



UNIVERSITE IBN ZOHR



ETABLISSEMENT :

FACULTE DES SCIENCES - AGADIR

CENTRE DES ETUDES DOCTORALES IBN ZOHR

Formation doctorale :

Mathématiques, Informatique et applications

THESE

Présentée par

Hicham MOUTACHAOUIK

pour l'obtention de grade de
DOCTEUR de l'Université Ibn Zohr

Spécialité : Informatique

**Méthodes et outils pour l'amélioration de la recherche
d'information sur le web et application en e-Learning**

Soutenu Le 02 Juillet 2013

Devant la commission d'examen composée de :

M. Mohamed WAKRIM,	Directeur de l'ENSA d'Agadir, Président de jury et Rapporteur
M. Hassan DOUZI,	Professeur à la Faculté des Sciences d'Agadir, Directeur de thèse
M. Abdelaziz MARZAK,	Professeur à la Faculté des Sciences Ben M'Sik de Casa, Co-directeur de thèse
M. Abdelaziz KRIOUILLE,	Professeur à ENSIAS de Rabat, Rapporteur
M. Chihab CHERKAOUI,	Professeur à ENCG d'Agadir, Rapporteur
M. Brahim OUHBI,	Professeur à ENSAM de Meknès, Examineur
M. Hicham BEHJA,	Directeur adjoint de l'ENSEM de Casablanca, Examineur

A mes Parents

A ma petite famille

A ma grande famille

A vous tous

Remerciements

Je souhaite, avant tout, remercier Dieu de m'avoir soutenu et permis l'achèvement de cette thèse.

La réalisation de cette thèse de doctorat a été une occasion en or de rencontrer et d'échanger avec de nombreuses personnes. Ce travail n'aurait pas pu être réalisé sans l'intervention d'un certain nombre de personnes qui m'ont apporté une assistance précieuse.

J'adresse mes sincères remerciements à mon directeur de thèse Monsieur Hassan Douzi, pour la patience, la gentillesse et la disponibilité dont il a fait preuve. Si j'arrive aujourd'hui au dénouement de ce travail c'est grâce à ses conseils et ses remarques constructives. Qu'il trouve ici l'expression de ma très grande gratitude.

Je remercie également mon second directeur de thèse Monsieur Abdelaziz Marzak qui a été d'une disponibilité bienveillante et fructueuse. Je le remercie de m'avoir bénéficié de sa propre expérience, par ses précieux conseils et sa ferme volonté de collaboration.

Je tiens à présenter mes vifs remerciements et reconnaissance à Monsieur Hicham Behja, Directeur adjoint de l'École Nationale Supérieure d'Electricité et de Mécanique (ENSEM) de Casablanca et Monsieur Brahim Ouhbi, Professeur à l'École Nationale Supérieure d'Arts et Métiers (ENSAM) de Meknès pour leurs remarques, leurs suggestions et surtout leurs idées qui m'ont été d'un apport considérable. A eux, je leur témoigne ma sincère reconnaissance pour tous les efforts déployés et les conseils objectifs apportés à mon égard au cours de la période de thèse.

Je remercie également Monsieur Abdelaziz Kriouile, professeur universitaire à l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS) de Rabat et Monsieur Chihab Cherkaoui, professeur universitaire à l'École Nationale de Commerce et de Gestion (ENCG) d'Agadir, d'avoir accepté de rapporter ce travail et pour l'intérêt qu'ils ont porté à ce travail.

J'aimerais également remercier Monsieur Mohamed Wakrim, Directeur de l'École Nationale des Sciences Appliquées (ENSA) d'Agadir qui m'a fait l'honneur de présider le jury et rapporter ce présent travail.

Mes plus sincères remerciements vont à tous les professeurs et doctorants du laboratoire « Image et Reconnaissance de Forme, Système Intelligent et Communication » et

spécialement à Monsieur Driss Mammass, Directeur de l'École Supérieure de Technologie (EST) d'Agadir le responsable du laboratoire.

Un remerciement très spécial à tous les membres du Centre des études doctorales et spécialement pour sa directrice Mme Amina Idrissi pour les efforts déployés ainsi que le suivi et la communication des informations.

Je tiens à présenter mes vifs remerciements à Mme Bouchera Frikh pour l'aide apportée durant les dernières années de la thèse.

Un mot de vérité, à tous les membres de l'équipe de recherche du projet « WRUM » : Mr Zemouri, Mme Sael, Mr Chhibi, Mlle Chakhmoune et Mlle Benghabrit,... pour leur amitié, leur soutien moral, leur aide et l'encouragement constant durant les années de thèse.

Un grand merci à Mr Hain pour les remarques objectives apportées à mon égard au cours de la période de thèse.

Un grand merci également à Mr Abouelfadel d'avoir eu l'amabilité de lire et de relire ma thèse.

Je remercie profondément ma chère épouse Sanaa pour ses prières, sa présence et son interminable soutien qui me sont d'une immense source d'énergie et d'enthousiasme.

Je tiens à remercier également mes amis, doctorants ou non, qui m'ont aidé au cours de ces années de thèse.

Enfin, je remercie du fond du cœur et avec un grand amour mes parents qui n'ont jamais cessé de croire en moi pendant toutes mes années d'études. Merci pour les sacrifices consentis à mon éducation, pour leur soutien et surtout leur patience. Merci aussi à ma sœur Nawal et mes frères qui m'ont toujours encouragé.

Table des Matières

Remerciements	3
Table des Matières	5
Table des figures.....	8
Table des tableaux.....	9
INTRODUCTION GENERALE	
	10
Chapitre I. ÉTAT DE L'ART	
	16
I.1. Objectif du chapitre	16
I.2. La recherche d'information	17
I.2.1. Les dates clés de la recherche d'information	18
I.2.2. Le processus de la recherche d'information.....	21
I.2.3. Système de recherche d'information.....	25
I.2.4. Catégorie des SRI.....	26
I.2.4.1. Les Annuaires.....	26
I.2.4.2. Les moteurs de recherche	29
I.2.4.3. Le méta-moteur de recherche	31
I.3. Les modèles de la recherche d'information	32
I.3.1. Le modèle Booléen.....	33
Le modèle booléen à base de logique floue	35
I.3.2. Le modèle p-norme	37
I.3.3. Le modèle vectoriel.....	39
I.3.3.1. Mesure de similarité :	40
I.3.3.2. Pondération	41
I.3.3.3. Exemple	42
I.3.4. Les modèles probabilistes	44
I.3.4.1. Définition	44
I.3.4.2. Formalisation.....	45
I.3.4.3. Le modèle de BIR	46
I.3.4.4. Le modèle de poisson	48
I.3.4.5. Exemple.....	49
I.3.4.6. Conclusion.....	50
I.3.5. Les modèles Bayésiens de RI.....	51
I.3.5.1. Architecture générale du modèle Bayésien	53
I.3.5.2. Les modèles basés sur les réseaux d'inférence	55
I.3.6. Conclusion.....	58
I.4. Algorithmes de classement	59

I.4.1. Algorithme PageRank	59
I.4.2. Algorithme HITS.....	60
I.5. Métriques d'évaluations des SRI	62
I.5.1. Métriques d'évaluation : le rappel.....	63
I.5.2. Métriques d'évaluation : la précision	64
I.5.3. La précision et le rappel dans un cadre multi-classe	64
I.5.4. Interprétation des résultats de précision et de rappel.....	65
I.5.5. La F-mesure	66
I.5.6. Exemple.....	66
I.6. Synthèse.....	67

Chapitre II. SIMILARITE ENTRE DEUX REQUETES DANS UN SRI.....	71
---	-----------

II.1. Introduction	72
II.2. Travaux connexes.....	74
II.2.1. Divergence de Kullback-Leibler	74
II.2.2. Indice de jaccard.....	75
II.3. Test χ^2	76
II.3.1.1. Mode de calcul de χ^2	76
II.3.1.2. Interprétation de χ^2	79
II.4. Similarité entre requêtes basée sur le Chir statistique	79
II.5. Similarité basée sur l'information mutuelle.....	83
II.5.1. Information mutuelle.....	83
II.5.2. Notre méthodologie d'utilisation de IM.....	85
II.6. Similarité basée sur une mesure hybride	87
II.7. Conclusion.....	88

Chapitre III. SYSTEME DE RECOMMANDATION POUR LA RECHERCHE D'INFORMATION	90
--	-----------

III.1. Introduction	91
III.2. Aperçu sur e-Learning.....	93
III.2.1. Historique.....	93
III.2.2. Les plates formes e-Learning.....	96
III.3. Les systèmes de recommandation	97
III.4. Type d'évaluations	99
III.4.1. Évaluations explicites	99
III.4.2. Évaluations implicites.....	99
III.5. Pourquoi Moodle :.....	101
III.6. Analyse du problème et conception de la solution.....	102
III.6.1. Utilisateurs du système	102
III.6.2. Séquencement des tâches assurées par le système.....	104
III.7. Processus de fonctionnement et calculs des critères d'extraction de connaissances.....	106
III.7.1. Mx-Search : Comment ça fonctionne ?.....	106
III.7.2. Calculs des critères d'extraction de connaissances.....	106
III.7.2.1. Comment calculer le Score d'Appréciation ?.....	107
III.7.2.2. Comment calculer la note d'appréciation pour un document web ?.....	108
III.7.2.3. Comment calculer <i>le Ratio</i> ?	108
III.7.2.4. Comment utiliser la formule de chan ?.....	108
III.7.2.5. Extraction de la connaissance à l'aide de la méthode alpha ?	109
III.7.2.6. Attribution du lien visité à une catégorie des cours ?.....	110
III.8. Conclusion.....	111

**Chapitre IV. VALIDATION DES APPROCHES PROPOSEES POUR AMELIORER
LA RECHERCHE D'INFORMATION SUR LE WEB 113**

IV.1. Introduction 114
IV.2. Corpus de test 115
 IV.2.1. Corpus OHSUMED..... 115
 IV.2.2. Corpus Reuters-21578 117
 IV.2.3. Corpus TREC 117
 IV.2.4. Corpus Cranfield 117
 IV.2.5. Corpus MEDLINE..... 118
 IV.2.6. Corpus NPL 118
 IV.2.7. Corpus LISA..... 118
 IV.2.8. Corpus CISI 118
 IV.2.9. Corpus TIME..... 119
 IV.2.10. Corpus CACM..... 119
IV.3. La similarité entre requêtes dans un SRI..... 119
IV.4. Fonctionnement du système de recommandation proposé 131
 IV.4.1. Assister l'apprenant pour parcourir le résultat de sa recherche sur le web..... 132
 IV.4.2. Guider l'administrateur de la plate forme à extraire des connaissances 134
Figure IV.10 : Résultat obtenu pour le module java selon le critère : Formule de chan 137
IV.5. Conclusion..... 137

Chapitre V. CONCLUSION ET PERSPECTIVES..... 140

V.1. Rappel du contexte de la thèse 140
V.2. Apport du présent travail..... 141
V.3. Perspectives de la thèse 142
V.4. Publications..... 143

BIBLIOGRAPHIE 146

LES ANNEXES : 155

Annexe 1 : Calcul de la similarité hybride 155

Annexe 2 : La similarité selon la divergence de Kullbak-leiber 172

Annexe 3 : La similarité selon la méthode de jaccard 176

RÉSUMÉ..... 178

RESUME..... 180

Table des figures

Figure I.1: Processus en U de la RI.....	24
Figure I.2: Annuaire Yahoo.....	27
Figure I.3: Exemple : Page jaunes.....	28
Figure I.4: Processus d'un méta-moteur de recherche	32
Figure I.5: Évaluation d'une conjonction.....	38
Figure I.6 : Évaluation d'une disjonction.....	38
Figure I.7: Représentation requête-document	40
Figure I.8: Exemple de Réseau Bayésien.....	53
Figure I.9: Architecture générale du modèle Bayésien	54
Figure I.10: Modèle générique d'un réseau d'inférence	55
Figure I.11: Modèle générique d'un réseau de croyance	57
Figure I.12: Mesures de performance dans la RI.....	63
Figure II.1: Processus de fonctionnement de notre méthode.....	74
Figure II.2: Utilisation de Chir pour le calcul de la similarité statistique.....	82
Figure II.3: Utilisation de l'information mutuelle pour le calcul de la similarité sémantique..	86
Figure III.1: Diagramme de cas d'utilisation.....	102
Figure III.2: Diagramme de classe	103
Figure III.3: Diagramme de séquence apprenant	104
Figure III.4: Diagramme de séquence administrateur	105
Figure III.5: Processus de fonctionnement de MX-Search.....	106
Figure IV.1: Interface de la plate forme e-learning moodle: LP Java/C++	132
Figure IV.2: Interface pour effectuer des recherches sur le web.....	132
Figure IV.3: Interface pour parcourir le résultat de la recherche	133
Figure IV.4: Interface pour visualiser le contenu du lien web	133
Figure IV.5: Interface pour choisir les critères d'extraction de connaissances	1334
Figure IV.6: Résultat obtenu pour le module java selon le critère : Score d'appréciation.....	135
Figure IV.7: Résultat obtenu pour le module java selon la combinaison des critères : Score d'appréciation, Ratio.....	135
Figure IV.8: Résultat obtenu pour le module java selon le critère : période de recherche.....	136
Figure IV.9: Résultat obtenu pour le module java selon le critère : format du document.....	136
Figure IV.10 : Résultat obtenu pour le module java selon le critère : Formule de chan.....	137

Table des tableaux

Tableau I.1: Table de vérité.....	37
Tableau I.2: Poids des termes.....	42
Tableau I.3: Classement des documents par cosinus.....	43
Tableau I.4: Poids des termes obtenus par la méthode de pivot.....	43
Tableau I.5: Classement des documents selon la normalisation par pivot	43
Tableau I.6: Poids des termes	50
Tableau I.7: Classement des documents.....	50
Tableau I.8: Tableau de pertinence	67
Tableau II.1: Tableau des effectifs observés	77
Tableau II.2: Tableau des effectifs attendus sous l'hypothèse H0	78
Tableau IV.1: Résultat de similarité selon notre méthode hybride	121
Tableau IV.2: Résultat de similarité selon la méthode de divergence de Kullbak-leiber.....	122
Tableau IV.3: Résultat de similarité selon l'indice de jaccard.....	124
Tableau IV.4: Comparaison du résultat de similarité.....	124
Tableau IV.5: Résultat de similarité selon notre méthode hybride	1218
Tableau IV.6: Résultat de similarité selon la méthode de divergence de Kullbak-leiber.....	1229
Tableau IV.7: Résultat de similarité selon l'indice de jaccard.....	12430
Tableau IV.8: Comparaison du résultat de similarité.....	12431

INTRODUCTION GENERALE

L'avènement de nouvelles technologies de communication à la fin du XXème siècle a suscité la curiosité et la soif du savoir et de l'information. Ces qualités étaient imprégnées dans l'esprit de l'Homme depuis longtemps par la tradition orale, par la découverte de l'imprimerie jusqu'à l'apparition de l'informatique, et de l'internet qui aujourd'hui se trouve à la portée de tout le monde. Le développement des réseaux a fait que la demande de l'information à travers le net s'est simultanément accompagnée d'une grande offre, une sorte de banques énormes d'informations qui affluent de tous les coins de la planète. Donc, le souci majeur est de réunir suffisamment de données relatives à la recherche d'information, de la localiser et de l'interpréter afin d'en extraire des connaissances intéressantes. Dans cette perspective, des approches et outils deviennent impérativement nécessaires pour la facilitation de l'accès à l'information qui n'est pas tout à fait aisément accessible car selon Alexandre Serres ¹ « *L'information disponible sur Internet est une information à la fois hétérogène, structurée et non structurée, diversifiée, instable, fragmentée, validée et non validée* ».

Par ailleurs, le besoin exprimé par l'utilisateur se traduit sous forme de requêtes dont l'aboutissement sera notamment un texte, un morceau de texte, une page Web ou une image, etc. L'information sera pertinente si elle exauce une satisfaction immédiate de l'utilisateur sinon la disponibilité de l'information demandée au milieu d'un nombre gigantesque de données serait considérée comme une entrave.

A priori, nous constatons que le nombre de documents disponibles sur le web est de l'ordre de dizaines de milliards. Avec cette augmentation exponentielle de nombres de pages Web et d'internautes, la recherche d'informations sur Internet devient le premier réflexe à

¹ Source : SERRES Alexandre, Recherche de l'information sur Internet : initiation, stage Urfist Bretagne-Pays de la Loire, 2002 (dernière MàJ :02/11/2004),http://www.uhb.fr/urfist/Supports/RechInfoInit/Rechinfo1_cadre.htm.

l'occasion de n'importe quelle requête. Cette taille colossale associée à la demande pressante des utilisateurs pose un défi à la communauté scientifique qui doit être en mesure de proposer des outils efficaces de la recherche d'information. Ce qui a donné automatiquement naissance à des moteurs de recherche permettant de retrouver facilement l'information attendue et comprise. Justement, l'intérêt de ce présent travail réside dans la recherche d'une performance aussi bonne que possible ainsi que le filtrage des informations réellement pertinentes. Nous essaierons tant soit peu de trouver des réponses aux déficiences existantes et de suggérer quelques solutions aussi modestes soient-elles dans ce domaine.

Comme nous l'avons cité ci-dessus, le souci de rendre service aux usagers, des moteurs de recherche sont donc nés depuis des dizaines d'années. À partir d'une requête simple, ils parcourent leur base de données pour trouver le plus rapidement possible les documents souhaités par l'utilisateur. A titre d'exemple, le moteur de recherche *Google* est certainement la dernière réussite américaine en ce domaine puisqu'il possède en mémoire plusieurs milliards de documents. Il fournit souvent une réponse à une requête sous forme des milliers de documents en un temps inférieur à la seconde. Ce sont là précisément, les points forts et les points faibles de ce type de méthode : l'utilisateur est submergé par des milliers de réponses présentées "en vrac", dont une partie ne correspond pas à la requête, une autre correspond à des pages qui n'existent plus, sans parler de nombreux types de documents qui ne sont pas pris en compte dans la mémoire du moteur. Dans bien des cas, l'utilisateur passe un temps assez long à analyser les résultats du moteur, sans aucune garantie de résultats.

Le principal handicap des moteurs de recherche et plus précisément les moteurs de recherche traditionnels est le défaut d'intelligence qui caractérise leur travail d'indexation. Rien de plus normal, vu que c'est un robot qui est chargé de cette opération délicate. L'avantage est que le moteur n'est pas loin d'être exhaustif dans le recensement et l'indexation des pages de la Toile.

Nous évoquons alors les inconvénients de la recherche d'informations sur le web : le premier inconvénient est « le bruit », c'est-à-dire, le nombre très élevé de documents proposés en réponse à une requête. Certaines sont très pertinentes, d'autres le sont beaucoup moins et c'est à l'utilisateur de faire le tri, ce qui peut demander un certain temps quand plus de 100 références sont affichées à l'écran. Le deuxième inconvénient est « le silence » représentant l'ensemble de documents pertinents que l'interrogation n'a pas pu retrouver.

C'est dans ce contexte de la recherche d'information que nous proposons de nouvelles méthodes pouvant être incorporées dans un système de recherche d'information (SRI) afin d'offrir à l'utilisateur les documents qui répondent à ses besoins traduits par une requête.

Les moteurs de recherche sur Internet doivent faire face à de très fortes contraintes. Les outils de recherche sont donc tenus de répondre à des millions de requêtes par jour alors que la quantité de documents qu'ils sont appelés à analyser est gigantesque. La détection d'une mise à jour de ces documents est également source de grandes difficultés. Le constat en est qu'il n'y a pas de centralisation de ces données et un moteur de recherche doit par conséquent scruter en permanence le réseau en vue de détecter ces changements.

Cette problématique est bien illustrée par une citation de Sergey Brin, un des inventeurs de Google et de l'algorithme PageRank (Brin et Page, 1998) : *«le Web est une vaste collection de documents hétérogènes complètement incontrôlés»*.

Dans ce contexte, des recherches ont été menées pour alléger la charge des machines composant les systèmes de recherche. Le principal facteur de ralentissement étant l'analyse des textes dans le dessein d'établir un tri de pertinence des résultats, d'autres voies ont été explorées : les meilleurs résultats ont été obtenus par des méthodes utilisant les propriétés déduites de l'analyse de la topologie d'Internet (Calvert et al., 1997). Néanmoins, le processus de recherche d'information soulève quelques déficiences (Carpio, 2010) :

1. La recherche d'information est devenue une tâche difficile qui demande un effort intellectuel considérable. L'utilisateur a besoin d'être assisté et aidé pendant son processus de recherche afin de retrouver et d'exploiter les réponses pertinentes répondant à son besoin ;
2. La quantité spectaculaire de ressources sur le Web a généré un besoin de structuration et d'organisation au point de rendre les informations facilement exploitables.

Pour récapituler, la problématique de notre thèse s'oriente sur la manière de fournir des méthodes et des outils répondant à la question suivante :

Comment répondre aux besoins des utilisateurs qui sont à la recherche d'information dans une grande masse de données ? Comment les aider, les assister à bon escient et leur offrir des outils pour obtenir des résultats pertinents et d'une façon conviviale ?

Plusieurs modèles pour la recherche d'information à savoir le modèle booléen, le modèle vectoriel, le modèle probabiliste et le modèle bayésien ont été proposés dans la littérature. Toutefois, ces modèles comme nous allons le voir ultérieurement contiennent des inconvénients même si chacun d'eux a tenté de surmonter les lacunes et les failles des autres.

Comme souligné avant, ces déficiences nous ont amenés à proposer d'autres méthodes et outils répondant au maximum possible aux besoins des utilisateurs pour la recherche d'informations.

Notre contribution vise donc à présenter aux utilisateurs, par le biais d'un système de recherche d'information, les documents susceptibles de répondre aux besoins exprimés par une requête. Autrement dit, cette contribution s'orientera selon deux axes à savoir

1. **Une nouvelle mesure hybride visant à calculer la similarité entre deux requêtes dans un système de recherche d'information :** la mesure de cette similarité entre une requête nouvellement reçue par le système exprimant le besoin de l'utilisateur et des requêtes candidates dont le système mémorise les documents pertinents.

Ce calcul de similarité passe par trois phases :

- Dans un premier temps, nous présenterons une statistique plus précise basée sur la version étendue de la statistique χ^2 , appelée la méthode statistique de CHIR pour sélectionner les requêtes positivement dépendantes par rapport à la requête donnée ;
- Dans un second temps, nous utiliserons l'information mutuelle pour mesurer la similarité sémantique entre la requête de l'utilisateur et la requête candidate du système ;
- Finalement, nous combinerons ces deux mesures, statistique et sémantique par le biais de notre méthode dite alpha pour prédire la requête candidate la plus proche à la requête donnée en termes de similarité.

2. **Système de recommandation pour améliorer le service de recherche d'information dans les plates-formes E-Learning :** Lors de cette contribution consistant à proposer une méthode de classement des documents web par une insistance sur l'appréciation de l'utilisateur du système de recherche d'information. Cette méthode a été insérée au sein d'un système de recommandation en s'inspirant de différentes méthodes, outils et techniques qui entrent en jeu à savoir : la recherche d'information, le filtrage d'information et le Web usage mining. Le système proposé est appliqué dans le domaine e-Learning pour tester son impact sur le processus de la recherche.

La suite de cette thèse sera composée de **cinq** chapitres :

1. **En premier lieu, au chapitre 1,** l'état de l'art où nous exposerons l'historique de la recherche d'information, les différents modèles utilisés pour répondre à une requête et enfin, nous clôturerons par une synthèse.
2. **En second lieu, le chapitre 2,** contiendra notre méthode pour le calcul de la similarité entre deux requêtes dans un SRI où figurent des travaux connexes, Test χ^2 , la similarité statistique basée sur la méthode de Chir et la similarité sémantique par le truchement de l'information mutuelle.
3. **En troisième lieu, au chapitre 3,** nous développerons notre système de recommandation pour la recherche d'information dans les plate forme e-Learning notamment nous décriverons : le e-learning, les systèmes de recommandation, les types d'évaluations, l'analyse du problème et les méthodes utilisées pour la recommandation.
4. **En quatrième lieu, le chapitre 4,** sera consacré à la validation des approches et méthodes proposées sur des corpus standards de la recherche d'information.
5. **En dernier lieu, au chapitre 5,** nous arriverons à la clôture de ce travail, l'apport du présent travail, les perspectives de la thèse et les publications.

Chapitre I. ÉTAT DE L'ART

I.1. Objectif du chapitre	16
I.2. La recherche d'information	17
I.2.1. Les dates clés de la recherche d'information	18
I.2.2. Le processus de la recherche d'information	21
I.2.3. Système de recherche d'information	25
I.2.4. Catégorie des SRI	26
I.2.4.1. Les Annuaire	26
I.2.4.2. Les moteurs de recherche	29
I.2.4.3. Le méta-moteur de recherche	31
I.3. Les modèles de la recherche d'information	32
I.3.1. Le modèle Booléen	33
Le modèle booléen à base de logique floue	35
I.3.2. Le modèle p-norme	37
I.3.3. Le modèle vectoriel	39
I.3.3.1. Mesure de similarité :	40
I.3.3.2. Pondération	41
I.3.3.3. Exemple	42
I.3.4. Les modèles probabilistes	44
I.3.4.1. Définition	44
I.3.4.2. Formalisation	45
I.3.4.3. Le modèle de BIR	46
I.3.4.4. Le modèle de poisson	48
I.3.4.5. Exemple	49
I.3.4.6. Conclusion	50
I.3.5. Les modèles Bayésiens de RI	51
I.3.5.1. Architecture générale du modèle Bayésien	53
I.3.5.2. Les modèles basés sur les réseaux d'inférence	55
I.3.6. Conclusion	58
I.4. Algorithmes de classement	59
I.4.1. Algorithme PageRank	59
I.4.2. Algorithme HITS	60
I.5. Métriques d'évaluations des SRI	62
I.5.1. Métriques d'évaluation : le rappel	63
I.5.2. Métriques d'évaluation : la précision	64
I.5.3. La précision et le rappel dans un cadre multi-classe	64
I.5.4. Interprétation des résultats de précision et de rappel	65
I.5.5. La F-mesure	66
I.5.6. Exemple	66
I.6. Synthèse	67

I.1. Objectif du chapitre

Dans ce chapitre, nous survolerons le domaine de la recherche d'information (RI), par la suite nous détaillerons les modèles utilisés pour répondre à la requête utilisateur qui constitue l'axe central de cette thèse. Puis, nous procéderons à la description des critères utilisés pour mesurer la qualité d'un système de recherche d'information. Enfin, nous conclurons par une synthèse.

I.2. La recherche d'information

Au début des années soixante, quelques années après l'invention de l'ordinateur, apparut la recherche d'information (RI) comme une réponse au besoin de gérer l'explosion de la quantité d'informations. C'est la science de la recherche de l'information dans des documents (dans les documents eux-mêmes, dans les métadonnées qui décrivent les documents ou encore, c'est le cas de cette étude, dans les relations qu'entretiennent les documents entre eux) qu'ils soient dans une base de données, dans une base documentaire ou sur le Web.

Le nom de « recherche d'information » (information retrieval) fut donné par Calvin Mooers en 1948 pour la première fois quand il travaillait sur son mémoire de maîtrise (Mooers, 1948). La première conférence dédiée à ce thème – International Conference on Scientific Information - s'est tenue en 1958 à Washington. On y comptait les pionniers du domaine, notamment, Cyril Cleverdon, Brian Campbell Vickery, Peter Luhn, etc....

Les premiers problèmes qui intéressaient les chercheurs portaient sur l'indexation des documents afin de les retrouver. Déjà à la « International Conference on Scientific Information », Luhn avait fait une démonstration de son système d'indexation KWIC qui sélectionnait les indexes selon la fréquence des mots dans les documents, et filtrait des mots vides de sens en employant des « stoplistes ». C'est à cette période que le domaine de RI a vu le jour.

La recherche d'information est traditionnellement définie comme l'ensemble des techniques permettant ainsi de sélectionner à partir d'une collection de documents, ceux qui sont susceptibles de répondre aux besoins de l'utilisateur. Gérer des textes de sorte qu'on puisse stocker, rechercher et explorer des documents pertinents (Salton, 1971).

En outre la nécessité de définir quelques concepts fondamentaux :

- Données : dans un environnement numérique, « L'unité élémentaire d'information est la donnée qui n'est qu'une chaîne de caractères ou octets constitués de bits (0 ou 1) » (Dherent, 2002). Une définition plus générale est présentée dans (Benayache, 2005) comme suit : « toute représentation à laquelle une signification peut être attachée ». Une donnée peut être qualitative ou quantitative mais elle n'a pas de sens en elle-même. Les données peuvent être récupérées, représentées et réinterprétées ;
- Information : une information est issue dès lors pour qu'on donne un sens à une donnée. L'information est donc une collection de données organisées pour donner forme à un message le plus souvent sous forme visible, imagée, écrite ou orale, de telle sorte à réduire une incertitude et transmettre quelque chose qui déclenche une action (Benayache, 2005) ;
- Connaissance : Debenham a défini la connaissance comme l'ensemble des associations fonctionnelles explicites entre les éléments de l'information et/ou des données (Kendal et Creen, 2006). Le grand dictionnaire terminologique définit le terme connaissance dans le domaine informatique comme « l'ensemble de faits, événements, règles d'inférence et heuristiques permettant à un programme de fonctionner intelligemment ».

I.2.1. Les dates clés de la recherche d'information

- L'année 1957 :

Bien que l'un des premiers travaux notables ce soit la proposition de H.P. LUHN en 1957, qui stipule qu'un système de recherche d'information se base sur une représentation des documents (et des requêtes) obtenue d'une façon automatique à partir des contenus de ces documents (Piwowski, 2003). Cette proposition est basée sur une approche statistique qui utilise la fréquence des données pour l'extraction des mots et des phrases dans une perspective d'indexation automatique des documents (Luhn, 1957).

Par conséquent, le domaine de la recherche d'information a connu plusieurs développements :

- Les années 60' :

Ces années ont vu une large gamme d'activités reflétant l'évolution de la recherche d'information en utilisant l'ordinateur. L'une des figures majeures de l'émergence de cette période a été Gerard Salton², qui a formé et dirigé un grand groupe RI, d'abord à l'Université de Harvard, puis à l'Université Cornell. Le groupe produit de nombreux rapports techniques et des idées qui sont encore les principaux domaines d'enquête aujourd'hui. Un de ces domaines est la formalisation des algorithmes pour classer les documents relatifs à une requête. On notera en particulier une approche où les documents et les requêtes ont été considérés comme des vecteurs dans un espace à N dimensions (N étant le nombre de termes uniques dans la collection en cours de recherche). Ce fut d'abord proposé par Switzer (Switzer, 1963) et, plus tard, la similarité entre un document et le vecteur de requête a été proposé par Salton au point de mesurer le cosinus de l'angle entre les vecteurs en utilisant le coefficient du cosinus (Salton, 1968).

