



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



UNIVERSTE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Master en Informatique

Option : conduite de projets informatiques

Thème

*Intégration des intérêts sociaux
dans le processus de la recherche
d'information sociale*

Proposé et dirigé par :
M^{me} L. BOUGCHICHE.

Réalisé par :
M^{lle} MEDDAH Nawel.

M^{lle} AMENOUCHE Ferroudja.

Promotion 2011/2012

Remerciement

Merci, Merci, Merci,...

*Pour le courage et la persévérance dont vous nous avez armés, on s'incline devant votre majesté et acclame votre nom majestueux « **dieu** »*

Nos plus vifs remerciements vont à notre promotrice M^{me} Bougchiche Lilia pour nous avoir proposé ce sujet et nous avoir dirigé tout au long de son réalisation. sa compétence, ces critiques et son bon sens nous ont largement aidé à réaliser ce travail, nous lui exprimons ici nos sincères reconnaissances.

Nous tenons aussi à remercier l'ensemble des membres du jury de soutenance pour l'honneur qu'ils nous ont fait en acceptant de juger ce modeste travail.

Nous tenons à remercier toute personne qui nous a aidé de près ou de loin.

∞ Dédicaces ∞

*Je préserve tout ce travail au nom de notre dieu miséricordieux
et de son prophète MOHAMMED « sala alaho alih wa salam ».*

Et je dédie ce modeste travail à :

Mes chers parents,

Mes chères sœurs,

Mes chers frères,

A toute ma famille paternelle et maternelle,

A tous mes amis,

A mon binôme « NAWEL ».

FERROUDJA

∞ Dédicaces ∞

*Je préserve tout ce travail au nom de notre dieu miséricordieux
et de son prophète MOHAMMED « sala alaho alih wa salam ».*

Et je dédie ce modeste travail à :

Mes chers parents,

Mes chères sœurs,

Mes chers frères,

A toute ma famille paternelle et maternelle,

A tous mes amis,

A mon binôme « FERROUDJA ».

NAWEL

Résumé :

L'explosion du Web 2.0 (blogs, wikis, sites de partage, réseaux sociaux, etc.) ouvre des perspectives inédites de partage et de gestion de l'information, l'accès à cette information est facilité par l'apparition des réseaux sociaux qui ont conduit à l'émergence d'une nouvelle branche de la Recherche d'Information (RI) : la RI sociale. Il s'agit d'adapter les modèles et les algorithmes de la RI classique afin d'exploiter les informations sociales propres à ce nouveau cadre. Ces informations sociales ont un impact sur le processus de la RI, dont l'objectif est d'améliorer les résultats de la RI classique. Les informations sociales (relations sociales, annotations, clicks, etc.) peuvent être exploitées au sein même du modèle de RI (modèle de document et de requête, fonction de pondération / de correspondance).

L'objectif principal de ce projet est l'amélioration de la recherche d'information classique telle qu'en intégrant les intérêts sociaux (les tags sociaux, dans notre cas) dans le processus de la recherche, de nouveaux documents peuvent apparaître sur la liste restituée par le système. Si ces nouveaux documents n'étaient pas restitués auparavant, c'est qu'ils ne contenaient pas de termes identiques aux mots clés de la requête. Avec notre solution, le système ne se contente pas d'une recherche dans la collection de documents mais extrait les intérêts sociaux explicites et fouille dans l'ensemble des tags sociaux qui sont assignés à ces documents afin de retrouver les tags (des termes supplémentaires et complémentaires au contenu textuel des documents) similaires aux termes de cette requête.

sommaire

Sommaire

<i>Introduction générale</i>	1
------------------------------------	---

Chapitre I : La recherche d'information sociale(RIS).

I.1 Introduction	2
I.2 la RI classique / La RI sociale	2
I.2.1 la RI classique	2
I.2.2 La RI sociale	6
I.2.2.1 Définition de la RI sociale	6
I.2.2.2 Les informations sociales	6
I.2.2.3 Un Modèle du Domaine pour la RI Social	9
I.3 Les modèles de la recherche d'information sociale	9
I.4 Conclusion	11

Chapitre II : Les intérêts sociaux.

II.1 Introduction	12
II.2 La notion d'intérêt	12
II.3 Type d'intérêt	13
II.3.1 intérêt implicite	13
II.3.2 intérêt explicite	13
II.4 Les indicateurs d'intérêts	13

II.4.1 les indicateurs d'intérêts explicites	15
II.4.2 Indicateurs d'intérêt implicites	17
II.5 Détection des centres d'intérêt	21
II.5.1 Le comportement de lecture	21
II.5.2 Le comportement de sauvegarde	22
II.5.3 Le comportement d'impression	22
II.6 Modélisation des centres d'intérêt de l'utilisateur	23
II.6.1 Modèle de Wen	24
II.6.2 Le modèle de Tamine	25
II.7 Maintenance des centres d'intérêt	27
II.7.1 Détection du basculement dans le sujet de la requête	27
II.7.2 Affinement des centres d'intérêt à court terme	27
II.8 Evolution des centres d'intérêt	28
II.9 Conclusion	30

Chapitre III : les intérêts dans la RI sociale - notre approche-

III.1 Introduction	31
III.2 Les objectifs de notre système	31
III .2.1 Description de notre approche	31

III.2.2 Architecture du système	32
III.2.3 Description des composants du système	33
III.2.4 Le module de requête.....	34
III.2.5 Le module de document	34
III.2.5.1 Le processus d'indexation	35
III.2.6 Le module de recherche.....	43
III.2.7 Le module d'intérêts sociaux	46
III.3 Conclusion.....	47
 <i>Chapitre VI : Implémentation et test.</i> 	
IV.1 Introduction.....	48
IV.2 Présentation de l'environnement du travail	48
IV.2.1 Aspect technique du matériel utilisé.....	48
IV.3 L'environnement de développement	48
IV.3.1 Le langage de programmation.....	48
IV.3.2 présentation de l'environnement eclipse.....	51
IV.4 La plate-forme Terrier	52
IV.5 Implémentation	53
IV.6 Evaluation expérimentale de notre approche	54
IV.6.1 Collection de tests.....	54
VI.7 Evaluation expérimentale de notre approche	56
IV.8 Conclusion.....	58
Conclusion générale.....	60

Annexe I	62
Annexe II	70
Annexe III	77
Bibliographie	

liste des figures

TABLE DES FIGURES

Chapitre I

Figure I.1 Modèle de la RI classique	3
Figure I.2 représentation des informations sociales.....	7
Figure I.3 Modèle de RI sociale : l'utilisateur producteur et consommateur d'informations	8

Chapitre II

Figure II.1 Les indicateurs d'intérêt.....	14
Figure II.2 page d'accueil du site del.icio.us	16
Figure II.3 Principe d'évolution des centres d'intérêt	28

Chapitre III

Figure III.1 Schéma général de notre approche	32
Figure III.2 Architecteur générale de notre système	34
Figure III.3 Processus d'indexation de Terrier.....	36
Figure III.4 Exemple de la structure d'un Index de document	38
Figure III.5 Exemple de la structure d'un Direct Index de document	39
Figure III.6 Exemple de la structure d'un Inverted Index de document	40
Figure III.7 Exemple de la structure d'un Lexicon de document.....	41
Figure III.8 Exemple de la structure d'un Meta Index de document.....	41
Figure III.9 Processus de recherche de Terrier	44
Tableau III.1 Paramètres de la fonction BM25.....	47

Chapitre IV

Figure IV.1 Interface « Eclipse GALILEO	52
Tableau IV.1 Description de la collection TIME.....	55
Tableau IV.2 Contenus des fichiers résultats de la recherche.....	57
Figure IV.2 Graphe représentant un Score finale pour $\alpha=0,3$	57

Introduction générale

Introduction générale

L'accès à une information pertinente, adaptée aux besoins et au contexte de l'utilisateur est un défi dans un environnement Internet, caractérisé par une prolifération des ressources hétérogènes (données structurées, documents textuels, composants logiciels, images), conduisant à des volumes de données considérables, et la quantité de données disponibles sur le web devient tellement volumineuse que l'utilisateur se trouve face à une masse d'informations difficile à appréhender. Elle induit souvent une surcharge cognitive chez l'utilisateur et rend difficile l'accès à l'information répondant à ses besoins. Par conséquent, la recherche d'une information spécifique sur le Web peut être une vraie gageure.

Dans un domaine en pleine effervescence comme les systèmes des réseaux sociaux, observer les activités des utilisateurs afin de trouver des similarités entre les individus, de faire des recommandations, d'améliorer le processus de la recherche d'information est une tâche qui n'est pas facile. Le nombre de chercheurs qui s'intéressent à ce domaine ne cesse de croître. Ce phénomène est lié au développement de l'Internet et au besoin de trouver des outils pour faciliter leurs activités quotidiennes.

D'un autre côté, les individus sont de plus en plus intéressés par les sites des réseaux sociaux, afin de stocker, de classer, de chercher et de partager, des documents, des photos, leurs liens favoris. Ainsi, les internautes ayant les mêmes centres d'intérêts peuvent consulter et sauvegarder les ressources que d'autres ont trouvé, les services des réseaux sociaux permettent aux utilisateurs d'organiser les ressources en utilisant des étiquettes (des tags) pour faciliter une recherche ultérieure. Le succès immédiat de tels systèmes (tel que del.icio.us, Facebook, MySpace, et YouTube), ont eu est dû au fait qu'aucune compétence de la part de l'individu n'est nécessaire pour les utiliser et pour obtenir des bénéfices immédiats. Ce succès ainsi que leur popularité ont provoqué un grand intérêt auprès de la communauté de la recherche.

Le regroupement des utilisateurs en communautés virtuelles se fait autour d'un centre d'intérêt commun et les encouragent à contribuer et partager plus de contenus. Découvrir ces intérêts sociaux et les intégrer dans le processus de la recherche d'information est un challenge vu la difficulté de détecter et de représenter ces intérêts.

Dans la littérature il n'existe pas d'approche pour la prise en compte des intérêts sociaux dans le domaine de la RI, vu que la plupart des travaux se basent sur les communications en ligne des utilisateurs et donc comment identifier l'intérêt commun des

utilisateurs qui interagissent en ligne ? Ces intérêts qui peuvent être bénéfiques dans plusieurs domaines tel que : l'apprentissage et la recherche d'information. De ce fait, comment les exploiter pour améliorer le processus de la recherche d'information?

Notre travail se situe dans le cadre de la recherche d'information contextuelle et plus précisément dans le cadre de la recherche d'information sociale permettant d'exploiter les informations sociales avec un objectif d'améliorer les résultats de la RI classique c'est-à-dire intégrer les intérêts sociaux au sein d'un réseau social afin de mieux répondre aux requêtes formulées par les utilisateurs.

Le présent manuscrit est constitué des quatre chapitres suivants :

- *Chapitre 1 : « ce chapitre traite la recherche d'information sociale », ils donnent une petite définition sur la recherche d'information sociale RIS, comme il cite les différentes informations sociales et enfin il présente quelque modèle de la recherche d'information sociale existant.*
- *Chapitre 2: « aborde les intérêts sociaux » décrit les intérêts ou il donne une définition du mot intérêts, il présente aussi les types d'intérêts et les différentes indicateurs d'intérêts ainsi, il présente quelque modèles d'intérêts existants.*
- *Chapitre 3 « : ce chapitre présente la phase de description de notre approche ».*
- *Chapitre 4 : Enfin, dans ce chapitre nous avons présenté «les résultats et expérimentations de notre approche ».*

Nous concluons notre travail et présentons quelques perspectives en fin du mémoire

Chapitre 1

Recherche d'information sociale

I.1 Introduction

Le développement d'Internet a permis de former différents types de réseaux sociaux(RS) à grande échelle. Une des caractéristiques fondamentale de ces réseaux, outre leur taille, est le caractère “explorable” des informations qu’ils contiennent, l’exploration qui peut être mené par des processus d’analyse automatique. Un réseau social sur le web est ainsi constitué d’un ensemble d’utilisateurs associés à des données (profils, documents, etc.) structurées par des liens qui sont créés lors des interactions sociales (référencement, mails, etc.). La présence de ces données permet de voir ce type de réseaux comme des “contenu de graphes ”. Qui sont maintenant reconnus comme un moyen important pour la diffusion de l'information [Salton 2010]. Il existe bien sur différents types de contenus que les utilisateurs peuvent partager dans ces réseaux : Textes, Documents (messages ou articles), Annotations, Commentaires, Images, Vidéos.

L'explosion du Web 2.0 (blogs, wikis, sites de partage, réseaux sociaux, etc.) ouvre des perspectives inédites de partage et de gestion de l'information, l'accès à cette information est facilité par l'apparition des réseaux sociaux qui ont conduit à l'émergence d'une nouvelle branche de la Recherche d'Information (RI) : la RI sociale. Il s'agit d'adapter les modèles et les algorithmes de la RI classique afin d'exploiter les informations sociales propres à ce nouveau cadre. Ces informations sociales ont un impact sur le processus de la RI, dont l'objectif est d'améliorer les résultats de la RI classique. Les informations sociales (relations sociales, annotations, clicks, etc.) peuvent être exploitées au sein même du modèle de RI (modèle de document et de requête, fonction de pondération / de correspondance).

I.2 la RI classique /La RI sociale

I.2.1 la RI classique

La RI traite différents aspects, notamment la représentation, le stockage, l'organisation et l'accès à l'information. Un système de RI se base sur une comparaison entre la représentation interne de la requête et la représentation interne des documents du corpus, comme montre la *Figure I.1*.

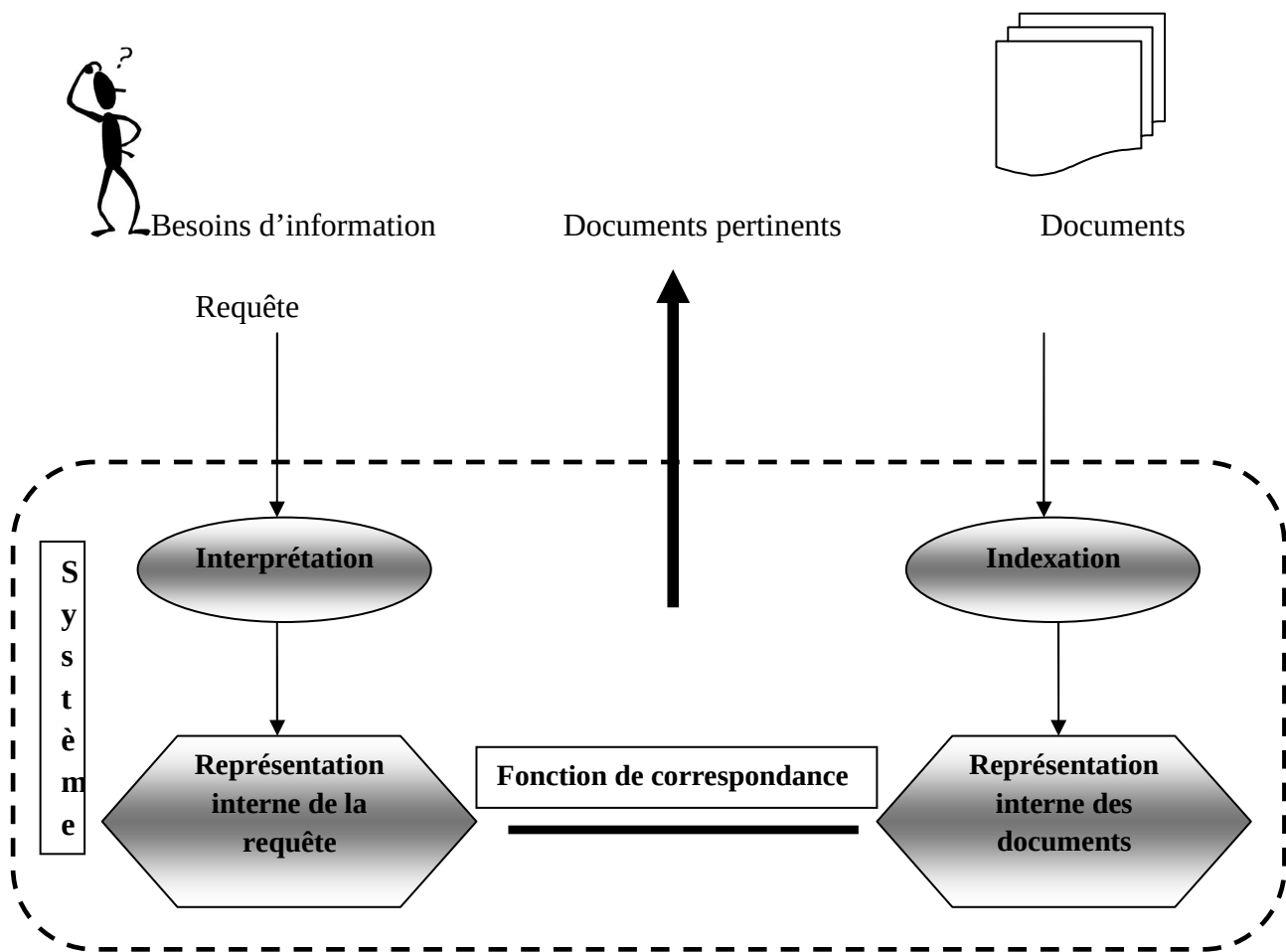


Figure I.1 Modèle de la RI classique.

• Recherche d'information

La Recherche d'Information (RI) est un domaine de recherche en informatique qui s'attache à définir des modèles et des systèmes dont le but est de faciliter l'accès à un ensemble de documents sous forme électronique (corpus), afin de permettre à un utilisateur de retrouver ceux qui sont pertinents pour lui, c'est-à-dire ceux dont le contenu correspond le mieux à son besoin d'information à un moment donné. Le domaine de la RI est donc centré autour des utilisateurs, notamment par le biais de la notion de pertinence, c'est-à-dire l'adéquation pour l'utilisateur entre le contenu d'un document et l'information recherchée. Avec l'émergence des réseaux électroniques et le développement du Web, la quantité d'information disponible pour les utilisateurs est sans cesse croissante, et la RI doit faire face

à de nouveaux défis d'accès à l'information, à savoir retrouver une information dans un espace diversifié et de taille considérable.

● Document

Le document représente le conteneur élémentaire d'information, exploitable et accessible par le SRI. Un document peut être un texte, une page web, une image, une bande vidéo, etc...Il s'agit de toute unité qui peut constituer une réponse à un besoin en information exprimée par un utilisateur.

● Requête

Une requête constitue l'expression du besoin en informations de l'utilisateur dans un langage de requête. Plusieurs systèmes utilisent des langages différents pour décrire la requête.

● Pertinence

La pertinence est la notion centrale en RI car toutes les évaluations s'articulent autour de cette notion. Elle définit le degré de correspondance entre un document et une requête. C'est une mesure d'informatisation du document à la requête. Il existe deux types de pertinences : la pertinence système, pertinence utilisateur.

● Besoin en information

On distingue 3 types de besoins en information :

Besoin vérificatif, Besoin thématique connu, Besoin thématique inconnu.

● Systèmes de recherche d'information

Les SRI servent d'interface entre une source (collection) contenant des quantités considérables de documents et des utilisateurs les cherchant, via des requêtes, des informations susceptibles de se trouver dans cette collection. Un SRI inclut un ensemble de procédures et d'opérations qui permettent la gestion, le stockage, l'interrogation, la recherche, la sélection et la représentation de cette masse d'informations. Il est défini comme suit :

Un Système de Recherche d'Informations (SRI) est un système informatique qui permet de retrouver à partir d'un ensemble de documents, les documents pertinents et précis pour un besoin en information d'un utilisateur, exprimé à l'aide d'une requête.

❖ Le processus de la RI

Un système de recherche d'information intègre trois(3) fonctions principales (représentation des documents, représentations de la requête, appariement 'document-requête') représentées schématiquement par le processus en U de recherche d'information [Belkin *et al.* 92] (**Figure I.1**)

De manière générale, la recherche dans un SRI consiste à comparer la représentation interne de la requête aux représentations internes des documents de la collection. La requête est formulée, par l'utilisateur, dans un langage de requêtes qui peut être le langage naturel, un langage à base de mots clés ou le langage booléen. Elle sera transformée en une représentation interne équivalente, lors d'un processus d'interprétation.

Un processus similaire, dit l'indexation, permet de construire la représentation interne des documents de la base documentaire. Le processus de recherche consiste alors à mettre en correspondance et à calculer le degré d'appariement des représentations internes des documents et de la requête. Les documents qui correspondent au mieux à la requête, ou documents dits pertinents, sont alors retournés à l'utilisateur, dans une liste ordonnée par ordre décroissant de degré de pertinence lorsque le système le permet. 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.

Ce processus fait ressortir trois mécanismes de base : le processus d'indexation (quelques fois dit processus d'interprétation pour les requêtes), le processus de recherche (appariement) et le processus de reformulation des requêtes.

❖ L'indexation

L'indexation est l'opération qui consiste à décrire et à caractériser un document à l'aide de représentations des concepts contenus dans ce document, c'est-à-dire à transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse.

Techniquement, l'indexation peut-être manuelle, automatique ou semi-automatique [Salton, 1988].

❖ Modèles de recherche d'information

Si c'est l'indexation qui choisit les termes pour représenter le contenu d'un document ou d'une requête, c'est au modèle de leur donner une interprétation. Étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit les deux rôles suivants:

- créer une représentation interne pour un document ou pour une requête basée sur ces termes.
- définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance (ou similarité).

Le modèle joue un rôle central dans la RI. C'est le modèle qui détermine le comportement clé d'un système de RI. On cite quelques modèles souvent utilisés dans la RI : le Modèle booléen, le Modèle logique, le Modèle probabiliste.

I.2.2 La RI sociale

I.2.2.1 Définition de la RI sociale

La recherche d'information sociale est définie comme l'incorporation d'information concernant les réseaux sociaux et les rapports dans le processus de la recherche d'information:

La recherche d'information social = réseaux sociaux + recherche d'information

La recherche d'information social a pour but: d'intégrer les concept de réseau social dans les processus de la recherche d'information avec exploration de réseau social pour accéder à l'information pertinente et estimer la pertinence d'un document à partir de son contexte social et intégrer deux domaines la recherche d'information qui représente et compare les documents et les requetes et l'analyse des réseau sociaux qui représente les entités sociales et estime la centralité d'un individu.

I.2.2.2 Les informations sociales

Les utilisateurs du Web produisent divers contenus, créent des annotations, manipulent des documents sur le Web et laissent des traces de leurs passages, etc. Par exemple, les *folksonomies*¹ représentent des informations d'une grande importance. La plupart des réseaux

sociaux RS sont modélisés par des structures de graphe social dont les nœuds sont les utilisateurs du réseau et les arcs représentent les relations entre ces utilisateurs.

Dans le cas des *folksonomies*, la structure sous-jacente peut être considérée comme un graphe tripartite : utilisateurs, annotations et nœuds de ressources.

On observe dans les RS l'existence de ces informations qui sont représenté comme suit

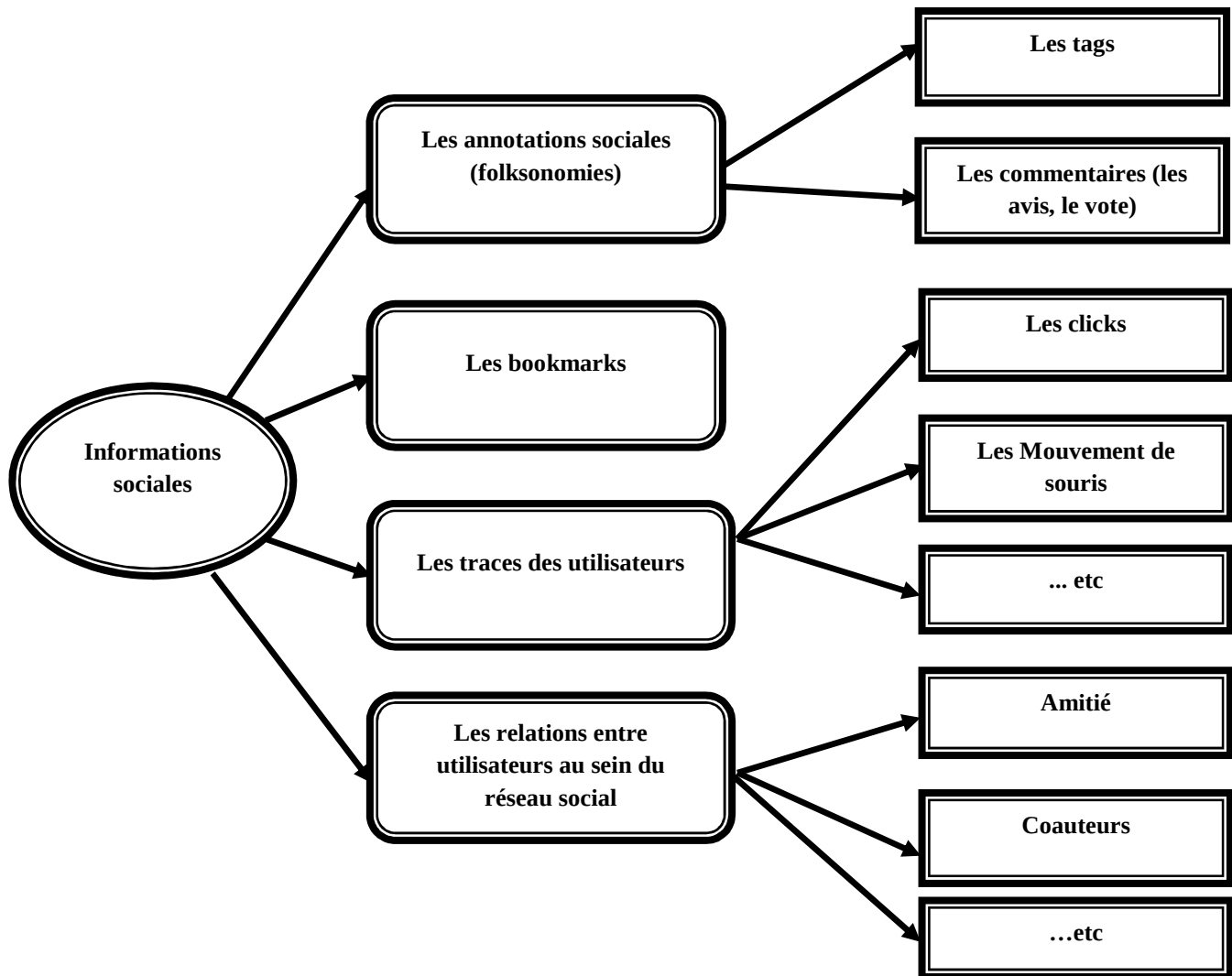


Figure I.2 représentation des informations sociales

¹ le terme *folksonomy* est une combinaison de «folk» et «taxonomy», décrivant le phénomène de classification sociale, les ressources, les utilisateurs, les tags, ainsi que l'activité d'un individu. La folksonomie est constituée par l'effort collectif des utilisateurs qui ont diverses connaissances et besoins quand ils interagissent avec un système de *social bookmarking*.

