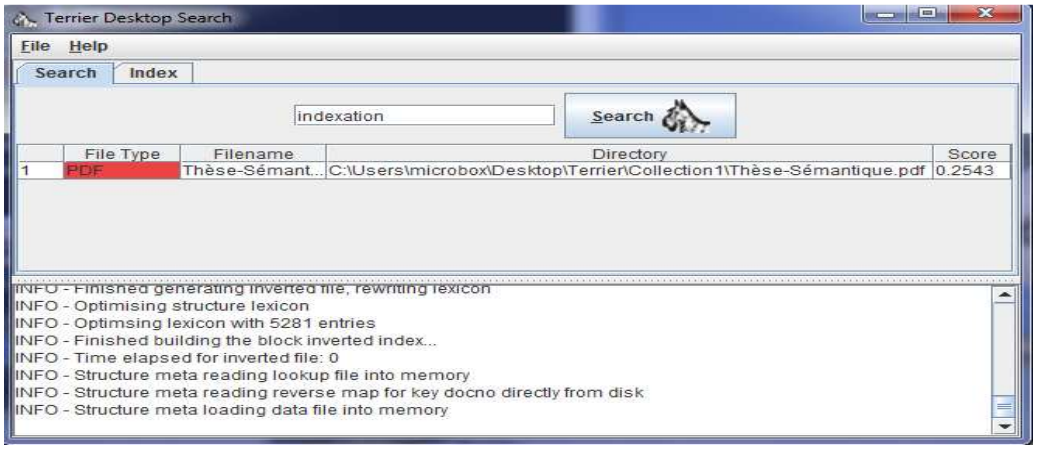
Desktop Terrier es une interface graphique pour l'indexation et la recherche de documents pertinents




**Interface Desktop Terrier**

Desktop Terrier propose deux onglets principaux :

- **Index :** Sèlèction des dossiersqui contiennet les documents à indexer et la création de l'index. Il prend en charge que les documents de type txt, text, pdf, MS Word, MS Excel, MS PowerPoint, HTML, XML, Tex, et les document Bip. Les formats complexes ne sont pas traités avec desktop terrier (examples : collections Trec).
- **Search :** Retourner les documents pertinents pour une requête saisie dans le champ texte à partir de l'index créé




Le desktop Terrier est un exemple d'application permettant de tester les caractéristiques de terrier. Pour effectuer les travaux complexes de l'indexation et la recherche, il est recommandé d'utiliser le Trec (Batch) Terrier de terrier en utilisant l'invité de commandes.

### ✓ TREC (Batch) Terrier

TREC Terrier est conçu principalement pour l'indexation, la recherche et l'évaluation des résultats sur des collections TREC

Un document du format TREC est délimité par les balises (tags) <DOC> </DOC>

### ✓ Interactive Terrier

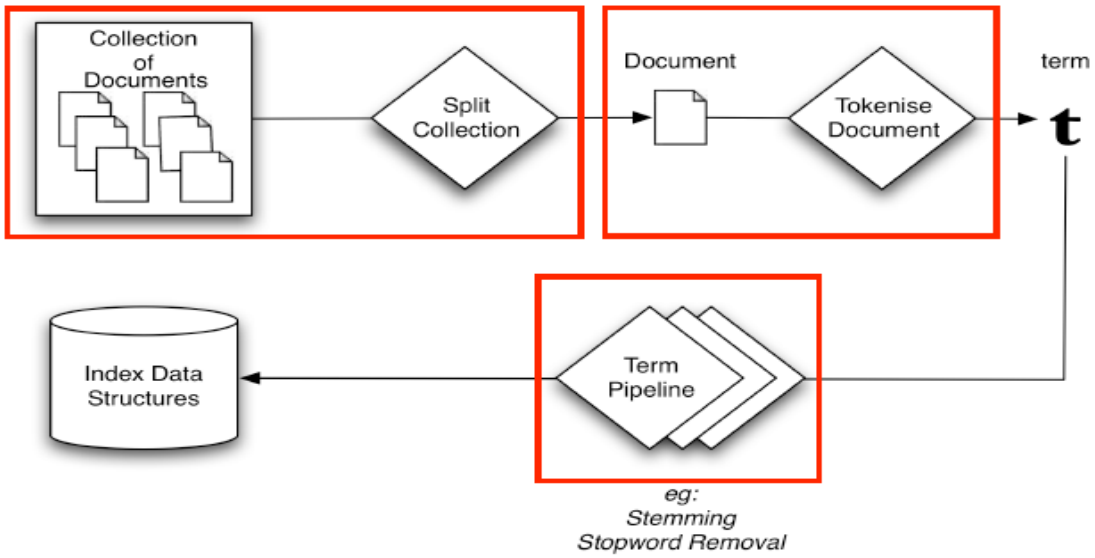
Interactive Terrier est une interface graphique pour la recherche de documents pertinents. Permet une recherche interactive en exécutants le script interactive-terrier. C'est un moyen facile de tester Terrier.