La collaboration entre Salton et ses étudiants durant la période 1968-1970 a conduit au développement du système SMART qui utilise le modèle vectoriel (Salton, 1971).

Une autre innovation importante à cette époque était l'introduction de feedback utilisateur connu aussi par retour de pertinence dans le processus de la recherche (Rocchio, 1965). Ce fut un processus pour soutenir la recherche itérative, où les documents précédemment récupérés pourraient être marqués comme pertinents dans un système de recherche d'information. La requête d'un utilisateur a été ajustée automatiquement en utilisant les informations extraites des documents pertinents. Les versions de ce processus sont utilisées dans les moteurs de recherche modernes.

Par ailleurs, d'autres améliorations RI examinés dans cette période, mentionnons le regroupement des documents avec un contenu similaire, l'association statistique des termes ayant une signification sémantique similaire, l'augmentation du nombre de documents retournés avec une requête en élargissant la requête avec les variations lexicales (dites les stems) ou à l'aide des relations d'association sémantiques entre les termes (Stevens, 1964 ; van Rijsbergen, 1979).

² 1927-1995 : Allemand, professeur informaticien à l'université Cornell, Leader du domaine de RI

- la période 1957-1967 :

Cyril Cleverdon a dirigé le projet Cranfield, dans le collège d'Aéronautiques (U.K.). L'évaluation Cranfield consiste en une collection de test constituée d'un ensemble de 18000 articles et d'un ensemble de 1200 requêtes (Nie, 2007). Les requêtes étaient d'abord évaluées par des experts afin de déterminer les réponses pertinentes. Par suite, les résultats d'une recherche automatique étaient comparés avec les réponses pertinentes pour mesurer la performance en termes de précision et de rappel.

Au constat du système SMART et de l'évaluation de Cranfield, il s'avère que la communauté de RI, a instauré, dès les premiers jours, une tradition d'expérimentation et d'évaluation pour tester n'importe quelle méthode d'indexation et de recherche de documents afin de connaître son effet en réalité.

- Les années 70' :

L'un des faits marquants de cette période était que Luhn a proposé le poids du terme de fréquence (tf) (sur la base des occurrences de mots dans un document). Cette proposition a été complétée par le travail du Sparck Jones sur la présence des mots à travers les documents d'une collection. Son article sur la fréquence inverse du document (idf) introduit l'idée que la fréquence d'occurrence d'un mot dans une collection de documents est inversement proportionnelle à son importance dans la recherche : mots peu courants ont tendance à se référer à des concepts plus spécifiques, qui étaient plus importants dans la recherche. L'idée de combiner ces deux poids ($tf * idf$) a été rapidement adopté (Salton et al., 1975) pour une première explorations de ces idées.

- Les années 70 et 80 :

Ces années ont été témoins de plusieurs développements en matière de modèles et techniques en RI. Salton synthétisé les résultats de son groupe de travail sur les vecteurs pour produire le modèle d'espace vectoriel (Salton, 1975). Un autre moyen de la modélisation de systèmes RI était l'idée de Maron, Kuhns et Ray d'utiliser la théorie des probabilités. Robertson a défini le principe de classement par probabilité. Ces modèles booléen, vectoriel et probabiliste constituent les modèles de base pour la RI classique. Ces modèles ainsi que leurs dérivés seront exposés en détail ultérieurement.

- Les années 90' :

Bien que Tim Berners-Lee ait créé le World Wide Web à la fin 1990, le nombre de sites web et de la quantité de pages était relativement faible jusqu'en 1993. Dans ces premières années, le catalogage manuel conventionnel du contenu suffit. Au milieu de 1993, tel qu'enregistrer par l'enquête de Matthew Gray³, il y avait près de 100 sites web ; six mois plus tard, il y avait le quadruple de ce nombre. Les moteurs de recherche sur le Web ont commencé à émerger à la fin 1993 pour faire face à cette croissance. L'arrivée de Web a lancé l'étude des nouveaux problèmes dans la recherche d'information.

Toujours dans une optique d'évaluation et d'expérimentation, des conférences de RI annuelles se sont organisées (Langville et al., 2006). SIGIR (Special Interest Group on Information Retrieval), TREC (Text REtrieval Information), CIR (Context-based Information Retrieval), sont des exemples. Elles se sont utilisées pour comparer différents modèles propriétaires des moteurs de recherche afin d'aider le domaine à progresser vers des moteurs de recherche meilleurs et efficaces, notamment pour des collections de données plus larges (par exemple la collection de test TREC primaire, en 1992, contient autour de 2 giga octets de textes, soit entre 500.000 et 1.000.000 des documents) (Voorhees, 2003).

Du point de vue contenu, ces conférences présentent des collections de tests comportant trois parties, un ensemble de documents, un ensemble de besoins informationnels (appelés topics dans TREC) et un ensemble de jugements de pertinence indiquant les documents répondant au mieux à un besoin informationnel donné. Voorhees donne plus de détails sur ces notions (Voorhees, 2003).

I.2.2. Le processus de la recherche d'information

Le processus de la recherche d'information se matérialise au moment où un utilisateur résume son besoin d'information sous une requête, le système va l'indexer. Simultanément la collection de documents est également indexée par le système et sera alors en mesure de construire les représentations et de mettre en correspondance la représentation de la requête avec les représentations des documents de la collection grâce à l'index (documents et

³<http://www.mit.edu/~mkgray/net/web-growth-summary.html>, la bibliographie de Matthew Gray sur : <http://www.mit.edu/people/mkgray/index.3.html>.

requêtes). Puis, le système retourne alors une liste de documents considérés par le système de la recherche d'information comme pertinents par rapport à la requête.

Avant de décrire en détail ces différentes fonctions du SRI, nous allons brièvement définir les deux acteurs nécessaires à son fonctionnement, à savoir d'une part l'information disponible, c'est à dire le corpus documentaire, et d'autre part l'utilisateur et son besoin en information exprimé au travers d'une requête (figure I.1).

- La requête : un besoin d'information, c'est le résultat d'une analyse conceptuelle du besoin d'information d'un utilisateur de façon plus ou moins précise. Elle est créée par l'utilisateur, c'est elle qui initie le processus de recherche. C'est une situation problématique qui amène l'utilisateur à formuler une requête ;
- Collection de documents : représente l'ensemble des informations ou de granules documentaires qui sont exploitables, compréhensibles et accessibles par l'utilisateur. Un granule définit la partie sélectionnée comme réponse à une requête de l'utilisateur ;
- Besoin en information : un besoin d'information est une représentation mentale qui amène l'utilisateur à rechercher, c'est l'intérêt de la recherche. La requête n'est qu'une représentation d'un besoin en information. donc on peut dire que la notion de la « requête » et « besoin » sont souvent confondus (Salton, 1971) ;
- Processus d'indexation : dans un SRI, dont l'objectif final est de retourner une liste de documents pertinents par rapport à une requête utilisée, il est nécessaire de pouvoir rechercher les documents de la collection dont le contenu ressemble ou correspond au contenu de la requête. La recherche implique une méthode de tri et la comparaison de contenu entraîne une analyse par défaut de pouvoir directement comparer les concepts véhiculés dans le document à ceux présents dans la requête.

Les mots « représentants » ces concepts sont comparés. Ceux qui sont des unités linguistiques porteuses d'un sens constituent les unités les plus souvent utilisées dans les systèmes actuels. Dès lors pour avoir un système de recherche de qualité, il est important que son index reflète au mieux le contenu de la collection originale.

Indexer un document, c'est élire ses termes représentatifs afin de générer la liste des termes et de l'ajouter à l'index de la collection. Par référence, un moyen de retrouver de façon non ambiguë des documents ou un document ou une partie de document où le terme apparaît.

- La pertinence : concept fondamental dans la recherche d'information. Les travaux de recherche récents s'accordent sur la difficulté de la définition de la pertinence, le principe de celle-ci se base sur la restitution des documents pertinents comme réponse à une requête utilisée. La pertinence des documents se mesure à partir de la perception de l'utilisateur, en plus elle est multidimensionnelle et évolue durant le temps d'une recherche. Les modèles de recherche d'information définis dans la littérature mesurent cette pertinence comme un score, cherchant à évaluer des documents par rapport à une requête. Cette pertinence est mesurée soit par la méthode de similarité basée sur la représentation document-requête (modèle vectoriel), ou une probabilité de pertinence des documents (modèle probabiliste) (Brini, 2005) ou par d'autres méthodes qu'on va définir ultérieurement ;
- Reformulation de requêtes : la requête initiale est vue en RI comme un moyen permettant d'initialiser le processus de sélection d'informations pertinentes. La reformulation rentre dans un processus plus général d'optimisation de la fonction de pertinence qui a pour but de rapprocher la pertinence système de celle de l'utilisateur. Ce processus permet de générer une requête plus adéquate que celle initialement formulée par l'utilisateur. Son principe est de modifier la requête de l'utilisateur par ajout de termes significatifs et/ou ré-estimation de leurs poids, selon deux approches (Brini, 2005) :
 1. la première est basée sur les techniques d'association de termes (méthode directe) ;
 2. la seconde est basée sur les jugements utilisateurs (méthode indirecte), communément appelée relevance feedback ou retour de pertinence.

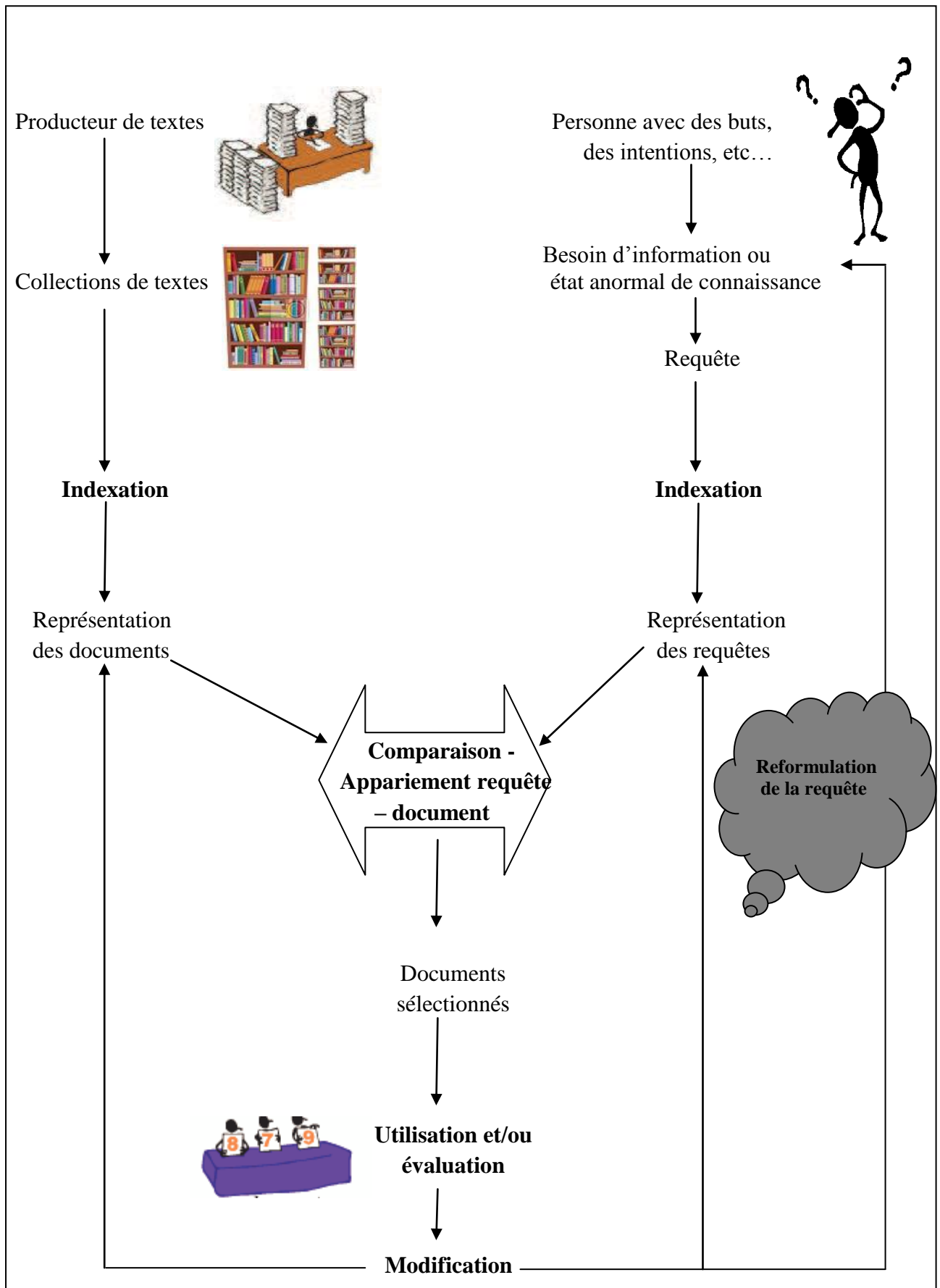


Figure I.1: Processus en U de la RI

Le principe de la méthode directe est d'ajouter à la requête initiale des termes sémantiquement proches. Cette proximité entre termes est obtenue selon différentes manières telles que des études sur le langage naturel, des mesures statistiques sur les documents de la collection, des calculs de corrélations entre termes.

La seconde méthode (indirecte) permet à l'utilisateur de juger les documents restitués par le système pour repondérer les termes de la requête initiale ou ajouter-supprimer des termes qui se trouvent dans les documents jugés pertinents/non pertinents.

I.2.3. Système de recherche d'information

Un système de recherche d'information (SRI) est un système qui intègre un ensemble de techniques et de mécanismes qui permettent de sélectionner les informations documentaires pertinentes dans l'objectif de répondre au besoin de l'utilisateur.

Un système de recherche d'information intègre trois fonctions principales représentées schématiquement par le processus en U de la recherche d'information (Razan, 2004). La Figure I.1 illustre l'architecture générale d'un système de recherche d'information.

D'un côté, on a l'information accessible dans le système. Elle est en général le résultat de la collecte de documents ou de sous collections de documents traitant d'un même domaine ou de domaines proches.

D'un autre côté, on a le besoin en information exprimé par l'utilisateur, en général sous forme de requête, une fois stabilisé.

Ensuite, l'information aussi bien que le besoin en information passe par des étapes de traitement pour qu'ils soient exploitables. Ces processus s'appuient sur un certain nombre de modèles permettant de sélectionner des informations pertinentes en réponse à une requête utilisateur. Il s'agit principalement des processus de représentation et de recherche.

Retrouver de l'information dans des documents suppose :

- Soit que l'ordinateur dispose des documents sur support électronique. Le système de recherche d'information construit un représentant de chacun des documents. Celui-ci se limite souvent à une liste de mots-clés. Le but escompté est d'obtenir une représentation synthétique de la sémantique des documents. L'ensemble complet de ces représentations est mémorisé sur un disque ;

- Soit que l'utilisateur dispose d'outils informatiques pour décrire et exprimer ce qu'il désire sous forme de requêtes écrites. Ces dernières sont analysées par le système de peur d'en tirer des représentations qui soient compatibles avec celles des documents. Disposant d'une représentation interne des requêtes et des documents, le système effectue un appariement afin de déterminer les documents qu'il juge pertinents « pertinence système » avec chacune des requêtes. Une fois l'appariement réalisé, le système sélectionne les documents les plus « prometteurs » et les présente à l'utilisateur. En fait, un système de recherche d'informations ne visualise souvent qu'une référence au document et pas le document lui-même. Sur la base de cette première réponse du système, l'utilisateur peut indiquer au système les documents qu'il juge réellement importants « pertinence utilisateur » et ceux qui n'ont aucun intérêt. À l'aide de ces informations, le système est capable de construire automatiquement une nouvelle requête (bouclage de pertinence ou rétroaction, relevance feedback (Salton, 1990) au point d'afficher une nouvelle liste de références. Ce processus, qui se révèle bénéfique et efficace, indique clairement que la recherche d'information doit toujours être vue comme un processus itératif.

I.2.4. Catégorie des SRI

I.2.4.1. Les Annuaire

Un annuaire est un répertoire de sites, organisé selon un classement thématique ou géographique fait par un éditeur. Chaque site est commenté par l'éditeur et classé par catégories.

L'annuaire est surtout utile pour explorer un sujet général, trouver des sites ressources dans un domaine ou encore trouver des sites similaires sur un même thème (exemple d'annuaire figure I.2).



Figure I.2: Annuaire Yahoo

Les modes de recherche

Deux modes de recherche sont possibles avec un annuaire :

- Par mot-clé : il s'agit de taper dans l'annuaire, le mot-clé sur lequel l'utilisateur souhaite faire des recherches. L'annuaire soumet alors la liste des sites répertoriant le mot recherché ;
- Par arborescence : l'annuaire propose une liste de thématiques et de sous thématiques. La recherche s'effectue alors en avançant de sous-thèmes en sous-thèmes jusqu'à obtenir une liste de sites correspondant au sujet.

En revanche, il existe d'autres types d'annuaires assez utiles pour la recherche d'information. Cependant, c'est à noter qu'ils ne couvrent que certains domaines bien déterminés comme des personnels associatifs, des entreprises ou souvent des personnes

ordinaires. On en cite le plus répandu, à savoir l'annuaire téléphonique qui répertorie les coordonnées selon des thématiques et assez souvent suivant un ordre précis cerné couramment en alphabétique.

L'annuaire téléphonique se trouve sur deux modèles distincts selon la nature des informations exposées : les pages jaunes et les pages blanches.

- Les pages jaunes regroupent les coordonnées des professionnels, des entreprises ou des administrations ;
- Les pages blanches regroupent les coordonnées sur quoi une personne, du moins qu'elle ne figure plus dans une liste rouge. Ladite liste sert pour un souci de sécurisation et de protéger les coordonnées personnelles.

The screenshot shows a search interface with a yellow header. Navigation tabs include 'Annuaire Professionnel', 'Annuaire Inversé', 'Annuaire Du monde', and 'Carte Itinéraire' (marked 'nouveau'). Search fields are labeled 'Qui, Quoi' and 'Où'. The search results section displays '2 résultat(s) pour votre requête.' and lists two entries for 'Pharmacie 2 Mars' in Casablanca, including addresses and phone numbers. A 'Booking.c' advertisement is visible on the right side.

Pharmacie du 2 Mars	05 22 28 57 84
575, avenue du Deux Mars Quartier: Nouvelle Médina 20550 CASABLANCA	
Plan d'accès >	
Pharmacie prolongement 2 Mars	05 22 21 46 73
avenue du Deux Mars, Lot. Safaa rue 10 n°158 Quartier: Ain Chok 20480 CASABLANCA	
Plan d'accès >	

Figure I.3: Exemple : Page jaunes

I.2.4.2. Les moteurs de recherche

On appelle moteur de recherche une base de données générée et traitée automatiquement par différents outils. Sa fonction première est de trouver de l'information sur Internet. Après saisie des mots-clés, le moteur de recherche retrouve toutes les pages web contenant de l'information en rapport avec les dits mots-clés. Ils présentent par suite une liste de résultats classée en fonction du rapport entre les mots-clés utilisés, le contenu et la popularité de la page. En marge de ces résultats apparaissent généralement :

- Les liens sponsorisés ;
- Nom de domaine : correspond à un emplacement sur le web et il désigne un site. En référencement, le nom de domaine est un élément important de l'URL, aussi il est important de définir un nom de domaine qui contiendra si possible les mots-clés principaux. Le nom de domaine contient toujours un suffixe de deux ou trois lettres désignant le pays ou le type d'organisation du domaine ;
- Optimisation : On appelle optimisation l'ensemble des techniques qui visent à classer le mieux possible la page ou le site sur un outil de recherche (les moteurs de recherche et les annuaires). Le terme optimisation est utilisé quand ces techniques sont autorisées par les outils de recherche.

Il existe par ailleurs des outils de recherche internes, lesquels permettent à l'internaute d'effectuer ses recherches à l'intérieur même du site sur lequel il se trouve.

Historique des moteurs de recherche :

Dans cette section, nous allons faire un bref rappel historique sur la naissance et l'évolution des moteurs de recherche. Nous n'allons bien sûr pas être exhaustifs mais nous essaierons de donner un bref aperçu de l'évolution des moteurs de recherche.

- 1990 - L'ancêtre "Archie" :

Tous les moteurs de recherche descendent d'Archie, un logiciel conçu pour rechercher des documents sur Internet et créé par Adam Emtage, étudiant à l'université McGill Québec.

- 1993 - Wanderer est né, les "spiders" attaquent :

Le premier moteur de recherche digne de ce nom naît avec le web : il s'agit du Wanderer ("le vagabond"), un robot mis au point par Matthew Gray. Une armée de spiders (logiciels chargés de sillonner le web) artisanaux déferlent sur la toile.

- 1994 - Un annuaire nommé Yahoo :

Deux étudiants de l'université Stanford, Jerry Yang et David Filo, ont une idée qui va changer la face du web : sélectionner et recenser humainement les meilleurs sites dans un annuaire Internet. Yahoo est né, il devient en quelques mois le portail le plus utilisé par les internautes.

- 1995 - Lycos et Excite :

Les machines à chercher se perfectionnent. Lycos, qui doit son nom à une araignée particulièrement rapide, est mis en ligne en juin 1995 par l'université Carnegie Mellon (Pennsylvanie). Quelques semaines plus tard, le moteur Excite, mis au point par des étudiants de Stanford, se lance à son tour.

- 1996 - Altavista :

Lancé en décembre 1995, Altavista est mis au point par le Français Louis Monnier pour les laboratoires Digitaux. Il est rapide, pertinent et exhaustif. Il propose de multiples fonctionnalités de recherche, notamment par langues. Jusqu'en juillet 1996, les seuls outils de recherche destinés aux francophones sont québécois. Gilles Ghesquière et Jean Postaire y remédient en créant Nomade.fr. Il faudra attendre 1998 pour que France Télécom lance son propre moteur de recherche, Voila.fr, créant ainsi le premier moteur de recherche français.

- 1997 – Inktomi conquérait le B to B (business to business) :

Alors que tous les moteurs de recherche tentent leur mue en portails grand public financés par la publicité, Inktomi se spécialise dans la fourniture de solutions de recherche internet aux entreprises. Yahoo choisit sa technologie pour se "motoriser".

- 1998 - Un bijou nommé Google :

L'université de Stanford produit deux nouveaux petits génies : concepteurs d'un moteur de recherche baptisé Google, Sergei Brin et Larry Page vont révolutionner le secteur. À la fois

pertinent et exhaustif, cet engin de nouvelle génération classe les résultats de recherche en fonction de leur popularité auprès des internautes. Le bouche-à-oreille fait le reste.

- 1999 - fast, le moteur venu du froid :

Inspirés par le succès de Google, des chercheurs de l'université d'Oslo créent Fast Technology : comme Inktomi, fast se positionne sur la fourniture de technologies pour les entreprises. Un challenger est né.

- 2000 - Yahoo consacre Google :

Élu "meilleur moteur de recherche de l'année 2000" par Searchenginewatch.com, Google est choisi par Yahoo comme moteur de recherche privilégié à la place d'Inktomi. Grâce à cette publicité, Google devient l'outil de recherche le plus utilisé par les internautes (notamment en France), et rêve désormais d'entrer en Bourse.

- 2002 - Google conquiert AOL :

La petite star des moteurs est à son apogée : le géant AOL (34 millions d'abonnés) vient de le choisir en lieu et place du malchanceux Inktomi. Parallèlement, Google s'attire les premières critiques en proposant des résultats de recherche "sponsorisés" par des annonceurs.

Comme nous pouvons le constater, les débuts des moteurs de recherche sont récents : une petite quinzaine d'années. De nombreux bouleversements ont déjà eu lieu et auront très certainement encore lieu. En effet, les moteurs de recherche ont encore un long chemin à parcourir avant d'arriver à la réponse parfaite. Il y a de fortes chances pour que les moteurs de l'avenir intègrent des outils d'analyse linguistique. C'est-à-dire qu'ils pourront lier des ensembles de mots entre eux. Par exemple, une requête sur "berger allemand" proposera des pages sur la race de chiens "berger allemand" en priorité.

I.2.4.3. Le méta-moteur de recherche

Un méta-moteur ou un méta-chercheur est un logiciel qui puise ses informations à travers plusieurs moteurs de recherche. Plus précisément, le méta-moteur envoie ses Requêtes à plusieurs moteurs de recherche, et retourne les résultats de chacun d'eux. Le méta-moteur permet aux utilisateurs de n'entrer le sujet de leur recherche qu'une seule fois mais d'accéder aux réponses de plusieurs moteurs de recherche différents (figure I.4).

Un méta-moteur élimine les résultats similaires ; par exemple, si Google et Yahoo renvoient sur les deux les mêmes liens, le méta-moteur ne va l'afficher qu'une seule fois dans la liste des résultats. Enfin un méta-moteur trie les résultats pour fournir en premier les pages fournies par plusieurs moteurs. Certains méta-moteurs permettant en outre de mélanger une fonction annuaire (les résultats sont classés par thème) et une fonction motrice. Cela permet d'avoir une double vue sur les résultats.

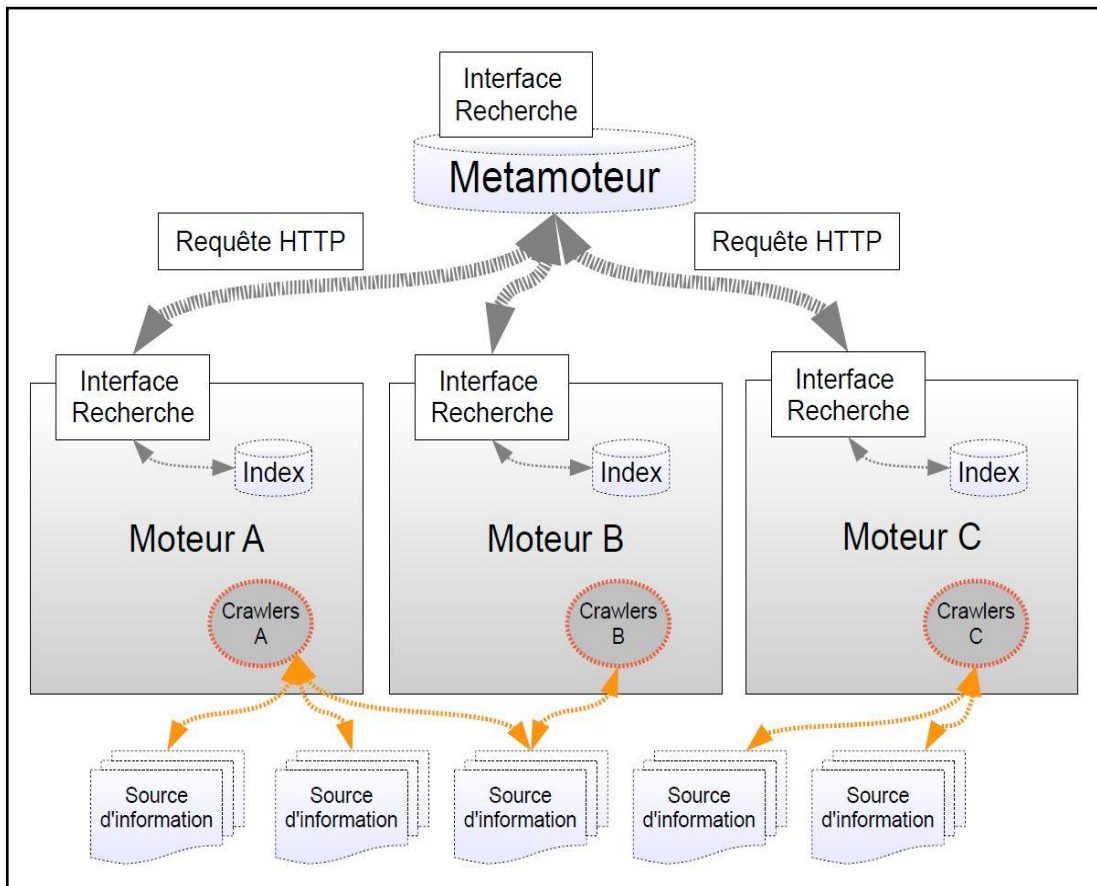


Figure I.4: Processus d'un méta-moteur de recherche

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

Nous allons examiner dans cette section les différents modèles permettant de spécifier la présence, l'absence ou encore la proximité de termes dans un document. Nous étudierons en premier lieu le modèle booléen et le modèle booléen étendu à l'origine des premiers moteurs de recherche. Nous détaillerons en second lieu d'autres méthodes autorisant la spécification des requêtes sous une forme différente comme le modèle vectoriel qui permet de

mesurer le degré de similarité entre deux documents. Nous nous intéresserons en dernier lieu aux modèles : probabiliste et bayesiens.

On a alors :

- un ensemble de M termes d'indexation : $T = \{t_1, \dots, t_M\}$;
- un ensemble de N documents : $D = \{d_1, \dots, d_N\}$;
- des requêtes : q_1, q_2, \dots, q_n .

Les documents et requêtes sont représentés par une combinaison de termes appartenant à T.

Il s'agit de construire des formalismes permettant :

- de représenter les documents et les requêtes ;
- de calculer la similarité entre un document et une requête ainsi représentés.

I.3.1. Le modèle Booléen

C'est le modèle le plus ancien de tous les modèles de la recherche d'information (Salton, 1971), il se base essentiellement sur la théorie des ensembles.

- Un document est représenté comme une conjonction logique de termes (non pondérés : termes d'indexation), par exemple, $d = t_1 \wedge t_2 \wedge \dots \wedge t_n$;
- Une requête est une expression logique quelconque de termes. On peut utiliser les opérateurs et (\wedge), ou (\vee) et non (\neg). Par exemple : $q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$.

Le processus de recherche mis en œuvre consiste à effectuer des opérations sur l'ensemble des documents afin de réaliser un appariement exact avec l'équation de la requête. La correspondance $R(d, q)$ s'appuie sur la présence ou l'absence des termes de la requête dans les documents et se concrétise de la façon suivante :

- $R(d, t_i) = 1$ si $t_i \in d$; sinon 0 ;
- $R(d, q_1 \wedge q_2) = 1$ si $R(d, q_1) = 1$ et $R(d, q_2) = 1$; sinon 0 ;
- $R(d, q_1 \vee q_2) = 1$ si $R(d, q_1) = 1$ ou $R(d, q_2) = 1$; sinon 0 ;

- $R(d, \neg q1) = 1$ si $R(d, q1) = 0$; sinon 0.