D'une manière générale, ces informations sociales peuvent être exploitées pour incorporer le domaine d'intérêt de l'utilisateur dans la RI sociale (ex. à partir de ses avis sur un thème précis) mais aussi pour améliorer l'évaluation des ressources sur le Web par rapport à un thème donné suite aux retours positifs que donne l'utilisateur.

Elles peuvent être prises en compte dans un modèle de RI sociale à différents niveaux, comme montre le schéma suivant :

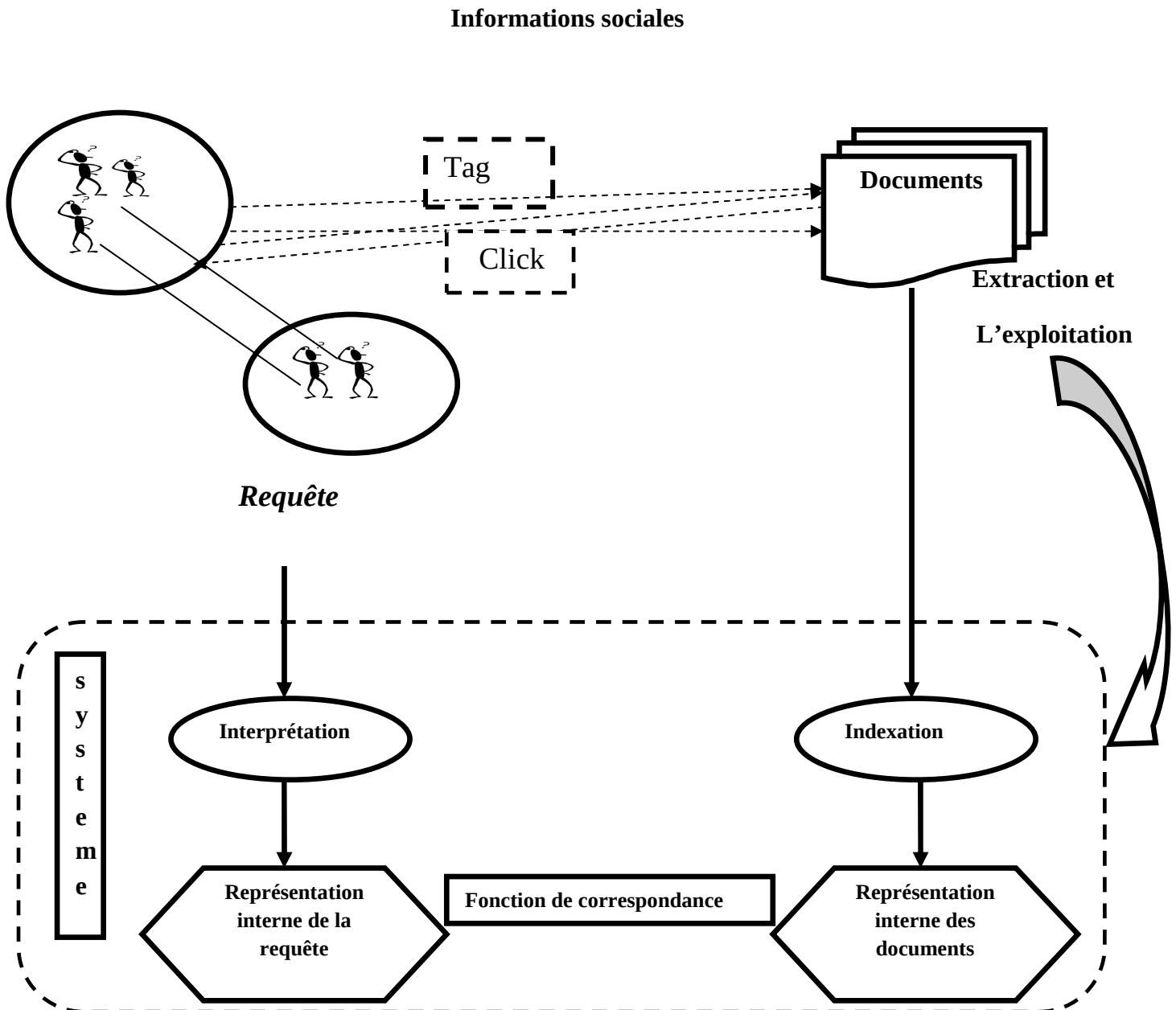


Figure I.3 *Modèle de RI sociale : l'utilisateur producteur et consommateur d'informations.*

I.2.2.3 Un Modèle du Domaine pour la RI Social

Les modèles traditionnels pour la recherche d'information concernent eux-mêmes des documents, des requêtes, et leurs relations de l'un à l'autre: Un document est pertinent à une requête, un document, référence autres documents, une requête est semblable aux autres requêtes. Également, l'analyse du réseau sociale modélise des individus et leurs relations de l'un à l'autre: amis et famille, connaissances, collaborateurs. Les Systèmes de recherche d'information traditionnelle ne modélise pas les individus, ni dans leur rôle comme les utilisateurs du système, ni comme auteurs des documents retrouvés, et les réseaux sociaux n'incorporent pas le contenu recouvrable.

La RI social combine les deux modèles avec l'un à l'autre. En incorporant des individus dans le modèle, nous gagnons une plus grande perspicacité dans leur rôle dans la recherche d'information et le processus de production. Les Nouvelles associations entre les entités deviennent apparentes: Les individus paraissent dans leur rôle comme les producteurs ou consommateurs de l'information, les requêtes sont en rapport avec les besoins en information d'un individu, ou décrit un sujet dont lequel l'individu possède la connaissance.

Un système de RI social est caractérisé par la présence de tous ces trois types d'entités: documents, requête, et individus. La plupart des systèmes utiliseront seulement un sous-ensemble possible d'associations entre les entités, selon le domaine du système. La Modélisation des relations entre les individus est obligatoire pour un système de RI social; tous les autres types d'associations sont facultatives, aussi long que toutes les trois entités qui ont une association avec au moins un autre.

La motivation pour la RI social a enraciné dans la croyance qu'un producteur de l'information et son produit ne peut pas être séparé: L'information ne saute pas dans l'existence spontanément.

I.3 Les modèles de la recherche d'information sociale

Il existe plusieurs modèles dans la recherche d'information sociale et parmi ces modèles on trouve les modèles les plus cités :

1. Modèle1 de ressources bibliographiques

Dans ce modèle une nouvelle approche est proposée, basée sur les réseaux sociaux, pour l'accès aux ressources bibliographiques. Il introduit un modèle d'information sociale

dont les auteurs sont les principales entités et les relations sont extraites à partir des liens de coauteur et de citation. En effet, ces relations sont pondérées tenant compte des interactions entre les auteurs et des annotations sociales produites par les utilisateurs. Dans ce modèle, la pertinence d'un document est estimée par combinaison de la pertinence thématique et de la pertinence sociale, qui est à son tour dérivée de l'importance sociale des auteurs associés. La viabilité de ce modèle est évaluée sur une collection d'articles scientifiques dont les annotations sociales sont extraites depuis le réseau social académique **CiteULike.org**. Les résultats obtenus montrent la supériorité des performances de modèle par rapport à la recherche d'information traditionnelle.

2. Le modèle de recherche de tweets

Le modèle proposé est un modèle social pour la recherche de *tweets* basé sur les réseaux bayésiens de croyance. Ce modèle intègre trois sources d'évidence qui consiste en l'évidence thématique, l'évidence temporelle et l'évidence sociale. La pertinence thématique est estimée selon une fonction de fréquence *tf* modifiée. Ainsi, la présence de l'ensemble des termes de la requête est favorisée. La pertinence temporelle des *tweets* est déduite de la distribution des termes dans les fenêtres temporelles.

La pertinence sociale est estimée par l'autorité du blogueur correspondant dans le réseau social de *tweet*. L'évaluation expérimentale menée sur la collection TREC Microblog 2011 montre que la combinaison de ces différentes sources d'évidence permet de mieux évaluer la qualité des *tweets*. Les résultats obtenus dépassent la médiane des résultats officiels avec un gain de 24%.

3. Le Modèle de microblogs

Les principales entités et relations du réseau social de Twitter ont été modélisées. En outre, il propose de représenter l'influence au sein du réseau à travers les relations de rediffusion des messages et il introduit un modèle de recherche d'information sociale qui évalue la pertinence en tenant compte de l'influence du blogueur et de son expertise par rapport au sujet de la requête. Plus précisément, son modèle combine la pertinence thématique des publications avec un premier score d'influence sociale obtenu par l'application de l'algorithme PageRank sur le réseau social de rediffusion et un deuxième score d'expertise obtenu par un modèle de langue. Il a évalué l'efficacité de son modèle sur une collection de 237000 microblogs

publiés sur Twitter. Les résultats obtenus montrent l'intérêt de l'intégration de la structure du réseau dans l'évaluation de l'importance des bloggeurs.

I.4 Conclusion

Nous avons vu dans le présent chapitre la définition donnée à la recherche d'information sociale RIS ou ils décrits aussi les différentes informations sociale qui peut être bénéfique dans le processus de la RI, et parmi toutes ces informations sociales, nous nous intéressons aux tags que nous allons utiliser pour améliorer la RI classique. Ces tags sont considérés dans le cadre des réseaux sociaux comme des intérêts sociaux des utilisateurs.

La détection des centres d'intérêts d'un utilisateur permet de les exploiter dans le processus de la recherche d'information sociale. Le chapitre suivant portera sur les intérêts sociaux.

Chapitre II

Les intérêts sociaux

II.1 Introduction

L'intérêt stratégique porté à l'information sous ses différentes facettes, combiné à l'avènement explosif d'Internet et autres services de l'information sont des facteurs déterminants qui justifient la multiplication de directions de recherche ayant pour objectif de mettre en œuvre des processus automatiques d'accès à l'information, sans cesse plus performants. Pour aider les utilisateurs à naviguer sur le web, les chercheurs avaient développé des techniques intelligentes pour établir des besoins d'utilisateur basés sur les pages web qui sont à l'intérêt de différents utilisateurs.

La détermination des intérêts d'un utilisateur peut être effectuée explicitement en demandant aux utilisateurs de les exprimer, ou implicitement en observant leurs comportements. Les indicateurs implicites sont habituellement moins précis que les indicateurs explicites. Cependant, les indicateurs implicites n'exigent aucune heure ou effort supplémentaire de l'utilisateur et peuvent s'adapter aux changements des intérêts de l'utilisateur avec le temps. Pour mesurer implicitement l'intérêt de l'utilisateur il faut identifier les indicateurs implicites.

II.2 La notion d'intérêt

Nous donnerons ci-dessous quelques définitions autour de la notion d'intérêt :

➤ Définition¹

Un intérêt est ce qui convient, ou procure un avantage à quelqu'un. Il peut être une attention, une curiosité ou bien ce qui importe ou convient à son utilité. C'est aussi le profit tiré par un prêteur, sous la forme d'une rémunération de l'argent prêté à un emprunteur. Il peut être aussi un attrait pour quelque chose qui veille et captive l'esprit.

Dans notre étude, nous allons nous intéresser au centre d'intérêts dans la recherche d'information sociale au sein des réseaux sociaux.

➤ Définition d'un centre d'intérêt en RI

Le centre d'intérêt est l'ensemble des besoins (récurrents) en information de l'utilisateur où il exprime son domaine d'expertise, ses connaissances ou son périmètre

¹ Selon http://www.toupie.org/Dictionnaire/Interet_general.htm

d'exploration au cours de ses sessions de recherche ; cours de ses sessions de recherche. Il peut être défini par un ensemble de mots clés (concepts) ou un ensemble d'expressions logiques (requêtes). Dans de nombreuses approches l'importance de chaque concept est définie par une pondération des mots clés du centre d'intérêt. Une ontologie² de domaine peut compléter la définition du centre d'intérêt en explicitant la sémantique de certains termes. Le centre d'intérêt peut être 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.

Le centre d'intérêt peut être corrélé avec les données personnelles et s'enrichir par déduction de certaines informations.

II.3 Typologie des centres d'intérêt

Les centres d'intérêts des utilisateurs sont sous deux formes, intérêts implicites et intérêts explicites.

II.3.1 Intérêt implicite

Les intérêts implicites sont indiqués ou sont déduite par les interactions antérieures des utilisateurs dans le système avec divers types d'information, tels que le contenu d'utilisateur qu'il créé ou le contenu des pages web, des documents et des emails.

II.3.2 Intérêt explicite

Les intérêts explicites sont basée sur une collecte d'informations directement fournies par l'utilisateur via l'interface du système, car les utilisateurs expriment leurs intérêts explicitement en les spécifiant dans leur profile.

II.4 Les indicateurs d'intérêts

Les indicateurs d'intérêts sont nombreux. On trouve ceux qui résultent des usages au sein des réseaux sociaux (les traces d'usages telles que les clics souris), ceux-là sont extraits

² **Une ontologie:** est un ensemble de termes du savoir qui inclut le vocabulaire, les interconnexions sémantiques ainsi que les règles d'inférence et la logique d'un domaine particulier.

en utilisant des applications dédiées. D'autres qui sont explicites et déposés par l'utilisateur lui-même, on trouve dans cette catégorie, les tags sociaux qui servent de descripteurs sur des informations, les thématiques (les topics) d'intérêt des usagers ainsi que les bookmarks créés afin de classer leurs favoris. La figure suivante montre une classification de ces indicateurs qui ne se veut pas exhaustive.

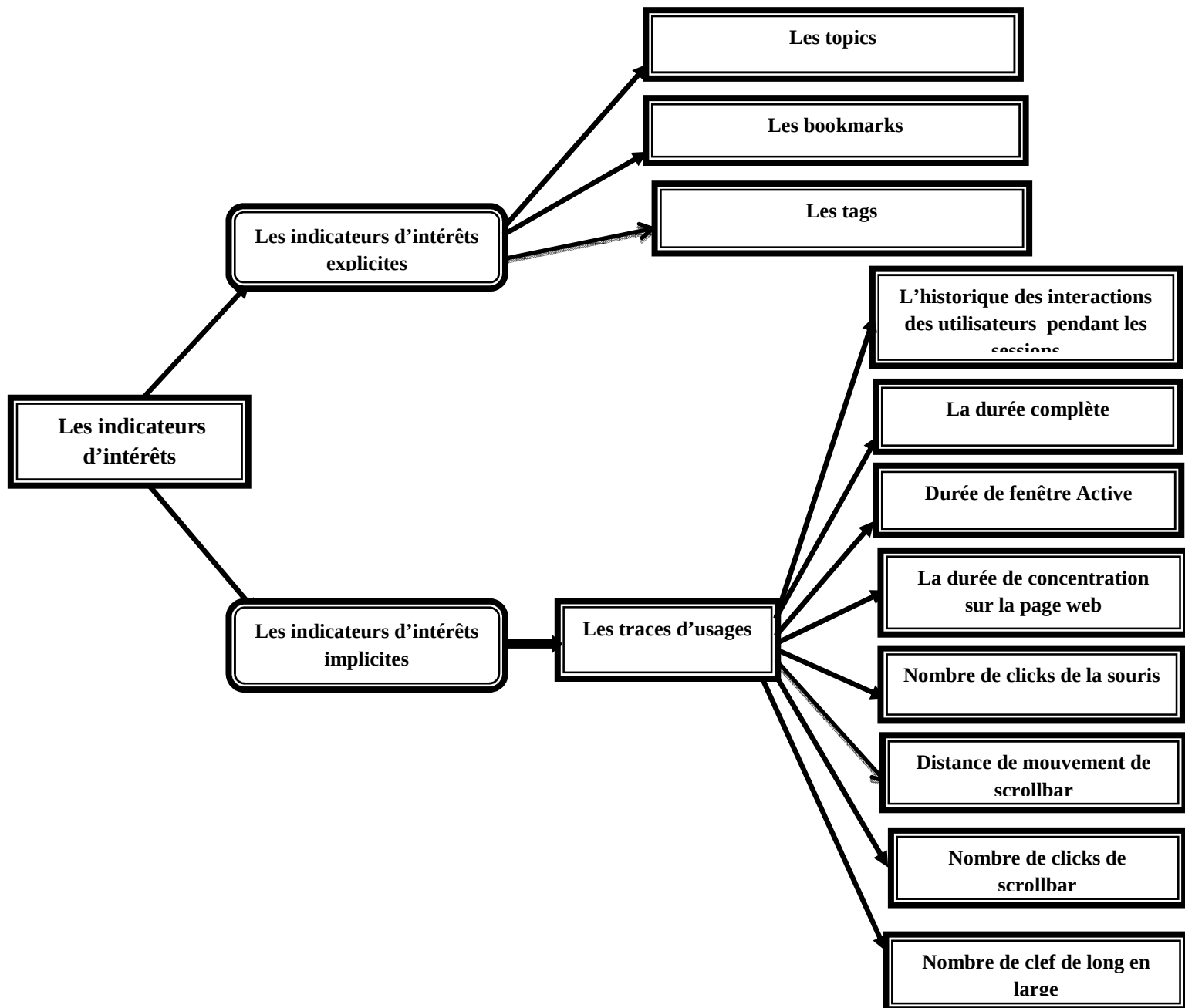


Figure II.1 Les indicateurs d'intérêts.

II.4.1 les indicateurs d'intérêts explicites

Ces indicateurs d'intérêts peuvent être exprimé par la spécification de la thématique d'intérêt, par l'action du tagging, par le mouvement des yeux ou bien sont exprimés à partir du nombre de clicks sur un lien ou une page.

➤ **les topics** (la thématique ou le sujet)

La thématique d'intérêt de l'utilisateur est souvent spécifié dans son profile exprimant ainsi soit son domaine d'expertise, sa spécialité, son domaine de recherche

➤ **Les bookmarks** (Appelés aussi les **signets** ou les **marques-pages**)

Cette notion a été adoptée aussi dans nos jours avec un sens commun. Les bookmarks sont utilisés comme des « espaces d'information personnelle de stockage de l'information du web » pour aider les gens à se rappeler et à récupérer des pages Web. Un utilisateur habituellement signe des pages web pour les visiter encore plus tard. On lit un article, un billet, une étude, une vidéo, quoi que ce soit qui semble avoir un minimum d'intérêt on le sauvegarde. Les utilisateurs signent les pages web qui sont intéressantes [Li *et al*, 1999] . Les utilisateurs sauvegardent des pages Web importantes dans leur disque dur en utilisant l'ordre “ Comme Sauvegarder ”. Cela implique aussi que ces pages Web sauvegardées sont intéressantes aux utilisateurs [Liber, 1995].

Les bookmarks représentent des outils simples pour construire des sous-ensembles d'information personnalisés où des pages Web identifiées par des URLs utiles ou intéressantes peuvent être stockées afin de servir pour une utilisation ultérieure. Les utilisateurs gardent la trace des liens vers des pages en créant une liste des bookmarks.

Plus de 100 sites du réseau social se trouvent sur Internet qui permet de signets des pages web tel qu'on trouve CiteULike, BibSonomy et Connotea. Ainsi, on a choisi comme exemple de définir le site Del.icio.us.

❖ *Del.icio.us*

Le site du réseau social Del.icio.us a été développé par Joshua Schachter en 2003 et acquis par Yahoo ! en 2005. Pour sauvegarder un bookmark, l'utilisateur saisit l'URL du site Web qu'il veut conserver, L'individu doit saisir les informations suivantes qui concernent l'URL :

le titre, un ou plusieurs tags (défini juste après) qu'il considère spécifiques pour la ressource, des notes (commentaire). Il choisit aussi le niveau de visibilité du bookmark : privé ou public

La page d'accueil (voir la figure II.2 ci-dessous) de Del.icio.us affiche les bookmarks les plus récents qui ont été ajoutés, les tags qui leur sont associés, les utilisateurs qui les ont créés, et le nombre des personnes ayant ces bookmarks en commun. Sur la page d'accueil, il y a aussi un lien vers les bookmarks populaires (présentés notamment avec les auteurs qui les ont créés) et un lien vers les pages des autres utilisateurs. Le lecteur peut sélectionner les utilisateurs qui correspondent à un tag donné.

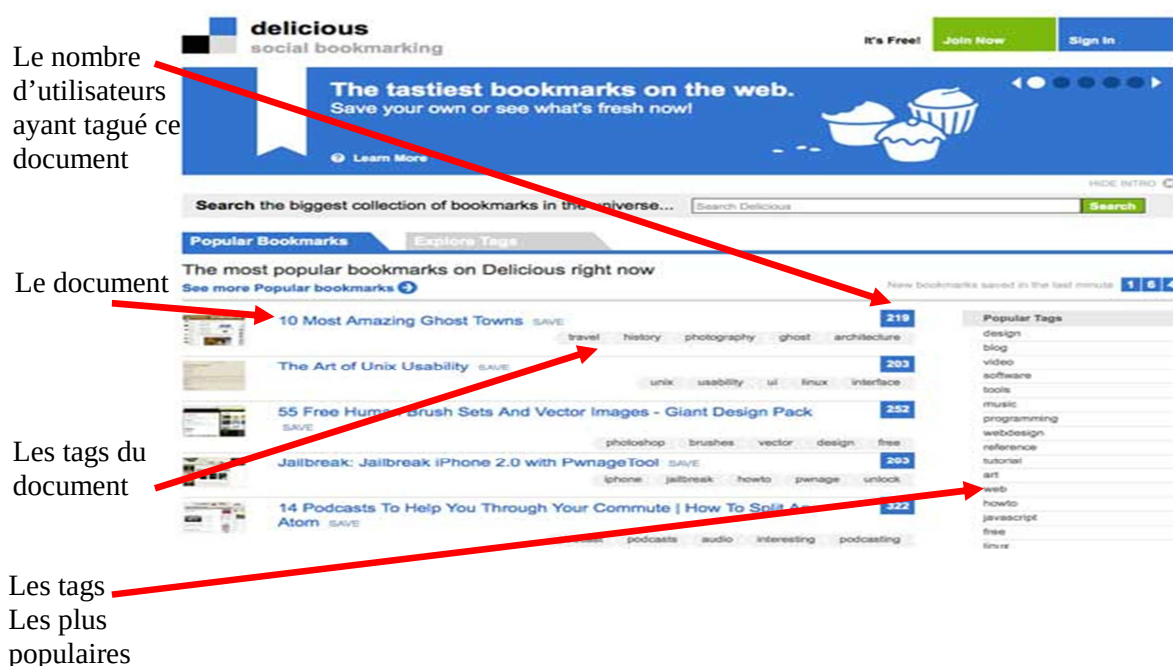


Figure II.2 page d'accueil du site del.icio.us.

Grâce aux pages personnelles et aux pages populaires, les individus ont une idée de ce en quoi les autres individus sont intéressés. En naviguant sur différents tags et personnes, les utilisateurs peuvent trouver des ressources qui présentent un intérêt pour eux. En plus, ils peuvent trouver des personnes qui ont des intérêts communs [Golde, 2005].

L'accès aux bookmarks dans un site de réseau social réalisé par d'autres personnes qui ont les mêmes intérêts que soi ;

– la possibilité de consulter les bookmarks des amis et de les ajouter dans sa propre liste (Bibliothèque de bookmarks) ;

– taguer avec des mots spécifiques pour faciliter la recherche d'un article qu'il a ajouté ou qui ont été rajoutés par la communauté.

➤ **Les Tags**

Le terme « tag » (en français « étiquette ») représente un mot-clé ou une expression associée ou assignée aux ressources. Il décrit ainsi l'objet et lui permet d'être retrouvé par navigation, par filtrage ou par recherche. L'activité des individus consistant à attribuer des métadonnées aux ressources s'appelle «tagging» [Golde, 2005]. L'ensemble des tags d'un individu forme une collection qui s'appelle « personomy »[Hotho et al,2006]. L'ensemble des personomies constitue la «folksonomy».

Les tags, sont utilisés pour annoter des bookmarks, des pages Web, des images, etc. Ce type d'informations peut être considéré comme un avis de l'utilisateur, généralement positif, à propos des différentes ressources annotées. Il est aussi envisageable d'exploiter ces données pour en extraire des informations thématiques à propos des ressources annotées (ex : ressources liées à un domaine d'intérêt).

Plus un document s'y est taggué plus l'utilisateur s'intéresse, et vice versa, un document qui n'a aucun tag peut signifier qu'il n'intéresse personne.

II.4.2 Indicateurs d'intérêt implicites

Ils sont essentiellement basés sur les traces d'usages, parmi ces traces on trouve le nombre de clicks sur un lien ou une page, la durée de fenêtre active, la durée de concentration sur la page web, la distance de mouvement de scrollbar, le nombre de clicks de scrollbar, les mouvements de souris, les pages consultés, le temps de consultation d'une page.

• L'historique des interactions des utilisateurs pendant les sessions

Les interactions passées de l'utilisateur représentent une source pour prédire ses intentions et lui recommander des objets. Cette dimension contient ce qu'on appelle communément le 'feedback' de l'utilisateur. Elle regroupe l'ensemble des informations collectées sur le comportement de l'utilisateur, que ces informations soient directement fournies par lui (feedback explicite) ou qu'elles soient récupérées ou dérivées à son insu (feedback implicite). 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.

- **la durée complète**

L'utilisateur peut tendre à passer plus de temps sur les pages qu'il trouve intéressante, ainsi la durée dépensée sur une page Web peut être enregistré. La durée complète est définie comme l'intervalle de temps entre le moment où un utilisateur ouvre et laisse une page Web. Quelques pages web contiennent beaucoup d'images qui retardent le temps de téléchargement, ainsi nous commençons à mesurer la durée après que la page entière soit chargée.

De cette manière, la durée complète ne sera pas affectée par la vitesse de raccordement, la quantité du trafic d'Internet, ou la vitesse d'unité centrale de traitement. La durée complète pour une page Web peut être calculée en soustrayant la période de finir le téléchargement de la page Web courante dès laisser la page Web. La durée complète est différente de la durée employée par [Jung, 2001]. Sa durée inclut la période de téléchargement d'une page Web.

$$\text{La durée_complète} = \text{periode_laisser_page_web} - \text{periode_finir_telecharg_page_web}$$

- **Durée De Fenêtre Active**

La plupart des systèmes d'exploitation modernes permettent au utilisateur la multitâches, ou courent plusieurs applications en même temps. Un utilisateur peut écrire un rapport ou chatter tout en passant en revue une page Web. Ces autres applications peuvent être indépendantes du contenu d'une page Web. Si un utilisateur passait une heure écrivant un papier de travail avec un web browser réduit, la durée complète de la page Web pourrait être une heure.