## 5. API d'indexation de Terrier

L'indexation dans Terrier est divisée en quatre étapes, et à chaque étape des plug-ins (des classes java) peuvent être ajouté pour la personnalisation du système. Les quatre étapes sont présentées comme suit :

a. Extraction de l'objet *Document* de la *Collection* qui générée par l'ensemble des corpus reçu en entrée par Terrier.
b. Parcourir chaque document de la collection pour en extraire les termes ainsi que les informations relatives.
c. Traduction des termes extraits en utilisant TermPipelines (mot vide, lemmatisation).
d. La construction de l'index via l'utilisation de la classe *Indexer*.

La figure suivante montre les déférentes étapes du processus d'indexation dans Terrier.




**Architecture d'indexation**

Dans se qui suit on présente les quatre étapes de ce processus :

### 5.1. Collection

Cette composante englobe le concept le plus fondamental de l'indexation avec terrier qui est la « collection ». C'est un objet qui représente le corpus, c'est-à-dire un ensemble de documents. *Collection* est une interface qui se trouve dans le package org.terrier.indexing, (..\src\core\org\terrier\indexing). Utilisée généralement par Terrier pour intégrer de nouvelles

sources de données (nouveau corpus) et cela en ajoutant une nouvelle classe java qui implémente cette interface.

Plusieurs Classes implémentent interface collection selon le format du document.

Exemples:

- SimpleFileCollection: PDF, TXT, HTML,…..
- TrecCollection
- SimpleXMLCollection : XML

### 5.2. Document

C'est une interface qui se trouve dans le package org.terrier.indexing, (..\src\core\org\terrier\indexing). Cette composante englobe le concept de Document.

Les classes qui implémentent cette interface s'occupe de parcourir les documents et dont extraire les différents termes. Terrier possède plusieurs parseurs de documents, par exemple : HTMLDocument, FileDocument, MSExcelDocument,etc…

### 5.3. TermPipeline

C'est une interface qui se trouve dans le package org.terrier.terms, (..\src\core\org\terrier\terms), cette composante reçoit l'ensemble des termes extraient des documents. Elle est considérée comme étant un pipeline qui traite et transforme chaque terme.

### 5.4. Indexer

Cette composante est chargée de la gestion du processus d'indexation. Entre autre la construction de l'index et de son écriture dans la structure de données appropriée. Terrier offre deux types d'indexeur : BasicIndexer,BlockIndexer.

### 5.5. Les structures d'Index (Index Data Structures)

Après l'indexation les termes sont stockés en structures de données suivante :

❖ **Lexicon :** informations de chaque terme de la collection

- Terme
- Id terme

- Nombre Docs qui contiennent le terme
- Fréquence terme dans la collection
- Offset terme dans le fichier inverse

❖ **Inverted Index :** Fichier inverse

- Id terme
- Id document
- Fréquence terme dans le document
- #filds (# of fields bits)

❖ **Direct index :** Index

- Id terme
- Fréquence terme
- #filds (#of fields bits)