Les avantages du modèle booléen :

1. Les formalismes de description des documents et des requêtes font partie du même langage ;
2. Le modèle est plus facile à implémenter et nécessite relativement peu de ressources (Salton, 1990) ;
3. Un formalisme précis, la logique des propositions ;
4. Le langage de requête booléen est plus expressif que celui des autres modèles à savoir le modèle vectoriel, probabiliste, etc....(Crof, 1987).

Les inconvénients du modèle booléen

1. La décision binaire sur laquelle est basée la sélection d'un document ne permet pas d'ordonner les documents renvoyés à l'utilisateur selon un degré de pertinence ;
2. La sélection d'un document est basée sur la décision binaire, un document est soit pertinent ou non. En conséquence, le système détermine un ensemble de documents non ordonnés comme réponse à une requête. Il n'est pas possible de dire quel document est mieux que l'autre. Cela crée beaucoup de problèmes aux usagers, car ils doivent encore fouiller dans cet ensemble de documents non ordonnés pour trouver des documents qui les intéressent ;
3. La notion de pondération des termes n'est pas prise en compte, un terme a un poids égal à 1 s'il appartient au document et 0 s'il n'appartient pas ;
4. Le modèle ne permet pas de retourner un document s'il ne contient qu'une partie des mots de la requête (si le connecteur ET est utilisé) ;
5. Pour une requête qui est une longue conjonction, un document qui satisfait la majorité de termes est aussi mauvais qu'un document qui ne satisfait aucun terme ; pour une requête qui est une longue disjonction, un document qui satisfait un terme est aussi bon qu'un document qui satisfait tous les termes ;

6. Il est difficile aux utilisateurs de formuler une requête combinant plusieurs opérateurs logiques, notamment pour les questions complexes. L'importance relative des mots-clés ne peut pas être exprimée. En particulier, les opérateurs booléens ne correspondent pas exactement aux connecteurs linguistiques.

Comme nous pouvons le remarquer, ce modèle booléen standard n'est utilisé que dans très peu de systèmes de nos jours. Si on utilise un modèle booléen, c'est plutôt une extension de ce modèle qu'on utilise pour remédier à ces inconvénients. Parmi ces modèles, nous citerons Le modèle booléen à base de logique floue.

Le modèle booléen à base de logique floue

Il s'agit d'une extension pour pallier aux inconvénients du modèle booléen qui vise à tenir compte de la pondération des termes dans les documents en utilisant le formalisme de la logique floue et des ensembles flous proposée par ZADEH en 1965, où un élément possède un degré d'appartenance à un ensemble.

Avec ce formalisme proposé par ZADEH :

- Un terme peut indexer totalement, partiellement ou pas du tout un document. Le degré d'indexation est un réel dans l'intervalle $[0, 1]$;
- Un document peut être totalement, partiellement ou pas du tout similaire à une requête. Le degré de similarité est un réel dans l'intervalle $[0, 1]$.

Du côté requête, elle reste toujours une expression booléenne classique. Avec cette extension, un document est représenté comme un ensemble flou de ses termes pondérés (Buell, 1982) : $d = \{ (t_1, a_1), \dots, (t_i, a_i), \dots \}$, Où a_i est le degré d'appartenance du terme t_i au document d , qui est choisi de façon à refléter le degré de représentativité du terme par rapport au document (Waller et al., 1979 ; Buell et al., 1981). En général, ce poids est principalement basé sur le nombre d'occurrences d'un terme dans le document.

Dans les modèles flous obtenus, l'appariement n'est plus binaire, mais graduel ; on obtient un degré compris entre 0 et 1. Cet appariement entre requêtes et documents et le degré de correspondance ainsi obtenu, que nous assimilons au degré de pertinence, permet d'ordonner les documents.

La fonction de correspondance entre une requête et un document peut être formalisée comme suit :

- $R(d, t_i) = a_i$;
- $R(d, q_1 \wedge q_2) = \min(R(d, q_1), R(d, q_2))$;
- $R(d, q_1 \vee q_2) = \max(R(d, q_1), R(d, q_2))$;
- $R(d, \neg q_1) = 1 - R(d, q_1)$.

Dans cette évaluation, les opérateurs logiques \wedge et \vee sont évalués par min et max respectivement. C'est une des évaluations classiques proposées par L. Zadeh dans le cadre des ensembles flous. Cependant, cette évaluation n'est pas parfaite. Par exemple, on n'a pas $R(d, q \wedge \neg q) \equiv 0$ et $R(d, q \vee \neg q) \equiv 1$, ce qui signifie, quand on évalue une requête en forme de conjonction, on s'intéresse à la requête dont le degré d'appartenance aux documents est faible, tandis que, quand on évalue une requête en forme de disjonction, le composant ayant un degré d'appartenance plus élevé est pris en compte.

Un autre formalisme est proposé par l'évaluation de Lukaswicz (Lukaswicz, 1963) défini comme suit :

- $R(d, t_i) = a_i$;
- $R(d, q_1 \wedge q_2) = R(d, q_1) * R(d, q_2)$;
- $R(d, q_1 \vee q_2) = R(d, q_1) + R(d, q_2) - R(d, q_1) * R(d, q_2)$;
- $R(d, \neg q) = 1 - R(d, q)$.

Dans cette évaluation, les opérateurs logiques de conjonction ou de disjonction (\wedge et \vee) jouent toutes les deux un rôle en contribuant en même temps à l'évaluation contrairement au formalisme classique proposé par L. Zadeh dans le cadre des ensembles flous. Cependant, cette évaluation n'est pas aussi parfaite en remarquant qu'elle a le même problème (on n'obtient toujours pas $R(d, q \wedge \neg q) \equiv 0$ et $R(d, q \vee \neg q) \equiv 1$). En plus, ($R(d, q \wedge q) \neq R(d, q) \neq R(d, q \vee q$). Mais il reste convenable.

Même si, ces extensions présentent quelques problèmes, elles ont constitué pendant la fin des années 1970 et au début des années 1980, un standard pour les modèles booléens. Le plus important c'est de mesurer le degré de similarité entre un document et une requête dans l'intervalle $[0, 1]$. Ainsi, peut-on ordonner les documents dans l'ordre décroissant de leur pertinence avec la requête. L'utilisateur peut donc parcourir la liste ordonnée des résultats

renvoyés et décider où s'arrêter. Au niveau de la représentation, nous avons également une représentation plus raffinée nous permettant d'exprimer le degré d'importance d'un terme dans le document.

I.3.2. Le modèle p-norme

Le modèle p-norme introduit par salton (Salton et al., 1983) est aussi proposé pour pallier à certaines déficiences observées dans le modèle booléen standard (Picarougne, 2004). Ce modèle mesure les correspondances de la conjonction et de la disjonction par attribution d'une pondération au terme des documents et des requêtes ainsi qu'aux opérateurs booléens AND et OR.

L'idée de base réside dans l'observation de la table de vérité de la conjonction et de la disjonction (voir table I.1). Pour la conjonction : dans la colonne $A \wedge B$, la meilleure correspondance est atteinte dans le cas de la dernière ligne. Tandis que, pour la disjonction : dans la colonne $A \vee B$, la pire correspondance correspond à la première ligne qu'il faut éviter.

A	B	$A \wedge B$	$A \vee B$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Tableau I.1: Table de vérité

Ainsi, l'évaluation d'une conjonction ou d'une disjonction consiste à calculer une sorte de distance entre le point à atteindre ou à éviter. L'idée de base correspond aux figures suivantes : dans ces schémas (figure I.5 et I.6), les axes des abscisses correspondent à l'évaluation du document A et les axes des ordonnées correspondent à l'évaluation du document B.

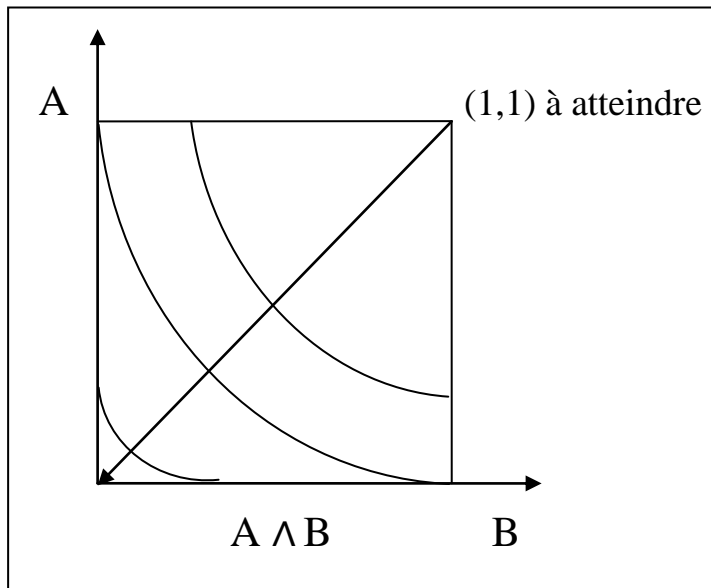


Figure I.5: Évaluation d'une conjonction

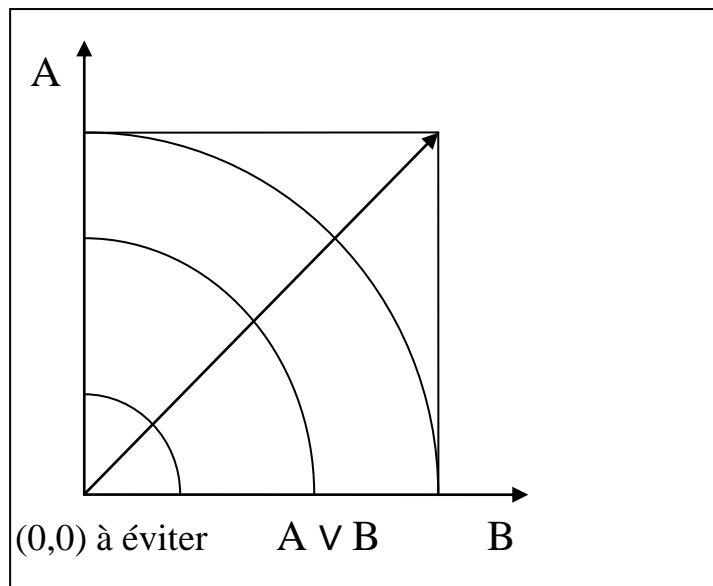


Figure I.6 : Évaluation d'une disjonction

- Dans la première figure de conjonction ($A \wedge B$), on cherche à évaluer dans quelle mesure ce point c défini par l'évaluation d'un document A et d'un document B est proche de (1, 1) - le point à atteindre. Ce rapprochement peut être mesuré par le complément de la distance entre le point c et le point (1,1). Plus cette distance n'est grande, moins $A \wedge B$ est satisfait.

- Dans la deuxième figure de disjonction (AVB), on cherche à éviter le point (0,0). Plus on est loin de (0,0), plus $A \vee B$ est satisfait.

Dans cette optique, SALTON proposa les évaluations normalisées suivantes en admettant la pondération de termes dans les documents : p_i est le poids de t_i dans d .

- $R(d, t_i) = p_i$;
- $R(d, q_1 \wedge q_2) = 1 - ([(1-R(d, q_1))^2 + (1-R(d, q_2))^2] / 2)^{1/2}$;
- $R(d, q_1 \vee q_2) = [(R(d, q_1)^2 + R(d, q_2)^2) / 2]^{1/2}$;
- $R(d, \neg q_1) = 1 - R(d, q_1)$.

Le principal avantage de ce modèle sur le modèle booléen classique réside dans la mesure du degré de correspondance ou de similarité entre un document et une requête dans [0, 1]. On peut ainsi ordonner les documents dans l'ordre décroissant de leur pertinence avec la requête. L'utilisateur peut donc parcourir la liste ordonnée des résultats renvoyés et décider où s'arrêter.

I.3.3. Le modèle vectoriel

Le modèle vectoriel est un modèle algébrique qui se base sur la représentation des documents et des requêtes par des vecteurs dans un espace multidimensionnel dont les dimensions sont les termes issus de l'indexation (Salton, 1983). Comme on l'a vu précédemment, la création de l'index implique le parcours de la collection, la recherche des termes pertinents, le traitement lexical des termes retenus et enfin l'analyse statistique de la distribution de ces termes dans les documents et dans la collection pour leur attribuer un poids. Ainsi, les documents et la requête sont représentés comme des vecteurs dans le repère des termes. La comparaison de la requête au document est effectuée en comparant leurs vecteurs respectifs. On ramène ainsi une proximité sémantique à une mesure de distance géométrique (exemple d'une représentation figure I.7).

Soit R l'espace vectoriel défini par l'ensemble des termes : $\langle t_1, t_2, \dots, t_n \rangle$. Un document d et une requête q peuvent être représentés par des vecteurs de poids comme suit :

- $D \longrightarrow \langle wd_1, wd_2, \dots, wd_n \rangle$;

- $Q \rightarrow \langle wq_1, wq_2, \dots, wq_n \rangle$;
- w_{di} et w_{qi} correspondent respectivement aux poids du terme t_i dans le document d_i et dans la requête q et n correspond au nombre de termes de l'espace.

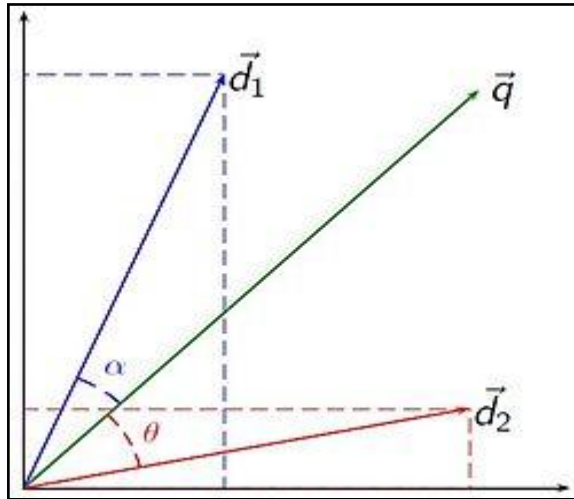


Figure I.7: Représentation requête-document

I.3.3.1. Mesure de similarité :

L'attribution d'une note de pertinence (système) aux documents, ce qui revient à faire qu'une comparaison de vecteurs, peut se réaliser de différentes manières : par exemple par un produit scalaire :

$$CS(Q, D_j) = \sum_{i=1}^t w_{Qi} * w_{Dij}$$

Avec :

- w_{Qi} : le poids du terme i dans la requête Q ;
- w_{Dij} : le poids du terme i dans le document j .

Lorsque les poids appartiennent à l'intervalle $[0, 1]$, CS mesure en fait la cardinalité de l'ensemble $Q \wedge D_j$.

La similarité peut être mesurée par le calcul de l'angle qui forme les vecteurs, la méthode la plus connue et la plus utilisée est :

$$CS \cos (Q, Dj) = \frac{\sum_{k=1}^t W_{Q_k} * W_{Dkj}}{\sqrt{\sum_{k=1}^t (W_{Dkj})^2 \sum_{k=1}^t (W_{Q_k})^2}}$$

D'autres méthodes géométriques qui ont été expérimentées dans ce modèle, telles que la distance euclidienne, le coefficient de Jaccard, de Dice etc. Salton et Buckley ont expérimenté leurs mesures en utilisant des représentations de documents plus complexes en tenant compte des termes reliés, phrases, termes de thesaurus. Cependant, ces méthodes n'ont pas montré une nette amélioration.

Singhal a défini une fonction dite normalisation de pivot, contient deux paramètres : le pivot et le gradient qui se basent sur une transformation de normalisation par rapport à la longueur du document pour augmenter les scores des documents longs et diminuer les scores des documents courts. Ainsi, le calcul de similarité est-il représenté par la formule suivante :

$$CS \text{ Normalisée } (Q, Dj) = \sum_{i=1}^{UT_Q} \frac{w_{Qi}}{(1-s) \times p + s \times UT_Q} \frac{\frac{W_{Dj}}{1 + \log(\text{avg}Dj)}}{(1-s) \times p + s \times UT_{Dj}}$$

Avec :

UT_Q et UT_{Dj} : Le nombre de termes ayant un nombre d'apparitions égal à 1 dans Q et Dj respectivement ; $\text{avg}Dj$: Le nombre moyen d'apparitions de termes dans le document Dj ; p : le nombre moyen de termes dans les documents, $(\frac{\sum_{j=1}^N |Dj|}{N})$, $|Dj|$ la cardinalité du document Dj et N le nombre de termes) ; S : constante trouvée expérimentalement, fixée à 0.2.

I.3.3.2. Pondération

Pour désigner l'absence et la présence d'un terme dans un vecteur on utilise respectivement 0 et 1. La mesure qui a semblé donner les meilleures performances pour un terme i d'un document j , notée W_{ij} , est :

$$W_{ij} = \frac{w_{ij}}{\sum_{k=1}^t [w_{kj}^2]}$$

Avec :

- $w_{ij} = (1 + \log t_{fij}) * \text{idf}_i$ Et $\text{idf}_i = \log (N+1)/n$ ou $\log(N/n)$;
- t_{fij} : nombre d'apparitions du terme i dans le document j et t le nombre de termes dans le document j ;
- N_i : nombre de documents contenant le terme t_i et N le nombre de documents de la collection.

I.3.3.3. Exemple ⁴

Soit la collection suivante :

- $D1 = \{4t_1, 6t_4\}$; $D2 = \{20t_2, 10t_3, 15t_5, 5t_6\}$; $D3 = \{t_2, t_3, t_5\}$;
- $D4 = \{t_2, 15t_3, 10t_5\}$; $D5 = \{15t_1, 15t_2, 15t_3\}$;
- $Q = \{t_2, t_3, t_6\}$.

Dans cet exemple de collection, le document D1 contient les termes t_1 et t_4 . tel que le terme t_1 a un nombre d'apparitions égal 4 fois et le terme t_4 6 fois.

	t2	t3	t6
wi1	0	0	0
wi2	2.30	2.00	1.69
wi3	1	1	0
wi4	1	2.17	0
wi5	2.17	2.17	0
wQi	0.09	0.09	0.69

Tableau I.2: Poids des termes

⁴ Exemple extrait de (Brini, 2005)

D2
D3 ; D5
D4

Tableau I.3: Classement des documents par cosinus

La méthode de normalisation par pivot nécessite le calcul des variables suivant :

$$p = 3, s = 0.2 ; 1 + \log (avg_{tf_{D_1}}) = 1.69 ; 1 + \log (avg_{tf_{D_2}}) = 2.09 ; 1 + \log (avg_{tf_{D_3}}) = 1 ; 1 + \log (avg_{tf_{D_4}}) = 1.93 ; 1 + \log (avg_{tf_{D_5}}) = 2.17.$$

	t2	t3	t6
wi1	0	0	0
wi2	3.97	1.98	0.99
wi3	0.33	0.33	0
wi4	0.19	2.97	0
wi5	2.87	2.87	0
wQi	0.03	0.03	0.23

Tableau I.4: Poids des termes obtenus par la méthode de pivot

Nous avons utilisé t_f car cela a semblé donner des meilleurs résultats que l'utilisation de $1 + \log (t_f)$.

D2
D5
D4
D3

Tableau I.5: Classement des documents selon la normalisation par pivot

Les avantages du modèle vectoriel

1. Le modèle vectoriel est relativement simple à appréhender (algèbre linéaire), facile à implémenter et nécessite relativement peu de ressources (Salton, 1990) ;
2. Il permet de retrouver assez efficacement des documents dans un corpus non structuré (recherche d'information) ;
3. Son efficacité dépendant pour une grande part à la qualité de la représentation (vocabulaire et schéma de pondération) ;
4. La représentation vectorielle permet aussi une mise en correspondance des documents avec une requête imparfaite.

Les inconvénients du modèle vectoriel

1. Ce modèle suppose que les termes représentatifs sont indépendants ;
2. Dans un texte, l'ordre des mots n'est pas pris en compte ;
3. Dans sa version la plus simple, il ne prend pas non plus en compte les synonymes ou la morphologie des contenus.

I.3.4. Les modèles probabilistes

I.3.4.1. Définition

Le modèle probabiliste de pertinence est une méthode probabiliste de représentation du contenu d'un document, proposée en 1976 par Robertson et Jones (Robertson et al., 1976). Elle est utilisée en recherche d'information pour exprimer une estimation de la probabilité de pertinence d'un document par rapport à une requête, et ainsi classer une liste de documents dans l'ordre décroissant d'utilité probable pour l'utilisateur. L'une des applications directes de ce modèle est la méthode de pondération Okapi BM25, considérée comme l'une des plus performantes dans le domaine.

I.3.4.2. Formalisation

Probabilité de pertinence tente de répondre dans ce système, pour chaque document et chaque requête, à la question : Quelle est la probabilité que ce document soit pertinent pour cette requête ? Ainsi, deux événements sont possibles :

- L, D est pertinent pour Q ;
- \bar{L} , \bar{D} est non pertinent pour Q.

La théorie des probabilités forme donc la base de ces modèles, et tout particulièrement le théorème de Bayes :

$$P(L/D) = \frac{P(D/L)P(L)}{P(D)}$$

Après un développement par l'élimination de $P(D)$ non utile et l'ajout de \log , on obtient :

$$\log \frac{P(L/D)}{P(\bar{L}/D)} = \log \frac{P(D/L)P(L)}{P(D/\bar{L})P(\bar{L})} = \log \frac{P(D/L)}{P(D/\bar{L})} + \log \frac{P(L)}{P(\bar{L})}$$

Donc, Le score d'appariement entre le document D et la requête, noté $P(D)$ est :

$$P(D) = \log \frac{P(L/D)}{P(\bar{L}/D)} - \log \frac{P(L)}{P(\bar{L})}$$

Remarque :

- Si $P(L/D) = \frac{P(D/L)P(L)}{P(D)} > 1$ ou $\log \frac{P(D/L)P(L)}{P(D)} > 0$ donc D est pertinent.

Les avantages du modèle probabiliste

1. Les modèles probabilistes constituent un outil puissant pour les modèles de RI, car ils permettent de traiter d'une manière efficace l'incertitude intrinsèque au processus de RI ;

2. Selon Savoy (Savoy, 1994), le modèle de recherche probabiliste est plus efficace que le modèle de recherche booléen, mais **moins** performant que le modèle de recherche vectoriel.

L'inconvénient du modèle probabiliste

- Il n'existe pas de méthode d'estimation de la pertinence des termes avant toute extraction de document pertinent. Cette estimation se fait a posteriori.

I.3.4.3. Le modèle de BIR

Le modèle a été défini par la considération des attributs définissant l'univers du discours des documents de la collection comme indépendants. Ici nous parlons pas de termes t_i , mais plutôt d'un attribut A_i du document.

La probabilité de pertinence (ou non pertinence) d'un document noté $P(D/L)$ (respectivement $P(D/\bar{L})$) est donnée :

$$P\left(\frac{D}{L}\right) = \prod_i P(A_i = a_i/L)$$

$$P\left(\frac{D}{\bar{L}}\right) = \prod_i P(A_i = a_i/\bar{L})$$

Donc :

$$P(D) = \sum_i \log \frac{P(A_i = a_i/L)}{P(A_i = a_i/\bar{L})}$$

Les attributs manipulés dans le modèle caractérisent des termes. Soit $p(A_i = 1 | L)$, notée par p_i ; et $p(A_i = 1 | \bar{L})$, notée par \bar{p}_i .

La fonction W donne le poids, w_i , pour la présence de l'attribut i :

$$w_i = \log \frac{p_i(1 - \bar{p}_i)}{\bar{p}_i(1 - p_i)}$$

I.3.4.3.1. Pondération

Une première estimation des variables mesure l'importance d'un terme dans la collection. Le poids affecté au terme est égal à la fréquence inverse normalisée, w_i du terme, t_i :

$$w_i = \text{idf}_i$$

Ici $\text{idf}_i = \log(N/n_i)$

- N : est le nombre de documents dans la collection ;
- n_i : le nombre de documents contenant le terme t_i .

Une seconde pondération proposée dans la littérature, donnant le poids du terme i , notée par w'_i , est :

$$w'_i = \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

Après plusieurs améliorations les développeurs ont défini une fonction de pondération d'un terme t_i plus performante :

$$w''_i = \log \frac{r_i(N - n_i - R + r_i)}{(R - r_i)(n_i - r_i)}$$

Avec :

- R : le nombre de documents pertinents pour la requête traitée ;
- r_i : le nombre de documents pertinents contenant le terme.

Pour simplifier, nous notons, $p(A_i = 1 | L) = p$ et $p(A_i = 1 | \bar{L}) = \bar{p}$, alors, pour pallier les problèmes d'incertitude, les auteurs proposent d'ajouter la valeur 0.5 aux valeurs centrales de la table de contingence :

$$w'''_i = \log \frac{(N - n_i - R + r_i + 0.5)(r_i + 0.5)}{(R - r_i + 0.5)(n_i - r_i + 0.5)}$$

I.3.4.4. Le modèle de poisson

En 1994, Robertson et Walker (Robertson et al., 1994) ont proposé le modèle probabiliste basé sur la distribution de poisson pour pallier aux inconvénients du modèle BIR.

La pondération d'un terme est donnée par :

$$w = \log \left(\frac{P_{tf} \bar{p}_0}{\bar{P}_{tf} p_0} \right)$$

Avec :

- P_{tf} et \bar{P}_{tf} : signifiant la probabilité qu'un terme apparaisse avec la fréquence tf dans un document pertinent (respectivement non pertinent) ;
- p_0 et \bar{p}_0 : représentent l'absence du terme dans un document pertinent et non pertinent respectivement.

$$w_{ij} = \frac{tf_{ij}(k_1 + 1)}{k_1 + tf_{ij}} \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

Ou,

- tf_{ij} : est la fréquence du terme t_i dans le document d_j ;
- k_1 est une constante qui détermine à quel point le poids attribué réagit par rapport à une augmentation de tf .

Une autre pondération a été proposée afin de tenir compte de la longueur des documents.

Ainsi :

$$w'_{ij} = \frac{tf_{ij}(k_1 + 1)}{k_1 \times \left((1 - b) + b \frac{ld_j}{\text{avg} - ld_j} \right) + tf_{ij}} \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

Avec :

- ld_j : la longueur du document d ;
- $\text{avg} - ld_j$: la longueur moyenne du document d_j ;

- n_i : le nombre de documents contenant le terme t_i ;
- N : le nombre de documents de la collection ;
- T : le nombre de termes de la collection ;
- Les expérimentations ont montré que $b = 0.75$ donne des résultats de recherche satisfaisantes.

Les fonctions de pondération proposées ont été nombreuses, et celle qui a donné les meilleurs résultats est celle utilisée dans OKAPI BM25 (BM pour Best Match). Le poids du terme t_i de la requête, noté w_{Qi} , tient compte du nombre d'apparitions du terme t_i dans la requête, tf_{Qi} , et d'un paramètre k_2 , de valeur égale à 8 (trouvée expérimentalement) (Brini, 2005). Ainsi :

$$w_{Qi} = \frac{tf_{Qi} \times (k_2 + 1)}{k_2 \times tf_{Qi}}$$

I.3.4.5. Exemple⁵

L'utilisation du modèle de Poisson tient compte des poids tels que définis dans les formules ci-dessus. L'appariement entre un document et une requête est obtenu par le produit des poids des termes de la requête communs à ceux du document. Le tableau I.6, donne les poids des termes de l'exemple présenté dans la section I.3.3.3. Pour cet exemple, nous avons substitué $\log \frac{N-n+0.5}{n+0.5}$ par $\log \frac{N}{n}$. parce que le premier attribue des poids est négative. Dans cette application, nous avons utilisé les valeurs suivantes : $k_1 = 2$; $k_2 = 8$; $b = 0.75$; $avg - ld = 26.8$. Par exemple le poids du terme t_2 dans le document d_2 , de longueur égale a 50 est égale à $w_{22} = (20 \times 3) / (2 \times 0.75 + 0.75 \times ((50) / (26.8)) + 20) \times 0.09691001 \approx 0.2655 \approx 0.27$.

	t2	t3	t6
wi1	0	0	0
wi2	0.27	0.24	1.52
wi3	0.18	0.18	0

⁵ Exemple extrait de (Brini, 2005)

w_{i4}	0.13	0.27	0
w_{i5}	0.26	0.26	0
w_{Qi}	1.125	1.125	1.125

Tableau I.6: Poids des termes

D2
D5
D4
D3

Tableau I.7: Classement des documents

Remarques :

- Toutes les valeurs données dans le tableau I.6 sont données à deux chiffres après la virgule et arrondies au supérieur ;
- Le classement des documents selon le modèle de poisson est identique à celui obtenu par la méthode pivot du modèle vectoriel.

I.3.4.6. Conclusion

Le modèle de recherche probabiliste est basé sur le principe de classement des probabilités. Ce dernier, stipule un système de récupération d'informations censé classer les documents en fonction de leur probabilité de pertinence à la requête. Étant donné que toutes les preuves sont disponibles, le principe prend en considération que l'existence d'une incertitude dans la représentation de la nécessité de l'information peut contenir une variété de sources de données utilisées par les méthodes de récupération probabilistes.

I.3.5. Les modèles Bayésiens de RI

Les modèles probabilistes constituent un outil puissant pour les modèles de RI, car ils permettent de traiter d'une manière efficace l'incertitude intrinsèque au processus de RI. Or, ces modèles (les modèles probabilistes classiques) souffrent de la difficulté d'établir un compromis raisonnable entre le nombre de probabilités de base à estimer et les hypothèses d'indépendances nécessaires à la réduction de ce nombre (Desjardins, 2006).

Les réseaux bayésiens (RB) sont la combinaison des approches probabilistes et de la théorie de graphes. Par ailleurs, les réseaux bayésiens doivent leurs noms aux travaux de Thomas Bayes (1702,1761) au dix-huitième siècle sur « la probabilité des causes », travaux repris plus tard par Laplace et Condorcet. Ils visent à faciliter la description d'une collection de croyance en rendant explicite les relations de causalité et de l'indépendance conditionnelle et à fournir un moyen plus efficace pour mettre à jour les forces de croyances (distribution conjointe de probabilité) lorsque des nouvelles évidences sont observées (Kim et Pearl, 1987).

Basés sur cette intuition, des travaux récents ont permis d'exploiter l'apport des réseaux bayésiens (RBs) pour définir des modèles pour la RI. L'avantage apporté par l'utilisation de ces réseaux a été principalement de pouvoir combiner des informations provenant de différentes sources pour restituer les documents qui seraient les plus pertinents lors d'une requête.