C'est très probable de fournir des indications incorrectes d'intérêt d'utilisateur. Afin d'éviter d'être affecté par ce problème, nous déterminons si un navigateur web est actif ou pas. Le temps qu'un navigateur web est inactif est soustrait de la durée complète. Nous appelons cette durée la durée de fenêtre active depuis que nous comptons seulement le temps quand un navigateur web est actif.

$$\text{Durée_de_fenetre_active} = \text{la durée_complete} - \text{le temps_navigateur_web_inactif}$$

- **la Durée de concentration sur la page web**

Les utilisateurs ne lisent pas toujours de page Web quand le navigateur web est actif. Ils peuvent parler facilement aux amis ou ayant une pause-café, pendant que le navigateur web est actif. La durée de la fenêtre active peut être facilement plus de 30 minutes si un utilisateur laisse le navigateur actif et va pour une pause-café. Nous pouvons être capables de détecter l'absence de l'utilisateur en détectant l'action de mouvement de la souris. Cependant, une meilleure solution est d'utiliser un appareil-photo qui détecte l'orientation du visage d'un utilisateur. Un appareil-photo peut vérifier même si un utilisateur regarde le navigateur web ou si son attention est détournée.

Cette durée sera plus exacte que la durée de la fenêtre active quant à vérifier l'attention de l'utilisateur d'une page Web. Depuis que cette durée compte le temps qu'un utilisateur regarde le navigateur web, nous l'appelons apparence à lui la durée (look at it duration). Regarder à lui la durée peut être calculée en soustrayant le temps quand un utilisateur ne regarde pas le navigateur de la durée de fenêtre active.

$$\text{Durée_de_concentration} = \text{durée_de_fenetre_active} - \text{durée_de_deconsentration}$$

- **Distance de Mouvement de la Souris**

La plupart des utilisateurs déplacent leurs souris en lisant le contenu d'une page Web. Le mouvement de la souris peut se produire en regardant une image intéressante, ou quand en pointant à des objets intéressants. Nous supposons que plus la souris se déplace, plus un utilisateur s'y'intéresse à la page Web. Cet indicateur a aussi été examiné par [Jung, 2001]. la distance écrite ici est un petit peu différente du sien dans un sens de détecter le mouvement de la souris total. Il a compté seulement sur le mouvement de la souris quand le point de la souris est à l'intérieur du navigateur actif. La distance de mouvement de la souris est détectée par ses coordonnées x et y sur un moniteur chaque 100 milliseconde. La formule est :

$$\text{Mouvement_Souris}(\text{pixels}) = \sum_{i=1}^{t-1} \text{Dist}(\mathbf{P}(t_i) - \mathbf{P}(t_{i-1}))$$

Où le chronomètre t est la durée de la fenêtre active, l'intervalle de temps, $t_i - t_{i-1}$, est **100** millisecondes, $\mathbf{P}(t_i)$ est un emplacement de la souris par apport à l'axe des x et des y au temps t_i , et la fonction **Dist** est une distance Euclidienne.

• Nombre de clicks de la Souris

Les personnes utilisent le “click ” pour lier un hypertexte à une autre page Web. De plus, cliquer peut être considéré comme un comportement habituel [Jung, 2001]. Cliquer peut être une façon d'exprimer nos émotions telles que la joie de trouver un produit qu'on cherchait (par exemple, un livre), alors on peut cliquer à maintes reprises l'objet plusieurs fois. Cet indicateur a été examiné dans Kixbrowser, Curious Browser, Goeck Browser [Goec et al, 2000] et Letizia [Liber, 1995].

Nous utilisons l'hypothèse, que plus le nombre de click de la souris sur une page Web est grand, plus un utilisateur est intéressé [Jung, 2001]. Le nombre de clicks de la souris est compté chaque fois qu'un bouton de la souris est cliqué.

• Distance de Mouvement du Scrollbar

Un utilisateur peut défiler aussi de long en large une page Web en traînant un scrollbar. Ces événements traînants peuvent avoir lieu plusieurs fois pendant qu'un utilisateur lisant une page Web. La distance de mouvement du scrollbar pour un moment E , peut être calculé en mesurant le mouvement de la souris chaque **100** millisecondes. En additionnant toutes les distances de mouvement du scrollbar pour tous les moments, la distance d'un mouvement du scrollbar pour une page Web peut être calculée. La formule est :

$$\text{Mouvement_Scrollbar}(\text{pixels}) = \sum_j^E \sum_{i=1}^{E(j)-1} |P(t_i) - P(t_{i-1})|$$

Où E est la durée que le scrollbar est pressé, le temps $E(j)$ est la durée que le scrollbar est traîné dans un seul événement traînant, et $t_i - t_{i-1}$, est **100** millisecondes. Nous supposons que plus le mouvement du scrollbar est grand est correspondu avec plus d'intérêt de l'utilisateur dans une page Web.

• Nombre de clicks de Scrollbar

La longueur de beaucoup de pages Web est plus longue que la hauteur d'un moniteur. Si un utilisateur trouve une page Web intéressante, il peut lire la page Web plus bas, ainsi un utilisateur peut défiler en bas d'une page Web en cliquant ou en traînant le scrollbar. Ces événements sont comptés séparément. Le nombre de clicks du scrollbar est compté chaque fois qu'un utilisateur click un scrollbar. Un utilisateur défile de long en large une page Web

en augmentant, le nombre de clicks du scrollbar. [Jung, 2001] [Goec *et al*, 2000], et [Clayp *et al*, 2000] cet événement est mesuré et rapporté que c'est un bon indicateur. Nous supposons que nous voulons aussi découvrir que le nombre de clicks du scrollbar est correspondu avec l'intérêt d'un utilisateur dans la page Web.

- **Le nombre de Clef de long et de large**

Pour défiler une page Web, quelques utilisateurs utilisent les clefs “en haut” et “en bas” au lieu du scrollbar (les flèches qui se trouvent sur les extrémités du scrollbar). Cet indicateur est semblable au nombre de clicks du scrollbar et la distance de mouvement du scrollbar. L'hypothèse est que plus le nombre de fois qu'on presse ces clefs est élevées, plus un utilisateur s'y'intéresse à la page Web. Cet événement est mesuré en augmentant à chaque fois un compteur quand un utilisateur commence en bas de clefs. Curious Browser [Clayp *et al*, 2000] et [Jung, 2001] ont mesuré les activités du clavier mais ils n'ont pas mesuré de long en large la clef pour mesurer le mouvement du scrollbar.

II.5 la détection des centres d'intérêt

La détection implicite des préférences et des centres d'intérêt se fait à travers les comportements observables collectés par le système lors de l'interaction de l'utilisateur avec l'environnement (session de travail, consultation d'un document,...). Dans ce contexte plusieurs comportements peuvent être examinés, et l'observation de ces comportements s'effectue à l'aide d'indicateurs implicites :

II.5.1 le comportement de lecture

C'est un comportement trivial, considéré comme le premier observable lors de l'activité de recherche de l'utilisateur. En effet, si un document a été lu, ceci indique qu'il a suscité un certain intérêt chez l'utilisateur. Ainsi la connaissance de l'action de lecture d'un document permet de se prononcer sur la susceptibilité de sa pertinence. Parmi les facteurs qui permettent de capturer ce comportement nous avons :

- ✓ **La durée de lecture**

Est considéré comme le plus important des indicateurs de comportement de lecture. Le temps effectif que prend l'utilisateur pour lire les documents dépend fortement de la longueur

des documents. Dans le sens, où plus le document est long plus la valeur de cet indicateur est grande. Cependant, si cette valeur est élevée pour les documents longs elle ne traduit pas un intérêt plus grand par rapport au document de plus petite taille.

✓ nombre de click de la souris

Plus le nombre est grand plus le document est susceptible d'intéresser l'utilisateur.

✓ Les mouvements de la souris

Permettent de détecter la présence d'une activité sur une page donnée. Le mouvement de la souris se mesure par la distance parcourue de la souris sur l'écran de l'utilisateur.

✓ Le mouvement de scroll bar

L'utilité de cet indicateur est que lorsque l'utilisateur juge qu'un document pertinent, il sera intéressé par le lire entièrement et en visualisant son contenu en le faisant défiler en utilisant la souris et /ou les différentes touches de direction de clavier (→←↑↓).

I.5.2 Le comportement de sauvegarde

La sauvegarde d'un document est un comportement de faible fiabilité pour indiquer l'intérêt de l'utilisateur s'il est considéré comme une action isolée.

On peut enregistrer un document par faute de temps. Néanmoins, en le combinant avec les autres comportements (lecture), il s'avère être un bon indicateur pour renforcer le degré d'intérêt à inférer.

Il n'existe qu'un seul indicateur à savoir *l'action de sauvegarde*.

II.5.3 Le comportement d'impression

Considérée isolément peut ne pas traduire un intérêt réel de l'utilisateur. De manière analogue au comportement de sauvegarde, l'impression sera combinée avec les deux autres comportements. À titre d'exemple si un document est lu et ensuite imprimé, ceci traduit l'intérêt suscité pour l'utilisateur.

L'indicateur existant est *la commande d'impression*.

II.6. La modélisation des centres d'intérêt de l'utilisateur

Dans le domaine de la recherche d'information RI, la modélisation des centres d'intérêts des utilisateurs peut être caractérisée par trois éléments clés [Tam, 2008] qui sont : la durée de vie des centres d'intérêt, les sources d'informations et le modèle de base pour la représentation.

– *Durée de vie des centres d'intérêt* : selon ce critère les centres d'intérêt à court terme des centres d'intérêt à long terme sont distingués.

- ✓ *Les centres d'intérêt à court terme* décrivent un intérêt de courte portée lié à la session de recherche en cours [Dum et al, 2003],[Shen et Zhai, 2005.] L'identification du centre d'intérêt requiert dès lors la définition d'heuristiques pour caractériser le basculement dans les sujets d'intérêt d'une session de recherche à l'autre [Srir et al, 2004], [Jans et al, 2006]. Ou tout simplement un centre d'intérêt à court terme (courant) peut être défini comme un besoin en information à court terme exprimé sur une courte période d'interactions avec le système de recherche d'information SRI ; il est représenté par un vecteur de termes pondérés, ordonnés par leur degré de représentativité, extrait à partir de la matrice représentative de l'historique.
- ✓ *Les centres d'intérêt à long terme* traduisent en revanche des sujets d'intérêt récurrents de l'utilisateur, des domaines d'expertise dans le cas d'application en RI dédiées [Tan et al, 2006], [Liu et al, 2004], [Sieg et al, 2007] Notons que des stratégies d'évolution sont généralement sous-jacentes aux stratégies de construction des centres d'intérêt à long terme. Autrement Un centre d'intérêt à long terme traduit un besoin en information récurrent observé à travers l'historique de recherche de l'utilisateur sur des sessions non nécessairement successives dans le temps.

– *Sources d'informations* : les sources d'information sont tout d'abord déterminées par l'approche de modélisation. En effet, la source peut être simplement l'utilisateur dans le cas d'une approche de modélisation explicite [Pazza et al, 1996] Autrement, dans le cas d'une approche implicite, la source peut être l'historique des clicks, les données de navigation [Teev et al, 2005], [Shen et Zhai, 2005], le mouvement des yeux [Joac et al, 2007] les pages et les sites favoris [Gowan , 2003] les applications locales [Dum et al,2003].

– **Modèle de base pour la représentation** : de très nombreux modèles ont été proposés pour la représentation des centres d'intérêt de l'utilisateur. Les plus simples sont ceux basés sur un vecteur de mots-clés [Pazza et al, 1996] un ensemble de vecteurs de mots-clés [Chen et al, 1998] des classes de mots-clés [Gowan, 2003]. Des modèles plus riches sont également proposés, basés sur les réseaux sémantiques [Gent et al, 2003] les hiérarchies de concepts [Begg et al, 1993] issus des documents consultés, ou par appariement avec une ontologie de référence [Liu et al, 2004]. Modélisation du contexte de recherche [Sper et Gauch, 2005], [Sieg et al, 2007].

II.6.1 Modèle de Wen

Ce modèle propose de modéliser le centre d'intérêts implicite d'un utilisateur :

- **Modèle d'Intérêt Implicite de l'Utilisateur**

Parce que le contenu contribué des utilisateurs révèle leurs intérêts implicites, les intérêts de l'utilisateur sont modélisés comme un ensemble de sujets latents extrait de leurs données de communication et contribue le contenu social. Il a utilisé (LDA) Latent Dirichlet Allocation, un modèle probabiliste génératif pour extraire les sujets. Donnant un corpus de document D , LDA modélise chaque document d comme un mélange fini sur un ensemble sous-jacent de sujets où chaque sujet **subj** est caractérisé comme une distribution fini de mots. Un paramètre postérieur Dirichlet $\gamma(d, subj)$ peut être associé avec le document d et le sujet **subj** pour indiquer la force de **subj** dans d . Comme résultat, le document d peut être réduit à un Vecteur $\vec{\gamma}_d = (\gamma(d, subj_1), \gamma(d, subj_2), \dots, \gamma(d, subj_T))$ où T est le nombre total de sujets. Empiriquement T est choisi $T = 1200$, dans l'ordre d'équilibrer le besoin de couvrir divers contenu et la complexité computationnelle.

Après l'extraction du sujet, les intérêts de l'utilisateur sont décrits par une matrice S d'ordre $U \times T$ ou U est le nombre total des utilisateurs utilisant les sujets. Un élément s_{ij} dans S dénote le degré d'intérêt de l' $i^{\text{ème}}$ utilisateur par le $j^{\text{ème}}$ sujet extrait. Tel que s_{ij} est calculé par l'agrégation des forces du sujet dans tous les contenus de l' $i^{\text{ème}}$ utilisateur.

Explicitement $s_{ij} = \sum_{d \in D_i} \gamma(d, subj_j)$ ou D_i est l'ensemble du contenu de l' $i^{\text{ème}}$ utilisateur et $\gamma(d, subj_j)$ est un paramètre de Dirichlet postérieur décrit la force du $j^{\text{ème}}$ sujet dans un document d . Après la normalisation de S par :

$$s_{ij} = \frac{s_{ij}}{\sum_j s_{ij}}$$

Finalement, Les recherches pratiques et les applications de recommandation peuvent demander une représentation basée sur un mot clé des top-N d'intérêts d'un utilisateur. Pour les top-N intérêts, l'utilisateur devrait être hautement intéressé à eux (basé sur s_{ij}).

De plus les top-N intérêts devraient être distincts de l'un à l'autre (basé sur le chevauchement de mot). Pour incorporer les deux contraintes, la fonction objective pour les top-N intérêts T_N est défini :

$$obj(T_N) = \frac{\sigma_1}{N} \sum_{j=1}^N s_{ij} - \frac{\sigma_2}{N^2} \sum_{j=1}^N \sum_{k=1}^N \cos(s_{ij}, s_{ik})$$

Où le s_{ij} est le $j^{\text{ème}}$ sujet, $\cos(s_{ij}; s_{ik})$ est la ressemblance du cosinus, les poids $\sigma_1 = \sigma_2 = 0.5$ mais peut être ajusté dans différents scénarios de l'application. Une méthode avide est utilisée pour sélectionner les top-N intérêts. Alors, pour chaque intérêt un top N_w des mots-clés est produit (par exemple, $N_w = 20$).

II.6.2 Le modèle de Tamine

➤ Méthodologie de construction des centres d'intérêt

Elle se base sur l'hypothèse qu'un utilisateur a divers centres d'intérêt et qu'il peut basculer d'un centre d'intérêt vers un autre en cours de sessions de recherche successives. Le procédé de construction est fondé sur un cycle de trois principales étapes qui sont : **L'historique de l'information issue des sessions de recherche, la détection du basculement dans le centre d'intérêt à court terme et l'apprentissage des centres d'intérêt à long terme** [Tam, 2008].

• Historique de l'information issue des sessions de recherche

Cette étape est réalisée en combinant l'importance intrinsèque de l'information portée par le contenu des documents jugés par l'utilisateur et l'importance relative de cette même information en tenant compte du profil courant de l'utilisateur. Pour cela, on définit formellement l'opérateur d'agrégation comme suit :

$$H^0(d, t) = S^0(d, t)$$

$$H^{s+1}(d, t) = H^s(d, t) \oplus S^{s+1}(d, t) = \begin{cases} \alpha * w_{t,d} + \beta * S^{s+1}(d, t) & \text{si} \\ & t \notin T(R_u^{(s)}) \\ \alpha * H^s(d, t) + \beta * S^{s+1}(d, t) & \text{si} \\ & t \in T(R_u^{(s)}) \text{ et } d \in R_u^{(s)} \\ H^s(d, t) & \text{sinon} \end{cases}$$

$$(\alpha + \beta = 1), s > s_0$$

La définition de l'opérateur d'agrégation $\oplus$ est fondée sur l'hypothèse que les termes associés aux centres d'intérêt de l'utilisateur sont récurrents.

L'idée est alors d'affiner les descripteurs des documents déjà jugés par :

- expansion éventuelle avec des termes associés présents dans des documents pertinents,
- combinaison de l'importance classique de ses termes (relativement à la collection de documents) et de leur pertinence relative au profil, calculée à l'aide du coefficient de pertinence d'un terme t dans un document d $CPT(t, d)$ en cours des sessions de recherche passées.

• **Détection du basculement dans le centre d'intérêt à court terme**

Dans le but d'atteindre cet objectif, un opérateur est appliqué qui détermine la corrélation thématique entre les centres d'intérêt à court terme extraits à deux périodes successives de l'historique de recherches de l'utilisateur. Le basculement dans le sujet couvert par la requête se traduit à notre sens, dans une redistribution de l'importance relative des termes dominants associés. Du point de vue formel, on exploite des opérateurs statistiques pour la mesure des corrélations des rangs des termes dans les descripteurs des centres d'intérêt à court terme.

• **Apprentissage des centres d'intérêt à long terme**

Les centres d'intérêt à long terme sont dérivés selon le principe de récurrence des centres d'intérêt à court terme. Formellement, définir un seuil statistique ΔI aux corrélations thématiques entre les centres d'intérêt, notées ΔI et l'utilisons comme base de détection de

nouveaux centres d'intérêt ou indicateurs d'évolution de centres d'intérêt déjà identifiés. Plus précisément, le procédé est comme suit :

- (a) $\Delta I > \mathbf{6}$. Les sessions de recherche sont inscrites dans le même contexte : pas d'indication sur l'évolution des centres d'intérêt de l'utilisateur.
- (b) $\Delta I < \mathbf{6}$. Détection d'un changement de contexte, deux configurations se présentent : découverte d'un nouveau centre d'intérêt ou évolution d'un autre préalablement découvert.

II.7 Maintenance des centres d'intérêt

La notion de maintenance, ici est prise dans le sens « maintenir dans le temps ». La maintenance porte aussi bien sur les centres d'intérêt à court terme que sur les centres d'intérêt à long terme. La maintenance des centres d'intérêt à court terme étant basée sur la récurrence du sujet de la requête à travers une session de recherche, nous procédons alors à la détection du basculement dans le sujet de recherche puis l'affinement du centre d'intérêt courant dans le cas où le basculement n'a pas été détecté. [TAM, 2008].

II.7.1 Détection du basculement dans le sujet de la requête

L'approche pour détecter ce basculement est basé sur la mesure de corrélation thématique appliquée entre le centre d'intérêt courant (dérivé à partir des requêtes précédentes), soit C^s , et le profil de la requête en cours d'évaluation, soit $q_c^{\vec{s}+1}$

La corrélation thématique entre le profil de la requête $q_c^{\vec{s}+1}$ et le centre d'intérêt C^s est ensuite calculée puis souillée. Une valeur inférieure à un seuil prédéterminé, signifie un basculement dans le sujet des requêtes. A l'inverse une valeur de seuil supérieure au seuil signifie que les requêtes adressent le même sujet général. La détermination de ces seuils critiques est effectuée de manière expérimentale.

I.7.2 Affinement des centres d'intérêt à court terme

Soient C^s et $q^{\vec{s}+1}$ les représentations sémantiques associées respectivement au centre d'intérêt d'une requêtes courants, la maintenance des centres d'intérêt est exprimée par des opérations de mise à jour de la représentation sémantique du centre d'intérêt courant en considérant la distribution des poids des concepts communs. Sur ce principe, on calcule le nouveau poids du concept c_j dans le centre d'intérêt à court terme C^s comme suit :

$$sw_{C^{s+1}}(C_j) = \begin{cases} \beta * sw_{q^{s+1}}(C_j) + (1 - \beta) * sw_{C^s}(C_j) & \text{si } C_j \in C^s \\ \beta * sw_{q^s}(C_j) & \text{sinon} \end{cases}$$

Où $sw_{C^s}(C_j)$ est le poids du concept C_j dans le centre d'intérêt C^s , $sw_{q^s}(C_j)$ est le poids du concept C_j dans q^s .

Outre l'évolution des centres d'intérêt à court terme véhiculés par les sessions de recherche, comme détaillée ci-avant, l'évolution des centres d'intérêt traduit, selon notre approche l'expression de la récurrence des centres d'intérêt à court terme pour faire émerger des représentations génériques des centres d'intérêt à long terme.

I.8 Evolution des centres d'intérêt

Il se base sur l'hypothèse qu'un utilisateur a divers centres d'intérêt et qu'il peut basculer d'un centre d'intérêt vers un autre au cours des sessions de recherche successives. Dans le but d'évaluer le degré de changement des contextes liés aux centres d'intérêt de l'utilisateur, on compare le contexte courant cc avec le contexte extrait à la période précédente pc comme illustré sur la figure II.3.

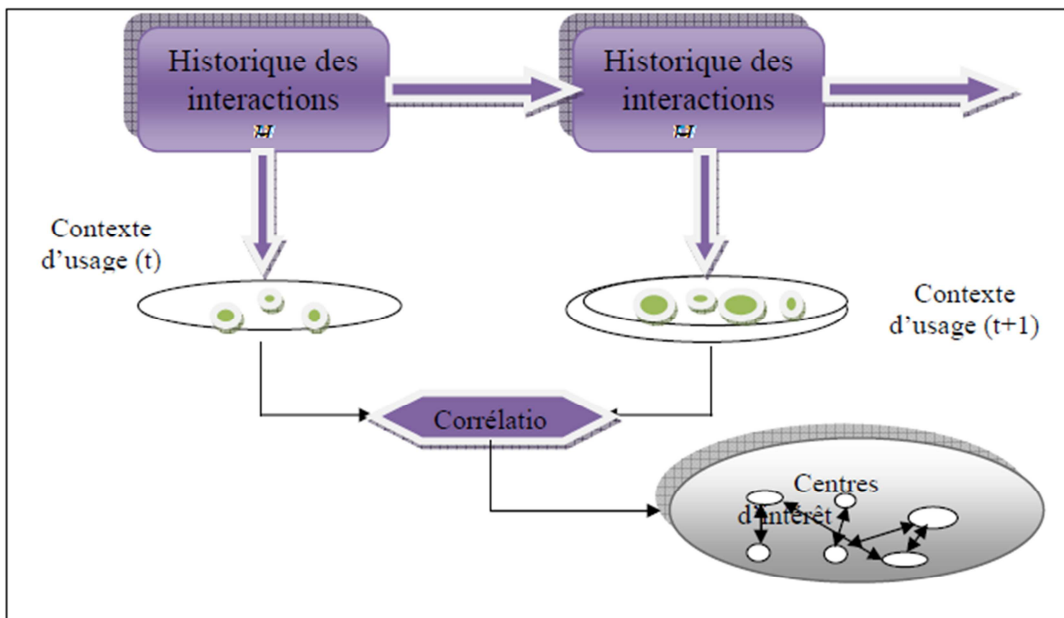


Figure II.3 : Principe d'évolution des centres d'intérêt.

L'objectif est de noter qu'on est davantage intéressé par l'évolution de l'ordre d'importance des termes dans la description des contextes d'usage extraits au cours des sessions, que par l'évolution de leur importance absolue. C'est pourquoi, une méthode statistique qui scrute le changement dans les centres d'intérêt à court terme est adoptée, en utilisant le coefficient de corrélation des rangs de Kendall.

Ce coefficient permet en effet de savoir si deux variables aléatoires évoluent dans le même sens ou dans des sens différents autrement dit, non corrélés. On définit alors le degré de ce changement est définit comme suit :

$$\Delta I = \frac{\sum_t \sum_{t'} S_{tt'}(pc) * S_{tt'}(cc)}{\sqrt{(\sum_t \sum_{t'} S_{tt'}(pc)^2) * (\sum_t \sum_{t'} S_{tt'}(cc)^2)}}$$

Où :

$$S_{tt'}(pc) = \text{signe}(pc(t) - pc(t')) = \frac{pc(t) - pc(t')}{|pc(t) - pc(t')|}$$

$$S_{tt'}(cc) = \text{signe}(cc(t) - cc(t')) = \frac{cc(t) - cc(t')}{|cc(t) - cc(t')|}$$

La valeur du coefficient ΔI est inscrite dans l'intervalle $[-1..1]$, où une valeur proche de -1 signifie que les contextes sont non similaires alors qu'une valeur proche de 1 signifie que les contextes sont proches sémantiquement.

L'évolution des centres d'intérêt, qui a pour effet la mise à jour éventuelle de la bibliothèque, est alors déterminée par le résultat de comparaison de ΔI relativement au seuil de corrélation σ . Plus précisément, la stratégie est la suivante :

1. $\Delta I > \sigma$. Les sessions de recherche sont inscrites dans le même contexte : pas d'indication sur l'évolution des centres d'intérêt de l'utilisateur;
2. $\Delta I < \sigma$. Détection d'un changement de contexte ; deux configurations se présentent : découverte d'un nouveau centre d'intérêt ou évolution d'un autre préalablement découvert.

I.9 Conclusion

Au cours de ce chapitre nous avons décrit les différentes intérêts sociaux, Nous avons en particulier exposé les principaux indicateurs d'intérêts, à savoir les indicateurs d'intérêts explicites et implicites, la détection et la modélisation des centre d'intérêts avec le modèle de Tamine et le modèle de Wen ainsi que la maintenance des centres d'intérêts et enfin on a terminé par l'évolution de ces derniers.

L'intégration des intérêts sociaux dans la recherche d'information sociale n'est traité par personne pour cela le chapitre qui suit décrit notre approche qui intègre les intérêts sociaux et plus particulièrement les tags sociaux dans le processus de la recherche d'information.

Conception

III.1 Introduction

Après avoir présenté dans les chapitres précédents, les différents concepts nécessaires à l'accomplissement de notre travail, nous passons maintenant à la partie principale où se situe notre contribution.

Notre objectif consiste en le développement d'une application basée sur l'intégration des intérêts sociaux dans le processus de la recherche d'information. Nous avons vu, dans le chapitre précédent que les intérêts des utilisateurs peuvent être représentés par différents types de méta-données (les tags, les traces d'usage, etc.). Nous choisissons, pour ce qui est de notre proposition, d'utiliser les tags sociaux car ils représentent le mieux les centres d'intérêts. Nous savons bien que pour qu'un utilisateur se donne de la peine pour taguer un document c'est bien parce qu'il y est intéressé. L'activité d'attribution de tags aux documents requiert un effort cognitif non négligeable. Ces tags sociaux ont peut faire bénéficier le processus de la recherche d'information et améliorer la liste des documents restitués pour un besoin donné en informations.