❖ **DocumentIndex :**

- Id document
- Longueur document
- Byte offset dans Direct Index

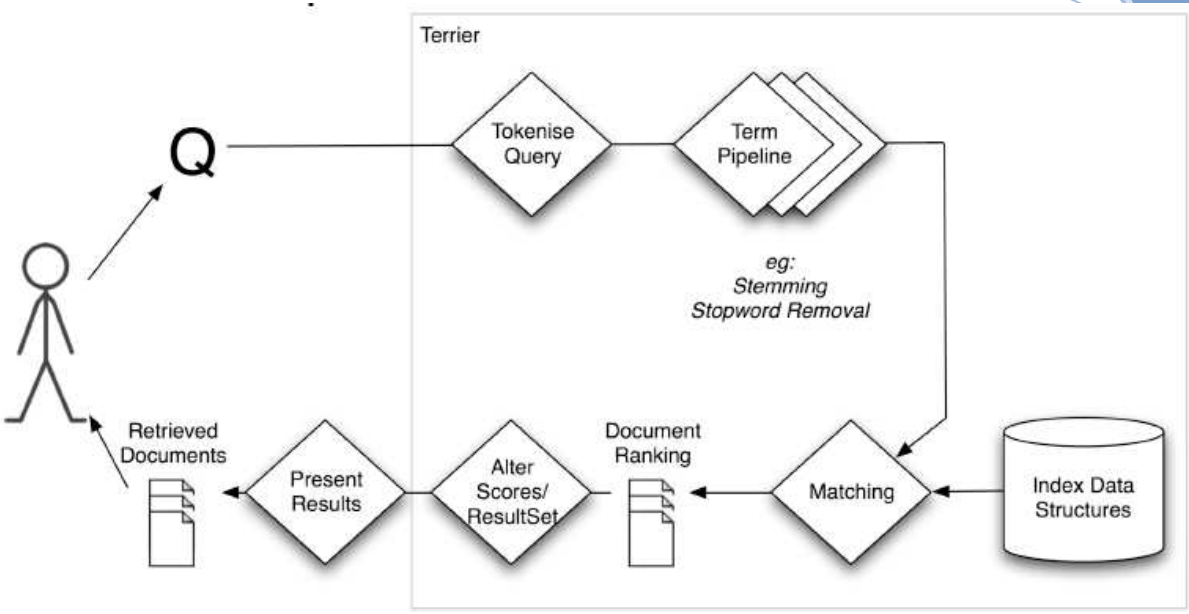
❖ **Propriétés Index :**

- Index sur les propriétés d'indexation (exemple : nombre de tokens dans l'index).

L'indexation en terrier peut être configurée en changeant les propriétés appropriées dans le fichier etc\terrier.properties. Chaque étape citée auparavant possède ses propres propretés.

## 6. L'API de recherche de terrier

Terrier utilise trois composants principaux pour la recherche : Query, Manager, set-results qui seront décrit ci-dessous :




**Présentation du processus de recherche dans Terrier**

## 6.1. Query

C'est l'entrée que l'application offre à terrier, le module *matching* est responsable de déterminer quel document correspond à une requête spécifique et attribut un score au document qui représente la requête.

Query est une classe abstraite, qui se trouve dans le package org.terrier.querying.parser (..\src\core\org\terrier\querying\parser), Un objet Query est créé pour chaque requête.

## 6.2. Manager

C'est le modèle qui chargé de la gestion de la recherche. Dans un premier niveau il lit chaque requête en utilisant le parseur approprié. Puis il crée un second niveau de gestion associé à chaque requête en récupérant l'ensemble des paramètres nécessaires et en spécifiant un modèle de pondération. Puis il crée un fichier résultat (result file) où il associe à chaque requête les documents qui lui sont pertinents. Le résultat de chaque requête est donné par le second niveau de gestion.

Le second niveau de gestion est responsable de l'ordonnancement et de la coordination des opérations principales de haut niveau sur une seule requête. Ces opérations sont : pre-processing, Matching, Post-processing, Post-filtring.

- **Pre-processing :** Appliquer Tokeniser et termPipeline
- **Matching :** Déterminer les documents qui répondent à la requete en initialisant :
  - **WeightingModels :** Assigner un score pour chaque terme dans le document (pondération), il existe plusieurs modèles de pondération tel que : TF_IDF, BM25… qui se trouvent dans ..\terrier3.5\src\core\org\terrier\matching\models
  - **DocumentScoreModifiers :** modifier le score d'un document en fonction du langage de la requête (ou expansion de la requête).
- **Post-filtring :** Filtrer les documents pertinents pour la requête.
- **Post-processing :** Reclasser les documents pertinents s'il ya une expansion de la recherche.

### 6.3. Set-Results : Retourner les documents selon leur degré de pertinence.

## 7. Modification de terrier

L'une des caractéristiques principales de Terrier est son extensibilité, il permet la modification du code source pour intégrer tout changement nécessaire. Pour que toute modification soit prise en compte, il est nécessaire de recompiler tout le code source de Terrier.