Un réseau bayésien est défini par (Pearl, 1988) :

- Un graphe acyclique orienté $G, G = (V, E)$, où V est l'ensemble des nœuds de G , et E l'ensemble des arcs de G ;
- Un espace probabilisé fini (Ω, Z, p) ;
- Un ensemble de variables aléatoires définies sur (Ω, Z, p) , tel que :

$$P(V_1, V_2, V_3, \dots, V_n) = \prod_{i=1}^n P(V_i \setminus \text{Parents}(V_i))$$

Où $\text{Parents}(V_i)$ est l'ensemble des parents (causes) de V_i dans le graphe.

Exemple ⁶:

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

La représentation graphique du modèle causal utilisé est dans la figure I.8. Cette figure représente un réseau bayésien simple contenant quatre variables binaires, on peut écrire aussi :

$$P(A, B, C, D) = P(A). P(B). P(C| A, B). P(D|B)$$

Où :

- A : Arroseur en marche ;
- B : Il a plu pendant la nuit ;
- C : Herbe du jardin humide ;
- D : Herbe du jardin voisin humide.

⁶ Exemple extrait de (Pearl, 1988)

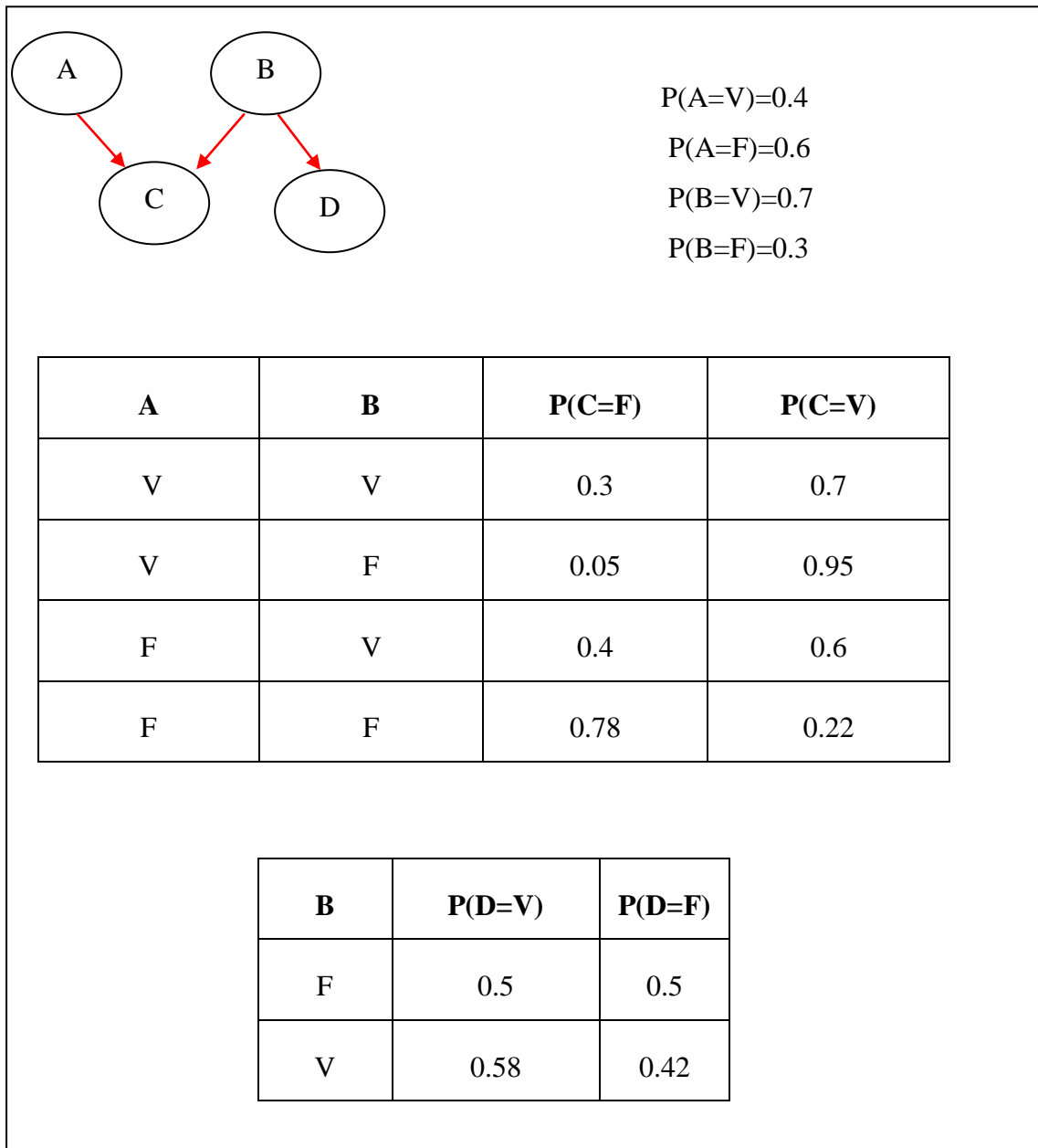


Figure I.8: Exemple de Réseau Bayésien

I.3.5.1. Architecture générale du modèle Bayésien

La figure I.9 présente l'architecture générale du modèle bayésien pour la recherche d'information qui se base sur les réseaux Bayésiens (Elayeb, 2009).

Les nœuds du réseau bayésien pour ce modèle appelé le modèle BNR (modèle de RI basé sur les réseaux Bayésiens) (De Campos et al., 2002; De Campos et al., 2003) ont été décomposés en deux ensembles de variables T et D :

- L'ensemble des termes $T = (TB1B, TB2B, \dots, TBMB)$, où M est le nombre de termes dans la collection ;
- L'ensemble des documents de la collection $D = (DB1B, DB2B, \dots, DBNB)$, où N est le nombre de documents dans la collection.

Les domaines des nœuds sont binaires {vrai, faux} signifiant que le nœud est instancié ou non.

T est l'ensemble des nœuds termes; une variable T_i associée à un terme prend ses valeurs dans le domaine $\text{dom}(T_i) = \{t_i, \bar{t}_i\}$, \bar{t}_i désigne le fait que le terme T_i est non pertinent et t_i désigne le fait qu'il est pertinent. Un terme est considéré comme pertinent si tous les documents qui le contiennent sont jugés pertinents par l'utilisateur et non pertinent sinon.

D est l'ensemble des nœuds documents, une variable D_j prend ses valeurs dans le domaine $\text{dom}(D_j) = \{d_j, \bar{d}_j\}$, \bar{d}_j signifie « le document D_j n'est pas pertinent » et d_j signifie « le document D_j est pertinent ». Un document est pertinent s'il répond au besoin utilisateur.

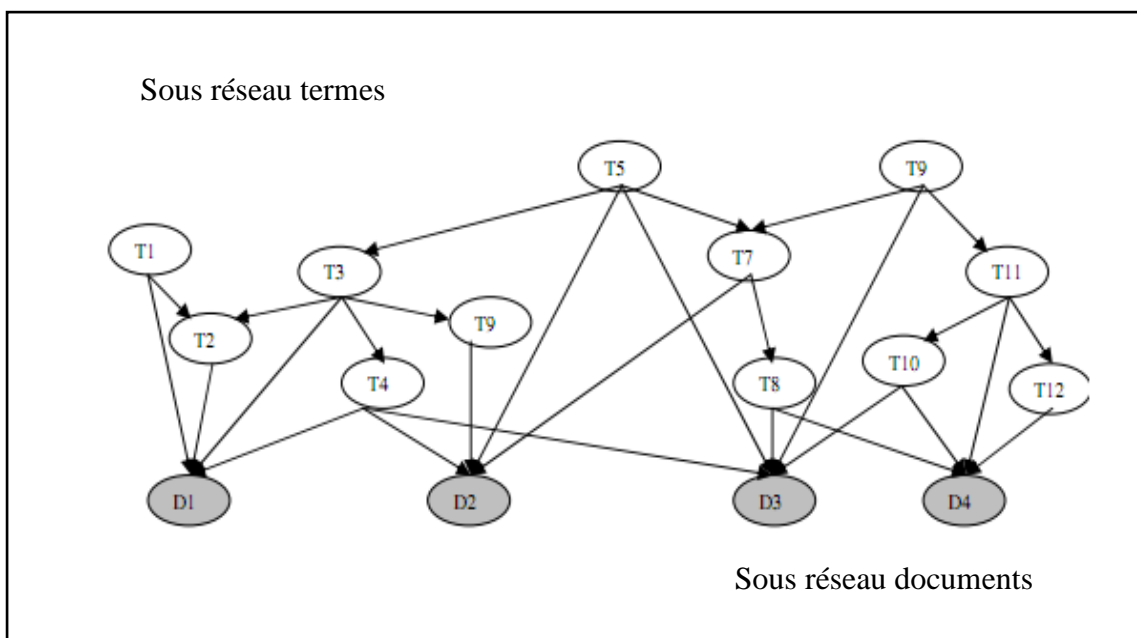


Figure I.9: Architecture générale du modèle Bayésien

I.3.5.2. Les modèles basés sur les réseaux d'inférence

Supposons que nous disposions d'un réseau bayésien défini par un graphe et la distribution de probabilité associée (G, P) . Supposons que le graphe soit constitué de n nœuds, notés $X=\{X_1, X_2, \dots, X_n\}$. Le problème général de l'inférence est de calculer $p(X_i|Y)$, où $Y \subset X$ et $X_i \notin Y$.

Un réseau d'inférence est donc un cas particulier des réseaux bayésiens. Il est représenté sous forme d'un graphe acyclique directionnel (orienté), où les nœuds représentent des variables propositionnelles ou des constantes et les arcs représentent les relations de dépendances entre les propositions (Turtle et Croft, 1990; Turtle, 1991; Desjardins, 2006). Si une proposition, représentée par un nœud p , implique une proposition, représentée par un nœud q , la relation entre p et q sera représentée par un arc orienté de p vers q .

Appliqué à la RI, le réseau d'inférence se divise en deux sous-réseaux, un pour les documents et l'autre pour la requête (Figure I.10.).

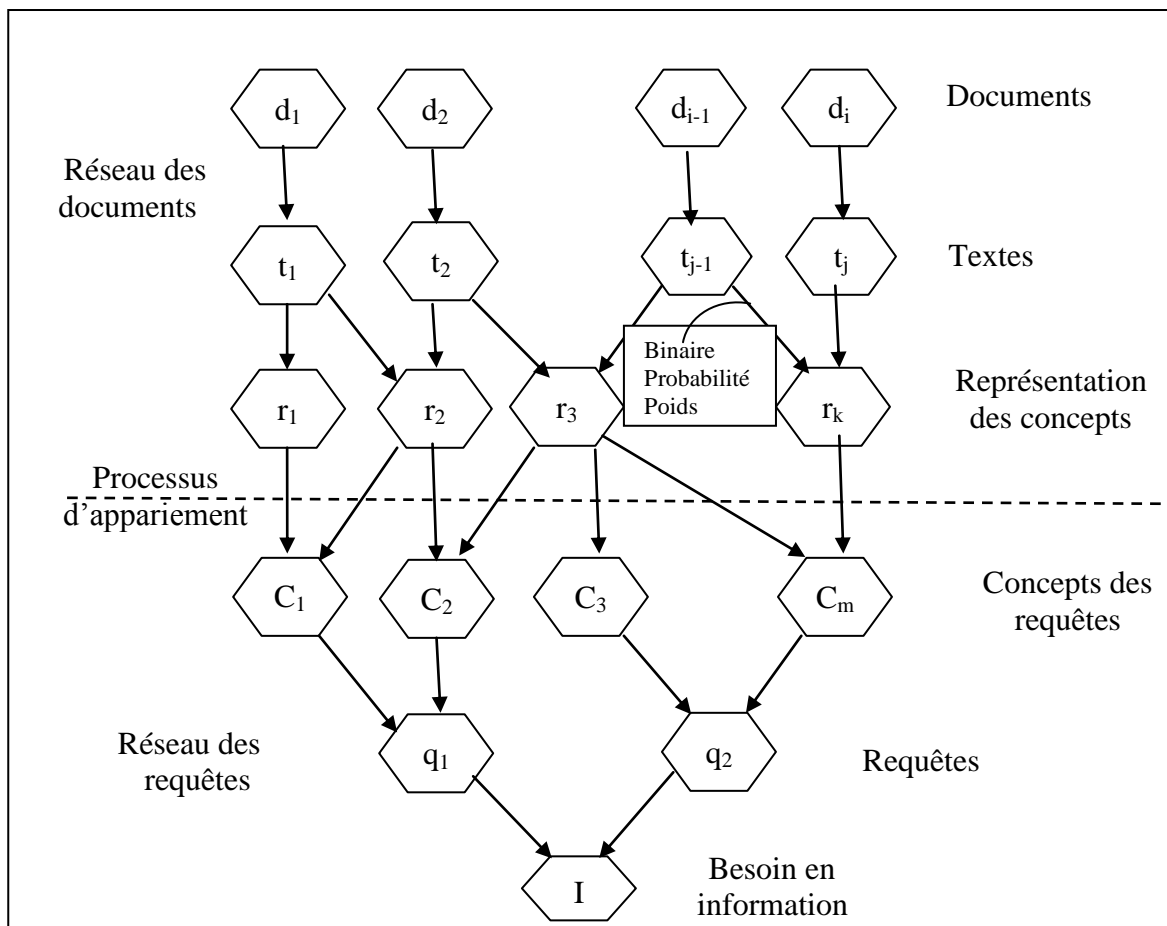


Figure I.10: Modèle générique d'un réseau d'inférence

Le sous-réseau document est composé de trois couches hiérarchiques :

- Les nœuds documents d_i qui correspondent à la probabilité d'observer un document de la collection. Ces probabilités sont habituellement initialisées à $(1/\text{nombre de documents de la collection})$;
- Les nœuds représentant les termes t_i qui correspondent à la probabilité d'observer un terme dans un document ;
- Les nœuds représentant les concepts r_i peuvent être générés par différentes techniques (Desjardins, 2006) : assignation manuelle des mots ou d'expression, extraction automatique, utilisation de thésaurus ou d'ontologies, etc... Ces nœuds correspondent à une probabilité conditionnelle $P(r_k/t_j)$ d'observer un concept étant donné l'ensemble de ses nœuds parents.

Le sous-réseau requête comporte une feuille unique pour représenter le besoin d'information et des racines qui correspondent aux concepts qui le représentent.

Les deux sous-réseau requête et document sont reliés par le biais des relations de dépendance entre les concepts de requêtes (c_i) et des concepts représentant la collection (r_k). Cette connexion modélise le processus d'appariement entre les concepts de la requête qui définit le besoin utilisateur et les concepts des documents de la collection.

Le SRI INQUERY (Turtle et al., 1990 ; 1991) est un exemple utilisant le modèle d'inférence (Callan et al., 1992).

Les avantages du modèle d'inférence :

1. Possibilité de représenter les documents sous plusieurs formes ;
2. Possibilité d'exprimer le besoin d'information par une combinaison de requêtes ;
3. Adaptation du processus d'appariement en offrant la possibilité d'intégrer différentes stratégies de recherche en parallèle (Moreau, 2006).

Inconvénient du modèle d'inférence :

- Le calcul des probabilités nécessite un temps exponentiel par rapport au nombre de termes de la requête.

I.3.5.2.1. Les modèles basés sur les réseaux de croyance

Les réseaux de croyance (Moreau, 2006), sont une généralisation des réseaux d'inférence. Ils s'en distinguent principalement par le sens des arcs. En effet, les valeurs se propagent de la requête vers les documents (Figure I.11).

Les réseaux de croyance (RC) (Ribeiro-Neto et al., 1996; Silva et al., 2000) ont été utilisés pour extraire des connaissances des requêtes du passé et les combiner avec le modèle vectoriel (Salton et al., 1994). La sélection d'un document s'appuie sur la similarité entre le document d_j et la requête Q en calculant la probabilité $P(d_j = 1 | Q = 1)$. En effet, $Q = 1$ et $d_j = 1$ signifient respectivement Q activé (choisi) et d_j activé (choisi).

Formellement, les réseaux de croyance utilisent la probabilité $P(d/q)$. Cette mesure peut être interprétée comme le rappel d'un document d_j par rapport à une requête q .

En pratique, seuls les termes des requêtes sont considérés pour un appariement. Dans ces circonstances, les réseaux de croyances définissent un espace commun et unique pour les concepts des documents et ceux des requêtes (Desjardins, 2006).

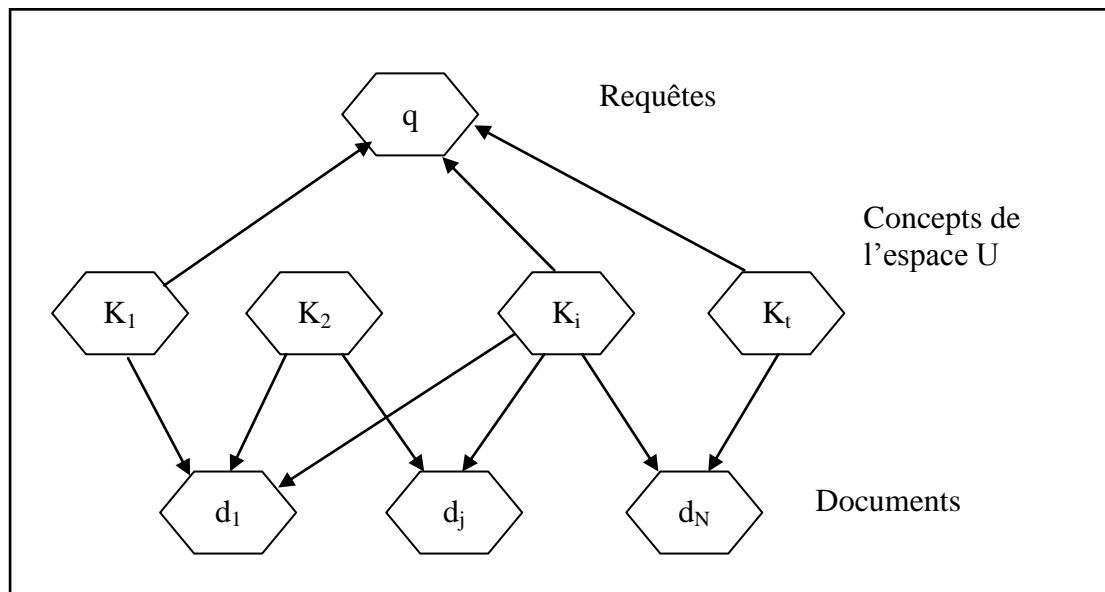


Figure I.11: Modèle générique d'un réseau de croyance

De plus, les réseaux de croyances utilisent les probabilités suivantes :

- $P(q) = \sum_u P(q/U) * P(U)$ comme le degré de couverture d'une requête q sur U ;

- $P(dj) = \sum_u P(dj/U) * P(U)$ comme le degré de couverture d'un document dj sur U.

Avec U : l'espace des concepts (termes).

En ce qui concerne le classement des documents résultat, les réseaux de croyances utilisent la mesure suivante (Desjardins, 2006) :

$$P\left(\frac{dj}{q}\right) = \eta \sum_u P\left(\frac{dj}{ki}\right) * P\left(\frac{q}{ki}\right) * P(ki)$$

Où les concepts $ki \in U$ sont, initialement, équiprobables : $p(ki) = (1/2)^t$, t est égale au nombre de termes dans U et η est une constante de normalisation.

I.3.6. Conclusion

Nous avons essayé tout au long de cette section de bien cerner les modèles utilisés dans le domaine de la recherche d'information et leurs intérêts dans le processus de la recherche. D'une part, ces modèles étaient centrés sur la représentation de la requête de l'utilisateur et du document, et sur la mise en correspondance directe entre ces deux représentations pour déterminer les documents pertinents selon la vision du système. Nous citons dans ce cadre : le modèle booléen, le modèle vectoriel et le modèle probabiliste. Afin d'enrichir ces deux représentations auxquelles sont associées deux types de connaissances : connaissances relatives aux documents et connaissances relatives à la requête, des extensions ont été proposées. Par ailleurs, ces extensions ont permis de fructifier le niveau d'analyse des documents, notamment en introduisant l'indexation sémantique latente, les domaines sémantiques, les réseaux d'inférence bayésiens. D'autre part, ces extensions ont concerné le niveau d'analyse de la requête, notamment le modèle booléen étendu, en introduisant des poids aux termes et des liens entre eux.

Ces modèles basées donc sur l'analyse de la présence de mots dans un corpus de texte, sont la base de la quasi-totalité des SRI actuels. Néanmoins d'autres groupes de chercheurs se sont lancés à la recherche d'une méthode permettant de "classer" les pages, non plus en fonction de leur contenu, mais en fonction des relations qui relient les différentes pages du site web. Les relations multiples entre les pages créées par l'apport de l'hypertexte constituent depuis l'origine l'un des éléments les plus caractéristiques du World Wide Web (WWW). Il

s'est avéré assez vite que l'étude des liens hypertextes permettait de tirer des informations utiles pour comprendre la structure du web.

Ces recherches ont abouti à deux algorithmes : pageRank et HITS qui feront le sujet de la section suivante.

I.4. Algorithmes de classement

I.4.1. Algorithme PageRank

Le PageRank créé par Lawrence E. Page et Sergueï Brin est l'algorithme d'analyse des liens concourant au système de classement des pages Web utilisé par le moteur de recherche Google, pour déterminer l'ordre dans les résultats de recherche qu'il fournit (Brin et al., 1998). L'algorithme du PageRank s'appuie donc sur un graphe hypertexte qui représente les liens hypertextes entre des ressources. La popularité d'une page sur le Web va donc être liée au nombre de fois que cette page est accédée à partir d'autres pages.

Le PageRank est un indice de popularité utilisé par Google. Cet indice est une note comprise entre 0 et 10 donnée à chaque page web. Plus la note est élevée et plus ce là signifie que la page est populaire. Ainsi, une page est considéré comme populaire lorsqu'il y a de nombreux liens internet qui pointent vers elle.

Le PageRank de la page A est défini comme suit :

$$\mathbf{PR(A)} = (1 - \mathbf{d}) + \mathbf{d} (\mathbf{PR(T1)}/\mathbf{C(T1)} + \dots + \mathbf{PR(Tn)}/\mathbf{C(Tn)})$$

- PR(A) : le PageRank de la page A ;
- PR(Tn) : le PageRank de la page Tn ;
- C(Tn) : le nombre de liens émis sur la page Tn ;
- d : tous les « votes » sont additionnés, mais pour en limiter l'importance, le total est multiplié par ce coefficient d'amortissement qui d prend ses valeurs dans l'intervalle [0-1] et est généralement placé à d=0.85 d'après des études statistiques menées par Lary Page dans ;

- $1 - d$: un petit peu de « magie mathématique » qui permet de garantir que la moyenne des PageRank de l'ensemble des pages du Web sera de 1.

Le paramètre d permet de faire converger l'algorithme de manière plus ou moins rapide. En effet, plus d est élevé, plus l'effet de l'ajout d'un lien entrant vers une page est accru et plus celui-ci se propagera dans toutes les pages d'un même site.

Le PageRank forme ainsi une distribution de probabilités des pages Web. Le calcul peut s'effectuer de manière itérative et converge vers une valeur asymptotique de manière assez rapide. En effet, le calcul effectué dans (S. Brin et al., 1998) sur un graphe de 26 millions de nœuds en considérant $d=0.85$ converge en seulement 52 itérations.

I.4.2. Algorithme HITS

L'algorithme HITS (Hyperlinked Induced Topic Search) (Jon M. Kleinberg, 1999) améliore la propagation de popularité en prenant en compte la pertinence des pages : "Une page référencée par un grand nombre de pages pertinentes est une bonne page", ou "une page qui référence un grand nombre de pages pertinentes est une bonne page". Contrairement à la technique du PageRank, qui assigne un score global à chaque page, l'algorithme HITS est une technique d'ordonnement dépendante de la requête. De plus, au lieu de donner un simple score, l'algorithme en donne deux : les scores d'autorité et de rayonnement.

L'algorithme HITS (Jon M. Kleinberg, 1999), conçu par John Kleinberg. Cet algorithme s'appuie sur un principe simple : tous les sites web n'ont pas la même importance, et ne jouent pas le même rôle. Certains sites sont des "sites de référence", leurs pages sont souvent citées dans d'autres sites. Ces sites de référence sont appelés "authorities" dans HITS. Alors que les "authorities" sont les véritables sites qui contiennent de l'information, d'autres sites appelés "Hubs" jouent un rôle tout aussi important, bien qu'ils ne contiennent, pas, à proprement parler, un contenu informatif... Il s'agit des sites qui contiennent des liens vers les "authorities", et qui permettent de "structurer" la Toile en indiquant où sont les pages intéressantes sur un sujet donné (Jon M. Kleinberg, 1999).

Voici comment fonctionne l'algorithme HITS. Soit la requête d'un utilisateur notée σ (Jon M. Kleinberg, 1999).

1. On fait d'abord une recherche classique (en utilisant par exemple un modèle vectoriel avec tf.idf). On note les pages trouvées les plus pertinentes R_σ ⁷ ;
2. À partir de l'ensemble des pages trouvées R_σ , on construit un plus grand ensemble S_σ qui contient :
 - les pages qui contiennent des liens vers R_σ ⁸ ;
 - les pages qu'on trouve à partir d'un lien sur une page se trouvant dans R_σ .
3. Une fois que S_σ et R_σ sont trouvés, on peut calculer la mesure « authority » ($a(p)$) ainsi que la mesure « hub » ($h(p)$) pour chaque page $p \in S_\sigma$. La mesure « authority » quantifie la qualité de la page en tant que page qui reçoit des liens, alors que la mesure « hub » quantifie le statut de la page en tant que page de liens.

On fixe d'abord $a(p) = 1$ et $h(p) = 1$ pour tout $p \in S_\sigma$. On note $q \rightarrow p$ la condition « q contient un lien pointant vers p ». On effectue alors les trois opérations suivantes en séquence, les répétant autant de fois que nécessaire, jusqu'à ce que les valeurs $a(p)$ et $h(p)$ convergent vers des valeurs stables.

1. $a(p) = \sum_{q \in S_\sigma, q \rightarrow p} h(q)$ (une bonne page de contenu obtiendra beaucoup de liens de la part de bonnes pages de liens) ;
2. $h(p) = \sum_{q \in S_\sigma, p \rightarrow q} a(q)$ (une bonne page de liens pointe vers de bonnes pages de contenu) ;
3. $a(p) = \frac{a(p)}{\sqrt{\sum_{q \in S_\sigma} a(q)^2}}$; $h(p) = \frac{h(p)}{\sqrt{\sum_{q \in S_\sigma} h(q)^2}}$ (on normalise $a(p)$ et $h(p)$ de manière à ce que la somme de leurs valeurs au carré soit unitaire).

On peut ainsi trouver les pages qui sont des « autorités » quant à la requête de l'utilisateur. Il suffit d'offrir à l'utilisateur les pages ayant le meilleur score $a(p)$.

⁷ R pour « ensemble-racine » ou *root*, en anglais.

⁸ Ce nombre peut être très élevé et, en pratique, on ne choisira qu'un sous-ensemble de toutes ces pages

Notons que, contrairement à PageRank, le calcul de $a(p)$ est fait pour chaque requête de l'utilisateur ! C'est donc évidemment plus coûteux et complexe.

Pour mesurer la qualité d'un système de recherche d'information, nous avons besoin de critères pour les utiliser dans cette perspective. Ces critères seront l'objet de la section suivante.

I.5. Métriques d'évaluations des SRI

Les systèmes de recherche d'information (SRI) sont composés essentiellement de deux modules. Un module d'indexation qui représente les documents, et un module d'interrogation qui représente la requête. La fonction de correspondance permet de calculer le degré d'appariement entre les termes de la requête et les termes d'indexation des documents afin d'évaluer la pertinence des documents par rapport à la requête. Avec le succès grandissant du web (Google recense plus de 8 milliards de pages web) le classement des réponses devient critique.

Le but de la RI est donc, de trouver des documents pertinents à une requête, et donc utiles pour l'utilisateur. La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles que l'utilisateur espère, mieux est le système.

Au fur et à mesure de l'évolution de ce domaine de recherche, des méthodes standard de mesure de qualité ont été mises au point afin de pouvoir comparer aisément les divers algorithmes de RI. La mesure de précision et de rappel est très utilisée sur des corpus textuels lorsque l'on connaît l'ensemble des éléments du corpus analysé.

Lorsqu'une personne interroge une base de données (que ce soit un logiciel documentaire ou un moteur de recherche), elle attend un ensemble de réponses (sous forme de documents) égal ou supérieur à un. À partir de l'ensemble de réponses obtenues mis en regard de l'attente de l'utilisateur, on mesure les performances de l'algorithme de recherche mis en œuvre pour retrouver un document. Cependant, il est indispensable de disposer de mesures quantitatives pour évaluer les performances des SRI : ces mesures des performances sont *le rappel et la précision*.

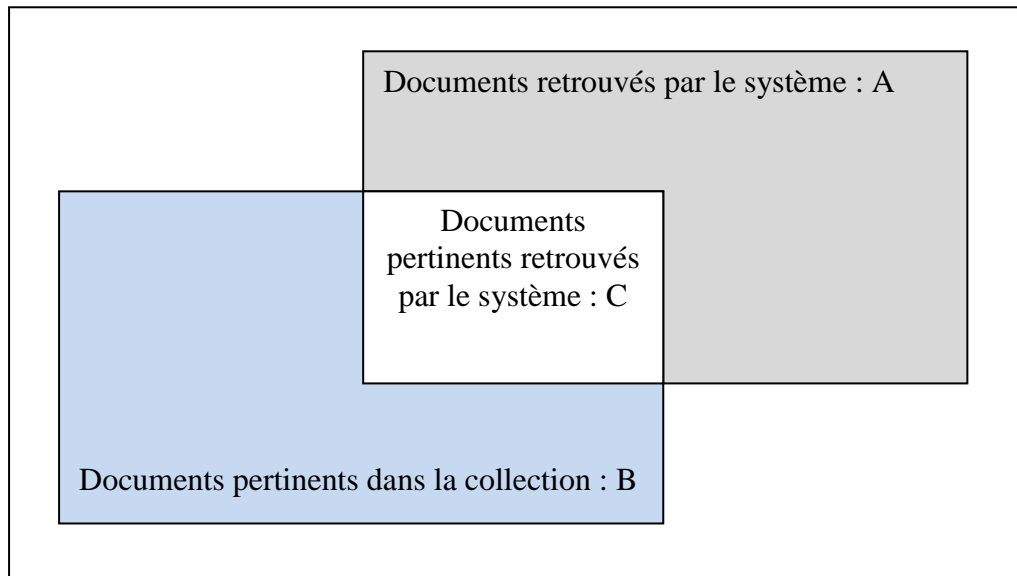


Figure I.12: Mesures de performance dans la RI

I.5.1. Métriques d'évaluation : le rappel

Le rappel est défini par le nombre de documents pertinents retrouvés au regard du nombre de documents pertinents que possède la base de données. Cela signifie que lorsque l'utilisateur interroge la base il souhaite voir apparaître tous les documents qui pourraient répondre à son besoin d'information. Si cette adéquation entre le questionnement de l'utilisateur et le nombre de documents présentés est importante alors le taux de rappel est élevé. À l'inverse si le système possède de nombreux documents intéressants mais que ceux-ci n'apparaissent pas on parle de silence. Le silence s'oppose au rappel.