Nous allons représenter, dans ce qui suit, la structure globale de notre solution que nous détaillerons au fur et à mesure d'avancer dans la description.

III.2 Les objectifs de notre système

L'objectif principal de notre étude est d'améliorer la Recherche d'Information Classique qui connaît des limites essentielles parce qu'elle se base sur l'indexation traditionnelle qui manipule les documents comme des sacs de mots. Nous allons, dans notre proposition, tenir compte de l'aspect social de la recherche. À côté du document, nous allons manipuler les intérêts sociaux explicites (les tags sociaux) extraits des interactions des utilisateurs au sein de leur communauté virtuelle.

Les tags permettent aux utilisateurs de retrouver rapidement les documents qui répondent mieux à leurs besoins.

Afin d'atteindre ces objectifs notre système doit accomplir un certain nombre de fonctions qui seront détaillées dans les sections suivantes.

III .2.1 Description de notre approche

La solution que nous proposons pour améliorer les performances du système de recherche d'information classique, est une approche hybride qui permet, d'un côté,

d'appliquer un modèle de recherche traditionnel basé sur le calcul du score textuel des documents à base de la BM25. En second lieu, on évalue un score social basé sur les tags sociaux à base de la TF_IDF pour qu'ensuite, combiner par lissage ces deux résultats intermédiaires dans une formule globale.

La prise en charge des intérêts sociaux des utilisateurs se fait explicitement à travers les tags laissés sur leur passage (pendant leurs interactions sur leur réseau social). Ces tags sociaux, une fois assignés aux documents de la collection, sont récupérés à l'étape d'extraction en utilisant des outils spécifiques (disponibles en Open Source sur le web) puis insérés dans la base d'intérêts sociaux. Vu que les tags sont des mots clés (des termes) semblables aux termes des documents, on suggère de les traiter de manière identique au contenu textuel des documents, c'est à dire on évalue leurs poids selon la formule de la TF_IDF.

III.2.2 Architecture générale du système

Nous adaptons l'architecture suivante à notre système :

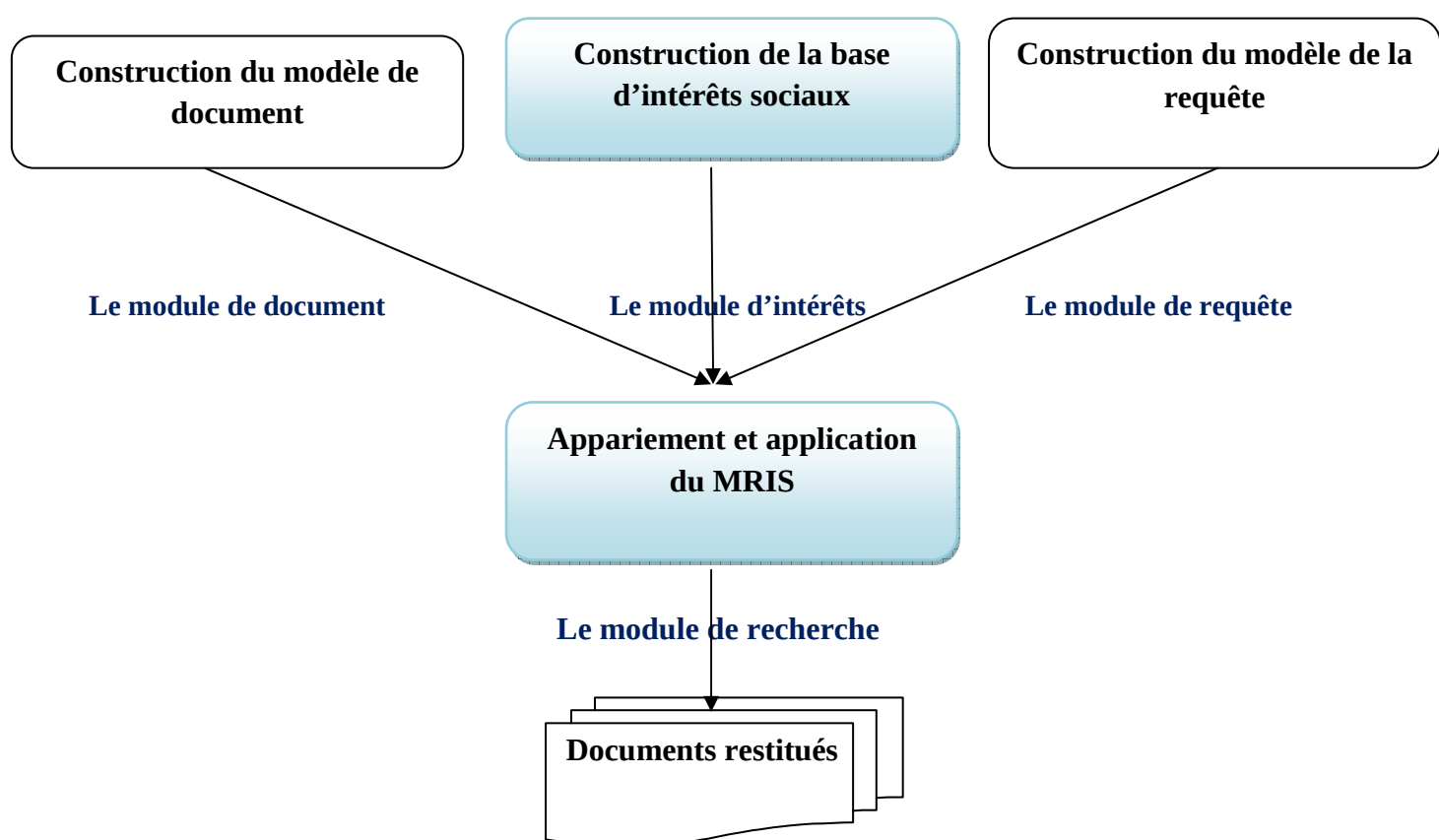


Figure III.1 : Schéma général de notre approche.

Notre système est composé des modules suivants :

- **Le module de requête** : qui sert à représenter la requête utilisateur.
- **Le module de document** : qui sert à indexer les documents.
- **Le module d'intérêts sociaux** : il s'agit d'extraire les tags sociaux qui sont assignés aux différents documents de la collection afin de construire la base d'intérêts sociaux.
- **Le module de recherche** : qui consiste à confronter les modèles de document, d'intérêts avec le modèle de la requête en calculant des scores intermédiaires (traditionnel et social) qui vont servir à évaluer le score de similarité global (le score final).

Notre contribution est au niveau des deux modules d'intérêt et de recherche.

III.2.3 Description des composants du système

Notre système est constitué de 03 modules de calcul des scores différents et 03 modules de représentation des données, comme montré sur *la figure III.2*.

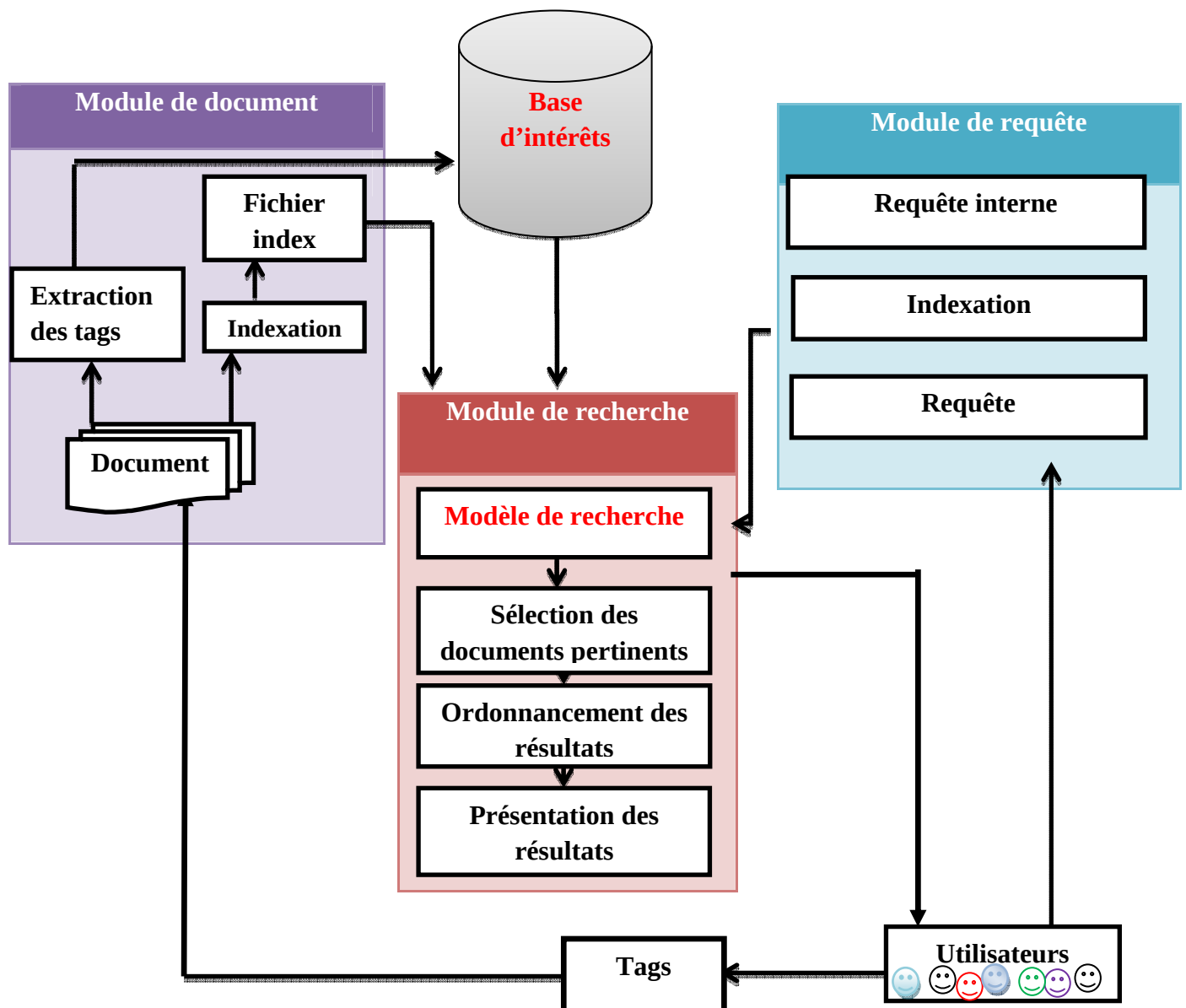


Figure III.2 : Architecteur générale de notre système.

III.2.4 Le module de requête

Ce modèle consiste en l'analyse de la requête afin de lui construire une représentation interne similaire à celles des documents. L'analyse de la requête est préalable à toute recherche effective dans la base.

Les requêtes que l'utilisateur peut formuler, sont composées de simples mots clés.

III.2.5 Le module de document

Il regroupe tous les traitements portant sur les documents de la collection, soit :

III.2.5.1 Le processus d'indexation

Le processus d'indexation s'occupe essentiellement de l'analyse des documents afin de les représenter sous forme concise, porteuse d'information et exploitable par les autres module.

La représentation des documents selon un langage d'indexation est un travail délicat, et pour ce faire, nous nous sommes basés sur l'indexation automatique qui permet de décrire les documents par un ensemble de termes paramétrés, susceptible de représenter leurs contenus, en vue de leur exploitation ultérieure. Les étapes de son processus sont :

- Extraction des mots simples.
- Elimination des mots vides.
- Normalisation.
- Pondération des termes de la liste.
- Stockage de la liste des termes sélectionnés.

Et pour cela nous avons utilisé la plate-forme terrier qui permet d'indexer une collection de document. Les principaux processus d'indexation de terrier sont les suivant:

➤ **Le processus d'indexation de terrier**

L'indexation dans terrier est divisée en quatre procédures et à chaque procédure des classes java peuvent être ajoutés pour la personnalisation du système.

Les quatre procédures sont :

1. Splitter la collection de document : consiste à parcourir l'ensemble du corpus reçu en entrée par Terrier et envoyer chaque document à l'étape suivante.
2. Extraction des termes (Tokenize Document) : qui consiste à parser chaque document reçu et en extraire les différents termes.
3. Traitement des termes extrait avec TermPipelines : consiste en l'élimination des mots vides et la lemmatisation des termes.
4. La construction de l'index.

Les différentes classes associées au processus d'indexation sont organisées dans un ensemble de packages dont on trouve :

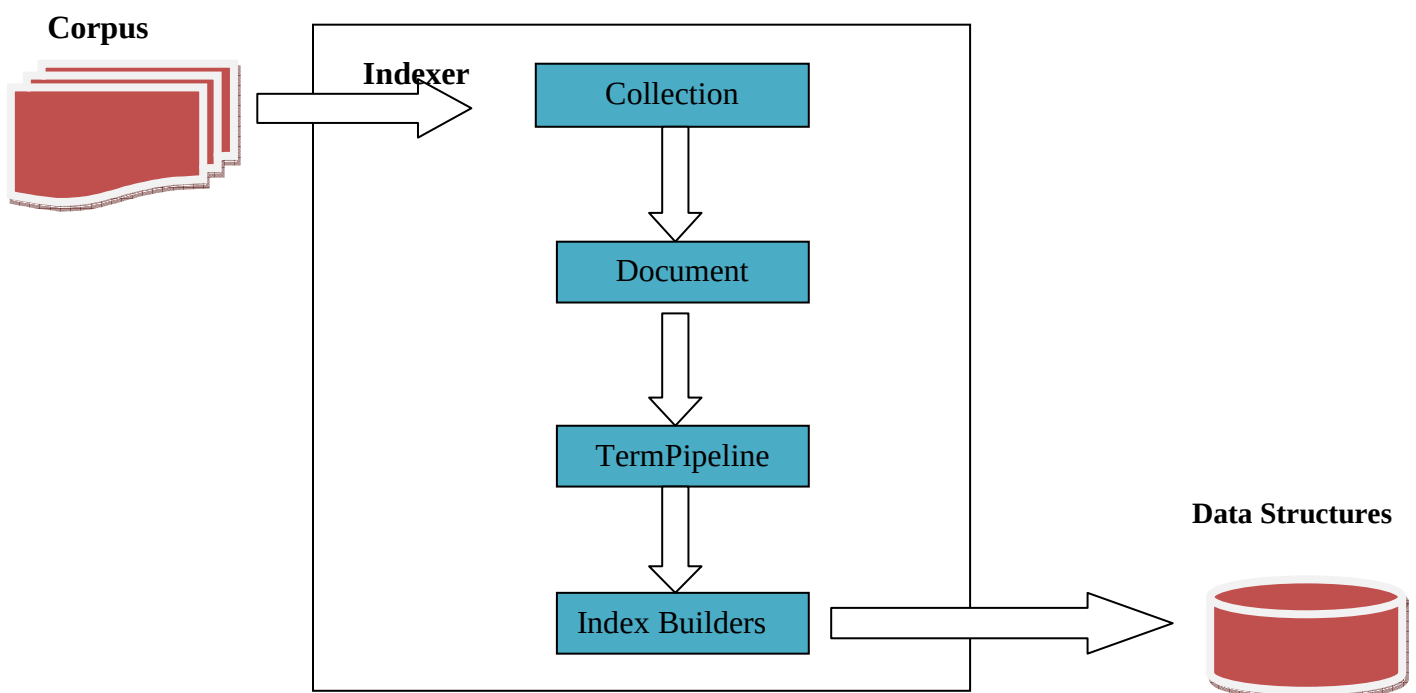
Org.terrier.indexing : ce package contient les différentes classes permettant de réaliser un ensemble d'opérations sur la collection des documents, dans le but d'extraire les termes de tous les documents de la collection.

Org.terrier.terms : Les classes qui se trouvent dans ce package permettent d'effectuer un ensemble de traitements sur les termes extraits. Parmi ces traitements l'élimination des mots vides, lemmatisation des termes, ...etc.

Org.terrier.structures : Les classes de ce package permettent la construction d'un ensemble de structures où un ensemble de données est stocké, parmi ces structures on a :

- ✓ Lexicon : qui stocke les informations de chaque terme de la collection ;
- ✓ Inverted Index : où le fichier inverse est stocké ;
- ✓ Document Index : qui contient des informations sur les différents documents de la collection ;
- ✓ ...etc.

La figure ci-dessous donne un aperçu de l'interaction des principaux composants impliqués dans le processus d'indexation.



FigureIII.3 : Processus d'indexation de Terrier.

a. Collection

Cette composante est une Interface dans le package **org.terrier.indexing**, elle représente un concept fondamental de l'indexation avec terrier. Son rôle est de Splitter une collection (préalablement construite à partir de corpus) en documents. Pour cela elle fait appel à plusieurs de ses méthodes telles que :

- ✓ `public Document getDocument()` ;

- ✓ public booleannextDocument()
- ✓ public String getDocid() ;
- ✓ public booleanendOfCollection() ;

Plusieurs Classes implémentent l'interface collection selon le format du document :

- ✓ SimpleFileCollection : PDF, TXT, HTML,.....
- ✓ TrecCollection
- ✓ SimpleXMLCollection / XML
- ✓ SimpleMedlineXMLCollection

b. Document

Ce composant est une interface qui se trouve dans le package **org.terrier.indexing**. Cette interface englobe un concept important, celui de document. Son rôle est de parcourir les documents et d'en extraire les *Termes* en utilisant Tokniser.

Plusieurs méthodes sont utilisées :

- ✓ public String getNextTerm() ;
- ✓ public booleanendOfDocument() ;

Plusieurs parseurs sont disponible selon le format du document

- ✓ HTMLDocument
- ✓ FileDocument

Dans notre cas on a utilisé des documents de format TXT

c. TermPepeline

C'est une interface qui se trouve dans le package **org.terrier.terms**. Son rôle consiste en le traitement des termes extraits. Il peut transformer les termes ou supprimer des termes qui ne devraient pas être indexés. En définitive il :

- ✓ Elimine les mots vides (Stopwords)
- ✓ Lemmatise les termes selon la langue : pour l'anglais l'algorithme de lemmatisation utilisé est celui de Porter (PorterStemmer)

d. Indexer

Ce composant est responsable de la gestion du processus d'indexation et entre autre de la construction de l'index. Il instancie les termes pipelines et initialise les builders qui sont en charge de l'écriture de l'index sur le disque dans la structure de données appropriée.

Terrier offre deux types d'indexeurs : *BasicIndexer*, *BlockIndexer*

e. Index Data Structures (structures d'index)

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

❖ Document Index

Ce fichier comporte trois colonnes précisant les noms des documents, leurs tailles et leurs adresses, comme suit :

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

Sur la figure suivante, voici un exemple du fichier Direct Index retourné par terrier après l'indexation et qui contient les informations décrites en haut

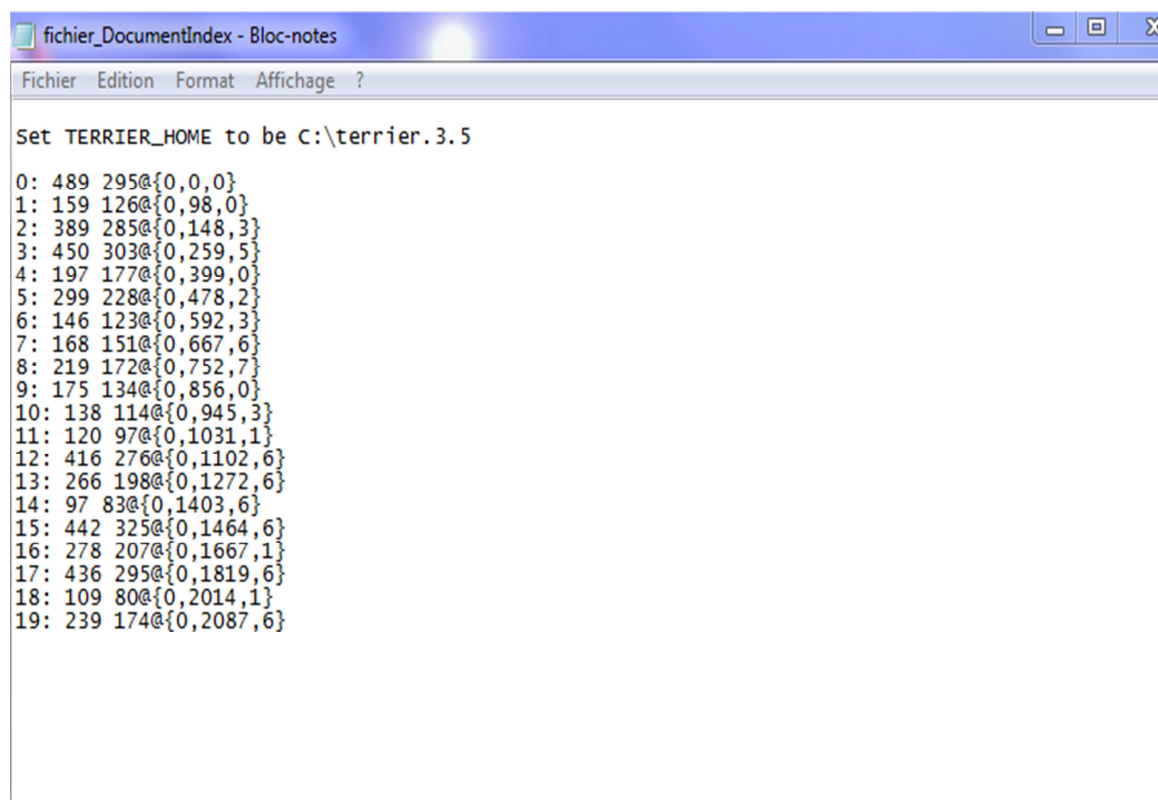


Figure III.4 : Exemple de la structure d'un Index de document.

❖ Direct Index

- Id Terme
- FréquenceTerme
- #Filds (#of fields bits)

Le voici un exemple du fichier DirectIndex



```

Set TERRIER_HOME to be C:\terrier.3.5
0 (0,2) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1) (7,1) (8,1) (9,4) (10,1) (11,1) (12,1) (13,1) (14
1,1) (142,1) (143,1) (144,2) (145,1) (146,2) (147,1) (148,1) (149,4) (150,3) (151,1) (152,1) (
269,1) (270,1) (271,1) (272,4) (273,2) (274,4) (275,1) (276,1) (277,1) (278,1) (279,1) (280,
1 (13,1) (19,1) (49,1) (57,1) (67,1) (68,1) (102,1) (144,1) (171,1) (182,1) (185,1) (193,1) (
2 (9,1) (23,1) (36,1) (46,1) (49,1) (57,1) (60,1) (68,2) (79,2) (95,1) (115,1) (119,4) (128,1
485,1) (486,1) (487,1) (488,1) (489,1) (490,2) (491,2) (492,1) (493,1) (494,1) (495,2) (496,2
(613,1) (614,2) (615,1) (616,1) (617,1) (618,1) (619,1) (620,1) (621,2) (622,1) (623,1) (624,
3 (4,1) (9,1) (19,1) (41,1) (51,1) (57,1) (60,1) (64,2) (67,1) (68,1) (75,1) (83,1) (104,1) (
2,2) (683,1) (684,2) (685,1) (686,1) (687,2) (688,1) (689,1) (690,1) (691,1) (692,1) (693,1) (
10,1) (811,1) (812,1) (813,1) (814,1) (815,1) (816,10) (817,1) (818,2) (819,1) (820,1) (821,1
4 (19,1) (44,1) (57,1) (120,2) (141,1) (199,1) (202,1) (216,1) (221,1) (231,3) (239,1) (243,1
946,2) (947,1) (948,1) (949,1) (950,1) (951,1) (952,1) (953,2) (954,1) (955,1) (956,2) (957,1
5 (5,1) (29,1) (32,1) (34,1) (42,2) (46,1) (57,4) (60,3) (73,1) (75,1) (83,1) (91,1) (93,1) (
1045,2) (1046,1) (1047,1) (1048,1) (1049,1) (1050,1) (1051,1) (1052,1) (1053,2) (1054,1) (10
6 (4,3) (41,1) (44,1) (57,1) (68,1) (79,1) (103,2) (108,1) (119,2) (179,1) (182,1) (183,1) (1
1216,2) (1217,1) (1218,1)
7 (16,1) (35,1) (44,1) (51,1) (57,1) (67,1) (103,1) (119,1) (177,1) (193,2) (199,1) (200,1) (
1288,1) (1289,1) (1290,1) (1291,1) (1292,1) (1293,1) (1294,1) (1295,1) (1296,1) (1297,1) (12
8 (3,1) (19,1) (21,1) (36,1) (51,2) (60,2) (76,1) (84,5) (103,1) (119,1) (120,1) (173,2) (179
1374,4) (1375,1) (1376,1) (1377,2) (1378,1) (1379,1) (1380,1) (1381,1) (1382,1) (1383,1) (13
9 (34,1) (57,1) (93,1) (102,1) (115,1) (116,1) (119,1) (126,1) (132,1) (143,1) (156,1) (164,2
) (1486,1) (1487,1) (1488,1) (1489,2) (1490,1) (1491,1) (1492,2) (1493,1) (1494,1) (1495,1) (
10 (19,1) (55,1) (67,1) (110,1) (119,1) (141,1) (143,1) (171,1) (182,1) (185,1) (197,1) (207,
11 (10,1) (17,1) (57,4) (67,1) (119,2) (170,1) (182,2) (193,1) (200,1) (239,2) (243,2) (266,1
12 (4,1) (5,8) (7,1) (13,2) (16,4) (19,1) (30,1) (38,1) (41,1) (42,1) (43,2) (46,1) (49,1) (5
,1) (1211,1) (1222,1) (1246,1) (1305,1) (1341,1) (1348,1) (1373,1) (1379,2) (1397,1) (1402,1)
00,1) (1701,1) (1702,1) (1703,2) (1704,1) (1705,1) (1706,1) (1707,2) (1708,1) (1709,1) (1710,
13 (4,4) (23,1) (24,1) (30,1) (67,1) (86,1) (117,1) (120,1) (128,1) (159,1) (166,2) (182,1) (
756,2) (1757,1) (1758,1) (1759,1) (1760,1) (1761,2) (1762,2) (1763,1) (1764,1) (1765,2) (1766
14 (15,1) (28,1) (103,1) (144,1) (155,1) (185,1) (219,1) (221,1) (274,1) (280,2) (325,2) (381
15 (3,1) (9,1) (19,3) (23,1) (35,1) (41,1) (44,1) (49,1) (51,1) (57,3) (64,2) (76,1) (79,1) (
2) (683,1) (698,1) (704,2) (707,3) (724,1) (737,1) (756,1) (764,1) (765,1) (789,2) (811,1) (8
,1) (1930,1) (1931,1) (1932,1) (1933,1) (1934,1) (1935,1) (1936,1) (1937,3) (1938,1) (1939,1)
16 (4,1) (13,2) (30,1) (35,1) (51,1) (53,2) (57,3) (67,2) (68,3) (102,1) (106,1) (112,1) (116
908,1) (1911,1) (1927,2) (1937,1) (1999,1) (2008,1) (2009,1) (2010,1) (2011,1) (2012,1) (2013
17 (16,1) (27,1) (35,2) (44,1) (51,1) (57,2) (75,1) (89,2) (95,1) (102,1) (110,1) (117,1) (11
1554,1) (1565,1) (1577,1) (1590,1) (1594,1) (1610,2) (1627,1) (1631,1) (1634,1) (1654,1) (168
1) (2173,1) (2174,1) (2175,2) (2176,1) (2177,1) (2178,1) (2179,1) (2180,1) (2181,1) (2182,21)
18 (60,2) (67,1) (103,1) (116,1) (119,1) (121,1) (143,1) (155,1) (161,1) (171,2) (182,1) (193
19 (4,1) (10,1) (30,2) (36,1) (57,2) (60,1) (69,1) (75,1) (96,1) (116,1) (119,3) (143,1) (146
) (2283,3) (2284,1) (2285,1) (2286,1) (2287,1) (2288,1) (2289,1) (2290,1) (2291,1) (2292,7) (

```

Figure III.5 : Exemple de la structure d'un Direct Index de document.