# ANNEXES

# *Bibliographies*

**[Per ,88] :**Peral J., Probabilistic reasoning in intelligent systems, San Mateo, California: Morgan Kaufmann Publishers.

**[Jam ,95]** Jameson, A., Numerical uncertainty management in user and student modeling: an overview of systems and issues, user modeling and user adapted interaction 5(3-4), 193:251, 1995

**[ZA, 01]** I. Zuckerman, D.W. Albercht, Predictive statistical models for user modeling and user adapted interaction, 11 5:18, 2001

**[Salton, 1971]** Salton .G. A comparison between manual and automatic indexing methods. Journal of American Documentation, 20(1): pages 61–71, 1971.

**[BADR, 2007]** Georges BADR. "Recherche d'Information sur le Web: Impact de la structure des documents sur la pertinence des résultats". Maitrise, IRIT, Université Paul Sabatier & Institut National Polytechnique de Toulouse. 2007.

**[Urfist, 04] :**Alexandre SERRES, ”Cours de problématique générale de la recherche d’information ‘’, URFIST Bretagne pays de Loire, 2002.

**[Ingwersen., 92]** P. Ingwersen. Information retrieval interaction. London, Taylor Graham, 1992.

**[Saracevic, 70]** Saracevic T. “the concept of “relevance” in information science: a historical review”, dans T Saracevic (dir), Introduction to Information science. R.R. Bowker, New York, pages: 111-151, 1970.

**[Denos, 97]** Denos N. Modélisation de la pertinence en recherche d'information: modèle conceptuel, formalisation et application. Thèse de Doctorat de l'Université Joseph Fourier-Grenoble I, 1997

**[BADR, 2007]** Georges BADR. "Recherche d'Information sur le Web: Impact de la structure des documents sur la pertinence des résultats". Maitrise, IRIT, Université Paul Sabatier & Institut National Polytechnique de Toulouse. 2007.

**[Fox, 1992]** Fox C. Lexical analysis and stoplists. In: Information retrieval. Upper Saddle River, NJ: Prentice-Hall; 1992.

**[Porter, 1997]** Porter MF. An algorithm for suffix stripping. In: Readings in information retrieval. San Francisco, CA: Morgan Kaufmann Publishers Inc.. p. 313-316. 1997.

**[Salton, 1975]** Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Communications of the ACM :613–620.1975.

**[Singhal & al., 96]** A. Singhal, C. Buckley, M. Mitra. Pivoted document length normalization. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. Zurich, Switzerland .Pages: 21 - 29. 1996

**[Robertson, 1997]** Robertson SE, Walker S. On relevance weights with little relevance information. SIGIR :16–24. 1997.

**[Bai & al., 2006]** Jing Bai, Jian-Yun Nie, Guihong Cao. "Context Dependent Term Relations for IR". EMNLP 06. 2006.

**[Khan, 2000]** Latifur R. Khan, "Ontology -based Information Selection, Phd Thesis", Faculty of the Graduate School, University of Southern California. August 2000.

**[Baziz & al., 2003]** Mustapha Baziz, Nathalie Aussenac-Gilles, Mohand Boughanem, "Exploitation des Liens Sémantiques pour l'Expansion de Requêtes dans un Système de Recherche d'Information " Dans : XXIème Congrès INFORSID. Pages : 121-134, 2003.

**[Voorhees, 93]** E. Voorhees, "Using WordNet to Disambiguate Word Senses for Text Retrieval" Proceedings of the 16thAnnual Conference on Research and Development in Information Retrieval, SIGIR'93, Pittsburgh, PA, 1993

**[Järvelin & al., 2004]** Airio, E, Järvelin, K, Saatsi, P, Kekäläinen, J, & Suomela, "CIRI - An Ontology-based Query Interface for Text Retrieval". In Hyvönen, E, Kauppinen, T, Salminen, M, Viljanen, K & Ala-Siuru, P, eds. Web Intelligence. Helsinki: Finnish Artificial Intelligence Society. Pages: 73-82. 2004.

**[Bhogal & al., 2006]** J. Bhogal, A. Macfarlane, P. Smith. "A review of ontology based query expansion" Information Processing and Management 43 pages: 866–886. 2006.

**[Robertson, 94]** ROBERTSON S. E., WALKER S., « Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval », Proceedings of SIGIR 1994, pages: 232-241, 1994.