La formule appliquée pour calculer le rappel est :

$$\text{Rappel } i = \frac{\text{nombre de documents pertinents renvoyés}}{\text{nombre documents pertinents de la base}}$$

D'après la figure ci-dessus :

$$\text{Rappel } i = \frac{C}{B + C}$$

I.5.2. Métriques d'évaluation : la précision

La précision est le nombre de documents pertinents retrouvés rapporté au nombre de documents total proposé par le moteur de recherche pour une requête donnée.

Le principe est le suivant : quand un utilisateur interroge une base de données, il souhaite que les documents proposés en réponse à son interrogation correspondent à son attente. Tous les documents retournés superflus ou non pertinents constituent du bruit. La précision s'oppose à ce bruit documentaire. Si elle est élevée, cela signifie que peu de documents inutiles sont proposés par le système et que ce dernier peut être considéré comme "précis". On calcule la précision avec la formule suivante :

$$\text{Précision } i = \frac{\text{nombre de documents pertinents retrouvés}}{\text{nombre de documents renvoyés par le système}}$$

D'après la figure ci-dessus :

$$\text{Précision } i = \frac{C}{A + C}$$

I.5.3. La précision et le rappel dans un cadre multi-classe

Dans le cadre multi-classes (où i est supérieur à 1), les moyennes globales de la précision et du rappel sur l'ensemble des classes i peuvent être évaluées par la macro-moyenne qui calcule d'abord la précision et le rappel sur chaque classe i suivie d'un calcul de la moyenne des précisions et des rappels sur les n classes :

$$\text{Précision} = \frac{\sum_{i=0}^n \text{Précision } i}{n}$$

$$\text{Rappel} = \frac{\sum_{i=0}^n \text{Rappel } i}{n}$$

$$\text{silence} = 1 - \text{rappel}$$

$$\text{bruit} = 1 - \text{précision}$$

Le silence représente donc, l'ensemble des documents pertinents que l'interrogation n'a pas pu retrouver, tandis que *le bruit* représente l'ensemble des documents non pertinents (selon l'utilisateur) que l'interrogation a restitué.

Par exemple, si on utilise un moteur de recherche qui se contente des mots-clés « livre » et « Hugo » pour chercher « les livres de Hugo », le système sera confronté à deux types de problèmes. D'abord, il ne saura pas faire la différence entre le nom de famille Hugo, le prénom Hugo ou nom de la rue Hugo : c'est le bruit. Ou encore, s'il rencontre le terme « Roman », le système ne saura pas qu'il est pertinent pour la requête, car il cherche le mot « Livre ». Par contre, si on explique au système que roman ou livre est un sous type du concept document, et la relation « est auteur de » qui peut lier une personne à un document, alors le système peut inférer qu'un roman est un livre, un livre est un document, donc un roman est un document, et que la réponse « Hugo a écrit le roman Notre Dame de Paris » peut-être valide (Gandon, 2006).

I.5.4. Interprétation des résultats de précision et de rappel

Un système de recherche documentaire parfait fournira des réponses dont la précision et le rappel sont égaux à 1 (l'algorithme trouve la totalité des documents pertinents - rappel - et ne fait aucune erreur - précision). Dans la réalité, les algorithmes de recherche sont plus ou moins précis, et plus ou moins pertinents. Il sera possible d'obtenir un système très précis (par exemple un score de précision de 0,99), mais peu performant (par exemple avec un rappel de 0.10, qui signifiera qu'il n'a trouvé que 10% des réponses possibles). Dans le même ordre d'idée, un algorithme dont le rappel est fort (par exemple 0.99 soit la quasi-totalité des documents pertinents), mais la précision faible (par exemple 0.10) fournira en guise de réponse de nombreux documents erronés en plus de ceux pertinents : il sera donc difficilement exploitable.

Par exemple, un système de recherche documentaire qui renvoie la totalité des documents de sa base aura un rappel de 1 (mais une mauvaise précision). Tandis qu'un système de recherche qui renvoie uniquement la requête de l'utilisateur aura une précision de 1 pour un rappel très faible. La valeur d'un classifieur ne se réduit donc pas à un bon score en précision ou en rappel.

I.5.5. La F-mesure

Les deux métriques ne sont pas indépendantes (précision et rappel), il y a une forte relation entre elles, quand l'une augmente, l'autre diminue. Il ne signifie rien de parler de la qualité d'un système en utilisant seulement une des métriques : En effet, pour avoir 100% de rappel, il suffit de renvoyer tous les documents de la base comme réponse pour la requête de l'utilisateur. Cependant, la précision dans ce cas sera très basse. De même, on peut augmenter la précision en donnant très peu de documents en réponse, mais le rappel souffrira. Il faut donc utiliser les deux métriques ensemble. Une des méthodes utilisées est de maximiser la moyenne harmonique de la précision et du rappel.

Plusieurs indicateurs de synthèse ont été créés à partir de deux mesures de Rappel et de la Précision, mais le plus célèbre est la F-mesure. Cette mesure correspond à une moyenne harmonique de la précision et du rappel. Cette moyenne diminue lorsque l'un de ses paramètres est petit et augmente lorsque les deux paramètres sont proches tout en étant élevés (Van Rijsbergen, 1979) :

$$F_1 = \frac{2 * (\text{Precision} * \text{Rappel})}{(\text{Precision} + \text{Rappel})}$$

Ceci est connu comme mesure F_1 , car la précision et le rappel ont la même importance. Donc $\beta = 1$. Il s'agit d'un cas particulier de la mesure générale F_β (pour des valeurs réelles positives de β) :

$$F_\beta = \frac{(1 + \beta^2)(\text{Precision} * \text{Rappel})}{(\beta^2 * \text{Precision} + \text{Rappel})}$$

I.5.6. Exemple

Pour calculer les critères d'évaluation, Nous supposons qu'on a un ensemble de documents ainsi que leur jugement (pertinent ou non pertinent) :

Document	Pertinent
1	*
2	*
3	
4	*
5	*
6	*
7	
8	*
9	*
10	

Tableau I.8: Tableau de pertinence

- Ici le nombre des documents pertinents est égale à : 7 ;
- Le nombre des documents renvoyés par le système est égale à : 10 ;
- Le nombre des documents pertinents dans la collection est égale à : 17.

Alors, la précision, le rappel ainsi que le F1-mesure vaut :

$$\text{Rappel} = 7/17$$

$$\text{précision} = 7/10$$

$$F1 = \frac{2 * \left(\frac{7}{17} * \frac{7}{10} \right)}{\left(\frac{7}{17} + \frac{7}{10} \right)} = \frac{98}{189}$$

$$F1 = 0,51$$

I.6. Synthèse

Durant ce chapitre, nous avons survolé le domaine de la recherche d'information en y effectuant un historique laconique et précis, puis nous avons décrit en détails les différents acteurs qui entrent en jeu pour accomplir le processus de la recherche. Ensuite, nous avons mentionné les modèles utilisés pour récupérer les documents qui répondent mieux aux besoins de l'utilisateur avec leurs avantages et inconvénients.

Enfin, nous avons souligné les critères utilisés dans le domaine de la recherche d'information pour mesurer la qualité des systèmes de recherche à savoir la précision, le rappel et le F-mesure.

De là, nous pouvons conclure que toutes les méthodes et outils qui se conjuguent pour la recherche d'information tentent de surmonter les problèmes majeurs de cette opération à savoir : le bruit et le silence. En effet, lorsque l'utilisateur interroge la base, il souhaite voir apparaître tous les documents qui pourraient répondre à son besoin d'information. Si le système possède de nombreux documents intéressants mais que ceux-ci n'apparaissent pas il s'agit du silence. Par contre, si tous les documents retournés superflus ou non pertinents alors dans ce cas c'est du bruit.

Nous avons essayé tout au long de ce chapitre de bien cerner les modèles utilisés dans le domaine de la recherche d'information ainsi que leurs intérêts dans le processus de la recherche. D'une part, ces modèles étaient centrés sur la représentation de la requête de l'utilisateur et du document, et d'autre part, sur la mise en correspondance directe entre ces deux représentations pour déterminer les documents pertinents selon la vision du système. Nous citons dans ce cadre : le modèle booléen, le modèle vectoriel et le modèle probabiliste. Afin d'enrichir ces deux représentations auxquelles sont associées deux types de connaissances : celles relatives aux documents et celles à la requête, des extensions ont été proposées. Par ailleurs, elles ont permis d'enrichir le niveau d'analyse des documents, notamment par l'introduction de l'indexation sémantique latente, les domaines sémantiques et les réseaux d'inférence bayésiens. D'autre part, ces extensions ont concerné le niveau d'analyse de la requête, notamment le modèle booléen étendu, par l'inclusion des poids aux termes et des liens entre eux.

Donc, les algorithmes de recherche font l'objet de très nombreuses investigations scientifiques. Les moteurs de recherche les plus simples se contentent de requêtes booléennes pour comparer les mots d'une requête avec ceux des documents. Mais cette méthode atteint vite ses limites sur des corpus volumineux. Les moteurs les plus évolués utilisent la formule TF-IDF pour mettre en perspective le poids des mots dans une requête avec ceux contenus dans les documents. Cette formule est utilisée pour construire des vecteurs de mots, comparés dans un espace vectoriel, par une mesure de Cosinus (la similarité Cosinus). Pour améliorer encore les performances d'un moteur, il existe de nombreuses techniques, la plus connue étant celle du PageRank de Google qui permet de pondérer une mesure de cosinus en utilisant un indice de notoriété de pages. Les recherches les plus récentes utilisent la méthode dite

d'analyse sémantique latente qui tente d'introduire l'idée de co-occurrences dans la recherche de résultats (le terme "voiture" est automatiquement associé à ses mots proches tels que "garage" ou un nom de marque dans le critère de recherche).

Subséquentement, d'après la description de ce domaine riche et varié en méthodes et outils visant à la satisfaction des gens dans la quête de l'information, que nous participons par la proposition d'autres méthodes qui permettent **d'augmenter considérablement le rendement et la capacité des systèmes d'information** tels que les moteurs de recherche.

En outre, grâce à l'apport de leurs aspects statistiques et sémantiques, nos méthodes et outils peuvent remédier aux problèmes tels que le bruit et le silence rencontrés par les moteurs de recherche utilisant des méthodes statiques.

Chapitre II. SIMILARITE ENTRE DEUX REQUETES DANS UN SRI

<u>II.1.</u>	<u>Introduction</u>	72
<u>II.2.</u>	<u>Travaux connexes</u>	74
	<u>II.2.1. Divergence de Kullback-Leibler</u>	74
	<u>II.2.2. Indice de jaccard</u>	75
<u>II.3.</u>	<u>Test χ^2</u>	76
	<u>II.3.1.1. Mode de calcul de χ^2</u>	76
	<u>II.3.1.2. Interprétation de χ^2</u>	79
<u>II.4.</u>	<u>Similarité entre requêtes basée sur le Chir statistique</u>	79
<u>II.5.</u>	<u>Similarité basée sur l'information mutuelle</u>	83
	<u>II.5.1. Information mutuelle</u>	83
	<u>II.5.2. Notre méthodologie d'utilisation de IM</u>	85
<u>II.6.</u>	<u>Similarité basée sur une mesure hybride</u>	87
<u>II.7.</u>	<u>Conclusion</u>	88

II.1. Introduction

Les systèmes de recherche d'information (SRI) sont conçus, à l'origine, pour répondre aux besoins d'utilisateurs vis-à-vis d'une requête. Néanmoins, cet objectif devient de plus en plus difficile à atteindre, en raison de, l'augmentation exponentielle des nombres de documents et d'internautes existant sur le web. Ceci, a poussé les chercheurs dans ce domaine à faire un arrêt pour explorer d'autres défis, à savoir :

- La connaissance sur le profil ou le modèle utilisateur ;
- La connaissance sur les documents (représentation ou indexation) ;
- La connaissance sur le domaine d'application pour reformuler la requête qui exprime le besoin utilisateur.

Nous ajoutons notre pierre à l'édifice pour arriver à la satisfaction des utilisateurs des SRI. Cette satisfaction ne peut s'exaucer que si les systèmes de recherche d'informations retournent des documents jugés pertinents pour la requête utilisateur.

D'un autre côté, nous avons mentionné dans le chapitre précédent que les moteurs de recherche utilisent des méthodes statiques et souffrent de problèmes tels que le bruit et le silence.

Ce chapitre préconise une nouvelle méthode hybride pour répondre à la problématique : **Comment répondre aux besoins des utilisateurs qui sont à la recherche d'information dans une grande masse de données ? Comment les aider, les assister à bon escient et leur offrir des outils pour obtenir des résultats pertinents et d'une façon conviviale ?**

En effet, dans ce chapitre, nous répondrons autrement à la requête de l'utilisateur du système de recherche d'information que lesdits modèles et méthodes proposés dans l'état de l'art. C'est-à-dire, au lieu de fouiller le corpus de documents pour calculer la correspondance entre la requête et les documents et par la suite classer ces documents par ordre de pertinence décroissant, nous proposons une nouvelle mesure de similarité entre deux requêtes. Cette similarité est mesurée entre, une nouvelle requête que le système vient de recevoir et qui exprime bien sûr un besoin utilisateur, et les requêtes candidates dont le système possède les documents pertinents.

Alors, grâce à notre méthode proposée, nous répondrons à la question majeure de la recherche d'information et par conséquent nous surmonterons les problèmes relatifs au domaine à savoir le bruit et le silence.

Ce calcul de similarité passe par trois phases (Figure II.1) : Dans un premier temps, nous présenterons une statistique plus précise basée sur la version étendue de la statistique χ^2 , appelée la méthode statistique de CHIR (Li et al., 2008) pour sélectionner les requêtes positivement dépendantes par rapport à la requête entrée par l'utilisateur. Dans un second temps, nous utiliserons l'information mutuelle (Brun et al., 2002) pour mesurer la similarité sémantique entre la requête de l'utilisateur et la requête candidate du système. Finalement, nous combinerons ces deux mesures, statistique et sémantique à l'aide de notre méthode dite d'alpha pour prédire la requête candidate la plus proche à la requête donnée en termes de similarité.

Le reste du chapitre est organisé de la façon suivante :

- Dans la section 2, nous présenterons un aperçu sur le statistique χ^2 ;
- Dans la section 3, nous développerons notre méthodologie pour mesurer la similarité statistique entre deux requêtes à l'aide de la méthode de Chir ;
- Dans la section 4, nous décrirons notre méthodologie pour mesurer la similarité sémantique en se basant sur l'information mutuelle ;
- Dans la section 5, nous associerons les deux mesures (statistique et sémantique), pour le calcul du score final de chaque requête candidate par rapport à la requête entrée par l'utilisateur ;
- Enfin, la section 6 correspond à la conclusion du chapitre.

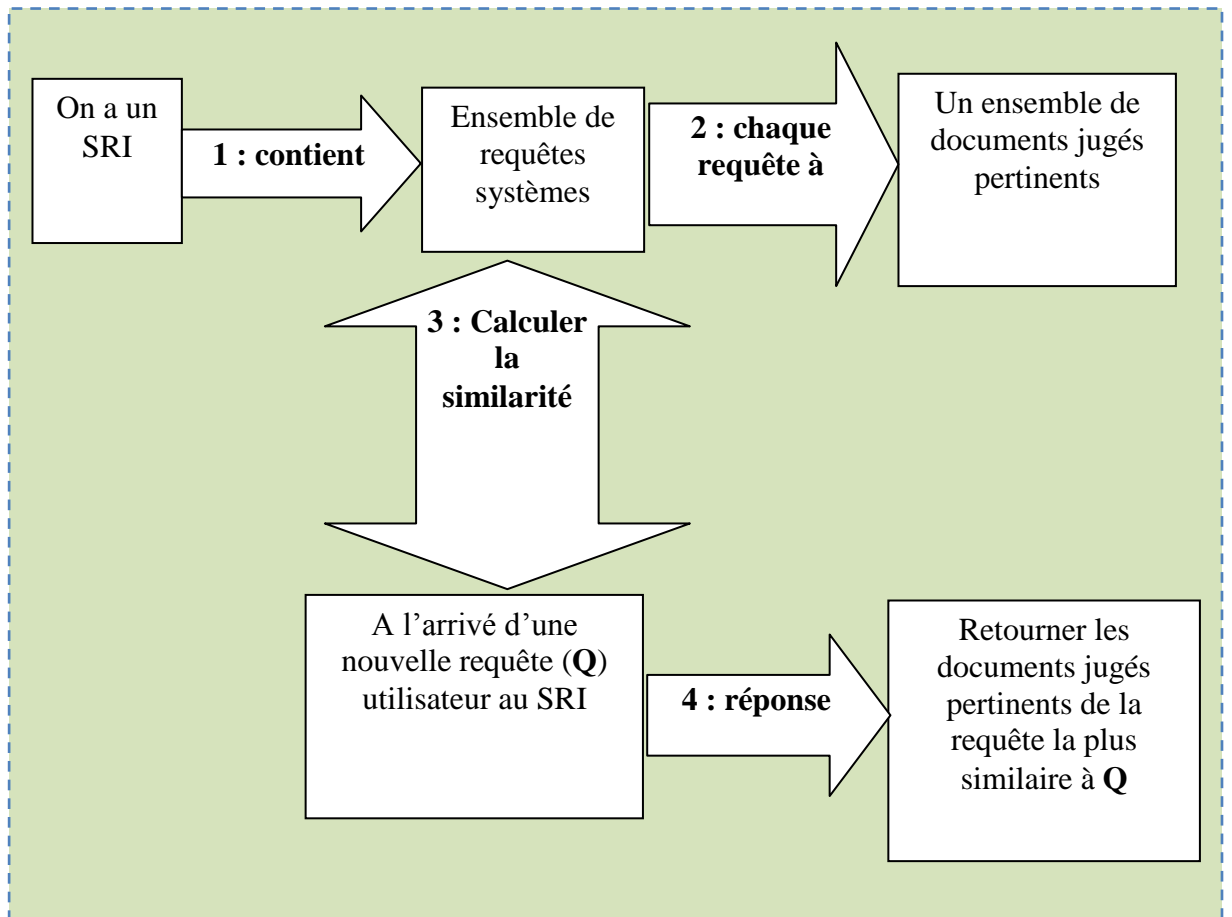


Figure II.1: Processus de fonctionnement de notre méthode

II.2. Travaux connexes

II.2.1. Divergence de Kullback-Leibler

Selon le cadre du travail proposé dans (Lafferty et al., 2001), la similarité entre une requête donnée Q et une requête candidate R_K du système peut-être exprimée par la mesure de divergence de Kullback-Leibler. La KL-divergence exprime la distance entre les distributions mises en relation. Ainsi, une fonction de similarité peut être définie comme suit :

$$\text{sim}(Q, R_K) = -\text{KL}(\theta_Q || \theta_{R_K}) = \sum_t p(t|\theta_Q) \log \frac{p(t|\theta_{R_K})}{p(t|\theta_Q)} \propto \sum_t p(t|\theta_Q) \log p(t|\theta_{R_K})$$

Où θ_Q représente la requête de l'utilisateur (Q), généralement estimé par fréquence relative des mots-clés dans la requête. Cette fonction de score peut aussi être vue comme une entropie croisée.

Afin d'éviter une probabilité nulle, ainsi que pour modéliser la spécificité des termes de la requête, il s'avère essentiel de lisser le modèle de la requête candidate par un modèle de retrait, généralement le modèle de la collection (Zhai et al., 2001). Une des stratégies fréquemment utilisées est de lisser le modèle de la requête par le lissage de Jelinek-Mercer, soit:

$$p(t|\theta_{R_K'}) = (1 - \gamma)p(t|\theta_{R_K}) + \gamma p(t|\theta_C)$$

Avec θ_C : représente le corpus des termes.

II.2.2. Indice de jaccard

Pour mesurer la similarité entre une requête Q qui définit un besoin utilisateur et une requête candidate R_K du système de recherche d'information, on fait appel à l'indice de Jaccard défini comme le rapport entre le nombre de mots communs à Q et R_K (l'intersection) et le nombre total de mots figurant dans Q et R_K (l'union) (Jaccard, 1901).

$$J(Q, R_K) = \frac{|Q \cap R_K|}{|Q \cup R_K|}$$

L'indice de Jaccard est normalisé (entre 0 et 1), plus il est proche de 1 (ou 100%) plus les deux requêtes comparées sont similaires.

Exemple de calcul de L'indice de Jaccard

Supposons qu'on a deux ensembles : $A = \{7, 3, 2, 4, 1\}$ et $B = \{4, 1, 9, 7, 5\}$. L'union des deux ensembles est : $A \cup B = \{1, 2, 3, 4, 5, 7, 9\}$ et l'intersection $A \cap B = \{1, 4, 7\}$.

Comme l'indice de Jaccard est égal au rapport entre le nombre des nombres communs a A et B (l'intersection) et le nombre total des nombres figurant dans A et B (l'union), on déduit que cet indice est égal :

$$S_{AB} = \frac{|A \cap B|}{|A \cup B|}$$

$$S_{AB} = \frac{3}{7} = 0,429$$

En définitive, pour calculer la similarité, ces méthodes se basent sur la fréquence du terme (TF) ou l'existence de celui de la requête usager dans les requêtes du système. Ce qui exige une correspondance parfaite entre chaque chaîne figurant dans Q et Rk. Or, notre mesure est différente et nouvelle dans le sens où elle emploie une technique hybride pour calculer la similarité : Nous utiliserons d'abord le CHIR statistique pour relever les requêtes qui sont positivement dépendantes par rapport à la requête utilisateur. Puis, nous introduirons des relations plus complexes entre les termes (relation sémantique) et, par conséquent, le score de similarité attribué à la requête est plus significatif que l'utilisation seule de Chir.

II.3. Test χ^2

Le test de χ^2 s'applique à un tableau croisant deux variables qualitatives. Il vise à tester l'indépendance des lignes et des colonnes de ce tableau (c-à-d si les deux variables étudiées ne sont pas indépendantes). Par exemple : *est-ce qu'il y a indépendance entre la couleur des yeux et la couleur des cheveux ?*

Par indépendance on peut dire :

- le fait d'appartenir à la modalité d'une des deux variables n'a aucune influence sur la modalité d'appartenance de l'autre variable ;

On définit alors les hypothèses :

- H0 : les deux caractères sont indépendants ;
- H1 : les deux caractères ne sont pas indépendants.

II.3.1.1. Mode de calcul de χ^2

Notre tableau croisé se présente sous la forme suivante :

		Caractère A					
		Modalité 1		Modalité i		Modalité p	Total
Caractère B	Modalité 1	n_{11}		n_{i1}		n_{p1}	$n_{.1}$
	Modalité j	n_{1j}		n_{ij}		n_{pj}	$n_{.j}$
	Modalité q	n_{1q}		n_{iq}		n_{iq}	$n_{.q}$
Total	$n_{1.}$		$n_{i.}$		$n_{p.}$	$n_{..} = N$	

Tableau II.1: Tableau des effectifs observés

- L'effectif n_{ij} correspond au nombre d'individus ayant la modalité i du caractère A et la modalité j du caractère B tel que $1 \leq i \leq p$ et $1 \leq j \leq q$;
- L'effectif $n_{i.}$ est la somme des effectifs de la colonne i ;
- l'effectif $n_{.j}$ est la somme des effectifs de la ligne j ;
- l'effectif $n_{..}$ est l'effectif total de la table.

Notre objectif est de tester l'existence ou non d'un lien entre les deux variables en calculant sous l'hypothèse d'indépendance la valeur de χ^2 . A cet effet, on détermine le tableau des effectifs théoriques sous l'hypothèse d'indépendance :

		Caractère A				Fréquence	
		Modalité 1		Modalité i			Modalité p
Caractère B	Modalité 1	$\frac{n_{1.} * n_{.1}}{N}$		$\frac{n_{i.} * n_{.1}}{N}$		$\frac{n_{p.} * n_{.1}}{N}$	$\frac{n_{.1}}{N}$
	Modalité j	$\frac{n_{1.} * n_{.j}}{N}$		$\frac{n_{i.} * n_{.j}}{N}$		$\frac{n_{p.} * n_{.j}}{N}$	$\frac{n_{.j}}{N}$
	Modalité q	$\frac{n_{1.} * n_{.q}}{N}$		$\frac{n_{i.} * n_{.q}}{N}$		$\frac{n_{p.} * n_{.q}}{N}$	$\frac{n_{.q}}{N}$
	Fréquence	$\frac{n_{1.}}{N}$		$\frac{n_{i.}}{N}$		$\frac{n_{p.}}{N}$	1

Tableau II.2: Tableau des effectifs attendus sous l'hypothèse H0

Sous H0, l'effectif attendu t_{ij} correspondant à la modalité i du caractère A (A_i) pour l'échantillon j, peut être calculé de la façon suivante :

$$t_{ij} = (n_{i.} * n_{.j}) / N$$

Donc, tous les effectifs attendus sont obtenus par le rapport du produit des distributions marginales sur l'effectif total de la table.

Nous avons notre tableau d'effectifs observés et celui d'effectifs théoriques. Nous pouvons à partir de là, calculer les écarts entre les deux, mais pour raisonner à l'échelle du tableau entier, nous devons rendre les écarts comparables avec la prise en compte d'une part de leur signe (en les élevant au carré) et d'autre part du fait qu'ils ne se rapportent pas aux mêmes effectifs de départ (en les divisant par les effectifs théoriques). Nous procéderons donc au calcul d'un nouveau tableau dont les cases contiennent la valeur suivante :

$$\chi^2 \text{ partiel} = \frac{(\text{Effectif observé} - \text{Effectif théorique})^2}{\text{Effectif théorique}}$$

On calculera après, le χ^2 statistique du tableau tout entier en additionnant les χ^2 partiels.

$$\chi^2 = \sum_{i=1}^p \sum_{j=1}^q \chi^2 \text{ partiels} = \sum_{i=1}^p \sum_{j=1}^q \frac{(n_{ij} - t_{ij})^2}{t_{ij}}$$

Avec n_{ij} l'effectif observé et t_{ij} l'effectif théorique attendu sous H_0 .

II.3.1.2. Interprétation de χ^2

L'hypothèse testée est la suivante :

- H_0 : Indépendance entre le caractère A et le caractère B ;
- H_1 : Non indépendance entre le caractère A et le caractère B.

Nous avons jusqu'à présent d'un côté la valeur du χ^2 pour notre tableau, et de l'autre, son nombre de degrés de liberté qui signifie que la valeur calculée du χ^2 doit être rapportée au nombre de colonnes et de lignes du tableau en question.

De même, nous lisons sur la table de χ^2 , la valeur seuil (χ^2 seuil) si nous insistons sur le degré de liberté du tableau croisé des deux variables ainsi qu'un risque d'erreur α fixé.

- Si $\chi^2 > \chi^2$ seuil : l'hypothèse H_0 est rejetée au risque d'erreur α : il n'y a pas indépendance statistique entre les deux caractères étudiés dans la population ;
- Si $\chi^2 < \chi^2$ seuil : l'hypothèse H_0 est acceptée : les deux caractères étudiés dans la population sont statistiquement indépendants.

II.4. Similarité entre requêtes basée sur le Chir statistique

Nous avons déjà cité dans la section précédente que l'objectif de la statistique χ^2 est de tester l'hypothèse, si deux variables ne sont pas indépendantes statistiquement.

Dans cette section, nous relèverons nos principaux objectifs concernant la méthode visée :

- L'un consiste à calculer le χ^2 pour chaque requête candidate par rapport à la requête entrée par l'utilisateur du système de recherche d'information ;
- Le deuxième à identifier les requêtes candidates qui sont positivement dépendantes vis-à-vis de la requête usagée ;
- Le troisième à calculer le score statistique de dépendance de chaque requête candidate.

Le point culminant de notre contribution au niveau de la statistique dite χ^2 , réside dans le fait que deux requêtes sont statistiquement dépendantes si et seulement si, le cluster des documents jugés pertinents pour la requête candidate, est obligatoirement pertinent pour la requête utilisateur. Dans ce cas-là, au lieu de mesurer la similarité entre la requête de l'utilisateur et celle candidate, nous chercherons à calculer cette similarité statistique χ^2 par rapport au cluster des documents jugés pertinents pour la requête candidate (figure II.2).