❖ Inverted Index : Fichier inverse

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

Voici un exemple du fichier Inverted Index

```

Set TERRIER_HOME to be C:\terrier.3.5
0 (1,2) (2,3) (4,1) (6,1) (9,1) (10,2) (15,3) (16,3) (17,1)
1 (0,1) (5,2) (6,1) (9,1) (12,1)
2 (1,1) (7,1) (9,1)
3 (3,2) (4,1) (8,1) (10,1) (13,1)
4 (2,1)
5 (13,1)
6 (15,1)
7 (2,1)
8 (1,1)
9 (8,1)
10 (2,1) (3,1) (5,1) (7,1) (16,1) (17,1)
11 (4,1)
12 (10,1)
13 (5,1)
14 (15,1) (16,1)
15 (9,1)
16 (8,1)
17 (12,1)
18 (19,1)
19 (5,1)
20 (17,1)
21 (17,1)
22 (17,1)
23 (0,1) (11,1)
24 (16,1)
25 (8,1) (16,1)
26 (9,1)
27 (9,1) (11,1) (17,1)
28 (9,1)
29 (0,1)
30 (15,1) (16,2)
31 (9,1) (11,1)
32 (6,1) (12,1)
33 (12,1)
34 (0,1)
35 (3,1)
36 (5,1) (7,1) (8,1) (18,1)
37 (1,1) (2,1) (6,1) (9,2) (13,1) (15,1)
38 (7,1) (9,1) (15,2) (17,1)
39 (3,1) (8,1) (16,1)
40 (7,1) (13,1)
41 (7,1)
42 (15,1)
43 (0,2)
44 (2,1) (9,1) (15,1)
45 (15,1)
46 (10,1) (15,1) (16,1)
47 (3,1) (8,1) (11,1)

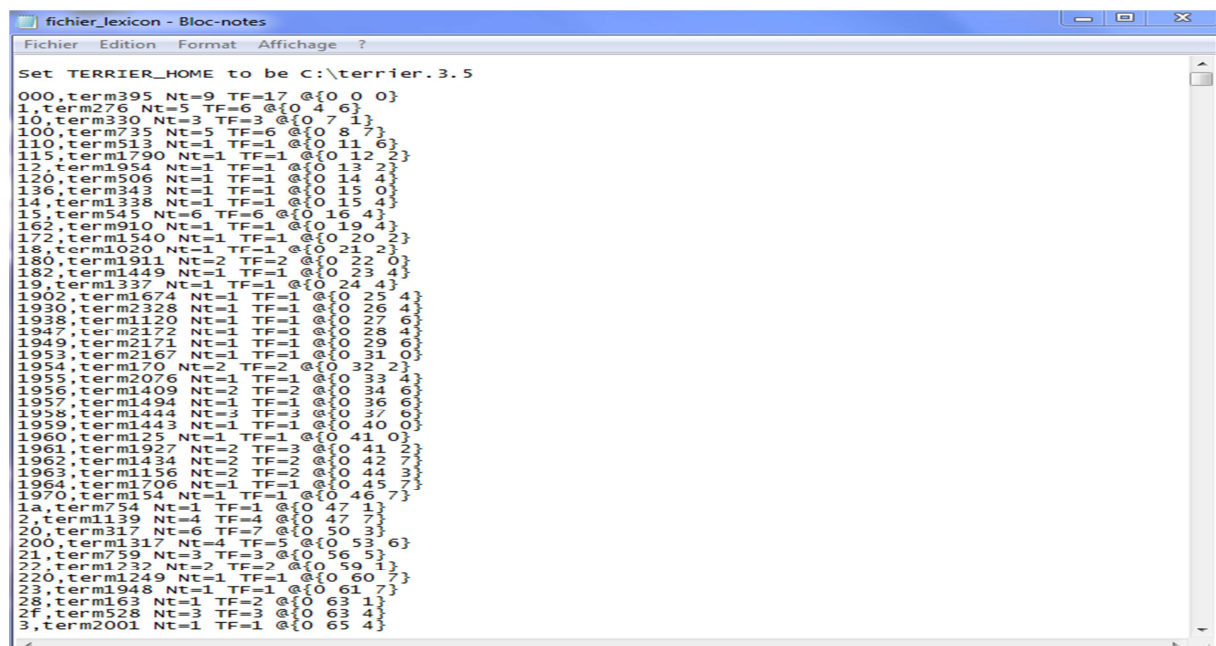
```

Figure III.6 : Exemple de la structure d'un Inverted Index de document.

❖ **Lexicon** : Informations sur chaque terme de la collection:

- Terme
- Id terme
- Nombre documents qui contiennent le terme
- Fréquence terme dans la collection
- Offset terme dans le fichier inverse

Voici les différentes informations contenues dans le fichier lexicon



```

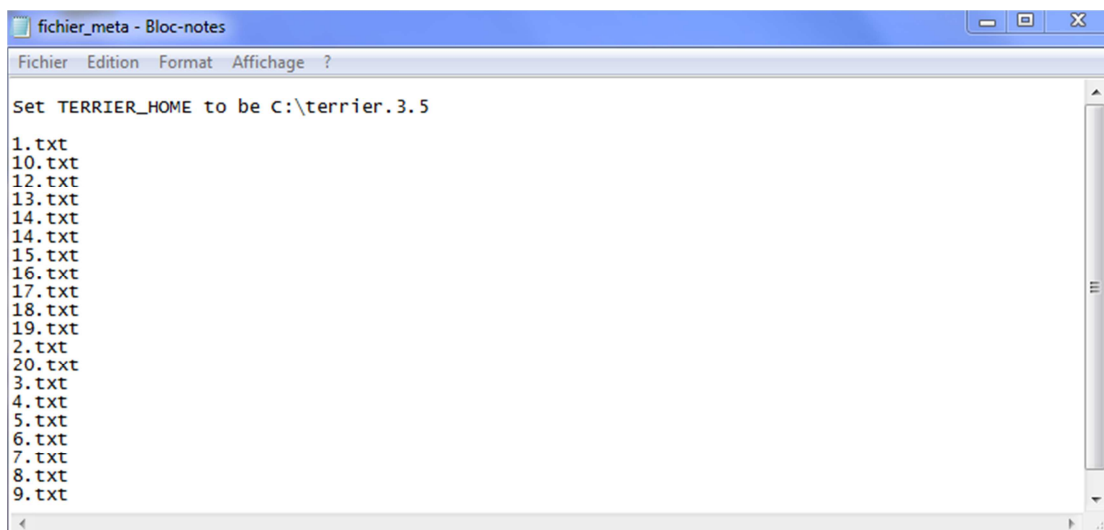
Set TERRIER_HOME to be C:\terrier.3.5
000,term395 Nt=9 TF=17 @{{0 0 0}}
1,term276 Nt=5 TF=6 @{{0 4 6}}
10,term330 Nt=3 TF=3 @{{0 7 1}}
100,term735 Nt=5 TF=6 @{{0 8 7}}
110,term513 Nt=1 TF=1 @{{0 11 6}}
115,term1790 Nt=1 TF=1 @{{0 12 2}}
12,term1954 Nt=1 TF=1 @{{0 13 2}}
120,term506 Nt=1 TF=1 @{{0 14 4}}
136,term343 Nt=1 TF=1 @{{0 15 0}}
14,term1338 Nt=1 TF=1 @{{0 15 4}}
15,term545 Nt=6 TF=6 @{{0 16 4}}
162,term591 Nt=1 TF=1 @{{0 19 4}}
172,term1540 Nt=1 TF=1 @{{0 20 2}}
18,term1020 Nt=1 TF=1 @{{0 21 2}}
180,term1911 Nt=2 TF=2 @{{0 22 0}}
182,term1449 Nt=1 TF=1 @{{0 23 4}}
19,term1337 Nt=1 TF=1 @{{0 24 4}}
1902,term1674 Nt=1 TF=1 @{{0 25 4}}
1930,term2328 Nt=1 TF=1 @{{0 26 4}}
1938,term1120 Nt=1 TF=1 @{{0 27 6}}
1947,term2172 Nt=1 TF=1 @{{0 28 4}}
1949,term2171 Nt=1 TF=1 @{{0 29 6}}
1953,term2167 Nt=1 TF=1 @{{0 31 0}}
1954,term170 Nt=2 TF=2 @{{0 32 4}}
1955,term2076 Nt=1 TF=1 @{{0 33 4}}
1956,term1409 Nt=2 TF=2 @{{0 34 6}}
1957,term1494 Nt=1 TF=1 @{{0 36 6}}
1958,term1444 Nt=3 TF=3 @{{0 37 0}}
1959,term1443 Nt=1 TF=1 @{{0 40 0}}
1960,term125 Nt=1 TF=1 @{{0 41 0}}
1961,term1927 Nt=2 TF=3 @{{0 41 2}}
1962,term1434 Nt=2 TF=2 @{{0 42 2}}
1963,term1156 Nt=2 TF=2 @{{0 44 3}}
1964,term1706 Nt=1 TF=1 @{{0 45 7}}
1970,term154 Nt=1 TF=1 @{{0 46 7}}
1a,term754 Nt=1 TF=1 @{{0 47 1}}
2,term1139 Nt=4 TF=4 @{{0 47 7}}
20,term317 Nt=6 TF=7 @{{0 50 3}}
200,term1317 Nt=4 TF=5 @{{0 53 6}}
21,term759 Nt=3 TF=3 @{{0 56 5}}
22,term1732 Nt=2 TF=2 @{{0 59 1}}
220,term1249 Nt=1 TF=1 @{{0 60 7}}
23,term1948 Nt=1 TF=1 @{{0 61 7}}
28,term163 Nt=1 TF=2 @{{0 63 1}}
2f,term528 Nt=3 TF=3 @{{0 63 4}}
3,term2001 Nt=1 TF=1 @{{0 65 4}}

```

Figure III.7: Exemple de la structure d'un Lexicon de document.

❖ **Meta Index** : informations additionnelles (méta informations) sur chaque document comme son URL.

L'ensemble des documents indexés sont affichés dans le fichier meta comme paraît dans la figure suivante :



```

Set TERRIER_HOME to be C:\terrier.3.5
1.txt
10.txt
12.txt
13.txt
14.txt
14.txt
15.txt
16.txt
17.txt
18.txt
19.txt
2.txt
20.txt
3.txt
4.txt
5.txt
6.txt
7.txt
8.txt
9.txt

```

Figure III.8: Exemple de la structure d'un Meta Index de document.

Un corpus est représenté sous la forme d'un objet de Collection. Les données de texte brut seront représentées sous la forme d'un objet Document.

L'indexeur est chargé de gérer le processus d'indexation. Il parcourt les documents de la collection et envoie chaque terme trouvé par un composant TermPipeline.

Un TermPipeline peut transformer ou supprimer des termes qui ne doivent pas être indexés. Un exemple d'une chaîne TermPipeline est : `termpipelines = mots vides, PorterStemmer`, qui supprime les termes du document en utilisant les objets `mots vides`, et applique ensuite un algorithme de PorterStemmer en anglais qui radicalise les termes (PorterStemmer).

Une fois que les termes ont été traités par le TermPipeline, ils sont regroupés et les structures de données suivantes sont créées par leurs DocumentBuilders correspondant : `DirectIndex`, `DocumentIndex`, `Lexicon`, et `InvertedIndex`.

L'indexation en Terrier peut être configurée en changeant les propriétés appropriées dans le fichier `etc\terrier.properties`. Chaque procédure citée auparavant possède ces propres propriétés.

o Utilisation de Terrier : « TREC Terrier »

L'utilisation de TREC Terrier pour l'indexation, la recherche et l'évaluation sur le système d'exploitation Windows 7. Sont faites comme suit

Indexation

Pour indexer des documents avec Terrier on peut utiliser cette méthode :

➤ Dans l'invite de commande :

- Spécifier le chemin vers Terrier : `cd <le path vers Terrier>\bin`
- Spécifier la collection de documents à indexer : `trec_setup<le path de la collection à indexer>`. Si on ouvre le fichier `collection.spec` on s'aperçoit que Terrier a mis dans ce dernier tous les chemins vers les fichiers contenus dans le répertoire donné automatiquement.
- L'indexation est lancée par la commande suivante :
 - « `trec_terrier -i` » Pour l'indexation ;

Remarque : pour indexer une collection autre qu'une Collection de TREC, il est nécessaire d'apporter quelques modifications au fichier `terrier.properties`.

S'il n'y a pas eu d'erreurs, on trouvera dans `var\index` les fichiers résultants de l'indexation.

Recherche

- Pour faire une recherche sur une collection de documents déjà indexés on utilise la commande « `trec_terrier -r` ». mais avons, on doit :
 - Donner à Terrier les requêtes et ce en modifiant `etc\terrier.properties` comme suit : `trec.topics=<le path vers les requêtes>`.
 - Spécifier le modèle de pondération (ex : `TF_IDF`) à utiliser dans le fichier `etc\trec.models`.

Dans notre cas nous avons utilisé le modèle `BM25` dans la recherche classique et le modèle `TF_IDF` dans la recherche sociale

- Les résultats de la recherche sont stockés dans :
« `var\result\nom_fonction_pondération.res` »

III.2.6 Le module de recherche

Ce modèle permet d'effectuer l'appariement entre la requête interne et les représentations des documents de la base afin de restituer les informations pertinentes en intégrant la composante sociale.

Le processus de recherche de Terrier

Durant le processus de recherche, chaque requête doit passer par les étapes suivantes :

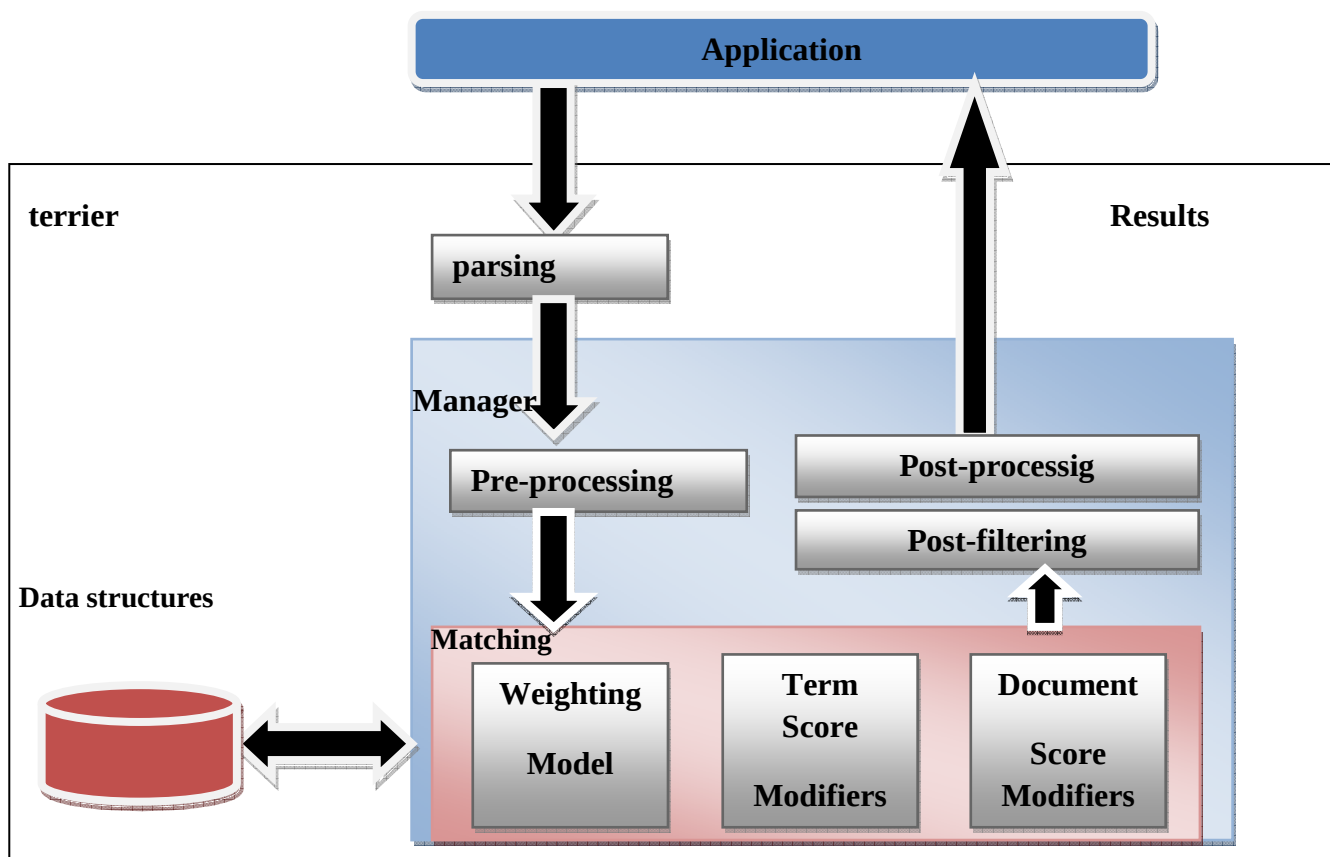
1. Parsing : qui se charge de tokenizer la requête.
2. Pré-processing : qui applique le `TermPipeline` à la requête. Elimine les mots vides et les lemmatise.
3. Matching : qui est responsable de l'initialisation du `WeightingModel` et du calcul des scores entre la requête et les documents.
4. Post-filtrage : va filtrer les résultats.
5. Post-traitement : peut modifier le `ResultatSet`, par exemple, par un procédé `QueryExpansion` afin de générer un meilleur classement de documents.

Parmi les packages où se trouvent les classes associées au processus de recherche on a :

Org.terrier.Matching : contient les classes qui permettent de déterminer les documents répondant à la requête.

Org.terrier.matching.models : on trouve dans ce package les différents modèles de pondération dont TF-IDF, BM25, ...etc ;

La figure suivante donne un aperçu de l'interaction des composants Terrier dans la phase de recherche :



FigureIII.9 : processus de recherche de Terrier.

a. Query

C'est une Classe abstraite dans le package `org.terrier.querying.parser` qui modélise la requête. Un objet Query est créé pour chaque requête.

Terrier supporte Trois modèles de requête:

- ✓ *SingleTermQuery*: modèle de requête avec un seul terme.
- ✓ *MultiTermQuery* : modèle de requête avec plusieurs termes.
- ✓ *FieldQuery* : Terme qualifié par un champ (Exemple: dans le titre du document)

b. Manager

Ce module est chargé de la gestion de la recherche il fait appel à quatre sous modules :

1. **Pre-processing**: Appliquer Tokeniser et TermPipeline
2. **Matching**: Déterminer les documents qui répondent à la requête en initialisant:
 - a. **WeightingModels** : Assigner un score pour chaque terme de la requête dans le document (Pondération). Plusieurs Modèles de pondération sont disponibles dans `org.terrier.matching.models` dont : `TF_IDF`, `BM25`.
 - b. **DocumentScoreModifiers** : Modifier le score d'un document en fonction du langage de la requête (ou expansion de la requête)
3. **Post-filtrring**: Filtrer les documents pertinents pour la requête.
4. **Post-processing**: Reclasser les documents pertinents s'il y a une expansion de la recherche.

c. Set-Results

Se charge de récupérer les résultats de Post-processing et de retourner les documents selon leur degré de pertinence.

Une application, `TrecTerrier`, émet une requête au système `Terrier`.

Dans une première étape, la requête sera analysée et une instanciation d'un objet de requête aura lieu.

Ensuite, la requête sera remise à la composante `Manager`. Le gestionnaire d'une part prétraite la requête, en l'appliquant au `TermPipeline`.

Après le prétraitement, la requête sera remise à la composante responsable de son traitement.

Cette dernière est responsable de l'initialisation du « `WeightingModel` » (modèle de pondération) approprié et « `DocumentScoreModifiers` ». Une fois tous ces éléments instanciés le calcul de score de documents à l'égard de la requête commencera.

Ensuite, le post-traitement et post-compilation ont lieu.

Enfin, le Résultat sera retourné à la composante d'application.

III.2.7 Le module d'intérêts sociaux

Ce score est calculé par le processus d'appariement classique (document-requête) estimé par le score de similarité **Score_C**, est augmenté par les informations contenues dans la base d'intérêt d'utilisateur. Le principe de base consiste à définir la fonction d'appariement calculant un score de pertinence d'intérêt **Score_S** qui est combiné dans la formule globale suivante :

$$Sim(d,q) = \alpha * Score_C + (1 - \alpha) * Score_S$$

Tel que :

α : est un paramètre défini par apprentissage, tel que $\alpha = \{0.3, 0.5, 0.8\}$.

Score-C : est le score classique qui est calculé à base du modèle de recherche BM25.

Score-S : est le score social qui est calculé à base de TF_IDF.

Les produits de cette étape sont constitués d'un ensemble de modèles qui sont une représentation détaillée du modèle objet de l'analyse, prenant en compte notamment les langages et les environnements de développement.

- **Fonction de pondération BM25**

La fonction de pondération BM25 est traitée par Roberson et al. En 1994 qui ont adapté la formule de BM25 afin de tenir compte de la structure des documents HTML. Les éléments de la structure sont pondérés suivant leur importance. Par exemple, ils considèrent le titre, le « heading », et par la suite leur affectent un poids supérieur à celui du body.

Ainsi, dans BM25, le poids de chaque terme est calculé à partir du nombre d'occurrences du terme dans le document, de la distribution du terme dans le reste du corpus, de la taille du document, etc...

Lors du traitement d'une requête, la pertinence d'un document est évaluée par la somme des poids BM25 des termes de la requête qu'il contient.

Etant donné un terme t , une requête q et un document d , le poids w de d selon q et le terme est calculé par la fonction BM25 suivante:

$$W = \frac{(k_1 + 1) \times tf}{k + tf} \times \log \frac{N - n + 0.5}{n + 0.5} \times \frac{(k_3 + 1) \times qtf}{k_3 + qtf} \times Idf$$

Le tableau suivant illustre les variables de la fonction BM25 :

Variables	Description
N	Le nombre total de document dans la collection
n	Le nombre de document contenant le terme t , ($n < N$)
tf	La fréquence du terme t dans le document d
dl	La longueur du document d
$avdl$	La longueur moyenne des documents dans la collection
k	$k_1 \times ((1.b) + b \times \frac{dl}{avdl}) + tf$
k_1, k_3, b	Paramètres constants
Idf	fréquence du terme dans le corpus

Tableau III.1 Paramètres de la fonction BM25.

III.3 Conclusion

Dans ce chapitre, nous avons présenté les caractéristiques fonctionnelles de notre système, notamment lors de l'indexation des documents et la recherche d'information.

Pour la recherche, notre système parcourt les structures qui se trouvent dans la base d'intérêt afin de déterminer les documents qui correspondent au mieux à la requête de l'utilisateur.

Le chapitre qui suit, s'intéressera à compléter ces caractéristiques fonctionnelles par des caractéristiques techniques relatives à l'outil de développement choisi.

Implimentation et tests

IV.1 Introduction

Nous avons décrit dans le chapitre précédent notre approche qui consiste en l'intégration des tags sociaux aux processus de la RI en suite de calculé un score finale entre un score généré par la RI classique et un score sociale généré par la RI sociale basé sur l'ensemble des tags sociaux.

Dans ce chapitre nous avons décrit l'implémentation de notre système dans un environnement orienté objet. Nous avons présenté d'abord l'environnement technologique de développement dans notre approche, puis nous avons présenté quelques aspects de son implémentation, ainsi que les détails des tests et évaluation que nous avons entrepris sur notre approche.

IV.2 Présentation de l'environnement du travail

IV.2.1 Aspect technique du matériel utilisé

Durant la réalisation de notre travail, nous avons utilisé une machine ayant les principales caractéristiques suivantes :

- Un microprocesseur Intelcore. i3 M370
- Fréquence d'horloge 2.1GHz.
- Mémoire vive 4Go DDR 3
- Disque dure de 250 GO.

Nous avons développé notre application sous un système d'exploitation Microsoft Windows 7 « Edition Intégrale »

IV.3 L'environnement de développement

Dans nos expérimentations on a utilisé la plate-forme Terrier qui est un open source entièrement écrit en java donc ce dernier est imposé à l'utilisation. Dans la section suivante nous présentons les outils utilisés.

IV.3.1 Le langage de programmation

Pour la programmation nous avons choisi le langage orienté objet JAVA.

a) Le langage JAVA

Le langage java, créé en janvier 1995, par l'équipe de James Gosling chez Sun, est un langage à objet qui permet d'écrire des programmes portables sur la majorité des plates-formes. Lié à l'essor World Wide Web .Il a été conçu en fonction des multiples exigences des développements informatique actuelle .il bénéficie d'une gigantesque bibliothèque de classes avec lesquelles il est aisé de composer des interfaces graphiques, de créer des applications multithread, de communiquer entre ordinateurs ... L'utilisation de Java s'étend de l'animation

d'une page HTML par une applet à des application complexes qui tourne à travers le réseau internet : forum de discussion, partage de page de données, jeux, etc...

Enfin, on trouve de plus en plus fréquemment le langage Java dans les systèmes embarqués

Les logiciels nécessaires au développement et à l'exécution de programme Java constitue ce qui est appelé chez SUN le JDK (Java Development Kit) qui comprend principalement :

- **Un compilateur** qui traduit un code source écrit en un langage intermédiaire entre le code source et le code machine appelé *byte code java*, code indépendant de la machine sur laquelle il sera interprété et exécuté.
- **Une machine virtuelle Java (JVM)** qui est destiné à exécuter un programme, interprète le *byte code Java* associé au fur et à mesure à l'exécution du programme pour le transformer en un code machine exécutable par la machine utilisée.

Une bibliothèque appelée API (pour application programming interface), formé d'une collection de classe.

L'ensemble formé de la machine virtuelle java et de l'API constitue la plate-forme Java.

- **Différents utilitaire** : pour fabriquer automatiquement de la documentation, pour visualiser les applets et pour fabriquer des archives java.