**[Rijsbergen, 79]** Van Rijsbergen C.J. Information Retrieval, Butterworths, londres, 1979.

**[Baeza-Yates & al., 99]** Ricardo A. Baeza-Yates, Berthier A. Ribeiro-Neto: Modern Information Retrieval ACM Press / Addison-Wesley 1999.

**[Soule, 2007]** Soule-Dupuy C., Recherche et exploration d'information, Objectifs de la RI, Cours Master 2IH, 2006/2007.

**[Kob.01]:** A. Kobsa, Genericuser modeling systems, User modeling anduser adapted intraction, 11 49 :63, Kluwer Academic Publishers. 2001.

**[Gre ,84]** : S. Greenberg, User modeling in interactive computer systems. M.Sc thesis, Dpt of computer science, University of Calgary, Calgary, Report 85/193/6, 1984

**[Pcg ,03]** :D.Poo, B. Cheng, JM. Goh, A hybrid approach for use profiling, 36th annual Hawai International Conference on System Sciences (HICSS'03)- Track 4, January 2003.

**[Riw, 94]:** Resnick, P., Iacovou, N. And Ward, B., An open architecture for collaborative filtering of netnews, In CSCW'94 Proceedings of the conference on computer Supported Collaborative Work, 175:186, 1994

**[Sal 1971]** G. Salton, The SMART retrieval system: experiments in automatic document processing, Prentice Hall Inc, NJ., 1971

**[PG 1999]** A. Pretshner, S. Gauch, Personalization on the web, Technical report ITTC-FY2000-TR-13591601? Information and telecommunication technology center, Deprtement of electrical engineering and computer science, University of Kansas, December 1999

**[SKW 2000]**: Scime, A., Kershberg, L., WebSifter: An Ontology-based personalizable search agent for the Web, In Proceedings of International Conference on Digital Librairies: Research and Practice (2000).

**[ZEM, 03/04] :** Vers le développement d'un système de recherche d'information personnalisé intégrant le profil utilisateur par Zemirli W.Nesrine Equipe SIG/RI

**[Turtl, 91]** H.R. Turtle. Inference Networks for Document Retrieval. PhD thesis, 1991.

**[HTBC, 91]** **H**oward **T**urtle and **B**ruce W. **C**roft. Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems, 9(3) :187_222, 1991.

**[Ricard, 99]** Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. Modern Information Retrieval. ACM Press / Addison-Wesley, 1999.

**[Hallouli, 2004]** Hallouli K., «Reconnaissance de caractères par méthodes markoviennes et réseaux Bayésiens», Thèse de Doctorat spécialité Signal et Images, Ecole Nationale Supérieur des Télécommunications, Télécom Paris, Mai 2004

**[SR, 75]** Saracevic, T. Relevance : A review of and a framework for the thinking on the notion in information science. Journal of the American Society for Information Science (JASIS) 32, 2 (1975), 321–343

**[CCH, 92]** Callan, J., Croft, W., and Harding, S. The INQUERY retrieval system. In Proc. of International Conference on Database and Expert
Systems Applications (DEXA) (1992), pp. 78–8

**[RT, 89]** van Rijsbergen, C. J. Towards an information logic. In Proc. Of the International ACM-SIGIR Conference (1989), pp. 77–8

**[CWB, 87]** Croft, W., and Bruce, W. Approaches to intelligent information retrieval. Information Processing and Management : an International
Journal 23, 4 (1987), 249–254

**[Sal, 88]** Salton, G. Syntactic approaches to automatic book indexing. In Proc. of the annual meeting on Association for Computational Linguis- tics (ACL) (1988), Department of Computer Science, Cornell University, Ithaca, New York, pp. 204–210.

**[SalB, 88]** Salton, G., and Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing & Management (IPM) 24, 5 (1988), 513–523.

**[Bri, 05]** ASMA HEDIA BRINI,Un Modèle de Recherche d'Information basé sur les Réseaux Possibilistes thèse Doctorat de l'Université Paul Sabatier 2005.

**[RNM, 96]** Ribeiro-Neto, B., and Muntz, R. R. A belief network model for ir. In Proc. of the International ACM-SIGIR Conference (1996), pp. 253–260.

**[SJW & AL 02]** A.Sprink, B.Janson, D.Wolfram, T.Saracevic. From E.sex to E-commerce: Web serche changes. IEEE Computer 35(3): 107-111, 2002