On définit alors :

	C	$\neg C$
q_i	$\Theta(q_i, c)$	$\Theta(q_i, \neg c)$
$\neg q_i$	$\Theta(\neg q_i, c)$	$\Theta(\neg q_i, \neg c)$

Tableau II.3: Tableau des effectifs observés

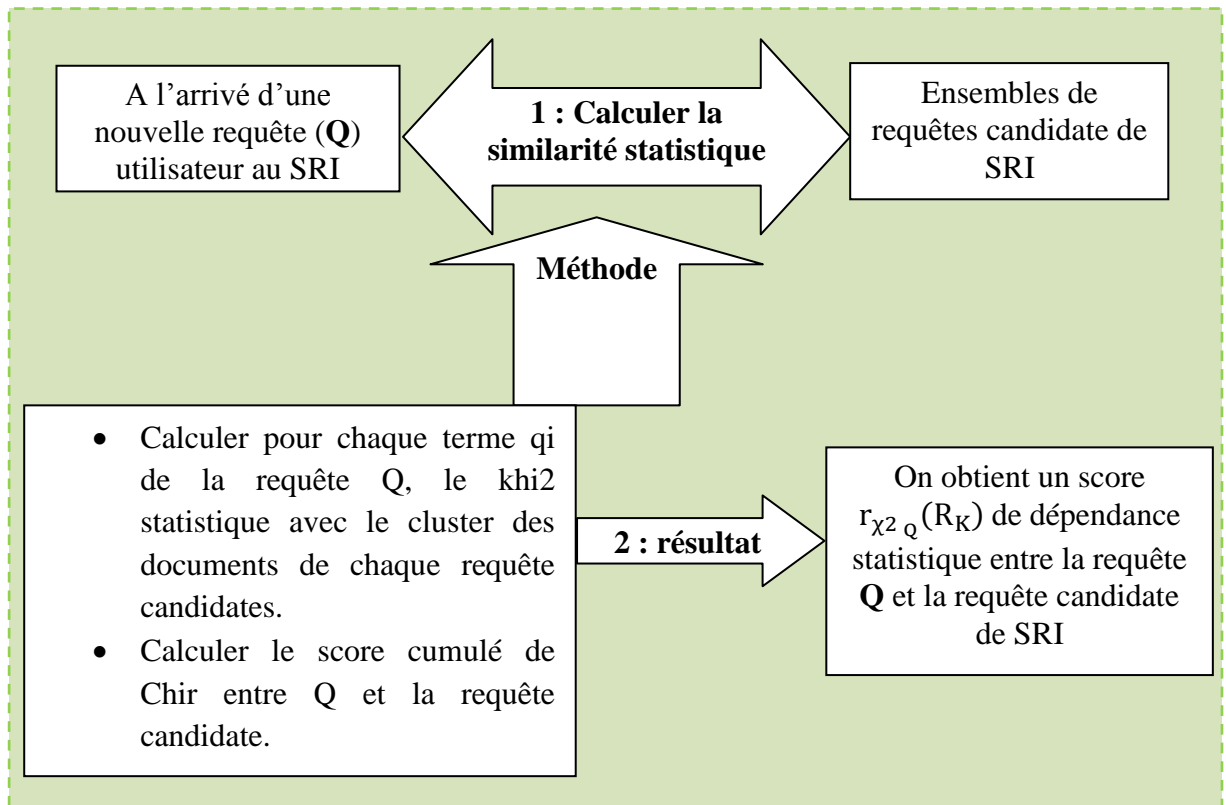


Figure II.2: Utilisation de Chir pour le calcul de la similarité statistique

Avec :

- $\Theta(q_i, c)$: Nombre de documents dans le cluster C qui contiennent le terme q_i de la requête de l'utilisateur ;
- $\Theta(\neg q_i, c)$: Nombre de documents dans le cluster C qui ne contiennent pas le terme q_i de la requête ;
- $\Theta(q_i, \neg c)$: Nombre de documents des autres clusters autre que C qui contiennent le terme q_i de la requête ;
- $\Theta(\neg q_i, \neg c)$: Nombre de documents des autres clusters autre que C qui ne contiennent pas le terme q_i .

	c	γc
qi	$e(qi, c)$	$e(qi, \gamma c)$
γqi	$e(\gamma qi, c)$	$e(\gamma qi, \gamma c)$

Tableau II.4: Tableau des effectifs théoriques

En somme, nous définirons le χ^2 statistique entre chaque terme de la requête utilisateur Q et celle candidate R_K , avec c le cluster des documents jugés pertinents pour la requête candidate R_K , par :

$$\chi^2_{qi,c} = \sum_{i \in \{qi, \gamma qi\}} \sum_{j \in \{c, \gamma c\}} \frac{(\theta(i, j) - e(i, j))^2}{e(i, j)}$$

D'ailleurs, la méthode de Chir est une extension d'une variante de la statistique χ^2 . Afin de surmonter les limites de celle-ci, le Chir statistique introduit un poids au variable χ^2 que nous définirons dans le contexte de calcul de la similarité statistique par :

$$p(R_{qi,c}) = \frac{R_{qi,c}}{\sum_{i=1}^n R_{qi,c}} \quad \text{avec} \quad R_{qi,c} > 1$$

Ainsi, $R_{qi,c}$ est défini par :

$$R_{qi,c} = \frac{\theta(qi, c)}{e(qi, c)} \quad \text{avec} \quad R_{qi,c} > 1$$

Comme $R_{qi,c}$ est le rapport entre $\theta(qi, c)$ et $e(qi, c)$: s'il n'y a pas de dépendance entre le terme qi et la catégorie c (c.-à-d $\chi^2_{qi,c}$ n'est pas statistiquement significative), $R_{qi,c}$ devra être proche de 1. S'il y a une dépendance positive, la fréquence observée doit être supérieure à la fréquence attendue, donc $R_{qi,c}$ doit être plus grand que 1. S'il y a une dépendance négative, $R_{qi,c}$ devra être plus petit que 1. Sur la base de la statistique χ^2 et $R_{qi,c}$, Li et al. (Li et al., 2008) proposent une nouvelle définition nommée *term-goodness* qui mesure le score d'un terme w dans un corpus de m classes comme suit:

$$r_{\chi^2}(w) = \sum_{j=1}^m p(R_{w,c_j}) * \chi^2_{w,c_j} \quad \text{avec} \quad R_{w,c_j} > 1$$

Où, $p(R_{w,c_j})$ est le poids de χ^2_{w,c_j} dans le corpus, définit par :

$$p(R_{w,c_j}) = \frac{R_{w,c_j}}{\sum_{j=1}^m R_{w,c_j}} \quad \text{avec} \quad R_{w,c_j} > 1$$

Cette nouvelle mesure de term-goodness, $r_{\chi^2}(w)$, est la somme pondérée des statistiques χ^2_{w,c_j} , quand il y a dépendance positive entre le terme w et la catégorie c_j . Un plus grand $r_{\chi^2}(w)$ indique que le terme est plus pertinent. Lorsque le terme w est en dépendance négative vis-à-vis de la catégorie c_j , son χ^2_{w,c_j} ne contribue pas au calcul de $r_{\chi^2}(w)$.

Par analogie, la définition du *term-goodness* représentant dans notre contexte la similarité entre la requête candidate R_K par rapport à la requête de l'utilisateur Q , présenté sous forme d'un score, par :

$$r_{\chi^2_Q}(R_K) = \sum_{i=1}^n p(R_{q_i,c}) * \chi^2_{q_i,c} \quad \text{avec} \quad R_{q_i,c} > 1$$

n : représente le nombre de termes de la requête donnée Q .

II.5. Similarité basée sur l'information mutuelle

Pour mesurer la similarité entre deux mots plusieurs mesures existent : Information Mutuelle, Gain d'Information (Quinlan, 1986), La divergence de kullback leibler (Maedche, 2002), rank corrélations (Strehl, 2002) ou le ratio log-likelihood (Resnik, 1999). La plus couramment utilisée est l'information Mutuelle.

II.5.1. Information mutuelle

L'information Mutuelle (IM) est née du fruit de travail de Claude Elwood Shannon en 1949 (Shannon et Weaver, 1949), peu après avoir introduit l'entropie différentielle qui porte maintenant son nom (Shannon, 1948).

La première mesure de l'information a été donnée par Hartley en 1928 (Hartley, 1928), un électronicien américain qui contribua aux fondations de la théorie de l'information. Sa mesure de l'entropie est concrétisée par :

$$H(x) = \log s$$

Où s le nombre de valeurs possibles pour chaque symbole du message x transmis. Cette mesure d'entropie est basée sur l'hypothèse que tous les événements sont équiprobables. Cette hypothèse est régulièrement mise en défaut dans les applications de télécommunications ou de traitement de signal. Pour pallier à cette difficulté, Shannon modifia la mesure de Hartley en pondérant chaque événement par son nombre d'occurrences (Shannon, 1948). Une fonction mathématique qui correspond à la quantité d'information contenue ou délivrée par une source d'information. Elle est originellement développée pour formaliser la nature statistique de l'information perdue dans les signaux des lignes téléphoniques. Cette mesure très informative a ensuite été reprise notamment en informatique, où elle est toujours utilisée de nos jours pour quantifier, par exemple, le nombre de bits sur lesquels on peut coder un fichier sans perte de données. L'entropie de Shannon, ou entropie différentielle associée à une loi continue pour une variable aléatoire x est donnée par :

$$H(x) = - \int p_x(u) \log(p_x(u)) du$$

Avec $p_x(u)$: la densité de probabilité de la variable aléatoire x .

On appelle information mutuelle entre deux variables aléatoires X et Y , la diminution de l'incertitude associée à une variable aléatoire due à notre connaissance (observation) de l'autre variable aléatoire :

$$I(X; Y) = \int p_{x,y}(u,v) \log \frac{p_{x,y}(u,v)}{p_x(u)p_y(v)} du dv$$

Dans le cas où la variable aléatoire est discrète, nous décrivons l'entropie conjointe entre deux variables aléatoires X et Y par :

$$H(X, Y) = - \sum_i \sum_j p(x_i, y_j) \log p(x_i, y_j)$$

On définit aussi l'information mutuelle entre deux variables aléatoires X et Y :

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

Soit, autrement dit :

$$I(X; Y) = \sum_i \sum_j p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

II.5.2. Notre méthodologie d'utilisation de IM

Dagan et al. (Dagan et al., 1999) ont introduit une mesure de similarité entre 2 mots x et y, évaluée. Il s'appuie sur leurs comportements respectifs en contexte (droit ; et gauche). Plus précisément, deux mots sont considérés comme similaires si leurs informations mutuelles avec l'ensemble des autres mots du vocabulaire sont proches. Cette similarité est évaluée de la manière suivante :

$$\text{similarite}(x, y) = \frac{1}{2V} \sum_{i=1}^{|V|} \frac{\min(I(z_i, x), I(z_i, y))}{\max(I(z_i, x), I(z_i, y))} + \frac{\min(I(x, z_i), I(y, z_i))}{\max(I(x, z_i), I(y, z_i))}$$

Où V est le vocabulaire et $I(z_i, x)$ est l'information mutuelle entre les mots z_i et x.

Cette mesure a été initialement développée dans le but d'estimer la probabilité de co-occurrences de mots, non observées dans l'apprentissage. Dans (Brun et al., 2002) l'équation ci-dessus a été adoptée en vue d'identifier le concept d'un document. Or Dans (frikh et al., 2011) cette équation a été utilisée pour identifier les termes qui sont sémantiquement pertinents pour construction d'ontologies de domaine.

Nous avons adopté cette mesure pour développer une méthode permettant le calcul de la similarité sémantique d'une requête donnée Q par rapport à une requête candidate R_k . Cette méthode est fondée sur l'information mutuelle I, calculée sur une fenêtre glissante de d mots. La nature de la similarité est donc plus sémantique que statistique.

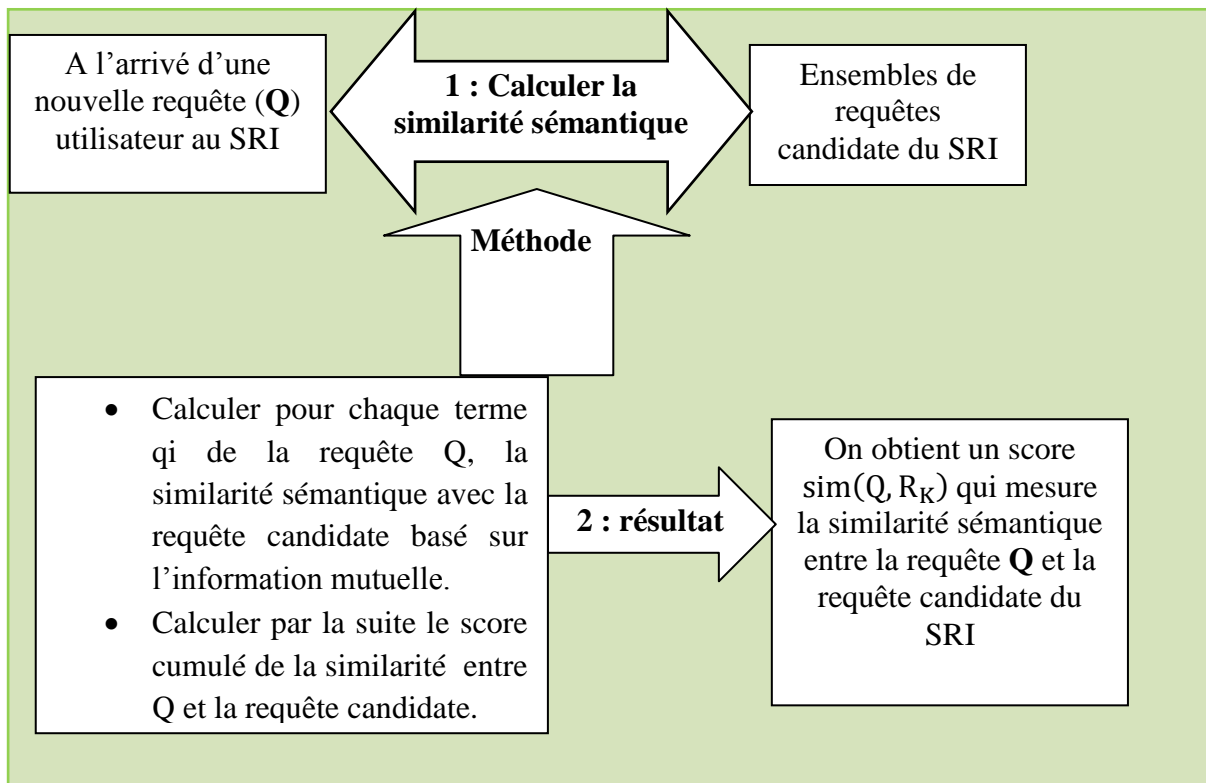


Figure II.3: Utilisation de l'information mutuelle pour le calcul de la similarité sémantique

$I(z_i, x)$, est évaluée de la manière suivante :

$$I(z_i, x) = Pd(z_i, x) \log \frac{Pd(z_i, x)}{d^2 P(z_i) P(x)}$$

Où d représente la distance ou la taille de la fenêtre glissante. $P(x)$ et $p(z_i)$ représentent respectivement la probabilité à priori du mot x et z_i . $Pd(z_i, x)$ est la probabilité de succession des mots z_i et x à une distance au plus d : cette probabilité peut être estimée par le ratio du nombre de fois que z_i est suivi par x au sein de la fenêtre et par le cardinal du vocabulaire.

$$Pd(z_i, x) = \frac{fd(z_i, x)}{|V|}$$

Où $fd(z_i, x)$ est le nombre de fois que z_i est suivi par x .

Ainsi, nous proposerons cette fonction pour calculer la similarité entre la requête donnée Q par rapport à la requête candidate R_k fondée sur l'information mutuelle par :

$$\text{sim}(Q, R_k) = \frac{\sum_{i=1}^n \text{sim}(q_i, R_k)}{\sum_{k'=1}^N \sum_{i=1}^n \text{sim}(q_i, R_{k'})} * \sum_{i=1}^n \delta_{iR_k}$$

$$\delta_{iR_k} = \begin{cases} 1 & \text{si } q_i \in R_k \\ 0 & \text{sinon} \end{cases}$$

Tel que la similarité entre chaque terme de la requête donnée Q et la requête candidate R_k est définie par la formule suivante :

$$\text{sim}(q_i, R_k) = p(q_i | R_k) * \frac{\sum_{k=1}^{|R_k|} \text{similarite}(q_i, y_k)}{\sum_{q=1}^{|V|} \sum_{k=1}^{|V|} \text{similarite}(q, y_k)}$$

Avec,

$$\text{similarite}(q_i, y_k) = \frac{1}{2V} \sum_{i=1}^{|V|} \frac{\min(I(z_i, q_i), I(z_i, y_k))}{\max(I(z_i, q_i), I(z_i, y_k))} + \frac{\min(I(q_i, z_i), I(y_k, z_i))}{\max(I(q_i, z_i), I(y_k, z_i))}$$

Sachant que notre vocabulaire V change d'une requête candidate R_k à une autre requête R_(k+1). V devient le vocabulaire du cluster C_k correspondant à la requête candidate R_k.

II.6. Similarité basée sur une mesure hybride

Notre objectif tracé depuis le départ est de proposer une nouvelle mesure pour la similarité entre deux requêtes utilisant le chir statistique ainsi que l'information mutuelle pour suggérer à l'utilisateur d'un système de recherche d'information ; le cluster des documents jugés pertinents de la requête candidate R_k la plus adéquate à la requête de l'utilisateur Q. Le choix de la requête R_k similaire à la requête donnée par l'utilisateur (Q) est défini selon le score le plus élevé de R_k. Ce score est déterminé par :

$$\text{score}(R_k) = \alpha r_{\chi^2_Q}(R_k) + (1 - \alpha) \text{Sim}(Q, R_k)$$

Où, $0 < \alpha < 1$

Pour maintenir la répartition du poids des variables et pour accorder la même importance, nous donnons à α la valeur 0,5(Djaanfar, 2011).

II.7. Conclusion

D'une part, ce chapitre décrit une nouvelle mesure hybride pour mesurer la similarité entre deux requêtes dans un système de recherche d'information. Cette mesure se base sur deux aspects : l'un est statistique dans le sens où il calcule le score statistique de dépendance positive entre la requête utilisateur Q et la requête candidate R_k en se basant sur la méthode de chir. L'autre aspect est sémantique qui cherche les requêtes candidates les plus similaires à la requête utilisateur en se servant de l'information mutuelle. Une telle mesure hybride indique sa performance en termes de satisfaction usager par rapport aux différentes méthodes et mesures mentionnées auparavant.

D'autre part, grâce à cette mesure proposée, nous pouvons dire que nous sommes arrivés à accroître considérablement le pourcentage du rendement et la capacité des systèmes de recherche d'informations notamment les moteurs de recherche web, ainsi que la diminution du bruit et du silence liés aux problèmes de la recherche d'information. Néanmoins, comme nous pouvons le remarquer que la méthode de similarité proposée n'inclut aucun paramètre qui reflète l'appréciation ou le feedback utilisateur vis-à-vis des résultats de la recherche.

Chapitre III. SYSTEME DE RECOMMANDATION POUR LA RECHERCHE D'INFORMATION

<u>III.1.</u>	<u>Introduction</u>	91
<u>III.2.</u>	<u>Aperçu sur e-Learning</u>	93
III.2.1.	Historique.....	93
III.2.2.	Les plates formes e-Learning.....	96
<u>III.3.</u>	<u>Les systèmes de recommandation</u>	97
<u>III.4.</u>	<u>Type d'évaluations</u>	99
III.4.1.	Évaluations explicites	99
III.4.2.	Évaluations implicites.....	99
<u>III.5.</u>	<u>Pourquoi Moodle :</u>	101
<u>III.6.</u>	<u>Analyse du problème et conception de la solution</u>	102
III.6.1.	Utilisateurs du système	102
III.6.2.	Séquencement des tâches assurées par le système.....	104
<u>III.7.</u>	<u>Processus de fonctionnement et calculs des critères d'extraction de connaissances</u>	106
III.7.1.	Mx-Search : Comment ça fonctionne ?.....	106
III.7.2.	Calculs des critères d'extraction de connaissances.....	106
III.7.2.1.	Comment calculer le Score d'Appréciation ?.....	107
III.7.2.2.	Comment calculer la note d'appréciation pour un document web ?.....	108
III.7.2.3.	Comment calculer le Ratio ?	108
III.7.2.4.	Comment utiliser la formule de chan ?.....	108
III.7.2.5.	Extraction de la connaissance à l'aide de la méthode alpha ?	109
III.7.2.6.	Attribution du lien visité à une catégorie des cours ?.....	110
<u>III.8.</u>	<u>Conclusion</u>	111

III.1. Introduction

L'objectif primordial de cette thèse est la satisfaction des utilisateurs des systèmes de recherche d'information. Cette satisfaction ne peut aboutir que si ces derniers retournent des documents qui répondent aux besoins des chercheurs de l'information.

Au cours du chapitre précédent (chapitre II), nous avons utilisé une mesure hybride basée sur le chi statistique et l'information mutuelle pour récupérer les documents pertinents pour une requête donnée sans aucune intervention de l'humain dans le processus de recherche d'information. Ainsi, d'après la figure I.1 qui exprime le processus en U de la recherche d'information et que nous avons exposé dans l'état de l'art, nous remarquons que le feedback utilisateur ou le retour de pertinence est utilisé pour reformuler la requête origine. En effet, ce fut un processus pour soutenir la recherche itérative, où les documents précédemment récupérés pourraient être marqués comme pertinents dans un système de recherche d'information. La requête d'un utilisateur a été ajustée automatiquement par l'utilisation des informations extraites des documents pertinents.

Dans ce chapitre, nous nous ne servons pas du feedback utilisateur pour reformuler la requête mais pour classer les documents web visualisés précédemment par les utilisateurs des systèmes de recherche d'information afin de les recommander pour d'autres usagers. Ce classement recourt, au-delà de l'appréciation directe de l'utilisateur donné sous forme de vote, à son appréciation indirecte estimée à partir de la formule de Chan que nous décrivons ultérieurement vis-à-vis des documents consultés.

La méthode de classement que nous proposerons est intégrée dans un système de recommandation pour avoir son impact sur le processus de la recherche. Ce système est divisé en deux parties :

1. La première partie : le système de recommandation que nous proposerons intègre un moteur de recherche pour garder la trace de navigation des utilisateurs ;
2. La deuxième partie : le système de recommandation suggéré emploie ladite méthode de classement en vue de présenter aux utilisateurs des documents exauçant leurs besoins d'information en tirant profit des ex-recherches effectuées.

La méthode suggérée s'appuie donc sur deux types d'évaluations :

- Le premier est explicite axé sur l'appréciation directe de l'utilisateur ;
- L'autre implicite, basé sur la trace et le comportement de l'utilisateur vis-à-vis des documents consultés.

Néanmoins, la validation de notre système est difficile pour la mise en œuvre à grande échelle. D'où la nécessité de le valider pour un domaine particulier. Le domaine choisi pour la validation de notre système de recommandation est l'e-Learning. En effet, chaque apprenant de la plate forme e-Learning effectue nécessairement des recherches sur internet pour avoir des ressources qui peuvent les aider à bien saisir et comprendre les cours dont lesquels il est inscrit. Or, cette tâche qui consiste à trouver des ressources pertinentes en un temps minimal est fastidieuse. D'où l'obligation de créer un système qui fait profiter des ex-recherches effectuées pour produire des recommandations dans le but d'aider ces apprenants durant leur apprentissage.

L'apport de ce travail réside principalement dans les points suivants :

1. Connaître les documents web préférés par les apprenants de la plate forme pour produire des recommandations selon différents critères :
 - Score d'appréciation : le nombre de points obtenus par vote pour un document particulier afin de calculer le degré de satisfaction des apprenants ;
 - Note d'appréciation : la note moyenne attribuée à un document basé sur le critère *Score d'appréciation* ;
 - Ratio : le temps moyen consacré par les apprenants pour visualiser une quantité d'informations ;
 - Type de document : le format du document (pdf, word, ppt, etc.) choisi par l'utilisateur pour extraire les meilleurs ressources ;
 - Formule de Chan : traduit le degré de satisfaction des utilisateurs pour un document web d'une manière implicite ;

- Période de recherche : indique les meilleurs documents existant pendant une période de recherche selon la combinaison de différents critères : *Score d'appréciation, Ratio ou formule de Chan*.
2. Connaissance sur les liens visités pour assister l'apprenant de la plate forme e-Learning pendant sa recherche. En effet, lors de l'affichage du résultat de la recherche, nous présenterons à l'apprenant des statistiques concernant les liens déjà sollicités par d'autres apprenants de la plate forme pour faciliter le parcours des documents renvoyés par le moteur de recherche web.

Les résultats obtenus indiquent la qualité des recommandations et son effet favorable sur l'apprentissage des apprenants tout en essayant de pallier les difficultés majeures des systèmes de recommandation.

La suite de ce chapitre est organisée comme suit :

- La section 2, nous exposerons un aperçu sur e-Learning ;
- La section 3, sera consacrée aux systèmes de recommandation ;
- la section 4, nous décrirons les types d'évaluations ;
- la section 5, nous justifierons le choix de moodle comme plate forme e-Learning pour la validation ;
- la section 6, nous présenterons la conception de la solution mise en place ;
- la section 7, nous décrirons le processus de fonctionnement de la solution proposée ainsi que les méthodes de calcul proposées pour les différents critères ;
- la section 8, fera l'objet d'une conclusion.

III.2. Aperçu sur e-Learning

III.2.1. Historique

Le développement des technologies éducatives et l'intégration de l'outil informatique ont permis d'introduire un potentiel éducatif illimité. En effet, les premiers systèmes

d'enseignement assisté par ordinateur sont apparus dès les années 70 ayant comme objectif, l'apprentissage en tant que transfert de connaissances. Ensuite une multitude de programmes éducatifs furent développés, mais vite délaissés à cause de leur contenu limité et leur utilisation rigide.

Les origines du e-Learning datent des années 80 lorsqu' un groupe de chercheurs a proposé la décentralisation spatio-temporelle des sessions de formations en utilisant l'ordinateur comme un moyen matériel permettant l'accès au contenu textuel des cours (Lackinger et al., 1984). Le nouveau type d'apprentissage a été vu comme une extension ou un supplément aux processus traditionnels d'apprentissage et peut être adopté dans les établissements d'enseignement spécialement équipés par des laboratoires d'informatiques. L'approche au départ ne visait pas à remplacer l'enseignement traditionnel, ni les supports papiers de cours, mais de bénéficier des avantages de ce mode d'enseignement. Citons entre autre : La communication asynchrone entre les apprenants et les enseignants, ainsi que entre apprenants, Les possibilités de suivi des formations sans que les formateurs soient dans la même place ni disponible en même temps que les apprenants.

Dix ans plus tard, dans les années 90, l'e-Learning s'est développé sous forme de formation assistée par ordinateur (Computer Based Training, CBT). L'apprentissage est présenté sur CD-ROM ou sur le Web à travers des séminaires (Aiken et al., (1998) et des sessions de formations. Néanmoins, Plusieurs questionnement commençaient à se poser par les professionnels du domaine éducatif (professeurs d'université, enseignants, institutions d'éducatifs...) (Mühlbacher, 1998): Est-ce que les nouveaux concepts du e-Learning sont en mesure de remplacer l'enseignement traditionnel, et surtout le rôle de l'enseignant ? Comment la qualité de l'apprentissage sera évaluée ? Comment l'efficacité et la rentabilité des enseignements seront assurées ? Toutefois et Malgré ces divers questionnements et doutes sur l'efficacité de ce dispositif de formation, il est devenu de plus en plus développé et une variété d'environnement e-Learning a été créé au cours des années suivantes.

A la fin des années 90, les environnements e-Learning ont développé de plus en plus l'aspect de la collaboration et ont mis l'apprenant au centre de tout processus de formation. Par exemple (Aiken et al., 1998) décrivent un scénario dans lequel un séminaire sur l'apprentissage interactif s'est tenue à Zurich (Suisse) et a été suivi par certains participants de Linz (Autriche). Durant cette expérimentation, plusieurs outils de communication ont été proposés citons par exemple : un groupe de discussion actif, des communications avec des experts via Internet, les apprenants pouvaient exploiter les ressources pédagogiques offertes et présenter des résultats sur les environnements e-Learning(Web). L'e-mail, les listes de

diffusion et un forum de discussion ont été utilisés comme canaux de communication. Le défi majeur de cette expérience était d'impliquer les apprenants dans l'expérience et les encourager à participer activement.

Au début des années 2000, la tendance vers le travail collaboratif s'est de plus en plus élargie en e-Learning. Parallèlement à une transformation des rôles des intervenants dans cet environnement : L'enseignant est devenu un tuteur et les apprenants sont transformés en des membres actifs dans le processus pédagogique au lieu que des simples consommateurs (Mühlbacher et al., 2002). Ainsi, l'enseignement est transformé d'un processus centré sur l'enseignant (Paris et al., 2001) vers un autre centré sur l'apprenant (Weimer, 2002). De plus, le développement des télécommunications et des réseaux impacte l'époque contemporaine par des plates-formes d'enseignement à distance et du campus virtuel. Ce dernier est défini par (Joye et al., 2003) comme un environnement unique intégrant différentes fonctions d'information, de communication (synchrone ou asynchrone), de collaboration, de gestion et d'apprentissage.

Selon les estimations des analystes de l'IDC⁹ (firme mondiale d'analystes et de consultants sur les TIC), le marché du e-Learning en Amérique du Nord était de 15 milliards de dollars en 2004 et la croissance annuelle du marché global de ce type d'apprentissage était de 69. Au Canada, il était estimé à 145 millions de dollars en 2000 et, selon IDC Canada, les dépenses de ce secteur devraient atteindre 1,5 milliard de dollars en 2006¹⁰.

Ces développements ont continué de révolutionner les concepts de l'apprentissage et d'apporter des évolutions dans la mise en œuvre des dispositifs de formation et des plates-formes e-Learning. En effet, durant ces dernières années, l'importance de l'e-Learning n'a cessé de croître. Les écoles, les universités et les autres établissements d'enseignement l'ont adopté comme un complément à l'enseignement traditionnel, des formations spécialisées sur l'utilisation du e-Learning et sur la création des contenus pédagogique en ligne commencent à s'enseigner et une diversité de formation et des cours en ligne se développent sur le web. Néanmoins, la nécessité d'une meilleure qualité du contenu d'apprentissage, d'accompagnement et de suivi des apprenants en session de formation est devenue une préoccupation primordiale des recherches dans ce domaine.

⁹ <http://www.idc.com>

¹⁰ Bulletin collégiale des technologies de l'information et des communications, Numéro 60, Janvier 2006, entretien avec *Gérald Roberge* -Conseiller pédagogique aux TIC, Collège *Gérald-Godin*

III.2.2. Les plates formes e-Learning

Une plate-forme pour le e-Learning est un logiciel qui assiste la conduite des formations ouvertes et à distance. Ce type de logiciel regroupe les outils nécessaires aux trois principaux utilisateurs : formateur, apprenant, administrateur, son objectif est d'assurer :

- la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et la télé - tutorat ;
- les fonctionnalités relatives aux référentiels de formation et à la gestion de compétences, à la gestion administrative, à la gestion des ressources pédagogiques et à la gestion de la qualité de la formation ;
- la possibilité de créer des contenus e-Learning riches en activités et ressources électroniques.

Une multitude de plates formes e-Learning sont disponibles sur le marché, nous citons à titre d'exemple :

ACOLAD (Apprentissage Collaboratif A Distance) :

ACOLAD Créé par l'université Louis Pasteur (Strasbourg I), elle repose sur les technologies employées sur Internet. Son interface graphique est fondée sur une métaphore spatiale qui met en scène les lieux habituels des formations. ACOLAD permet la mise à disposition de cours, mais aussi l'apprentissage en petits groupes et le développement de projets personnels par les étudiants.

WebCT :

C'est une plate-forme logicielle de télé-formation. Elle permet d'organiser un fonctionnement de classe sur le Web (à travers le réseau Internet). Par "classe" on entend "ensemble des étudiants inscrits à un cours donné".