L'API contient un grand ensemble de classes rangées en paquetages ; grosso modo, un paquetage de l'API fournit un ensemble de fonctionnalités réparties dans les classes de ce paquetage.

- ✓ Tout programme java se fonde sur l'utilisation essentielle de classes mais aussi d'interfaces
- ✓ Une classe modélise une entité à l'aide d'attributs et de méthodes.
- ✓ Un programme construit en générale des instances de classes ; une instance de classes est appelée objet.

b) Le choix du langage java

Le choix de java comme langage objet est facile à expliquer. Bien que très récent, car il a été introduit par la société Sun en 1995, il connaît un engouement extraordinaire. Son immense succès s'explique par ses nombreuses caractéristiques intéressantes :

- ✓ Java est un langage de programmation objet
- ✓ Il est portable sur la plus part des plates-formes
- ✓ C'est un langage généraliste ayant un très vaste champ d'application (réseau, base de données, calcul scientifique, etc.). il permet de développer des applications professionnelles de grande taille.
- ✓ Il emprunte sa syntaxe, en grande partie, aux langages C et C++ qui sont les références actuelle en matière de langages professionnels. Toutefois il est beaucoup plus simple à utiliser. Il est facile de passer de Java aux langages C et C++ et réciproquement.
- ✓ Il intègre une interface graphique de haut niveau
- ✓ Il est sûr, car de nombreuses vérifications sont faites, aussi bien à la compilation qu'à l'exécution, pour limiter le nombre et la gravité des erreurs. De ce fait, le développement de grosses applications en java est plus rapide qu'avec des langages comme C, C++.
- ✓ Il existe de nombreuses bibliothèques de programmes dans des domaines très variés, de sorte que ce langage devient un langage professionnel de premier plan.
- ✓ Le code produit (il s'agit d'un pseudo-code ou byte-code) est indépendant de la plateforme utilisée.
- ✓ Les programmes java peuvent être exécutés sous forme d'applications indépendantes ou distribuées à travers le réseau et exécutées par un navigateur internet sous forme d'applets.

En revanche, parmi les inconvénients les plus notables, il faut remarquer une certaine lenteur à l'exécution. Elle est particulièrement gênante pour les calculs scientifiques.

Toutefois, les éditeurs d'environnement de développement (EDI) s'attaquent sérieusement à ce problème. On peut raisonnablement espérer que java sera bientôt plus rapide que le C++

Lors du choix d'un langage d'application, il est primordial de chercher à minimiser les efforts pour :

- Développer le code
- Le vérifier (phase de test)
- Le maintenir
- Le faire évoluer

Comme plusieurs programmeurs peuvent intervenir dans ces différentes phases, la lisibilité du code est un critère déterminant et java y répond parfaitement. La facilité de vérification et de maintenance des applications résulte du caractère objet du langage.

Pour toutes ces raisons, java est un excellent choix pour le développement d'applications actuelles, d'autant que plusieurs Environnement de Développement intégrés de qualité sont disponibles chez différents éditeurs (Sun, Symantec, Borland, Asymetrix, etc.).

IV.3.2 présentation de l'environnement eclipse

Pour la réalisation de notre travail, nous avons choisis l'environnement de programmation « eclipse GALILEO » sous Windows7

❖ Eclipse (logiciel)

- **Eclipse IDE** est un environnement de développement intégré libre (le terme *Eclipse* désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

- La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.
- Plusieurs logiciels commerciaux sont basés sur ce logiciel libre, comme par exemple IBM Lotus Notes 8, IBM Symphony ou Websphere Studio Application Developer.

• Architecture d'Eclipse

La base de cet environnement de développement intégré est l'Eclipse Platform qui est composée de :

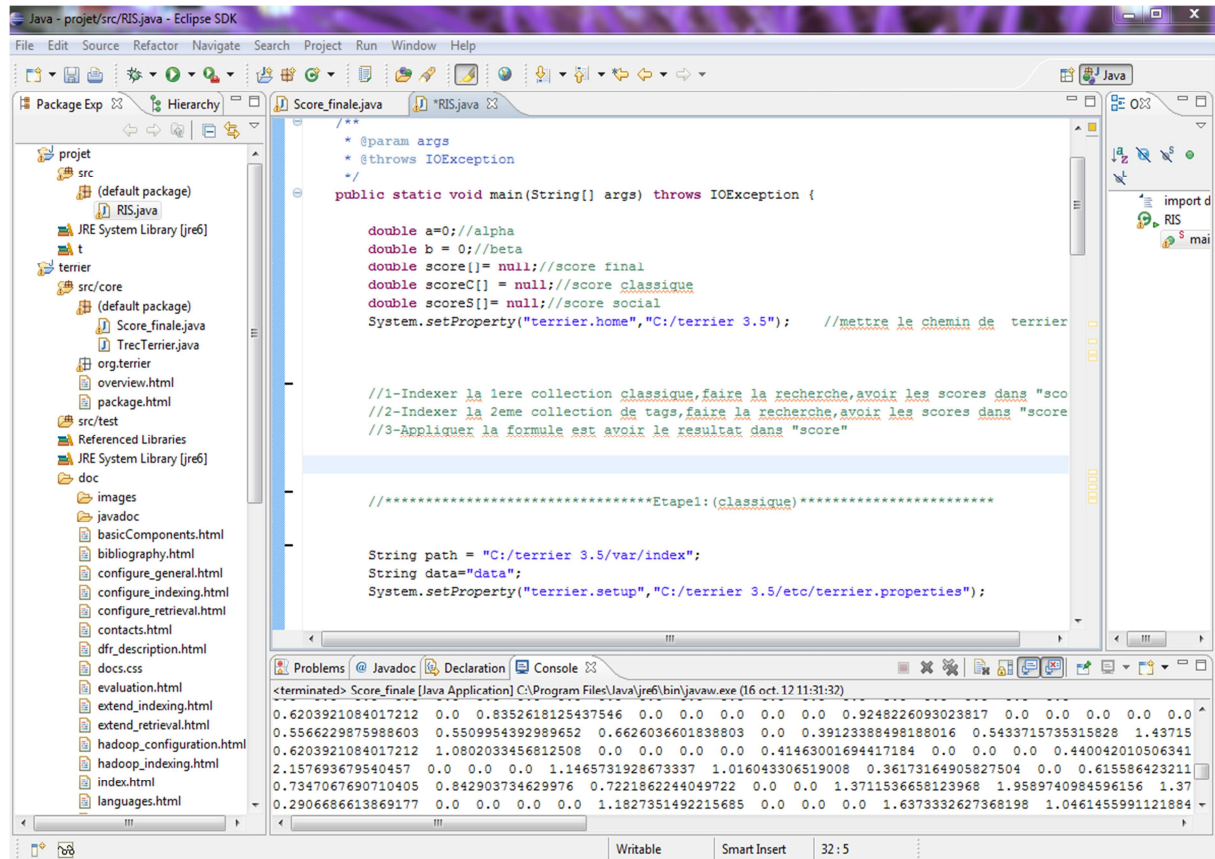
- Platform Runtime démarrant la plateforme et gérant les plug-ins
- SWT la bibliothèque graphique de base de l'EDI
- JFace une bibliothèque graphique de plus haut niveau basée sur SWT
- Eclipse Workbench qui est la dernière couche graphique permettant de manipuler des Composants tels que des vues, des éditeurs, des perspectives...

Ces composants de base peuvent être réutilisés pour développer des clients lourds indépendants d'Eclipse grâce au projet Eclipse RCP (Rich Client Platform).

L'ensemble des outils de développement Java sont ensuite ajoutés en tant que plug-ins. Ces plug-ins sont regroupés dans le projet JDT.

Ces plug-ins Eclipse sont architecturés selon les recommandations d'OSGi.

La figure ci-dessous illustre l'interface d'Eclipse **GALILEO**



FigureIV.1 Interface « Eclipse **GALILEO** ».

IV.4 La plate-forme Terrier

Terrier, *TeRabyteRetrIEveR*, est un moteur de recherche robuste et efficace, utilisé avec succès pour la recherche Ad-hoc, la recherche web et la recherche multilingue dans des environnements centralisées et distribuées. Terrier offre une plate-forme idéale destinée à l'indexation de volumes importants de documents: jusqu'à 25 millions de documents. Il est développé par le département informatique de l'université *Glasgow*. C'est un logiciel open source écrit en Java.

Comme tous les moteurs de recherche, terrier permet :

- **L'indexation classique** : extraction des mots clés des différents documents appartenant à une collection et les stocker dans un index.

- **Recherche des documents** : permet de généré des résultats aux requêtes formulés par l'utilisateur.
- **Evaluation des résultats de la recherche** : permet de mesurer le degré de pertinence des résultats de la recherche aux requêtes formulés par l'utilisateur.
-

IV.5 Implémentation

Nous présentons dans cette section un extrait du code implémentant notre classe

```
Public class Score_Finale {
```

```
.....
```

```
//charger la collection du collection.spec
```

```
TRECSetup trecsetup = new TRECSetup();
```

```
TrecTerrier trecerrier=new TrecTerrier();
```

```
//indexer
```

```
trecerrier.main(ind);
```

```
ApplicationSetup ap=new ApplicationSetup();
```

```
//modele de pondération
```

```
ap.setProperty("trec.model","BM25");
```

```
//rechercher
```

```
trecerrier.main(r);
```

```
Index indexclass = Index.createIndex(path,data);
```

```
DocumentIndexdoic = indexclass.getDocumentIndex();
```

```
CollectionResultSet resultclass=new CollectionResultSet(doic.getNumberOfDocuments());
```

```
intnbrclass=doic.getNumberOfDocuments();
```

```
//score classique
```

```
scoreC=resultclass.getScores();
```

```
.....
```

```
//charger la collection des tags
```

```
System.setProperty("collection.spec","C:/corpustag");
```

```
TRECSetup trecsetupsoc=new TRECSetup() ;
```

```

TrecTerrier trecTerriersoc=new TrecTerrier();
trecTerrier.main(ind);
//model de ponderation
ap.setProperty("trec.model","TF_IDF");
trecTerrier.main(r);
Index indexsoc=Index.createIndex(path,data);
DocumentIndex dois = indexsoc.getDocumentIndex();
CollectionResultSet resultsoc=new CollectionResultSet(dois.getNumberOfDocuments());
//score social
scoreS=resultsoc.getScores();
//choisir les valeurs de alpha et beta
float  $\alpha$ = 0,3;
//  $\beta = \alpha-1$ 
float  $\beta$ = 0,7;
//pour initialiser le tableau
score=scoreC;
for(inti =0;i<nbrclass; i++)
{
    score[i]=(a*scoreC[i])+(b*scoreS[i]);
    System.out.println(score[i]);
}
}
}

```

IV.6 Evaluation expérimentale de notre approche

L'objectif de cette évaluation est de mesurer les performances et la viabilité de notre approche de recherche d'information sociale. Nous présentons dans ce qui suit le cadre d'évaluation ainsi que les résultats expérimentaux préliminaires.

IV.6.1 Collection de tests

Les collections de tests ont été traditionnellement utilisées en RI pour évaluer les stratégies de recherche. Plus particulièrement, une collection de référence doit traduire la subjectivité de pertinence des utilisateurs. D'autre part, elle doit contenir une masse d'informations assez importante et variée pour constituer un environnement standard d'interrogation.

Différentes collections de tests sont utilisées en RI. Parmi elles, la collection TIME que nous avons utilisé pour nos expérimentations. Nous allons dans ce qui suit décrire cette collection en détail.

Collection TIME

Nom :	TIME magazine collection
Collection portant sur :	Divers domaines
Nombre de documents :	423
Nombre de requêtes:	83
Volume de la collection :	1.48 Mo

Tableau IV.1 : Description de la collection TIME.

La TIME est composé de documents au nombre de 423 tirés du magazine TIME de l'année 1963 et d'un nombre de requêtes raisonnable qui est de 83 accompagné d'un fichier de jugement de pertinence.

La collection TIME est une collection portant sur des domaines différents. Elle se caractérise par sa haute efficacité de recherche par rapport aux autres collections typiques.

Mais dans nos expérimentations on s'est restreint à 20 documents et 20 requêtes pour faciliter les tests

Un document est identifié comme suit : /Numéro du document/

Par exemple : « 15.txt ».

Les documents et les requêtes sont composés de textes simples. En voici un exemple :

Exemple de document (15.txt) :

```
<DOC>
<DOCNO>15.txt </DOCNO>
«RUSSIA PARTY TIME THE NEW YEAR'S EVE PARTY WAS GOING FULL BLAST IN THE BANQUET HALL ATOP THE KREMLIN'S PALACE OF CONGRESSES. COMMUNIST BIGWIGS MINGLED WITH DIPLOMATS, MILITARY LEADERS AND STARS OF THE SOVIET CULTURAL ELITE. EVERYONE WAS IN HIGH SPIRITS, INCLUDING SOVIET EX-PRESIDENT KLIMENT VOROSHILOV, 82, WHO BROKE INTO AN IMPROMPTU JIG WHEN THE BAND PLAYED A SNAPPY RUSSIAN MELODY. GENIAL HOST NIKITA KHRUSHCHEV ROARED HIS HEARTY APPROVAL. IT WAS NO OCCASION FOR DISHARMONIOUS WORDS, SO WHEN THE TIME CAME FOR SPEECHES, NIKITA WAVED AWAY THE LATEST 20,000-WORD ATTACK ON HIS POLICIES BY RED CHINA. EVERY FAMILY HAS TROUBLES, KHRUSHCHEV DECLARED, GESTURING AMIABLY AT THE WESTERN DIPLOMATS IN THE CROWD. "YOU JUST GET MARRIED AND YOU WILL SOON NOTICE THAT DIFFERENCES DEVELOP. "AS FOR HIS DIFFERENCES WITH THE WEST, KHRUSHCHEV HOPED THAT 1963 WOULD SOLVE "URGENT PROBLEMS FRAUGHT WITH NEW CRISES, "A BIT OF DOUBLETALK ABOUT BERLIN THAT COULD FIT ANY EVENTUALITY. THE FIRST WOULD PROBABLY COME AT NEXT WEEK'S CONGRESS OF THE EAST GERMAN COMMUNIST PARTY, WHICH KHRUSHCHEV WILL ATTEND. LAST WEEK, WHEN THE US MOVED 1,500 INFANTRYMEN BY HIGHWAY INTO THE DIVIDED CITY IN A ROUTINE SHIFT OF REGIMENTS, THERE WAS NOT A MOMENT OF OBSTRUCTIONIST DELAY AT THE RUSSIAN CHECKPOINT. READY TO GREET THE FRESH TROOPS WAS A NEW US WEST BERLIN COMMANDANT, MAJOR GENERAL JAMES H. POLK, 51. SAID POLK, IN A MESSAGE TO WEST BERLINERS:WE ARE HERE TO STAY.»
</DOC>
```

Exemple de requête :

<NUM : requête>

8:UN TEAM SURVEY OF PUBLIC OPINION IN NORTH BORNEO AND SARAWAK ON THE QUESTION OF JOINING THE FEDERATION OF MALAYSIA

13: PRECARIOUS TRUCE IN LAOS WHICH WAS BROUGHT UP BY BRITAIN BEFORE THE 14 NATIONS THAT AGREED ON THE TRUCE IN GENEVA LAST YEAR

Des jugements de pertinence sont associés aux requêtes selon le format suivant :

« Numéro de la requête / 0 /Numéro du document / 1 »

1 0 268.txt 1

1 0 326.txt 1

1 0 334.txt 1

1 0 288.txt 1

2 0 326.txt 1

1 0 304.txt 1

2 0 334.txt 1

IV.7 Evaluation de notre approche pour la recherche

Le tableau suivant récapitule les résultats de recherche obtenus pour (08) documents des trois scores classique S_Classique, le score social S_Sociale et le score finale S_Finale .pour $\alpha=0.3$

Les documents	Score de la Recherche d'information classique	Score de la Recherche sociale	Score finale
D1	0	1,95863495	1,37104447
D2	2,18979146	0,87267998	1,26781342
D5	1,30057872	0	0,39017362
D6	0	1,86217289	1,30352102
D7	0	1,55272333	1,08690633
D10	2,58427905	1,01486221	1,48568726
D11	1,63498102	1,31882869	1,41367439
D12	1,24888868	1,29923636	1,28413206
D14	1,22455441	0,49708263	0,71532417
D15	1,68077677	0,82936804	1,08479066
D16	1,5805759	0,79594212	1,03133225
D17	0	0,72793187	0,50955231

D19	0	1,96051776	1,37236244
D20	0	1,00475601	0,7033292

Tableau IV.2 contenus des fichiers résultats de la recherche.

Voici une figure qui résume les résultats obtenus dans le graphe suivant :

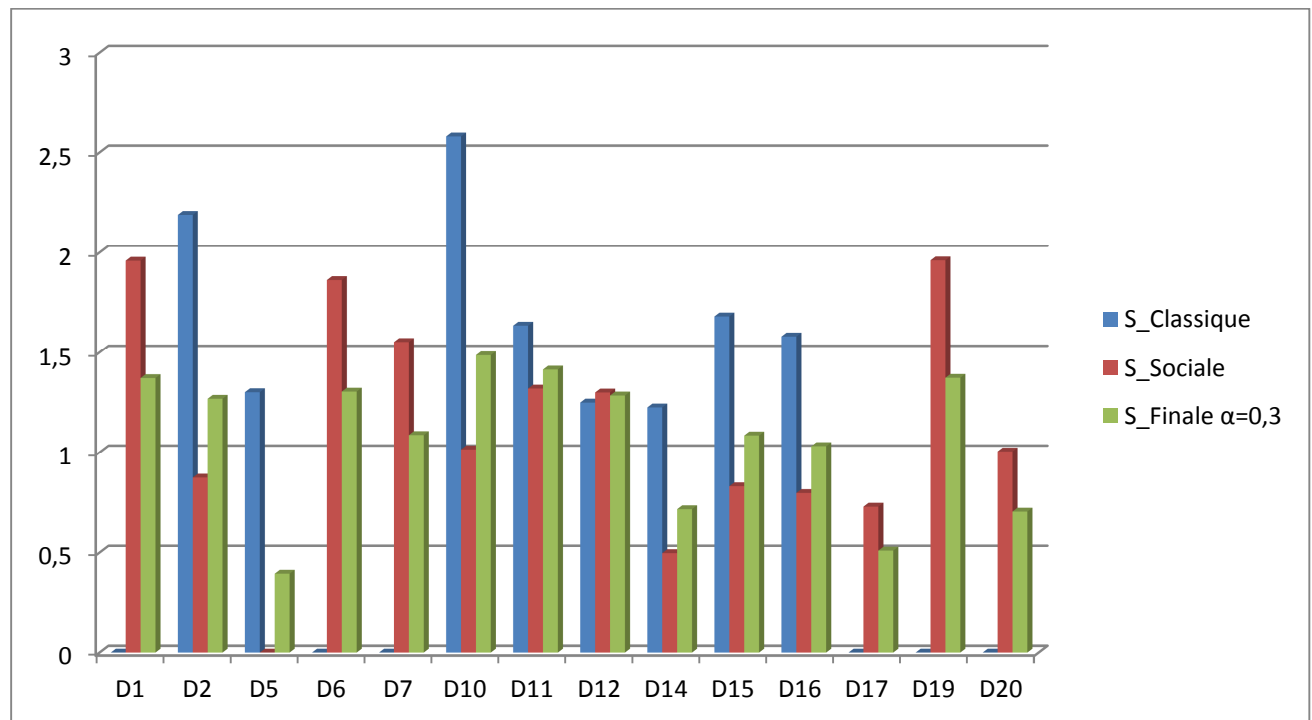


Figure IV.2 graphe représentant un Score finale pour $\alpha=0,3$.

• Discussion des résultats obtenus

D'après la figure n° VI.2 plusieurs cas se présentent :

- Nous observons que le système a restitué une liste de documents lors de la recherche classique (D10, D2, D15, D11, D16, D5, D12 et D14). Leur score social n'est pas nul ce qui explique que ces documents sont intéressants pour les utilisateurs qui les ont tagués. C'est le cas des documents D1, D6, D7, D17, D19 et D20 qui n'ont pas été restitués par le système lors de la recherche classique, mais avec notre algorithme ils ont été sélectionnés et sont classés parmi les premiers de la liste pour la requête 17.

- Pour le document D5, on voit que son score a diminué après qu'il ait été en tête de liste lors de la recherche traditionnelle. Ceci est dû au fait qu'il n'a pas de score social. A ce moment-là deux possibilités peuvent être considérées :
 - ✓ Soit le document n'a suscité l'intérêt d'aucun utilisateur (document non intéressant veut dire non tagué)
 - ✓ Ou que les tags qu'ils lui sont assignés ne correspondent pas aux termes de la requête.
- D'autres documents sont intéressants mais qui ne figurent pas parmi les résultats de la première recherche car ils ne contiennent pas (dans leur texte) de mots clés qui correspondent aux termes de la requête, mais grâce à notre approche, ils ont pu apparaître dans la liste puisque des utilisateurs ont exprimé leurs intérêts dessus (tags sociaux) qui correspondent aux termes de la requête. C'est le cas des documents D1, D6, D17, D19 et D20.

IV.8 Conclusion

Dans ce chapitre, nous avons en premier lieu, présenté l'environnement et les outils utilisés pour implémenter et réaliser notre solution. Ensuite, nous avons présenté les détails de l'implémentation ainsi que les expérimentations et résultats associés à notre approche d'intégration des intérêts (tags) sociaux. Selon les résultats obtenus, ils y a bien une amélioration des scores lorsqu'on utilise les tags sociaux, qui nous permettent d'obtenir un score social qui est lui-même rajouté au score classique en faisant varier un paramètre α pour obtenir en fin un meilleur score final.

conclusion générale

Conclusion générale

Le travail présenté dans ce mémoire se situe dans le cadre de la recherche d'information contextuelle, et plus particulièrement dans le cadre de la recherche d'information sociale (RIS) qui exploite les interactions sociales au sein des réseaux sociaux.

Les différentes approches de la recherche d'information classique ignorent l'influence des relations sociales et des interactions de l'utilisateur au sein de son contenu social, sur le processus global de la recherche d'information. Vu l'importance que peut apporter ce type de méta-informations sur la qualité de la recherche, nous avons pensé à une solution hybride qui tient compte des tags sociaux.

L'objectif principal de ce projet est l'amélioration de la recherche d'information classique telle qu'en intégrant les intérêts sociaux (les tags sociaux, dans notre cas) dans le processus de la recherche, de nouveaux documents peuvent apparaître sur la liste restituée par le système. Si ces nouveaux documents n'étaient pas restitués auparavant, c'est qu'ils ne contenaient pas de termes identiques aux mots clés de la requête. Avec notre solution, le système ne se contente pas d'une recherche dans la collection de documents mais extrait les intérêts sociaux explicites et fouille dans l'ensemble des tags sociaux qui sont assignés à ces documents afin de retrouver les tags (des termes supplémentaires et complémentaires au contenu textuel des documents) similaires aux termes de cette requête.

Les travaux effectués dans le cadre de la RIS montrent l'intérêt d'exploiter les informations sociales pour la recherche d'information mais n'utilisent pas les intérêts. Nous avons vu dans le présent mémoire deux représentations d'intérêts sociaux. Le premier modèle consiste en le modèle de Wen qui permet de déduire l'intérêt d'un utilisateur à partir de ces voisins sociaux. Le second modèle est celui de Tamine qui se base sur l'hypothèse qu'un utilisateur a divers centres d'intérêt et qu'il peut basculer d'un centre d'intérêt vers un autre en cours de sessions de recherche successives. Notre approche représente les tags sociaux d'un document par leurs poids dans le document lui-même en les considérant comme des termes ordinaires. Ensuite, on les rajoute au score classique par lissage, en variant un paramètre α .

Enfin, la réalisation de ce travail nous a donc permis de nous imprégner et même d'acquérir de nouveaux concepts dans plusieurs domaines aussi bien pratique que théorique entre autres la RI, la RIS où l'on manipule diverses informations sociales, avec la plate-

forme terrier ou nous avons pu maîtriser certain aspect . Dans le domaine de la programmation avec le langage orienté objet java.

Nous avons l'intention d'étendre notre modèle à d'autres types d'intérêts tels que Les clicks effectués sur le lien d'une page et même d'étendre le modèle de Tamine sur les intérêts dynamique (à court terme et à long terme)

D'un autre côté, on envisage d'intégrer d'autres informations sociales, comme par exemple les commentaires sociaux, les opinions.

Pour les tests, il serait préférable de faire la simulation sur un cas réel de réseau social.

Annexel

I.1 Introduction

Depuis quelques années nous assistons à une explosion du nombre de sites web qui nous permettent de rentrer en relation avec d'autres personnes. Que ce soit pour développer notre réseau d'amis ou professionnel, pour faire partager notre passion avec d'autres ou pour construire une base de connaissances ensemble, ils sont nombreux à se partager le marché. La plupart d'entre eux sont gratuits et accessibles à tous à condition d'accéder au web. Ils s'appellent des logiciels sociaux et font désormais partie incontournable de notre vie.

I.2 Définition d'un Réseau social

Le terme désigne un site internet permettant à l'internaute de s'inscrire et d'y créer une carte d'identité virtuelle appelée le plus souvent « profil ». Le réseau est dit social en ce qu'il permet d'échanger avec les autres membres inscrits sur le même réseau : des messages publics ou privés, des liens hypertexte, des vidéos, des photos, des jeux... L'ingrédient fondamental du réseau social reste cependant la possibilité d'ajouter des « amis », et de gérer ainsi une liste de contacts.

Un réseau social est une structure formée par des relations entre des personnes. Cette structure sociale est composée de nœuds, généralement représentés par des individus ou des organisations. Les nœuds sont reliés entre eux par diverses connaissances sociales qui peuvent aller d'une connaissance simple jusqu'à un lien familial très fort. Le terme « *réseau social* » a été utilisé pour la première fois en 1954 dans l'article « *Class and committees in a Norwegian island parish* » de l'anthropologue social anglais J. A. Barnes.

Aujourd'hui, avec le développement de l'informatique, nous parlons énormément des *réseaux sociaux en ligne*, c'est-à-dire sur internet pour connecter des amis, des professionnels et d'autres individus entre eux à l'aide de différents outils que nous appelons les *logiciels sociaux*.

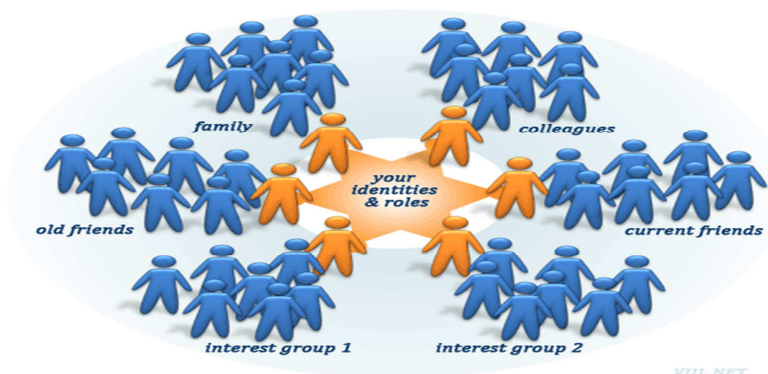


Figure I.1 : Réseau social à multiple relation.