Ganesha :

Ganesha est une plate-forme de téléformation (Learning Management System, LMS). Ce logiciel permet à un formateur ou un service de formation de mettre à la disposition d'un ou plusieurs groupes de stagiaires, un ou plusieurs modules de formation avec supports de cours, compléments, quiz et tests d'évaluation ainsi que des outils collaboratifs (webmail, forum,

chat, partage de documents) et d'assurer un tutorat en ligne. C'est un logiciel libre (sous licence GPL) et gratuit édité par la société Anéma Formation.

Moodle :

Moodle est une plate-forme d'apprentissage en ligne sous licence open source servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques. Nous allons voir cet environnement e-Learning avec plus de détail ultérieurement.

Claroline :

Claroline (Classroom Online) est une plate-forme de e-formation construite par l'Institut de pédagogie et des multimédias de l'Université catholique de Louvain. Le produit est en français, libre et accessible, et fonctionne sous PHP/MySQL/Apache. Il a été testé sous Linux-Mandrake 8.1 et Windows 98 et NT avec EasyPHP. Il peut accueillir jusqu'à 20 000 étudiants. Claroline permet de créer, d'administrer et d'alimenter des cours par Internet. Le logiciel offre un générateur de quiz, des forums, un calendrier, des documents partagés, un répertoire de liens, un système de suivi et de contrôle à l'entrée, etc. Une démo est présente sur le site.

Sakai

Campus Virtuel est un portail de formation web. Il dispose de toutes les fonctionnalités utiles pour construire et animer des parcours pédagogiques. Le Campus Virtuel dispose d'une interface Web, à partir de laquelle chaque acteur accède aux informations et outils dont il a besoin selon son rôle sur la plate-forme.

III.3. Les systèmes de recommandation

En raison de la démocratisation du web et l'augmentation exponentielle de la quantité de ressources disponibles et accessibles, les systèmes de recommandation ont vu leur popularité croître ces dernières années. Combinant des techniques de filtrage d'information, personnalisation, intelligence artificielle, réseaux sociaux et interaction personne-machine, les systèmes de recommandation fournissent à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnels. En effet, les systèmes de recommandation sont particulièrement sollicités dans les applications de commerce électronique. Par exemple, le site Amazon recommande toutes sortes de produits (films, musiques, livres, etc.) (Brun et al., 2010 ; Zaier, 2010).

Pour produire des recommandations plusieurs approches sont possibles : (i) l'approche par contenu (Pazzani et al., 2007) qui effectue des recommandations en comparant le contenu sémantique des ressources avec les goûts exprimés par l'utilisateur. (ii) l'approche à base de connaissances (Burke al., 1996) qui effectue des recommandations en exploitant les connaissances sur l'utilisateur et des heuristiques préétablies. et (iii) l'approche par filtrage collaboratif (Goldberg et al., 1992) qui effectue des recommandations par analyse à la fois des opinions de l'utilisateur sur les ressources qu'il a consulté ainsi que celles des autres utilisateurs sur les ressources qu'ils ont consultées.

Malgré leur popularité croissante, les systèmes de recommandation souffrent de plusieurs problèmes dont nous citons :

- *Masse critique* : cet aspect illustre la difficulté à gérer le fait qu'il existe peu d'articles effectivement évalués, ou peu d'utilisateurs qui procèdent à ces évaluations ;
- *Démarrage à froid* : souvent, on se retrouve confronté au problème qu'un utilisateur ne soit comparable avec aucun autre. Ce problème est dû au fait que peu ou pas d'utilisateurs ont évalué un article donné, ou qu'un utilisateur donné a évalué très peu ou pas d'articles ;
- *Collecte des préférences* : une des étapes les plus importantes et les plus difficiles des systèmes de recommandation sont la collecte des préférences des utilisateurs. En effet, l'obtention des évaluations de la part des utilisateurs sur une ressource donnée qui leur a plu, moins plu, ou pas du tout plus, est une tâche ardue. Ainsi, des techniques de collecte des préférences utilisateur, intrusives ou pas, ont vu le jour ;
- *Protection de la vie privée* : un autre problème qui touche les systèmes de recommandation est la protection des informations sensibles constituant le profil utilisateur (information personnelle, intérêts, goûts, habitudes, etc.). vu la nature de l'information, ces systèmes doivent assurer une telle protection. Aussi, des moyens, pour préserver l'anonymat des utilisateurs et chiffrer les données transmises, sont-elles nécessaires ;

- *Principe d'induction* : les systèmes de recommandation s'appuient sur le principe qu'un utilisateur qui a exhibé un comportement dans le passé tendra à exhiber un comportement semblable dans le futur. Cependant, ce principe n'est pas nécessairement valable dans le contexte réel.
- *Etc ...*

III.4. Type d'évaluations

III.4.1. Évaluations explicites

Les utilisateurs sont obligés de spécifier explicitement leurs préférences pour toutes les ressources. Généralement, les utilisateurs doivent indiquer leur degré d'appréciation sur une échelle de 1 à 5 ou 1 à 7. Des études ont mis en évidence l'impact du choix de l'échelle d'évaluation sur la qualité des recommandations (Ziegler, 2005).

III.4.2. Évaluations implicites

Les évaluations explicites imposent aux utilisateurs des efforts supplémentaires. En conséquence, les utilisateurs tendent souvent à éviter ce fardeau en quittant définitivement le système ou en fournissant des évaluations erronées (Ziegler, 2005).

À l'opposé, la déduction de ces évaluations par la seule observation du comportement des utilisateurs est beaucoup moins intrusive. Parmi les exemples typiques de ces observations pour des évaluations implicites, on peut citer l'historique d'achat pour Amazon, le temps de lecture pour Usenet News (Ziegler, 2005), et l'analyse des comportements de navigation sur Internet pour Quickstep (Gaul et Schmidt-Thieme, 2002 ; Middleton et al., 2004).

Un exemple réel de l'inférence des évaluations implicites est la formule, proposée par Chan (1999), pour prédire si une page web a été appréciée ou pas. Cette formule se base, en grande partie, sur les informations que nous pouvons récolter à partir des données du protocole de communication. En effet, elle est calculée en fonction de l'historique, le marque-page (bookmark), le contenu des pages et le journal des accès (access log). Plus précisément, l'historique d'un navigateur web maintient une trace du dernier moment qu'une

page a été visitée. Il est possible donc d'employer cette donnée pour calculer combien de fois un utilisateur passe sur une page et depuis quand il n'est pas retourné la visiter. De surcroît, nous serons également en mesure de supposer qu'une URL enregistrée dans le marque-page d'un utilisateur est considérée comme intéressante pour celui-ci. En outre, chaque page contient des liens vers d'autres pages. Nous présumons que si un utilisateur est intéressé par une page, il va probablement visiter les liens référencés par celle-ci. Ainsi, conjecturons-nous qu'un fort pourcentage de liens visités à partir d'une page dénote un intérêt particulier pour cette page.

Enfin, chaque entrée dans un journal des accès correspond à une requête HTTP, qui contient typiquement l'adresse IP du client, une marque du moment de connexion, des méthodes d'accès, une URL, un protocole, un statut, et une taille de fichier. Typiquement, une ligne de fichier de log se présente comme suit :

```
« 216.22.34.11 - [10/Ju//2008:04:04:54 +0200] "GET /uqam.Ca/index.htm/ HTTP/1.1"
200 2856 »
```

Où

- 216.22.34.11 correspond à l'adresse IP du client ;
- 10/Ju//2008:04:04:54 correspond au moment de connexion ;
- GET correspond à la méthode d'accès ;
- /uqam.Ca/index.htm/ est l'URL de la page qui a été visitée ;
- HTTP/1.1 correspond au protocole de communication ;
- 200 est la valeur de retour (ici, tout c'est bien passé) ;
- 2856 correspond à la quantité d'informations transférées (généralement la taille du fichier).

Grâce à ces informations, le temps passé sur chaque page peut être calculé. Cependant, le temps passé sur une page dépendant également de la longueur de cette page. l'intérêt d'un utilisateur pour une page sera calculé par le temps passé sur la page, normalisé par la taille de la page.

Une description *de la formule de chan* sera présentée dans le reste de ce chapitre.

III.5. Pourquoi Moodle :

Le choix de Moodle comme outil pour la conception de la plate forme e-Learning ainsi que le test du système proposé est justifié par les raisons suivantes :

- C'est un logiciel libre, ce qui signifie que les utilisateurs sont libres de le télécharger, l'utiliser, le modifier et même de le distribuer (Tortora et al., 2002 ; Zenha-Rela et al., 2006) ;
- Il s'agit d'un CMS & VLE qui permet aux enseignants de fournir et partager des documents, des missions classées, forums de discussions, etc. avec leurs élèves dans un format facile à apprendre et en haute qualité des cours en ligne (Dougiamas, 2008 ; Berry, 2005) ;
- Disponible en plusieurs langues (Cole et al, 2007 ; Williams et al., 2005), moodle donne aux utilisateurs la possibilité de poster des nouvelles, des affectations, et de recueillir les devoirs, etc. ;
- Moodle peut être utilisé sur presque tous les serveurs qui peuvent utiliser PHP. Les utilisateurs peuvent le télécharger et l'utiliser sur n'importe quel ordinateur et passe facilement d'une version à l'autre (Itmazi, 2005 ; Dougiamas, 2004) ;
- La clé de Moodle est qu'il est développé à la fois d'une manière pédagogique et technologique à l'esprit. Un des principaux avantages de Moodle par rapport aux autres systèmes est sa mise à la terre ferme dans la pédagogie socioconstructiviste et bons outils éducatifs (Al-Ajlan et al., 2008 ; Cheng-chao, 2005) ;
- Le logiciel Moodle est utilisé partout dans le monde par des professeurs indépendants, écoles, universités et les entreprises. La crédibilité de Moodle est très haute. Actuellement, il y a 3324 sites Web de 175 pays qui sont enregistrés officiellement dans le site de moodle, et il est disponible en 75 langues (Berry, 2005 ; Chavan et al, 2004) ;
- Moodle tourne sans modification sur n'importe quel système qui supporte PHP comme Unix, Linux et Windows. Il utilise MySQL, PostgreSQL et Oracle, et d'autres sont également soutenues (Al-Ajlan et al., 2008 ; Shearer, 2003) ;

- Il possède une excellente documentation et un soutien rigoureux pour la sécurité et l'administration (Al-Ajlan et al., 2008 ; Cheng-chao, 2005).

III.6. Analyse du problème et conception de la solution

III.6.1. Utilisateurs du système

Notre système est utilisé par trois types d'utilisateurs (figure III.1) :

- Apprenant : qui n'a le droit que de réaliser des recherches pour consulter des documents avec la possibilité de donner son vote ;
- Enseignant : qui a tous les droits d'un apprenant ainsi que la gestion des cours ;
- Un administrateur : prendre des décisions concernant les meilleurs documents web à ajouter dans la plate forme e-Learning moodle en se basant sur la recommandation du système proposé.

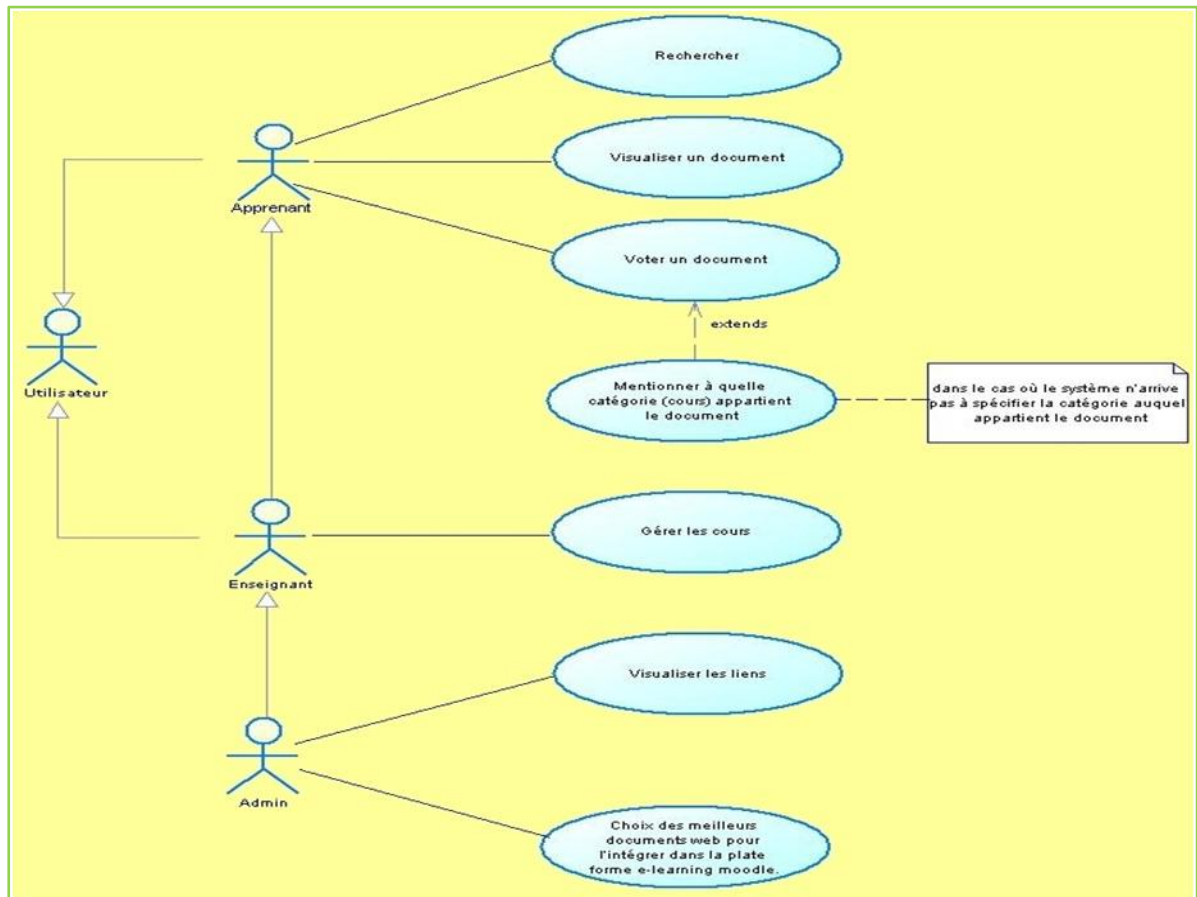


Figure III 1: Diagramme de cas d'utilisation

Nous utilisons Le diagramme de classe (figure III.2) pour montrer la structuration du modèle à adopter, en mettant en évidence les différentes associations et liens entre les classes.

- Chaque apprenant est assigné à un cours selon un rôle. Les cours et les liens recherchés par les apprenants sont classés par catégories ;
- Chaque apprenant donne son appréciation (vote) pour le document qu'il est entrain de visualiser, par la suite la durée de visite, le ratio, le nombre des liens visités et la date de la dernière visite sont calculé et archivi.

Commentaire :

Les classes dont le nom commence par mdl sont propres à moodle, par contre celles dont le nom commence par ext sont des classes externes que nous avons intégrées dans la base de données de moodle.

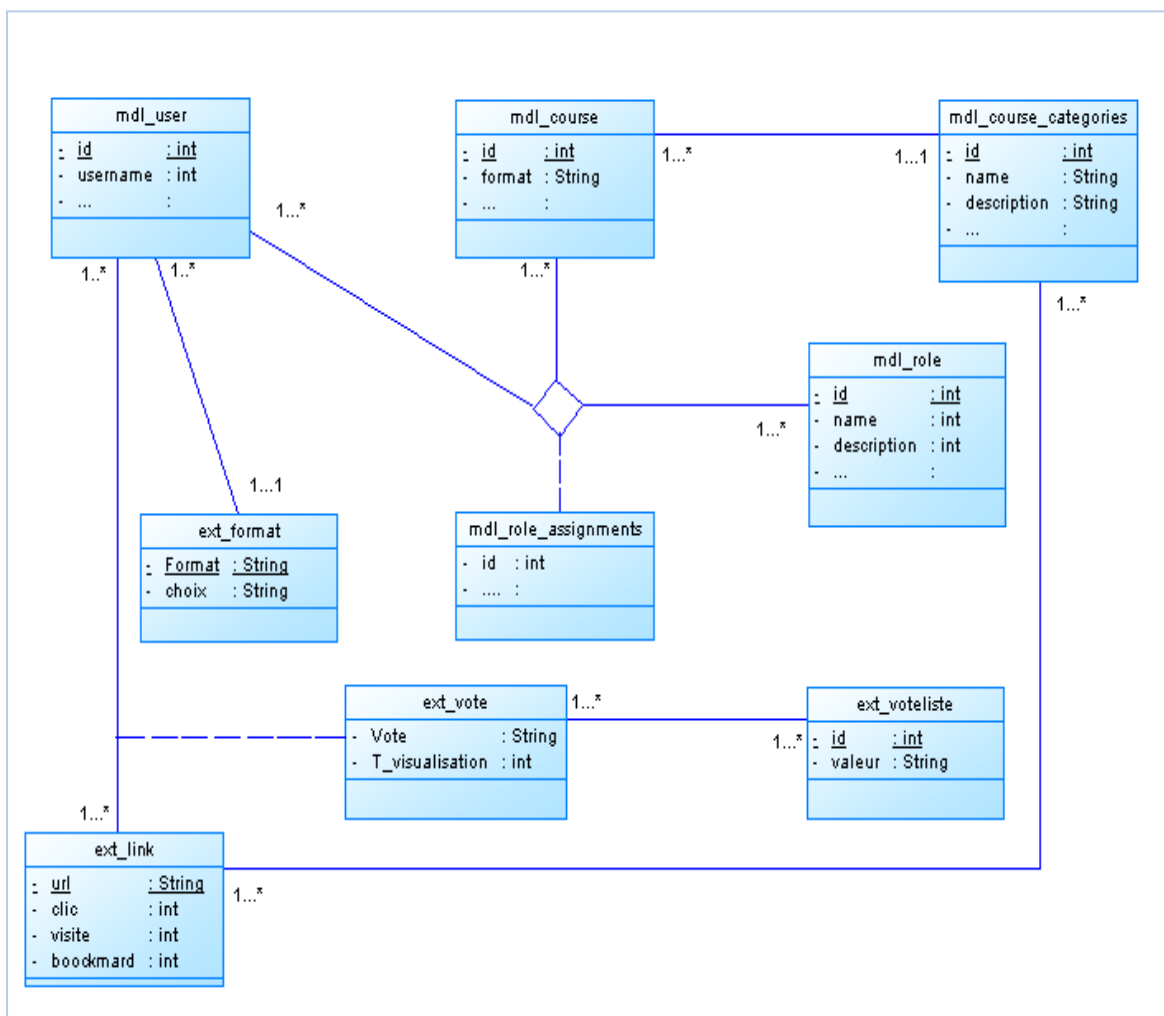


Figure III.2: Diagramme de classe

III.6.2. Séquencement des tâches assurées par le système

Pour mettre l'accent sur la chronologie des envois de messages par les différents acteurs, nous proposons des diagrammes de séquences (figure III.3 et III.4).

Acteur : Apprenant (figure III.3)

Après l'authentification de l'apprenant, ce dernier peut choisir les formats des documents désirés pour sa recherche. Lors du clic sur un document, un compteur de clic sera incrémenté en enregistrant la date de visite. Une fois l'apprenant quitte le lien, nous calculons à la fois, le temps consacré pour sa visualisation, l'appréciation (vote) pour ce document ainsi que l'affectation de ce dernier à une classe de catégorie des cours.

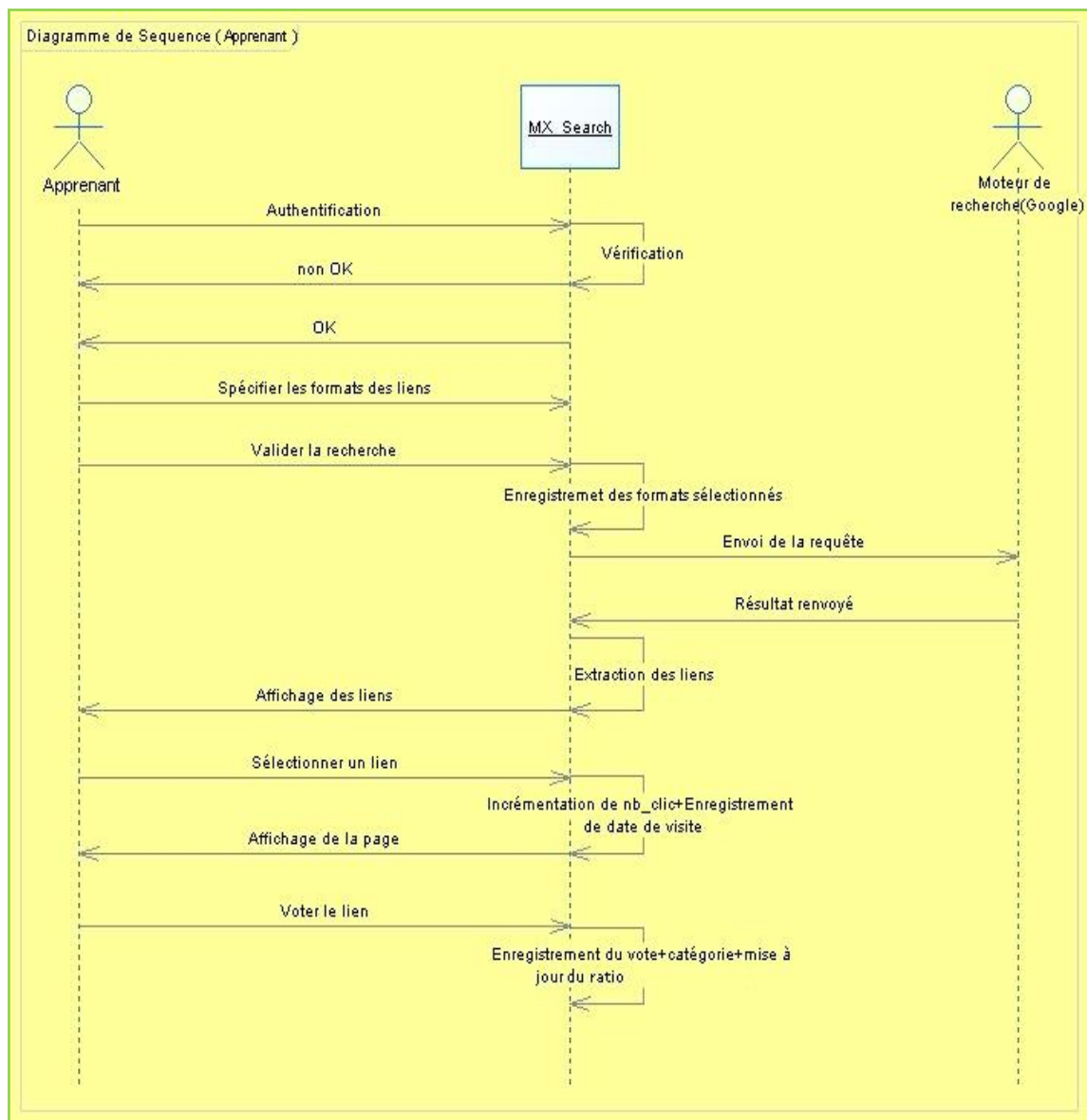


Figure III.3: Diagramme de séquence apprenant

Acteur : administrateur (figure III.4)

Après l'authentification de l'administrateur, il peut sélectionner la filière puis le module pour valider une recherche interne. Celle-ci permet d'extraire les meilleurs documents web selon le critère désiré (score d'appréciation, ratio, note d'appréciation. Formule de chan...). Son but est de prendre des décisions concernant les documents à ajouter dans la plate forme e-Learning afin que les apprenants puissent en profiter.

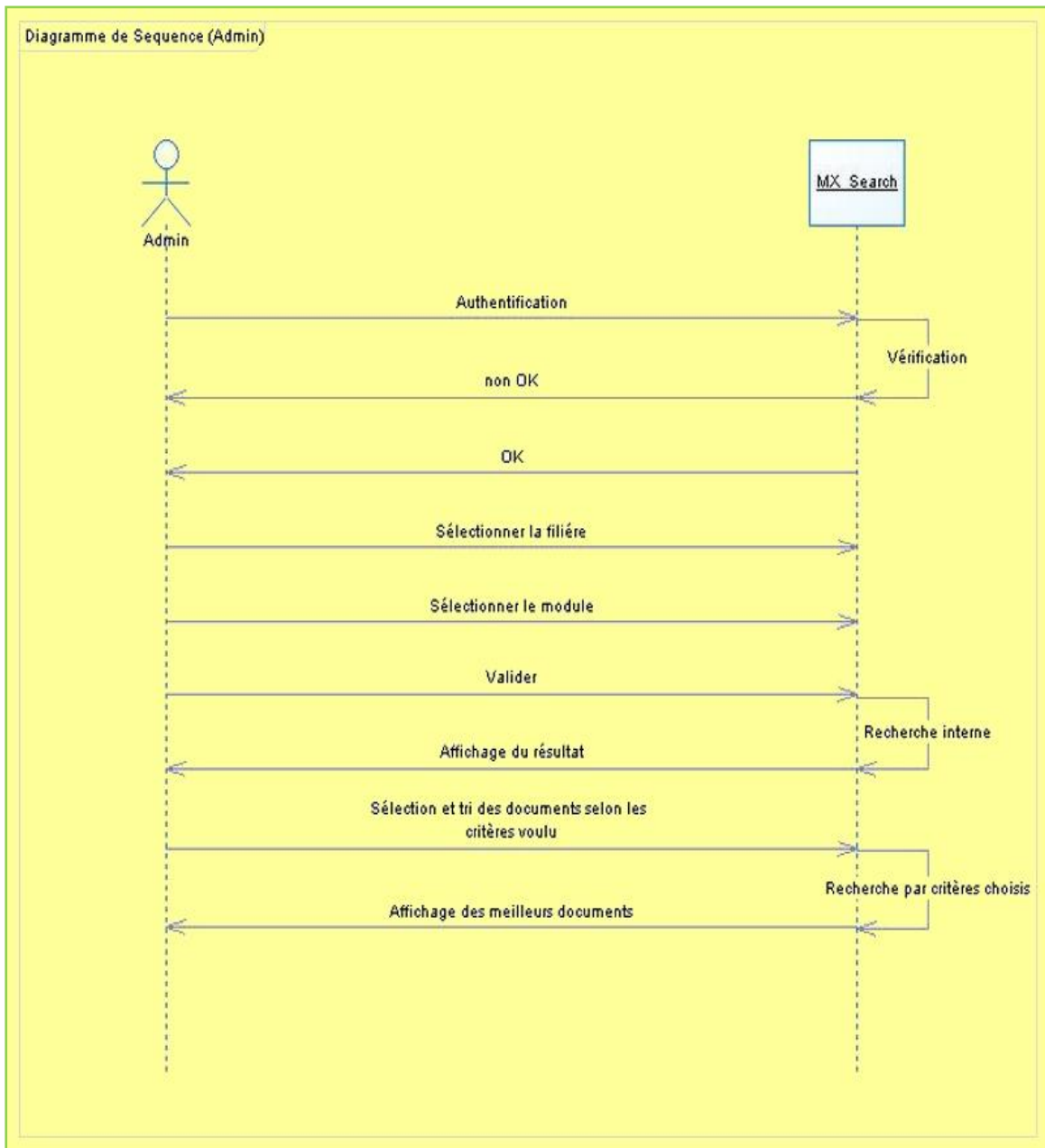


Figure III.4: Diagramme de séquence administrateur

III.7. Processus de fonctionnement et calculs des critères d'extraction de connaissances

Le système (plugin) qu'on a intégré dans la plate forme e-Learning moodle pour étudier le comportement des apprenants et produire des recommandations est nommé MX-Search.

III.7.1. Mx-Search : Comment ça fonctionne ?

MX-Search extrait et affiche le résultat de la recherche renvoyé par le moteur de recherche (google, yahoo ou Bing) tout en gardant la trace de l'apprenant pour effectuer des recommandations (figure III.5).

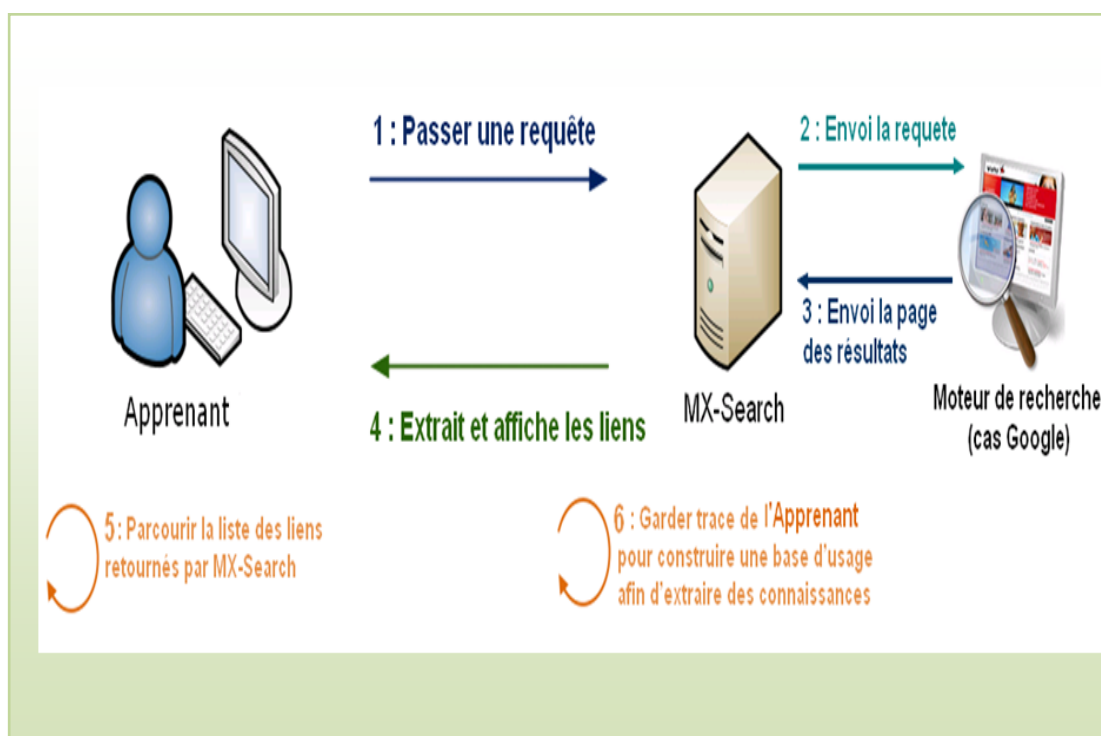


Figure III.5: Processus de fonctionnement de MX-Search

III.7.2. Calculs des critères d'extraction de connaissances

Dans cette section, nous présenterons les méthodes utilisées pour le calcul des critères pertinents de notre système pour extraire des connaissances sur les ressources web à ajouter dans la plate forme e-Learning.