I.3 Un réseau social en ligne

Un réseau social en ligne est un espace virtuel où les gens de même affinité peuvent se rencontrer et interagir. Les réseaux permettent aux internautes de communiquer entre eux de différentes manières, que ce soit par courrier électronique ou par messagerie instantanée, et de partager leurs informations personnelles. Ils sont de plus en plus utilisés par les entreprises, et non pas uniquement par les particuliers.

Les réseaux sociaux libres d'accès comme Facebook et MySpace sont les plus connus.

Les entreprises et les particuliers ont également la possibilité de créer leur propre site de réseau social sur mesure, à l'aide d'outils tels que Ning (www.ning.com).

I.4 Usages des réseaux sociaux : Les usages sont multiples.

- Retrouver des camarades de classe, de promotion ou d'anciens amis.
- Se faire des relations, un réseau.
- Partager des passions.
- Faire des rencontres, trouver de nouveaux amis.
- Organiser des événements.

I.5 Types des réseaux sociaux

En tapant réseau social sur un moteur de recherches sur Internet ou bien cherchant dans des livres traitant du sujet, on peut trouver des milliers de propositions répondant à ces critères. Les réseaux sociaux peuvent se diviser en plusieurs catégories.

Selon les différents avis, plusieurs classifications sont proposées. Nous allons présenter deux exemples déterminés par différents auteurs.

1) Tout d'abord, voici celle que l'on peut trouver sur Wikipédia, qui est la classification la plus simple :

- **Réseaux ouverts :** La plupart des réseaux sociaux sont des réseaux ouverts.

- **Réseaux sur invitation** : Il faut être invité par l'un de ses membres, cela permet au site de maîtriser sa croissance et aux individus de connaître au moins une personne.
- **Services en ligne de réseautage professionnels** : Ils favorisent les rencontres professionnels, les offres de poste et la recherche de profils.

2) Esther Dyson d'après sa définition sur les réseaux sociaux (« les réseaux sociaux fournissent des outils qui facilitent le processus de mise en relation autour d'un centre d'intérêt commun et permettent la prise de contact en ligne. »). Amène une classification des réseaux sociaux selon différents critères :

Les réseaux plate-forme de partage : Les plates-formes permettent de diffuser du contenu, souvent multimédia (vidéo et son), aux internautes. La mise en ligne et le partage de vidéo par exemple deviennent plus faciles car accessible par tous les internautes de la communauté. **Exemple** : Youtub, Dailymotion...

Les réseaux personnels et généralistes : Ils sont les plus souvent orientés vers un centre d'intérêt commun (lecture,...), avec le but de faire partager ses passions au reste de la communauté. On peut noter que les mises en relation sont rares dans ce type de réseaux. **Exemple** : MySpace, Skyblog, ...

- **Les réseaux personnels et thématiques** : Ils ont la plupart du temps le même fonctionnement que ceux de généralistes, mais ils sont orientés autour d'une thématique. **Exemple** : Boompa, Eonscom...
- **Les réseaux professionnels** : Les réseaux professionnels sont ceux les plus aboutis. Ils donnent la possibilité de mise en relation entre utilisateurs ainsi que le partage d'informations (information sur l'entreprise, coordonnées, CV...).

Exemples : 6nergies, Viaduc, LinkedIn, OpenBC...

I.6.1 Le développement des réseaux sociaux

Voici les points clés et primordiaux pour un bon développement d'un réseau social : communauté, modèle économique, technologie et fonctionnalités.

I.6.1.1 La communauté

Le développement des réseaux sociaux ne peut se faire sans une communauté solide et grandissante. Cette communauté consiste un groupe de personnes ayant des objectifs ou des intérêts similaires. Ils entrent en

contact et échangent des informations à l'aide des outils Web. Il faut qu'elles évoluent en même temps que leurs utilisateurs. Ainsi, il est important qu'un réseau social en ligne soit « ouvert » et « multi-culturel ».

Le but de tous réseaux sociaux que de rassembler un maximum d'utilisateurs ? Il est également essentiel, de répondre aux besoins des utilisateurs.

I.6.1.2 Le modèle économique

Un point important lors de la création d'un nouveau service ou une nouvelle entreprise, c'est bien celui de la réflexion sur le modèle économique. Cependant, dans le domaine du Web actuel, ce point vient souvent à la fin de la création mais comme le dit *Google* « les services d'abord, la monétisation viendra ensuite... ».

I.6.1.3 La technologie

Un point qui peut sembler être un détail mais qui, au contraire, s'il n'est pas bien analysé peut faire beaucoup de dégâts. Il faut tout d'abord savoir que le système d'un réseau social est bien plus complexe que n'importe quels autres sites, forums ou blogs. L'algorithme de mise en relation et de navigation au sein de ces liens demande beaucoup de ressources. De même les bases de données doivent gérer une énorme quantité d'informations et supporter une charge importante d'enregistrement. Dès lors, il est essentiel de bien dimensionner la puissance de son parc de serveurs et d'estimer correctement la bande passante nécessaire.

Effectivement, la technologie n'est pas critique pour débiter mais elle est indispensable pour atteindre le succès.

I.6.1.4 Les fonctionnalités

Certaines personnes se demandent comment *MySpace* a pu atteindre ce niveau avec une interface aussi peu agréable et austère. La réponse doit se situer au niveau des fonctionnalités et de la navigation sur le site. Effectivement, *MySpace* propose un nombre impressionnant de fonctionnalités et de services à valeurs ajoutées notamment le fait de pouvoir personnaliser entièrement son espace personnel et cela de manière très simple et sans aucune connaissance technique. C'est là, la preuve que les fonctionnalités et la simplicité d'utilisation sont primordiales dans le succès d'un réseau social.

D'autres théories ont été écrites sur les réseaux sociaux. D'après Robin Dunbar (anthropologiste) et Malcom Gladwell (auteur du livre *Le point de bascule*), le cerveau humain n'est pas équipé pour entretenir plus de 150 connexions sociales actives à un instant T. Ainsi, il est plus important de se concentrer sur la

qualité de ses connexions plutôt que sur le nombre. Une dernière théorie, dans le même sens que la précédente, est celle « des liens forts et des liens faibles » développée par Alain Lefebvre dans le livre blanc *“les réseaux sociaux pour professionnels”*. Les liens forts sont des liens quasi naturels (famille, amis et collègues proches) et facilement entre tenables. Les liens faibles sont quant à eux de court terme et très contextuels, voire occasionnels ou uniques. Cependant, ce sont les liens faibles les plus porteurs d’opportunités. Ainsi, les liens et la mise en relation sont les points clé de tous réseaux : construire son réseau de façon pertinente et l’entretenir régulièrement en plus du long terme sont deux facteurs de garantie pour un réseau performant.

I.7 Les principaux réseaux sociaux

- **Facebook** : ce réseau social, comme il se décrit lui-même, est apparu en 2004 sous la forme d’une communauté privée d’étudiants, qui s’est développée depuis en un outil reliant des personnes et des entreprises du monde entier. Sur Facebook, les utilisateurs peuvent se connecter et partager des informations de différentes façons. Sur Facebook on y crée un compte (aussi appelé « profil ») afin de rester en contact avec des membres de son entourage, Chaque membre y dispose d’une page personnelle pouvant être personnalisée de plusieurs manières : photo, texte de présentation, date d’anniversaire, profession, goûts musicaux, message s’affichant à côté du pseudonyme et reflétant l’humeur du moment. Pour ajouter une personne à son réseau et devenir son « ami » (terme employé par Facebook pour désigner les contacts), il est nécessaire de lui envoyer une demande. Chaque mise en relation est réciproque : si j’accepte une personne comme amie, elle fait alors partie de mon réseau et je fais automatiquement partie du sien. L’activité du réseau s’organise autour de la page d’accueil de chaque membre.



Figure I.2 : page d’accueil du réseau sociale Facebook.

- **Twitter** : Twitter est une plateforme de microblogging sur laquelle les messages (appelés tweets) sont limités à 140 caractères (plus ou moins l'équivalent du statut de Facebook). Cette spécificité en fait un réseau social à part, beaucoup plus épuré que Facebook en termes de contenu. Ici, l'identification d'un membre se fait uniquement via la photo, le nom, la localisation et éventuellement un lien vers un site Web. La plupart du temps publics, les profils peuvent également être privés (ils nécessitent un accord préalable de leur créateur pour être suivis). A chaque réseau sa terminologie : un membre de Twitter n'a pas d'amis ou de fans, il est abonné à un certain nombre de comptes (il s'agit du nombre d'Abonnements ou Following), et un certain nombre de personnes sont abonnées à son compte (il s'agit du nombre d'abonnés ou Followers). Pour suivre une personne, la démarche est très simple : il suffit de disposer d'un compte sur Twitter et de cliquer sur l'icône Suivre (Follow) en haut de la page du compte que l'on souhaite suivre. La personne en question est alors immédiatement ajoutée à mon réseau. Contrairement à Facebook, cette procédure n'est pas réciproque. Ce n'est pas parce que j'ajoute une personne à mon réseau sur Twitter que je fais partie du sien.

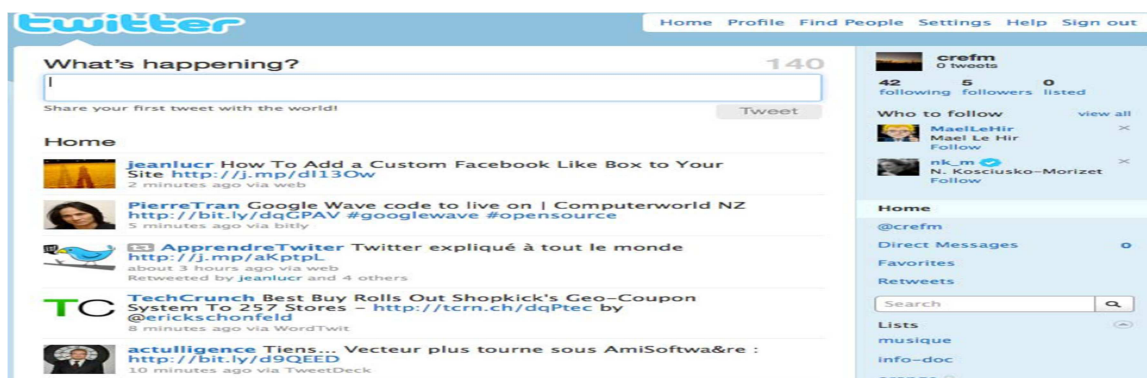


Figure I.3 : Page d'accueil du réseau sociale Twitter.

- **LinkedIn** : LinkedIn se décrit lui-même comme « un réseau interconnecté de professionnels expérimentés du monde entier. Les internautes l'utilisent généralement comme outil de recrutement ou de gestion des contacts professionnels, mais beaucoup d'autres utilisations sont possibles. L'adhésion au réseau est gratuite, mais des outils complémentaires payants sont disponibles si les internautes veulent profiter des fonctionnalités avancées.

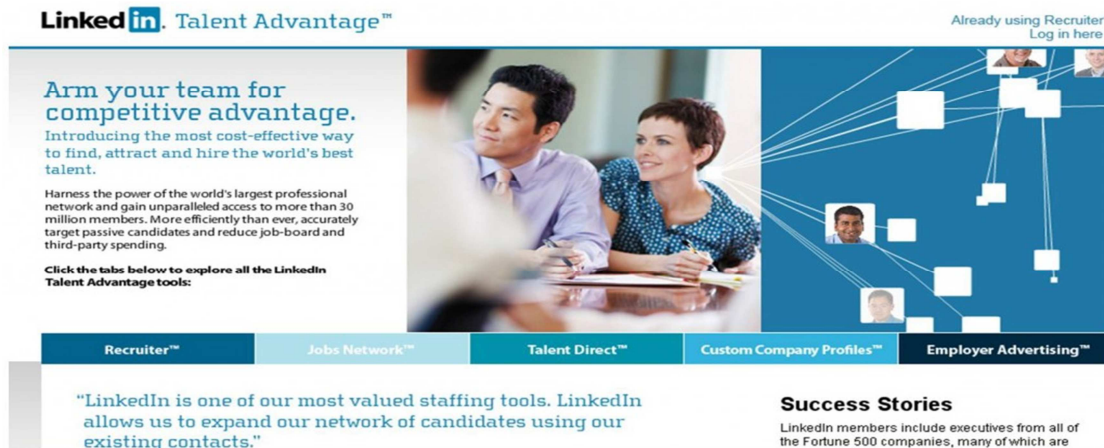


Figure I.4 : Page d'accueil du réseau sociale LinkedIn

- **Bebo** : Bebo, acronyme de « Blog early, blog often » (bloguez tôt, bloguez souvent), met en avant son rôle de « passerelle entre les internautes, leurs amis, et tout ce qui les intéresse ». Comme de nombreux réseaux sociaux, il permet aux internautes de se connecter et de partager des informations via des pages de profil personnel. Ils peuvent envoyer la musique ou les livres dont ils sont les auteurs pour les rendre publics. La plateforme Open Media de Bebo permet aux sociétés telles que CBS, Sky et la BBC de diffuser du contenu aux utilisateurs de Bebo via des pages spécialisées conçues pour la vidéo.

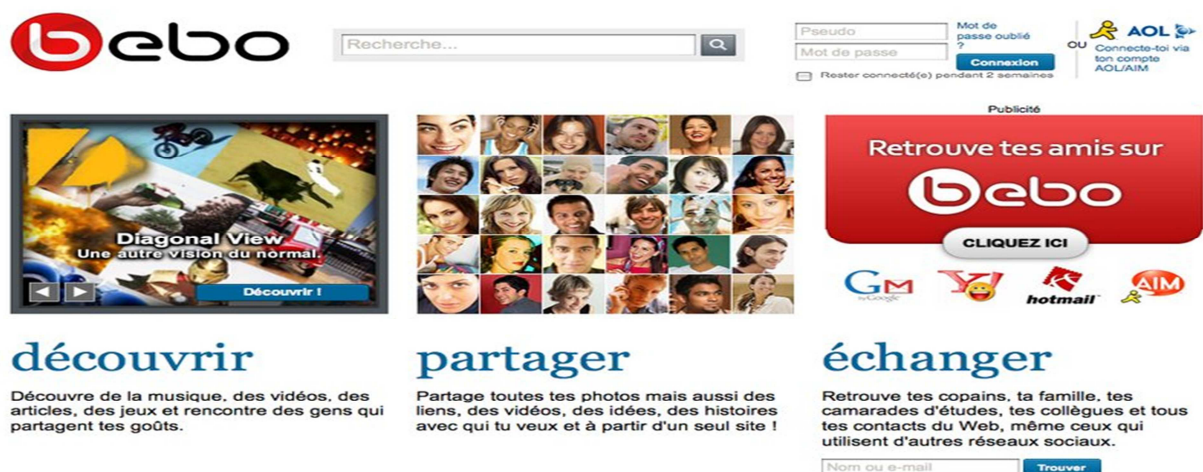


Figure I.5 : Image représentante le réseau sociale Bebo.

- **MySpace** : MySpace est une « communauté en ligne qui te permet de rencontrer les amis de tes amis ». Il s'agissait du réseau social le plus populaire aux États-Unis jusqu'en avril 2008, lorsqu'il a été détrôné par Facebook. Des personnalités politiques telles que Barack Obama et plusieurs partis ont utilisé MySpace pour se connecter à leur électeurat et augmenter leur nombre de membres. Myspace reste un

réseau social très populaire auprès d'une communauté jeune et reste la référence pour les groupes et musiciens souhaitant partager leur musique avec les internautes.



Figure I.6 : Page d'accueil du réseau sociale Myspace

I.9 Conclusion

C'est une tendance progressive que les gens révèlent de plus en plus de l'information très personnelle sur les sites des réseaux sociaux en particulier et dans le World Wide Web en général. Comme cette information devient de plus en plus publiquement disponible de ces plusieurs sites des réseaux sociaux et le web en général, les rapports sociaux entre les gens peuvent être identifiés. Cela permet l'extraction automatique des réseaux sociaux. Cette tendance est en outre conduite et appliquée par les initiatives récentes tel que facebook, la disponibilité de la donnée de MySpace et le Friend Connect de Google en rendant leurs données du réseau sociale disponible à n'importe qui.

Annexell

II.1 Introduction

La Recherche d'Information (RI) est un domaine de l'informatique qui s'intéresse à la collection, le stockage, l'organisation, la recherche et à la sélection d'informations répondant aux besoins des utilisateurs en information.

Ce domaine manipule différents concepts : la requête, le besoin en information, les documents, la pertinence...etc.

La recherche d'information (RI) vise à satisfaire les exigences des utilisateurs en informations en mettant en place un mécanisme d'appariement entre requête utilisateur et documents d'une base documentaire pour restituer les documents pertinents.

D'un point de vue utilisateur, l'accès à l'information peut être effectué de manière délibérée à travers un Système de Recherche d'Information (SRI) (on parle alors de collecte active de l'information), ou bien de manière passive à travers un système de filtrage d'information (c'est un processus qui permet d'extraire à partir d'un flot d'informations « News, e-mail, actualités journalières, etc. », celles qui sont susceptibles d'intéresser un utilisateur ou un groupe d'utilisateurs ayant des besoins en informations).

II.2 Définition de la RI (Recherche d'Information)

La recherche d'information selon [Salton, 1970] « est l'ensemble des techniques permettant de gérer des textes ».

D'après [Urfi, 2004], « La recherche d'information (RI) est un ensemble de méthodes et de procédures ayant pour objectif d'extraire d'un ensemble de document, les informations voulues. Dans un sens plus large, la RI est toute opération (ou ensemble d'opérations) ayant pour objectif la recherche, la collecte et l'exploitation d'informations en réponse à une question sur un sujet précis ».

II.3 Définition d'un SRI (Système de Recherche d'Information)

Un SRI est un ensemble de programmes informatiques qui a pour but de sélectionner des informations pertinentes répondant aux besoins des utilisateurs, exprimées sous forme de requêtes.

Selon [Salto et Gill, 1984] «Un système de recherche d'information traite de la représentation du stockage, de l'organisation et de l'accès aux éléments de l'information».

Afin de répondre aux informations voulues par l'utilisateur, un SRI met en œuvre un certain nombre de processus pour réaliser la mise en correspondance des informations contenues dans un fond documentaire d'une part, et des besoins en informations des utilisateurs d'autre part.

Le processus de recherche, couramment appelé Processus en U de Recherche d'Information est schématiquement représenté sur la figure 1.

Ce processus est composé de trois fonctions principales :

- l'indexation des documents et des requêtes ;
- l'appariement requête-document, permettant de comparer la requête et le document ;
- Reformulation de Requête qui permet de réécrire autrement la requête utilisateur ;

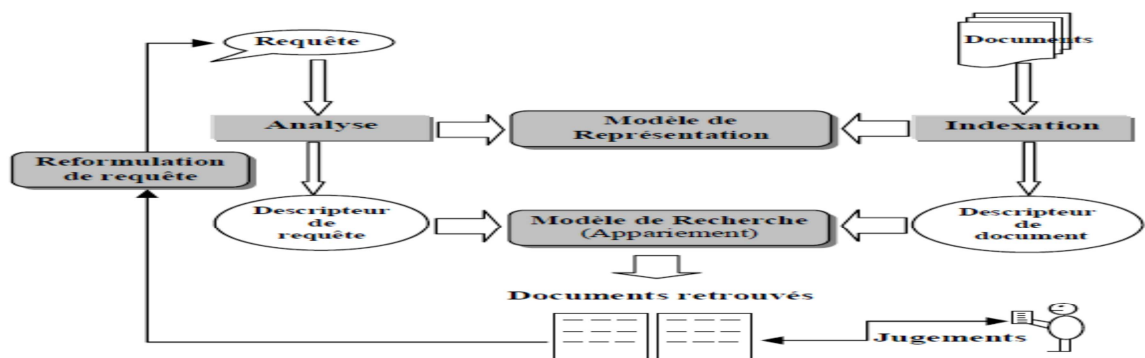


Figure I.1 : Processus en U de recherche d'information.

Cette figure représente les principaux concepts de base d'un SRI.

II.4 Notion de pertinence

La pertinence est la notion centrale dans la RI car toutes les évaluations s'articulent autour de cette notion, c'est aussi la notion qui constitue le cœur du SRI. Voyons ainsi quelques définitions de la pertinence :

- La correspondance entre un document et une requête, une mesure d'information du document à la requête.
- Un degré de relation (chevauchement, relativité, ...) entre le document et la requête.
- Un degré de la surprise qu'apporte un document, qui a un rapport avec le besoin de l'utilisateur.
- Une mesure d'utilité du document pour l'utilisateur.

Il existe deux formes de pertinence : système et utilisateur.

II.5 Le processus d'indexation

Pour que le coût de la recherche soit acceptable, il convient d'effectuer certains prétraitements sur les documents et les requêtes, c'est ce qu'on appelle l'indexation. L'indexation est une étape primordiale dans la conception d'un SRI, préalable à la recherche. Cette opération vise à construire une structure d'index qui permet de retrouver très rapidement les documents incluant des mots demandés. La structure d'index est de la forme suivante :

Mot $\longrightarrow$ { ..., Doc, ... }

C'est-à-dire, chaque mot est mis en correspondance avec les documents qui le contiennent.

L'indexation peut être :

- ✓ **manuelle** : chaque document est analysé par un spécialiste du domaine ou par un documentaliste. Elle consiste à analyser le document afin d'extraire une liste de mots clés les plus significatifs, cette méthode est coûteuse et très longue à mettre en œuvre notamment dans une base documentaire importante.
- ✓ **automatique** : il s'agit d'un processus automatique qui analyse les documents pour en extraire les mots clés les plus significatifs représentant le contenu.
- ✓ **semi-automatique** : Elle utilise l'indexation automatique pour déterminer les éléments d'indexation, mais le choix final revient au spécialiste ou au documentaliste, qui intervient souvent pour choisir d'autres termes significatifs.

L'indexation manuelle permet d'assurer une meilleure pertinence dans les réponses apportées par le SRI, elle présente toutefois plusieurs inconvénients :

Deux indexeurs différents peuvent présenter des termes différents pour caractériser un même document, et un indexeur à deux moments différents peut présenter deux termes distincts pour représenter le même concept.

Afin de pallier aux inconvénients des systèmes manuels, des procédures d'indexation automatiques ont été mises au point, on distingue : l'extraction automatique des mots des documents (l'analyse lexicale), l'élimination des mots vides, la lemmatisation (radicalisation ou normalisation), la pondération des mots.

Procédure de l'indexation automatique

- **L'analyse lexicale** : L'analyse lexicale est le processus qui permet de convertir le texte d'un document en un ensemble de termes. Un terme est une unité lexicale ou un radical [Fox, 1992].

L'analyse lexicale permet de reconnaître les espaces de séparation des mots, des chiffres, des ponctuations, etc.

- **L'élimination des mots vides** : Un des problèmes majeurs de l'indexation consiste à extraire les termes significatifs et à éviter les mots vides. Ces mots sont souvent des prépositions (" de ", " à "), pronom (" aucun ", " tout ", " on "), certain adverbess (" ailleurs ", " maintenant "), adjectifs (" certain ", " possibles "), etc.

Les mots vides peuvent aussi être des mots athématiques (les mots qui peuvent se retrouver dans n'importe quel document parce qu'ils exposent le sujet mais ne le traitent pas, comme par exemple contenir, appartenir, etc). On distingue deux techniques pour éliminer les mots

vides :

- L'utilisation d'une liste de mots vides (aussi appelée anti-dictionnaire),
- L'élimination des mots dépassant un certain nombre d'occurrences dans la collection.

- **La lemmatisation**

La lemmatisation désigne l'analyse lexicale du contenu d'une page (ou d'un document) regroupant les mots d'une même famille, chacun des mots d'un contenu se trouve ainsi réduit en une entité appelée lemme.

La lemmatisation regroupe les différentes formes que peut revêtir un mot, soit le nom, le pluriel, le verbe à l'infinitif, etc.

Par exemple on peut citer médecine, médical, médicalement, médecin, etc. et il n'est pas forcément nécessaire d'indexer tous ces mots alors qu'un seul suffirait à représenter le concept.

Frakes et Baeza-yates [Frak, 1992] distinguent cinq types stratégiques de lemmatisation, la table de consultation (dictionnaire), l'élimination des affixes (on peut par exemple citer l'algorithme de Porter [Port, 1980]), la troncature, les variétés de successeurs ou encore la méthode des n-gramme [Adam et Bore, 1974].

II.6 Pondération des termes

La pondération des termes permet de mesurer l'importance d'un terme dans un document. Cette importance est souvent calculée à partir de considérations et d'interprétations statistiques (ou parfois linguistiques). L'objectif est de trouver les termes qui représentent le mieux le contenu d'un document.

Si on dresse une liste de l'ensemble des mots différents d'un texte quelconque classés par ordre de fréquences décroissantes, on constate que la fréquence d'un mot est inversement proportionnelle à son rang de classement dans la liste.

Cette constatation est énoncée formellement par la loi de Zipf [Zipf, 1949] :

$$\text{Rang} * \text{Fréquence} = \text{Constante}$$

La loi de distributions des termes suit alors la courbe. Zipf explique la courbe hyperbolique de la distribution des termes par ce qu'il appelle le principe du moindre effort : il considère qu'il est plus facile pour un auteur d'un document de répéter certains termes que d'en utiliser de nouveaux. La relation entre la fréquence et le rang des termes permet de sélectionner les termes représentatifs d'un document : on élimine respectivement les termes de fréquences très élevées car ils ne sont pas représentatifs du document (on peut par exemple citer les mots outils, mots vides), En français, les mots "de", "un", "les", etc. sont les plus fréquents. Ce

processus est illustré sur la figure.2. En utilisant cette approche, le nombre de termes faisant partie de l'index d'une collection peut être réduit considérablement.

A partir de ces constatations, des techniques de pondération ont vu le jour.

La plupart de ces techniques sont basées sur les facteurs *tf* et *idf*, qui permettent de combiner les pondérations locales et globales d'un terme :

- **tf (term frequency)** : cette mesure est proportionnelle à la fréquence du terme dans le document (pondération locale). Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons ($\log(tf)$, présence/absence, . . .).
- **idf (inverse of document frequency)** : ce facteur mesure l'importance d'un terme dans toute la collection (pondération globale). Un terme qui apparaît souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent.

La mesure $tf * idf$ donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de taille homogène, cependant, elle ne tient pas compte d'un aspect important du document (sa longueur). En général, les documents les plus longs ont tendance à utiliser les mêmes termes de façon répétée, ou à utiliser plus de termes pour décrire un sujet. Par conséquent, les fréquences des termes dans les documents seront plus élevées, et les similarités à la requête seront également plus grandes.

Pour pallier à cet inconvénient, Robertson [Rober et al, 1994] et Singhal [Sing et al, 1996.] Proposent d'intégrer la taille des documents à la formule de pondération, on parle alors de facteur de normalisation.

II.7 Modèle de recherche d'information

Un modèle de RI a pour rôle de fournir une formalisation du processus de recherche d'information. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de la mesure de pertinence. Il existe plusieurs modèles de RI dont nous allons présenter quelques-uns :

II.7.1 Modèle booléen

Apparus dans les années 1950 se basent sur la théorie des ensembles, ainsi un tel modèle renvoyant un ensemble de documents jugés pertinents sans en proposer un ordonnancement est écarté.

Le modèle booléen [Salton, 1984] est le plus simple des modèles de RI, c'est aussi le premier qui s'est imposé dans le monde de la recherche d'information. Il est basé sur la

théorie des ensembles et l'algèbre de Boole. Le modèle booléen considère que les termes de l'index sont présents ou absents d'un document.

Ainsi, le modèle booléen affirme que chaque document est soit pertinent, soit non-pertinent, il n'y a pas de notion de réponse partielle aux conditions de la requête.

II.7.2 Modèle vectoriel

Le premier système vectoriel de recherche d'information apparaît dans les années 1970 avec le système SMART [Salton, 1970]. Dans le modèle vectoriel, les vecteurs de poids représentent le document et la requête. Chaque poids dans le vecteur désigne l'importance du terme correspondant dans le document ou dans la requête. Pour qu'un vecteur prenne une signification, il faut préalablement définir un espace vectoriel. L'espace engendré par les N termes d'indexation ($t_1, t_2, t_3, \dots, t_N$) que le système a rencontré durant l'indexation, c'est-à-dire l'ensemble des termes de la collection de documents.

II.7.3 Modèle de Langage (Modeling Language)

Les SRI utilisant les modèles de langages suivent une approche différente des autres modèles. En effet, dans la plupart des modèles, on cherche à comparer une représentation de la requête de l'utilisateur avec une représentation du document recherché, pour évaluer la pertinence de celui-ci. Ici, on part de l'observation que, l'utilisateur crée la requête à partir d'une représentation hypothétique qu'il se fait du document recherché.

La requête est donc, inférée par l'utilisateur à partir des documents voulus. Le but du modèle de langage est de générer des requêtes à partir des documents, et comparer celles-ci avec la requête de l'utilisateur. La pertinence d'un document est estimée en calculant la probabilité que la requête utilisateur soit inférée par celui-ci.

II.8 Evaluation d'un système de RI

L'évaluation d'un système de recherche d'information peut être appréhendée selon deux aspects : un aspect efficience et un aspect efficacité. L'aspect efficience dépend de l'évaluation cognitive de l'utilisateur, tels que la facilité d'utilisation du système, rapidité d'accès, temps de réponse à une requête, présentation des résultats, etc. L'aspect efficacité concerne la capacité du système à sélectionner le maximum de documents pertinents et un minimum de documents non pertinents. Nous nous intéressons dans cette section à présenter l'aspect efficacité, qui est souvent mesuré par deux paramètres Rappel/Précision.

- **Mesures de précision et rappel**

Ce sont les deux métriques les plus utilisées pour évaluer la performance d'un système de recherche d'information.

Les taux de rappel et de précision sont définis comme suit :

Taux de rappel : le rappel mesure la capacité du système de retrouver tous les documents pertinents répondant à une requête. Autrement dit, il mesure le pourcentage de documents pertinents, selon une liste de référence (ensemble des documents pertinents de la collection), que le système a trouvé :
$$\text{Rappel} = \frac{Pr}{R}$$

Taux de précision : la précision mesure la capacité du système de renvoyer tous les documents pertinents à une requête. Autrement dit, elle mesure le pourcentage de documents retrouvés qui sont pertinents :
$$\text{Précision} = \frac{Pr}{Dr}$$

- **Mesures de bruit et silence**

Le bruit : Il mesure le taux de documents pertinents non extraits par rapport à la totalité des documents extraits.
$$\text{Bruit} = \frac{\text{Nombre}(\text{Documents non extraits})}{\text{Nombre}(\text{Documents extraits})}$$

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

$$\text{Silence} = \frac{\text{Nombre}(\text{Documents Pertinents non extraits})}{\text{Nombre}(\text{Documents pertinents})}$$

- La performance d'un SRI peut être appréciée si d'un côté le taux de précision et de rappel est élevé, et de l'autre le taux de bruit et de silence est bas.
- Un rappel égal à 1 indique que tous les documents pertinents ont été retrouvés, une précision égale à 1 indique que tous les documents retrouvés sont pertinents.

II.9 Conclusion

Dans cette partie nous avons présenté les principales notions et concepts de base de la recherche d'information avec les différents modèles permettant de mesurer la similarité requête-document afin de déterminer les documents pertinents à l'utilisateur.

Annexe III

III.1 Historique de java

Le langage java trouve ses origines dans les années 90. A cette époque, quelques ingénieurs (innovateurs) de Sun Microsystems commencent à parler d'un projet d'environnement indépendant de hardware pouvant facilement permettre la programmation d'appareil aussi variés que les téléviseurs, magnétoscope etc... James Gosling (Sun Microsystems) développe un premier langage « Oak », permettant de programmer dans cet environnement. En 92, tout est prêt pour envahir le marché avec cette technologie mais la tentative se solde par un échec.

Bill Joy (co-fondateur de Sun Microsystems) sauve malgré tout le projet. Devant la montée en puissance de l'internet, il lui semble intéressant d'insister sur l'élaboration d'un langage indépendant des plates-formes et des environnements (les principaux problèmes rencontrés sur internet étant liés à l'hétérogénéité des machines et des logiciels utilisés).

Dès lors tout s'accélère. « Oak » est renommé (en 1995) **Java** est soumis à la communauté Internet grandissante. Une machine virtuelle, un compilateur ainsi que de nombreuses spécifications sont données gratuitement et java entame une conquête figurante. Aujourd'hui, après de nombreuses améliorations (parfois modification) Java n'est plus uniquement une solution liée à l'Internet.

De plus les qualités et la maturité du langage ainsi que la liste grandissante des bibliothèques des classes disponibles, on fait de java un acteur de plus incontournable dans la conception de solutions dédiées non seulement à internet mais aussi aux applications classiques, interface homme-machine, applications réseaux.

III.2 Les principales raisons du succès de java

Java a rapidement intéressé les développements pour 4 raisons principales :

- ✓ C'est un langage orienté objet dérivé du C, mais plus simple à utiliser et plus « pur » que le C++. On entend par « pur » le fait qu'en java, on ne peut faire que de la programmation orienté objet contrairement au C++ qui reste un langage hybride c'est-à-dire autorisant plusieurs styles de programmation. C++ est hybride pour assurer une compatibilité avec le C.
- ✓ Il est doté, en standard, de bibliothèque de classes très riches comprenant la gestion des interfaces graphiques (fenêtre, boîte de dialogue, contrôle, menu, graphismes). La programmation multithread (multitâche), la gestion des exceptions, les accès aux fichiers et au réseau...l'utilisateur de ces bibliothèques facilitent grandement la tâche du programmeur lors de la construction d'application complexe.

- ✓ Il est doté, en standard, d'un mécanisme de gestion des erreurs (les exceptions) très utile et très performant. Ce mécanisme, inexistant en C, existe en C++ sous la forme d'une extension au langage beaucoup moins simple à utiliser qu'en Java.
- ✓ Il est multi plates-formes : les programmes tournent sans modification sur tous les environnements où Java existe (Windows, Unix et Mac).

III.3 Les caractéristiques de Java

- Orienté objet
- Simple
- Robuste
- Sécurité
- Doté d'une bibliothèque de classe très complète
- Multithreads
- Indépendant de l'architecture matérielle
- Interprété
- Portable
- Dynamique
- Performant

Terminologie

En Java à l'exception de quelques primitives, tout est objet. Cette organisation des composants logiciels en objets, contribue à faciliter la modélisation et l'implémentation d'une solution ainsi que la réutilisabilité des composants développés.

Le rôle des technologies orientées objets, est de fournir un certain nombre de mécanismes destinés à organiser ces composants en objets. Dans ce qui suit nous allons présenter les différents concepts utilisés par une telle technologie.

III.3.1 Encapsulation

- ✓ **Notion d'objet :** C'est une entité qui possède une identité et caractéristique, et qui est susceptible d'effectuer des actions. Un objet est caractérisé par plusieurs notions :
- ✓ **L'identité :** l'objet possède une identité, qui permet de distinguer des autres objets, indépendamment de son état. On construit généralement grâce à un identifiant découlant naturellement du problème (par exemple un employé pourra repérer par un numéro de sécurité sociale).

- ✓ **Les attributs :** Il s'agit des données caractérisant l'objet. Ceux sont des variables stockant des informations sur l'état de l'objet.
- ✓ **Les méthodes :** Les méthodes d'un objet caractérisant son comportement, c'est-à-dire l'ensemble des actions (opérations) que l'objet est amené à réaliser. Ces opérations permettent de faire réagir l'objet aux sollicitations extérieures (ou d'agir sur les autres objets). De plus, les opérations sont étroitement liées aux attributs car, leurs actions peuvent dépendre des valeurs des attributs ou bien les modifier.
- ✓ **Notion de classes :** Elle peut être définie comme le type d'un objet, et comme un objet est capable de gérer à la fois ses données et ses actions, alors une classe doit être définie par une structure de données et un comportement. Les objets qui possèdent les mêmes caractéristiques (structures de données et comportements) sont regroupés en une classe.

Dans un programme, l'objet est créé à partir d'une classe on dit qu'il s'agit d'une instance de cette classe.

- ✓ **Un package :** C'est une bibliothèque, servant à structurer un ensemble de classes. Cette organisation facilite à la fois la recherche de l'emplacement physique des classes quand cela est nécessaire et la distinction entre différentes classes de même nom. Elle permet en outre, de nuancer des niveaux d'accès aux membres appartiennent au non au même package, et d'une manière globale, rend l'ensemble plus lisible.

II1.3.2 Le polymorphisme

Offre la possibilité d'associer à un comportement, une implémentation différente de l'objet auquel on se refait (ex : dessiner à l'écran un carré et un cercle implique dans les deux cas de manipuler un certain nombre de pixels, cependant la manière d'implémenter ce tracé se fera différemment). Le polymorphisme augmente la généricité, et donc la qualité du code.

II1.3.3 L'héritage

L'héritage permet à une classe d'objets de bénéficier de la structure de données et du comportement d'une classe « mère », afin de prendre en compte les spécificités de la classe « fille », sans avoir à redéfinir ce que les deux classes ont de commun.

II1.3.4 L'abstraction

Introduit le mécanisme favorisant la dissociation entre la déclaration d'une classe et son implémentation.

III.4 Le Kit de Développement Java(JDK)

Ecrire un programme Java ne nécessite pas d'outils spécifiques : un éditeur de texte tel Notepad suffit largement. Le programme Java, qui porte l'extension **.java** doit être compilé afin de générer un fichier bytecode, pourtant l'extension **.class**, qui sera interprétable par une machine virtuelle Java.

Pour se faire, il est nécessaire de télécharger sur le site de Sun (www.javasoft.com) Le JDK (Java Développement Kit). Ce Kit est disponible pour différentes plates-formes de développement. Il en existe, en outre, plusieurs versions : 1.0.x, 1.1.x, 1.2.x, 1.3.0 etc.

Ce Kit de développement comprend plusieurs outils (le nombre varié selon la version de JDK), ne disposant pas tous d'interface graphique, dont voici les principaux :

-**Javac** : il s'agit du compilateur java. Il traduit un fichier source d'extension **.java** en un fichier bytecode d'extension **.class**. une partie de ses fonctions concerne la sécurité afin que le code généré ne risque pas d'effectuer des instructions illégales.

-**java** : c'est un interpréteur java, c'est-à-dire une implémentation de la machine virtuelle java. Il traduit en langage machine les fichiers déjà compilés par **Javac**, après avoir effectué un certain nombre de contrôle (**ex** : s'assurer que l'exécution d'un programme en prévenance d'un serveur distant, ne porte préjudice à l'intégrité d'une machine locale).

-**AppletViewer** : ce programme est un interpréteur java qui possède la spécificité de ne pouvoir exécuter que des applets. Disposant de sa propre interface graphique, il permet de tester ces derniers sans avoir recours à un navigateur web.

-**JRE** : c'est un interpréteur allégé de java qui n'a pas besoin d'installer tous le JDK pour exécuter un programme compilé par le javac. Il est destiné à des plates-formes clientes qui ne développent pas d'application Java mais qui les utilisent.

-**Jdb** : ce débogueur permet de détecter les erreurs de programmation.

-**Javadoc** : cet utilitaire permet de construire, à partir des commentaires insérés dans des sources Java et sous condition que ces derniers respectent une certaine syntaxe, des fichiers HTML documentant les classes, méthodes,... développées dans des sources.

-**Jar** : cet utilitaire permet d'archiver plusieurs classes Java en un fichier d'archive Jar ou exécutable Jar (à partir de la version 1.1 du JDK).

Parmi les utilitaires utilisés, on a aussi : Java, Jit, Native2ascii, Rmic, Rmirejistry, javakey, Keytool, Jarsigner, Policytool,....

III.5 Les API Java

Les API de base sont fournies par le JD. Ces derniers sont structurés en package, elles contiennent des classes pouvant être utilisées pour construire des programmes plus complexes.

Les classes rassemblées sous un même package, présentent généralement des fonctions fortement connexe.

-Java.lang : ce package fournit des classes de base pour la conception du langage de programmation Java.

-Java.util : ce package contient des classes qui permettent de manipuler les vecteurs, les dates, des fichiers Zip, des fichiers d'archive jar.

-Java.applet : ce package fournit les classes nécessaires à la création d'une applet ainsi qu'à la communication entre une applet et son contexte.

-Java .awt : ce package contient toutes les classes pour créer des interfaces graphiques pré complexes, et pour manipuler des graphiques et des images.

-java.io : ce package fournit des classes pour les entrées /sorties du système.

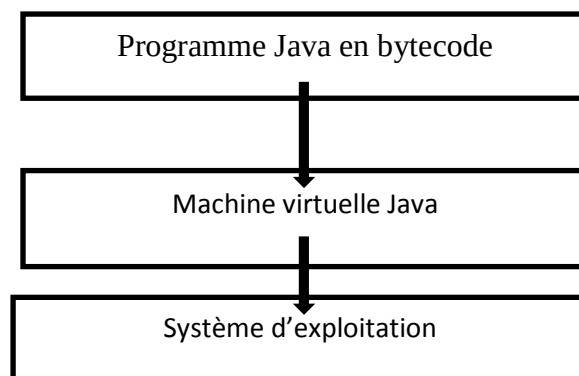
-Java .net : ce package fournit des classes pour implémenter des applications réseaux.

-Java.math : ce package fournit des classes utilisées pour l'arithmétique des nombreux entiers et des nombres décimaux à précision arbitraire.

-java.security : ce package fournit des classes permettant de mettre en œuvre de procédé de sécurité tel que la gestion des droits d'accès à différentes ressources (fichiers, réseaux, ...).

III.6 La machine virtuelle Java (JVM)

Une fois le programme source Java traduit en bytecode par le compilateur Javac, la machine virtuelle java se charge de l'exécuter sur la machine physique à travers son système d'exploitation (Windows, Unix, MacOS, ...).



Bibliographie

Bibliographie

- [Amer *et al*,2007] Amer-Yahia, S., M. Benedikt, et P. Bohannon (2007). Challenges in searching online communities. *IEEE Data Eng. Bull.* 30(2), 23–31.2007.
- [Adam and Bore, 1974] G. Adamson and J. Boreham, The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval*, 10: pages 253–60, 1974.
- [Bao *et al*, 2007] Bao S., Wu X., Fei B., Xue G., Su. Z., Yu Y., « Optimizing web search using social annotations », *World Wide Web Conference. 2007*
- [Benj *et al*, 2010] L. Ben Jabeur, L. Tamine et M. Boughanem, Un modèle de Recherche d'Information Sociale pour l'Accès aux Ressources Bibliographiques : Vers un réseau social », *Atelier Recherche et Recommandation d'Information dans les Réseaux Sociaux, REISO10*, 2010.
- [Benb ,2010] Benbunan-Fich R., (2010). Koufaris M., An empirical examination of the sustainability of social bookmarking websites, *Information Systems and e-Business Management*, Vol. n°8, Issue 2, Pages 131-148. January 2010.
- [Begg *et al*, 1993] M. Begg, J. Gnocato, et W. E. Moore. A prototype intelligent user interface for real-time supervisory control systems. Dans *Proceedings of the 1st international conference on intelligent user interfaces*, pages 211–214, New York, NY, USA, 1993.
- [Baeza *et al*,1999] Baeza-Yates, R. et B. Ribeiro-Neto. *Modern Information Retrieval*. New York: ACM Press. 1999.
- [Boug,2005] BOUGHANEM.M, "Recherche d'information" Université Paul Sabatier de Toulouse.2005.
- [Bour et Ande, 1979], Bourne & Anderson "Dialog: Labworkbook. Dans *LockheedvInformation Systems*", pages 640–644. PaloAlto, Californie (USA), 1979.
- [Chen *et al*, 1998] Chen et K. Sycara. Webmate : A personal agent for browsing and searching. Dans *Proceedings of the 2nd international conference on autonomoussagents and multi agent systems, Minneapolis*, pages 10–13, 1998.
- [Clayp, 2000] Claypool, M., Le, P., Wased, M., and Brown, D., Implicit interest indicators. In *Proc. 6th international conference on Intelligent User Interfaces*, 33-40.2001. 2000.
- [Dum *et al*, 2003] Dumais, Cuttrel E., J.J. Cadiz, G. Jancke, R. Sarin, et D.C. Robbins. Stuff i've seen: a system for a personal information retrieval and re-use. Dans *Proceedings of the 26th ACM SIGIR*, pages 72–79. Toronto, Juillet 2003b.
- [Dich *et al*,2008] Dichev et al., A study of community formation in collaborative tagging systems, 2nd Workshop on Collective Intelligence in Semantic Web and Social Networks (CISWSN 2008), 2008.
- [Farg et perny,2003] Fargier et P. Perny. A characterisation of generalised concordance rules in multicriteria decision making. In *International Journal of*

- Intelligent Systems (IJIS)*, 18 :751–774, 2003.
- [Fell, 2006] [S.FELLAG, Mémoire de magister “Recherche d’information dans les documents semi-structurés XML “, 2006
- [Fox, 1992] Lexical analysis and stoplists, pages 102–130. Frakes WB, Baeza-Yates R (eds) Prentice Hall, New jersey, 1992.
- [Frak, 1992] W. B. Frakes Stemming Algorithms, pages 131–160. Frakes W B, Baeza-Yates R (eds) Prentice Hall, New jersey, 1992.
- [Furuk et al,2007] Furukawa et al., Social networkds and Reading Behavior in the blogosphere, ICWSM 2007.
- [Gent et al, 2003] Gentili, A. Micarelli, et F. Sciarrone. Infoweb : An adaptive information filtering system for the cultural heritage domain. *Applied Artificial Intelligence*, 17(8-9) :715–744, 2003.
- [Clayp et al, 2000] Claypool, M., Le, P., Wased, M., and Brown, D., Implicit interest indicators. In *Proc. 6th international conference on Intelligent User Interfaces*, 33-40. 2001.
- [Goec et al, 2000] Goecks, J. and Shavlik, J.,. Learning users’ interests by unobtrusively observing their normal behavior. In *Proc. 5th international conference on Intelligent user interfaces*, 129-132. 2000.
- [Golde,2005] Golder S. A. et Huberman B. A. : The structure of collaborative tagging systems. *CoRR—Computing Research Repository*, abs/cs/0508082, 2005.
- [Gowan , 2003] McGowan. A multiple model approach to personalised information access. Master Thesis in computer science, Faculty of science, University College Dublin, 2003.
- [Hotho et al, 2006] Hotho A., Jäschke R., Schmitz C. et Stumme G. : Information retrieval in folksonomies: Search and ranking. *Dans Proceedings of the 3rd European Semantic Web Conference*, volume 4011 de LNCS, p. 411–426, Budva, Montenegro, June 2006.
- [Joac et al, 2007] Joachims, L. Granka, H. Hembrooke, F. Radlinski, et G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information systems*, 25(2) :7, April 2007.
- [Jans et al, 2006] J. Jansen, A. Spink, et V. Kathuria. How to define searching session’s onweb search engines. *Dans In WEBKDD’06*, pages 92–109, 2006.
- [Jung,2001] Jung, K., 2001. *Modeling web user interest with implicit indicators*, Master Thesis, Florida Institute of Technology.
- [Kons et al,2008] Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195{202, New York, NY, USA, 2009. ACM. 2008.
- [Kirc et al, 2008] Kirchhoff L., Stanoevska-Slabeva K., Nicolai T., et Fleck M., « Using social network analysis to enhance information retrieval systems », *Applications of Social Network Analysis (ASNA)*, 2008.
- [Kirsch,2005] Kirsch S.M., Social Information Retrieval, Thèse de doctorat, Université de Rheinische Friedrich-Wilhelms, 2005.
- [Kautz et al,1997] Kautz et al., ReferralWeb: Combining Social Networks and

- [Kal, 2001] Collaborative Filtering, *Communication of the ACM*, 40:3, p. 63, 1997.
- [Lelu et Fran, 1992]. Kim, J., Oard, D.W., and Romanik, K., *Using implicit feedback for user modeling in internet and intranet searching*. College of Library and Information Services, University of Maryland. 2001.
- [Liber, 1995] A.Lelu & C.François " Information retrieval based on a neural unsupervised extraction of thematic fussy clusters, neuro-nîmes 92 : les réseaux neuro-mimétiques et leurs applications", nîmes, France. 2-6 novembre 1992.
- [Li et al, 1999]]Liberman, H.,. Letizia: An Agent that assists web browsing. In *Proc. IJCAI*, 924-929. 1995.
- [Liu et al, 2004] Li, W.S., Vu, Q., Agrawal, D., Hara, Y., and Takano, H., 1999. PowerBookmarks: A System for personalizable web information organization, sharing, and management. In *Proc. of the 8th Intl. World Wide Web Conference*, Toronto, Canada.1999.
- [Lin et al, 1998] Liu et C. Yu. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on knowledge and data engineering*, 16(1):28–40, 2004.
- [Muts, 2001] Lin, C.S Shih, M.C Chen, J. Ho, M. Ko, et Y.M. Huang. Extracting classification knowledge of internet documents with mining termassociations : A semantic approach. Dans *In the 21th International SIGIRConference on Research and Development in Information Retrieval*, pages 241–249, 1998.
- [Marl,2004] Mutschke P., « Enhancing Information Retrieval in Federated Bibliographic Data Sources Using Author Network Based Stratagems », ECDL 2001, Darmstadt, Germany, 2001.
- [Pazza et al, 1996] C. Marlow. Audience, structure and authority in the weblog community. In *Proc. Communication Association Conference*, 2004.
- [Port, 1980]. J. Pazzani, J. Muramatsu, et D. Billsus. Syskill&webert : Identifying interesting web sites. Dans *Proceedings of the 30th National Conference on Artificial Intelligence*, pages 54–61, Portland, 1996.
- [Ponte ,1998] M. F. Port, An algorithm for suffix stripping. *Program* 14, 1980.
- [Rae et al., 2010] J. M. Ponte, W. B. C J. M. Ponte, W. B. Croft. "A language modeling approach to information retrieval", *SIGIR'98*. 1998.
- [Rober et al, 1994]. Rae et al., Improving Tag Recommendation Using Social Networks, *RIAO* 2010.
- [Salton, 1970] S. Robertson, S. Walker, S. Jones, and M. H.-B. and M. Gatford, 1994 Okapi at TREC 3. Dans *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, pages 109–126, 1994.
- [Salton et Gill, 1984] G. Salton, The SMART retrieval system: Experiments in automatic document processing. Prentice Hall, 1970.
- [Sieg et al, 2007] G. Salton and M. McGill, Introduction to modern information retrieval. McGraw-Hill Int. Book Co 1984.
- [Sing et al, 1996]. Sieg, B. Mobasher, et R. Burke. Web search personalization with ontological user profiles. Dans *Proceedings of the 16th ACM conference on NConference on information and knowledge management*, pages 525–534, New York, NY, USA, 2007.
- [Sing et al, 1996]. A. Singhal, G. Salton, M. Mitra, and C. Buckley, 1996 Document length normalization. *Information Processing and Management*, 32(5): pages 619– 633, 1996.

- [Srir et al, 2004] Sriram, X. Shen, et C. Zhai. A session-based search engine. Dans *Proceedings of the International ACM SIGIR Conference*, 2004.
- [Shen et Zhai,2005] Shen, B. Tan, et C. Zhai. Context-sensitive information retrieval using implicit feedback. Dans *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA, 2005.
- [Sper et Gauch,2005] Speretta et S. Gauch. Personalized search based on user search histories. Dans *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 622–628, 2005.
- [SIGIR ,2003], Workshop Report: Implicit Measures of User Interests and Preferences
- [Tam et al, 2008] Lynda TAMINE-LECHANIDe la recherche d'information orientée système vers la recherche d'information orientée contexte : Verrous, contributions et perspectives.
- [Tam ,2007b] L. Tamine, M. Boughanem, et W.N. Zemirli. Exploiting multi-evidence from multiple users' interests to personalizing information retrieval. Dans Youakim Badr, Richard Chbeir, et PitPichappan, éditeurs, *IEEE International Conference on Digital Information Management (ICDIM)*, Lyon, 28/10/07-30/10/07, pages 7–12, <http://computer.org/cspress>, octobre 2007b. IEEE Engineering Management Society.2007.
- [Tan et al, 2006] Tan, X. Shen, et Ch. Zhai. Mining long-term search history to improve search accuracy. Dans *KDD '06 : Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723, New York, NY, USA, 2006.
- [Tebr, 2004]. H.TEBRI "Formalisation et spécification d'un système de filtrage incrémental d'information", 2004.
- [Teev et al, 2005] Teevan et S.T. Dumais. Personalizing search via automated analysis of interests and activities. Dans *Proceedings of the 28th International SIGIR conference on research and development in information retrieval*, pages 449– 456, 2005.
- [Urfi, 2004] Urfist, Cours "Problématique Générale de la Recherche d'Information" URFIST Bretagne-Pays de Loire, Alexandre Serres, 2004 ;
- [Xu et al,2007] Xu S., Bao S., Cao Y., Yu Y., « Using social annotations to improve language model for information retrieval », *Conference on Information and Knowledge Management*, p. 1003-1006. CIKM 2007,
- [Zhou, 2008] Zhou D., Bian J.,Zheng S., « Exploring Social Annotations for Information Retrieval », *World Wide Web Conference*, p. 715-724. 2008.
- [Zipf, 1949] G. Zipf, *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.