III.7.2.1. Comment calculer le Score d'Appréciation ?

Pour tirer les préférences des utilisateurs, l'approche de filtrage collaboratif utilise soit la relation de préférence ou la fonction d'utilité (les votes par exemple). En effet, dans le second cas qui nous intéresse, nous proposons à l'utilisateur d'émettre son opinion selon une échelle de valeurs entières fixées et relativement réduite (généralement une valeur entre 1 et 5 ou 1 et 7). Néanmoins, cette tâche de vote par note est très dure pour l'apprenant. Ce qui nous a poussés à suggérer une autre échelle d'appréciation au lieu de celle des valeurs entières.

Dans cette perspective, l'apprenant sera appelé à répondre suivante :

Comment trouvez-vous le document ? Est-il : Nul, Pas mal, Moyen, Bien, Excellent ?

Alors la réponse sera plus proche du contexte réel et facilite l'appréciation de l'apprenant.

On affecte par la suite à chaque appréciation une note, par exemple (Nul=1, Pas mal=2, Moyen=3, Bien=4, Excellent=5) et pour chaque vote d'un apprenant nous appliquons l'algorithme ci-dessous.

Algorithme : Algorithme pour le calcul de Score d'appréciation

```
Algorithme Score_d_appréciation
variable vote: Chaîne de caractère;
Score_d_appréciation, val_ancienne: Entier ;
Début
val_ancienne ← val_ancienne(Score_d_appréciation)
Cas vote vaut
'Nul' : Score_d_appréciation ←val_ancienne+ ai *1 ;
'Pas mal' : Score_d_appréciation ←val_ancienne+ ai *2 ;
'Moyen' : Score_d_appréciation ←val_ancienne + ai *3 ;
'Bien' : Score_d_appréciation ←val_ancienne + ai *4 ;
'Excellent': Score_d_appréciation ←val_ancienne + ai *5 ;
Sinon: rien faire
Fincas
Fin
```

Avec ai : poids représente le niveau de l'apprenant.

III.7.2.2. Comment calculer la note d'appréciation pour un document web ?

$$\text{note d'appréciation} = \text{score d'appréciation} / \text{somme}(\alpha)$$

III.7.2.3. Comment calculer le Ratio ?

Nous ferons appel à la technique du filtrage de l'information basé sur le contenu pour extraire des informations supplémentaires (autre que le vote) sur les ressources sollicitées par les apprenants comme la durée de visite de la page, la taille du document, etc.....

Ceci permet au système de produire des recommandations (soit en se basant sur le filtrage appuyé sur le contenu ou en combinant avec le filtrage collaboratif) au cas où les apprenants n'effectueraient que peu ou pas d'évaluation sur les ressources, ce qui tolère en outre au système de surmonter les problèmes de *Masse critique et Démarrage à froid*.

Nous procéderons au calcul d'abord du temps consacré par chaque apprenant pour visualiser un document particulier.

$$T_{\text{visualisation_apprenant}} = \text{temps d'activation de la page}$$

Nous calculons par la suite, la moyenne du temps de visualisation pour tous les apprenants ayant visualisé le document :

$$T_{\text{visualisation_doc}} = \text{moyenne}(T_{\text{visualisation_apprenant}})$$

Enfin, nous comptons le ratio défini par :

$$\text{Ratio} = T_{\text{visualisation_doc}} / \text{taille du document}$$

III.7.2.4. Comment utiliser la formule de chan ?

Chan (1999) a défini le degré d'intérêt d'une page web par :

interest (*Page*)

$$= \text{Frequency}(\text{Page}) * (1 + \text{IsBookmark}(\text{Page}) + \text{Duration}(\text{Page}) \\ + \text{Recency}(\text{Page}) + \text{LinkVisitPercent}(\text{Page}))$$

Où,

- Frequency(Page) est la fréquence de visite de la page ;
- IsBookmark(Page)= 1 si la page appartient au bookmark de l'utilisateur sinon 0 ;

$$- \text{Duration}(Page) = \frac{\frac{\text{TotalDuration}(Page)}{\text{Size}(Page)}}{\frac{\max \text{TotalDuration}(Page)_{page \text{ evisted page}}}{\text{Size}(Page)}} ;$$

$$- \text{Regency}(Page) = \frac{\text{Time}(\text{LastVisit}) - \text{Time}(\text{StartLog})}{\text{Time}(\text{Now}) - \text{Time}(\text{StartLog})} ;$$

$$- \text{LinkVisitPercent}(Page) = \frac{\text{NumberOfLinksVisited}(Page)}{\text{NumberOfLinks}(Page)} .$$

Pour la solution proposée, nous avons pondéré la formule de chan par l'attribution d'un poids aux différents critères de la formule :

interest (Page)

$$\begin{aligned} &= \text{Frequency}(Page) * (1 + \alpha * \text{IsBookmark}(Page) + \beta \\ &* \text{Duration}(Page) + \delta * \text{Regency}(Page) + \gamma \\ &* \text{LinkVisitPercent}(Page)) \end{aligned}$$

Avec $\alpha, \beta, \delta, \gamma$: les poids des variables, $0 < \alpha, \beta, \delta, \gamma < 1$ et $\alpha + \beta + \delta + \gamma = 1$.

III.7.2.5. Extraction de la connaissance à l'aide de la méthode alpha ?

Il s'agit ici de combiner le critère explicite (l'appréciation de l'apprenant) avec le critère implicite (ratio, intérêt (page)) pour produire des recommandations.

Alors, nous proposons les documents selon l'ordre décroissant de pertinence.

$$\text{score de pertinence (D)} = \alpha \text{ note d'appréciation} + (1 - \alpha) \text{interest}(Page)$$

Avec α : le poids de la variable et $0 < \alpha < 1$.

Ainsi, pour maintenir la répartition du poids des variables et pour accorder la même importance, nous attribuons la valeur de 0,5 à α .

III.7.2.6. Attribution du lien visité à une catégorie des cours ?

Il s'agit ici de classer les liens renvoyés par le moteur de recherche web. Ceci conduira à classer tous les documents sollicités par tous les apprenants de la plate forme durant leur navigation.

Pour aboutir à cette classification, nous considérons en effet les catégories des cours dans lesquels l'apprenant est inscrit comme des classes que nous affectons des objets (les documents web visités par l'apprenant).

Algorithme : Algorithme d'attribution d'un lien à une catégorie de cours

Algorithme classification

```
variable A, B, C, D, E, hi : ensemble du chaîne de caractère ;  
          H : ensemble d'ensemble du chaîne de caractère ;
```

Début

```
H ←  $\cup$  hi ; / i ∈ [1...nombre des modules]
```

où hi est l'ensemble des mots-clés de chaque module

```
A ← extraire mots-clés (requête apprenant) ;
```

```
B ← extraire mots-clés (titre du lien visité) ;
```

```
C ← extraire mots-clés (lien visité) ;
```

```
D ← extraire mots-clés (description du lien visité) ;
```

```
E ← AUBUCUD ;
```

```
Calcul_Similariter (E, hi) ;
```

```
attribuer le lien visité à la classe (catégorie de cours ou  
module) qui correspond à hi ;
```

Fin

III.8. Conclusion

Dans ce travail, nous avons expliqué comment les techniques du web usage mining et celle de filtrage de l'information contribuent à produire des recommandations pour l'amélioration des services offerts par les plates formes e-Learning afin de guider et faciliter l'apprentissage des apprenants.

La solution proposée MX-Search est basée, sur un système de recherche d'information, sur les concepts du web usage mining et celle du filtrage de l'information en vue de garder la trace de navigation des apprenants de la plate forme e-Learning moodle durant leurs recherches sur le web avec leur participation effective : (vote, clic ...). Ceci aboutira à avoir une base d'usage qui sera utilisée par le système pour aider l'administrateur (Webmaster ou enseignant) à prendre des décisions concernant les meilleurs documents (d'après le point de vue apprenant bien sûr) existant sur le web pour les ajouter comme ressources supplémentaires dans la plate forme afin que ces apprenants ou d'autres promotions puissent en profiter.

Chapitre IV. VALIDATION DES APPROCHES PROPOSEES POUR AMELIORER LA RECHERCHE D'INFORMATION SUR LE WEB

IV.1. Introduction	114
IV.2. Corpus de test	115
IV.2.1. Corpus OHSUMED	115
IV.2.2. Corpus Reuters-21578	117
IV.2.3. Corpus TREC	117
IV.2.4. Corpus Cranfield	117
IV.2.5. Corpus MEDLINE	118
IV.2.6. Corpus NPL	118
IV.2.7. Corpus LISA	118
IV.2.8. Corpus CISI	118
IV.2.9. Corpus TIME	119
IV.2.10. Corpus CACM	119
IV.3. La similarité entre requêtes dans un SRI	119
IV.4. Fonctionnement du système de recommandation proposé	131
IV.4.1. Assister l'apprenant pour parcourir le résultat de sa recherche sur le web	132
IV.4.2. Guider l'administrateur de la plate forme à extraire des connaissances	134
Figure IV.10 : Résultat obtenu pour le module java selon le critère : Formule de chan	137
IV.5. Conclusion	137

IV.1. Introduction

Comment satisfaire quelqu'un qui cherche de l'information sur le web ? Cette question a donné naissance à des travaux et des publications pour la résolution de cette problématique. Celle-ci est définie par une question brève mais qui cache derrière elle une série de questions et de problèmes.

Les deux précédents chapitres, décrivent d'une façon théorique de nouvelles méthodes et outils pour répondre à cette problématique selon une vision parfois différente et parfois complémentaire aux méthodes et outils qui entrent en jeu.

Dans ce chapitre, nous montrerons des exemples concrets pour la validation des différentes méthodes et algorithmes proposés dans cette thèse par une amélioration du service de la recherche proposé par les systèmes de recherche d'information. La confirmation des travaux réalisés est faite suivant deux stratégies :

- Durant la première stratégie, nous avons testé notre méthode de similarité entre une requête qu'un utilisateur d'un système de recherche d'information a entré et les requêtes stockées dans la base afin de présenter les documents qui répondent mieux au besoin exprimé par la requête de l'utilisateur ;
- Durant la deuxième stratégie, nous avons introduit le système de recommandation dans une plate forme e-Learning pour suggérer et faciliter la recherche des ressources web pour les apprenants.

Ce chapitre est structuré comme suit. La section 2, présente les corpus utilisés dans le domaine de la recherche d'information. La section 3, présente en détaille les résultats obtenus par notre méthode hybride de calcul de similarité entre deux requêtes dans un système de recherche d'information. La section 4, décrit les résultats obtenus par le système de recommandation proposé. Une conclusion de ce chapitre sera donnée dans la section IV.5.

IV.2. Corpus de test

Un certain nombre d'ensemble de données (corpus) a été utilisé à des fins expérimentales d'évaluation pour le domaine de la recherche d'information.

Dans un corpus de test, on trouve :

- Un ensemble de documents ;
- Un ensemble de requêtes ;
- La liste des documents pertinents pour chaque requête.

Tous les ensembles de données ci-dessous ont été prétraitées par enlèvement de mots vides, lemmatisation des termes par l'utilisation de l'algorithme de Porter, et la pondération de TF-IDF (Manning et al., 2008).

IV.2.1. Corpus OHSUMED

La collection OHSUMED (Hersh et al., 1994) a été créée pour la recherche d'information. Il s'agit d'un sous-ensemble de données MEDLINE, une base de données sur les publications médicales. La collection se compose de 348.566 enregistrements à partir de 270 revues médicales au cours de la période de 1987-1991. La base de données de test est d'environ 400 mégaoctets. Les champs d'un enregistrement incluent le titre, le résumé, les termes d'indexation MeSH, l'auteur, la source et le type de publication.

Le recueil a été construit dans le cadre d'une étude visant à évaluer l'utilisation de MEDLINE par les médecins dans un contexte clinique (Hersh et al., 1994). Médecins débutants dans MEDLINE générés 106 requêtes. Avant ils ont fouillé, ils ont été invités à fournir une déclaration de renseignements à propos de leurs patients ainsi que leur besoin d'information. Chaque requête a ensuite été reproduite par quatre chercheurs, deux médecins expérimentés dans la recherche et deux bibliothécaires médicaux. Les résultats ont été évalués pour la pertinence par un autre groupe de médecins, en utilisant une échelle en trois points : définitivement, peut-être, ou non pertinentes. Il y avait 12.565 uniques requêtes de références paires. Plus de 10% des paires de requête-document ont été jugés en double exemplaire pour évaluer la fiabilité inter observateur.

Le corpus est ensuite utilisé dans des expériences avec le système de récupération SMART (Hersh et al., 2001). Comme on s'y attendait, SMART a récupéré un certain nombre

de paires. Une deuxième série de jugements de pertinence a été effectuée après ces expériences. Il y avait 3.575 nouvelles paires requêtes-documents.

Il y a donc 106 requêtes, chacune avec un certain nombre de documents associés. Une requête exprime un besoin de recherche médicale, elle est donc également associée à l'information du patient et du sujet de l'information. Les degrés de pertinence des documents par rapport aux requêtes sont jugés par les êtres humains, sur trois niveaux : certainement pertinents, partiellement pertinents ou non pertinents. On trouve en total 16.140 paires requête-document avec jugements de pertinence. Ceux-ci sont dans un fichier (judged), donné avec le jugement de pertinence. Il existe aussi des fichiers qui listent les documents pertinents pour chaque paire requête-document (drel.i, drel.ui, pdrel.i et pdrel.ui). Dans ces fichiers, seul le jugement pertinence d'origine est utilisé.

Les documents MEDLINE ont le même format de fichier que ceux du système SMART, avec chaque champ défini comme ci-dessous (NLM désignation entre parenthèses) :

- .I : Identifiant séquentiel (I)
- .U : Identifiant Medline (UI)
- .M : Termes MeSH assignés-homme (MH)
- .T : Titre (TI)
- .P : Type de Publication (PT)
- .W : Résumer (AB)
- .A : Auteur (AU)
- .S : Source (SO)

Pour chaque requête de la collection OHSUMED, l'information du patient et le sujet de la requête sont définis de la manière suivante :

- .I : Identifiant séquentiel
- .B : L'information des patients
- .W : L'information sur la requête

Nombreux travaux de recherche (Nallapati et al., 2004 ; Qin et al., 2007) ont été publiés à l'aide de la collection OHSUMED.

IV.2.2. **Corpus Reuters-21578**

Le corpus Reuters-21578¹¹ est composé de 21578 documents extraits du journal « Reuters » en 1987. . Il contient 21,578 documents dans 135 catégories. Ce corpus est souvent utilisé comme base de comparaison entre les différents outils de classification de documents, ceci d'une part. D'autre part, on retiendra que le Reuters-21578 est souvent qualifié de « corpus difficile » pour des traitements complexes.

IV.2.3. **Corpus TREC**

Le corpus TREC¹² est un vaste ensemble de données composées d'articles tirés d'une variété de fil de presse et d'autres sources. L'ensemble de données est à la base de la compétition TREC de la recherche d'information.

Cet ensemble de données est constitué de 528.155 documents couvrant un total de 165,363,765 termes d'un vocabulaire de taille (629 469 après suppression des mots vides), il fournit également un certain nombre de chaînes de requête composé de trois parties, le titre, la description et la narration. Les degrés de pertinence des documents par rapport aux requêtes sont aussi présentés sur deux niveaux : certainement pertinents ou non pertinents.

IV.2.4. **Corpus Cranfield**

Le corpus Cranfield est un corpus relatif au domaine de la recherche d'information composé de 1400 résumés sur des sujets d'ingénierie aéronautique. Les documents contiennent un total de 136935 termes à partir d'un vocabulaire de taille 4632 (après le retrait des mots vides).

Le corpus Cranfield contient également un ensemble de 225 chaînes de requête avec jugement de pertinence.

¹¹ Ce corpus est disponible à l'adresse : <http://research.att.com/~lewis/reuters21578.html>.

¹² Text REtrieval Conferenec. <http://trec.nist.gov/>.

IV.2.5. Corpus MEDLINE

Le corpus MEDLINE est un corpus relatif au domaine de la recherche d'information composé de 1,033 résumé sur des sujets médicaux. Les champs d'un enregistrement incluent juste le résumé.

Le corpus MEDLINE contient également un ensemble de 30 chaînes de requête avec jugement de pertinence.

IV.2.6. Corpus NPL

Le NPL (également connu sous le nom Vaswani) collection est une collection de près de 10.000 titres des documents relatifs au domaine de la recherche d'information. Les champs d'un enregistrement incluent juste le résumé.

Le corpus NPL contient également un ensemble de 93 chaînes de requête avec jugement de pertinence.

IV.2.7. Corpus LISA

Le corpus LISA (The Library and Information Science Abstracts) est un corpus relatif au domaine de la recherche d'information composé de 5,872 documents. Les champs d'un enregistrement incluent juste la description.

Le corpus LISA contient également un ensemble de 33 chaînes de requête avec jugement de pertinence.

IV.2.8. Corpus CISI

Le corpus CISI est un corpus relatif au domaine de la recherche d'information composé de 1,460 résumé sur des sujets médicaux. Les champs d'un enregistrement incluent le titre, résumé, auteur...

Le corpus CISI contient également un ensemble de 112 chaînes de requête avec jugement de pertinence.

IV.2.9. **Corpus TIME**

Le corpus TIME est un corpus relatif au domaine de la recherche d'information composé de 423 résumés sur des articles de la revue Time. Les champs d'un enregistrement incluent la description de chaque article.

Le corpus TIME contient également un ensemble de 83 chaînes de requête avec jugement de pertinence.

IV.2.10. **Corpus CACM**

CACM est un ensemble de résumés des articles publiés dans les communications de la revue ACM entre 1958 et 1979. Cette collection a été utilisée dans de nombreux travaux de la recherche d'information. Les champs d'un enregistrement incluent le titre, résumé, auteur, source et type de publication.

Il ya 64 requêtes, chacune avec un certain nombre de documents associés. Une requête exprime un besoin de recherche envers le système informatique. Les degrés de pertinence des documents par rapport aux requêtes sont jugées par les êtres humains, sur deux niveaux : certainement pertinent ou non pertinent. Il ya en total de 796 paires requête-document avec jugements de pertinence.

IV.3. **La similarité entre requêtes dans un SRI**

Le corpus choisi pour notre test est CACM.

Pour tester notre méthode hybride de calcul de la similarité entre deux requêtes dans un système de recherche d'information (SRI), nous avons joué le rôle d'un usager qui cherche une réponse à sa requête dans le corpus CACM. La réponse bien sûr est définie en proposant à l'utilisateur les documents pertinents de la requête candidate R_k la plus similaire à la requête donnée (Q) par cet usager.

Nous avons évalué deux requêtes :

La première requête que nous avons introduite au système est la suivante: "*I am interested in articles written by Schatzoff or Nielsen*" ;

Cette requête a été comparée avec toutes les requêtes du corpus c.-a-d avec 64 requêtes (le fichier *query.text* de la base CACM présente ces requêtes selon leur indice qui varie de 1 jusqu'as 64) :

Le résultat de la similarité a été comparé avec les différentes méthodes de calcul de similarité,

Indice de la requête	Score de la requête candidate du système
Requête1	72.34934061854007
Requête2	0.0
Requête3	0.0
Requête4	8.679906466270948
Requête5	0.0
Requête6	0.0
Requête7	1.3242241522032823
Requête8	0.0
Requête9	0.0
Requête10	0.18809587871166758
Requête11	0.45545382606868906
Requête12	10.288820340909595
Requête13	0.5527537752629486
Requête14	0.0
Requête15	0.0
Requête16	1.2938582499785642
Requête17	0.0
Requête18	3.722803924322979
Requête19	1.2908296616350527
Requête20	0.0
Requête21	0.0
Requête22	0.0
Requête23	5.0277569930216615
Requête24	0.0
Requête25	3.654771408407613
Requête26	0.3316679721776381
Requête27	1.4604683102311995
Requête28	0.0
Requête29	0.0
Requête30	2.8640768811264397
Requête31	0.0
Requête32	5.894112265354721
Requête33	3.2495704374184284E-5
Requête34	0.0
Requête35	0.0
Requête36	0.40265645106420744
Requête37	0.862373807448304
Requête38	0.20350784247619963
Requête39	6.062859375620152
Requête40	0.6747484729821925
Requête41	0.0

Requête42	2.3159721817133008
Requête43	0.7776997427641729
Requête44	0.0
Requête45	0.0
Requête46	0.0
Requête47	0.0
Requête48	0.0
Requête49	0.0
Requête50	0.0
Requête51	0.0
Requête52	0.0
Requête53	0.0
Requête54	0.0
Requête55	0.0
Requête56	0.0
Requête57	14.258653622175636
Requête58	0.0
Requête59	0.003921570026720721
Requête60	0.0
Requête61	0.0
Requête62	1.6695337560592376
Requête63	0.862373807448304
Requête64	0.0

Tableau IV.1: Résultat de similarité selon notre méthode hybride

Indice de la requête	Score de la requête candidate du système
Requête 1	0.030467177128182994
Requête2	0.08094719556433197
Requête3	0.0
Requête4	0.03502228346505972
Requête5	0.0
Requête6	0.056888113935025346
Requête7	0.03121688456558441
Requête8	0.0
Requête9	0.0
Requête10	0.0
Requête11	0.0
Requête12	0.0
Requête13	0.0
Requête14	0.0
Requête15	0.0
Requête16	0.0
Requête17	0.0
Requête18	0.0
Requête19	0.0
Requête20	0.0
Requête21	0.0

Requête22	0.03659372280019521
Requête23	0.0
Requête24	0.0
Requête25	0.0
Requête26	0.0
Requête27	0.0
Requête28	0.026982398521443986
Requête29	0.0
Requête30	0.03192812815065668
Requête31	0.038871983392228554
Requête32	0.08443197417107096
Requête33	0.09052816694543234
Requête34	0.06385625630131336
Requête35	0.0
Requête36	0.0
Requête37	0.04517745038377601
Requête38	0.03755039243599696
Requête39	0.0
Requête40	0.085722086771279
Requête41	0.04889725512335404
Requête42	0.0
Requête43	0.0
Requête44	0.0
Requête45	0.0
Requête46	0.0
Requête47	0.0
Requête48	0.0
Requête49	0.0
Requête50	0.0
Requête51	0.0
Requête52	0.0
Requête53	0.0
Requête54	0.0
Requête55	0.0
Requête56	0.03192812815065668
Requête57	0.0428610433856395
Requête58	0.04406756140034515
Requête59	0.0
Requête60	0.0
Requête61	0.040828208844335634
Requête62	0.0
Requête63	0.0
Requête64	0.015233588564091497

Tableau IV.2: Résultat de similarité selon la méthode de divergence de Kullbak-leiber

Indice de la requête	Score de la requête candidate du système
Requête1	0.07692307692307693
Requête2	0.3333333333333333
Requête3	0.0
Requête4	0.07142857142857142
Requête5	0.0
Requête6	0.23076923076923078
Requête7	0.09090909090909091
Requête8	0.0
Requête9	0.0
Requête10	0.0
Requête11	0.0
Requête12	0.0
Requête13	0.0
Requête14	0.0
Requête15	0.0
Requête16	0.0
Requête17	0.0
Requête18	0.0
Requête19	0.0
Requête20	0.0
Requête21	0.0
Requête22	0.05555555555555555
Requête23	0.0
Requête24	0.0
Requête25	0.0
Requête26	0.0
Requête27	0.0
Requête28	0.11764705882352941
Requête29	0.0
Requête30	0.07142857142857142
Requête31	0.05555555555555555
Requête32	0.08333333333333333
Requête33	0.07692307692307693
Requête34	0.15384615384615385
Requête35	0.0
Requête36	0.0
Requête37	0.03333333333333333
Requête38	0.05263157894736842
Requête39	0.0
Requête40	0.08
Requête41	0.02631578947368421
Requête42	0.0
Requête43	0.0
Requête44	0.0
Requête45	0.0
Requête46	0.0
Requête47	0.0

Requête48	0.0
Requête49	0.0
Requête50	0.0
Requête51	0.0
Requête52	0.0
Requête53	0.0
Requête54	0.0
Requête55	0.0
Requête56	0.07142857142857142
Requête57	0.038461538461538464
Requête58	0.03571428571428571
Requête59	0.0
Requête60	0.0
Requête61	0.043478260869565216
Requête62	0.0
Requête63	0.0
Requête64	0.14285714285714285

Tableau IV.3: Résultat de similarité selon l'indice de jaccard

La méthode	L'indice de la requête réponse dans le fichier query.text
Divergence de Kullback-Leibler	R33
Indice de Jaccard	R2
Notre méthode hybride	R1

Tableau IV.4: Comparaison du résultat de similarité

Discussion des résultats :

- Le résultat donné avec l'utilisation de notre méthode hybride basée sur le chi statistique et l'information mutuelle, montre que la première requête candidate R1 de corpus a obtenu un score plus élevé par rapport aux autres requêtes. Ce qui montre que cette requête est similaire par rapport à la requête entrée par l'utilisateur du système. En effet, si nous examinons bien la requête résultante R1 de corpus donné par notre méthode : ***“What articles exist which deal with TSS***

(Time Sharing System), an operating system for IBM computers ?”, nous remarquons facilement qu’elle n’existe pratiquement aucune relation entre les termes de la requête introduite par l’utilisateur du système de recherche d’information et la requête candidate R1 donnée par le système comme résultat avec notre méthode hybride. Or, si nous examinons bien, les documents pertinents de la requête R1, nous trouvons que ces documents sont des articles écrits soit par Schatzoff (le document dont l’ID est 1605) ou Nielsen (les documents dont l’ID est 1572 et 2020). Donc malgré l’inexistence de relation entre les termes des deux requêtes en question, nous apercevons bien que la requête réponse R1 est la plus adéquate en terme de similarité à la requête usagée (Q) et que le cluster des documents pertinents de R1 répond convenablement au besoin défini préalablement par l’utilisateur de SRI ;

- Le résultat donné par la méthode de KULLBAK-LEIBLER, décèle que la requête candidate R33 de corpus a obtenu un score plus élevé par rapport aux autres requêtes. Ce qui démontre que cette requête est similaire à la requête entrée par l’usager du système. En effet, si nous observons bien la requête résultante R33 de corpus donné par la méthode de Kullbak-Leibler : “ *Articles about the sensitivity of the eigenvalue decomposition of real matrices, in particular, zero-one matrices. I'm especibally interested in the separation of eigenspaces corresponding to distinct eigenvalues. Articles on the subject : C. Davis and W.M. Kahn, "The rotation of eigenvectors by a permutation;, SIAM J. Numerical Analysis, vol. 7, no. 1 (1970); G.W. Stewart, "Error bounds for approximate invariant subspaces of closed linear operators",SIAM J. Numerical Analysis., Vol. 8, no. 4 (1971)*”, nous nous apercevons facilement qu’il existe une relation partielle entre les termes de la requête introduite par l’utilisateur du système de recherche d’information et la requête candidate R33 donnée par le système comme résultat par la méthode de Kullbak-Leibler. Or, si nous examinons bien, les documents pertinents de la requête R33, nous nous rendons compte qu’il y a un seul document pertinent écrit par van der Sluis (le document dont l’ID est 2805). Donc malgré, la relation qui existe entre les termes des deux requêtes en question, on voit bien que la requête réponse R33 n’est plus adéquate en termes de similarité à la requête de l’usager (Q) et que le

cluster des documents pertinents de R33 ne répond pas au besoin défini préalablement par l'utilisateur du SRI ;

- Le résultat donné par l'indice de jaccard, montre que la deuxième requête candidate R2 de corpus a obtenu un score plus élevé par rapport aux autres requêtes. Ce qui implique que cette requête est similaire à la requête entrée par l'usager du système. En effet, si nous observons bien la requête résultante R2 de corpus donné par la méthode de jaccard : “ *I am interested in articles written either by Prieve or Udo Pooch* ”, nous remarquons facilement qu'il existe une relation partielle entre les termes de la requête introduite par l'utilisateur du système de recherche d'information et la requête candidate R2 donné par le système comme résultat en utilisant l'indice de jaccard. Or, si nous étudions bien, les documents pertinents de la requête R2, on s'aperçoit que ces documents sont des articles écrits soit par Prieve (les documents dont l'ID est 2434, 2863), Fabry (le document dont l'ID est 2863) ou Chattergy and Pooch (le document dont l'ID est 3078). Par conséquent et en dépit de la relation qui existe entre les termes des deux requêtes en question, on découvre bien que la requête réponse R1 n'est plus adéquate en terme de similarité à la requête usagée (Q) et que le cluster des documents pertinents de R2 ne répond pas au besoin défini préalablement par l'utilisateur du SRI.

La seconde requête que nous avons introduit au système est la suivante : “*I am interested in articles articles about Time Sharing System and written by Schatzoff or Nielsen*”.

Cette requête a été comparée avec toutes les requêtes du corpus c.-a-d avec 64 requêtes :

Indice de la requête	Score de la requête candidate du système
Requête1	69.01748587489311
Requête2	0.369162191257701
Requête3	0.05135401389159776
Requête4	6.078003315939318
Requête5	1.0919160632055709
Requête6	0.22001017504911383
Requête7	2.334572270475992
Requête8	3.0398217444827877
Requête9	4.033942055894058
Requête10	0.21825511840158318
Requête11	0.45545382606868906
Requête12	6.974320121696374
Requête13	0.5527537752629486

Requête14	0.0
Requête15	0.3959254024397866
Requête16	5.181036983324167
Requête17	0.0
Requête18	2.526981559822743
Requête19	1.1816157611471418
Requête20	0.02557986820577695
Requête21	3.649212770402028
Requête22	0.0
Requête23	4.172108955527073
Requête24	1.1621842206781958
Requête25	5.787748054191779
Requête26	4.320835733130966
Requête27	2.3959727822334393
Requête28	1.223006304809021
Requête29	0.0
Requête30	2.8407290439735555
Requête31	0.25420184554469455
Requête32	5.627160526738177
Requête33	0.6854173414665784
Requête34	0.0
Requête35	0.0
Requête36	0.40265645106420744
Requête37	0.9486853395716688
Requête38	0.20350784247619963
Requête39	5.4488344490858225
Requête40	0.6747484729821925
Requête41	0.0
Requête42	3.9816086976887304
Requête43	0.7776997427641729
Requête44	0.0
Requête45	0.33442308143736277
Requête46	0.0
Requête47	0.0
Requête48	0.0
Requête49	0.6825618026040506
Requête50	0.0
Requête51	0.0
Requête52	0.0
Requête53	0.0
Requête54	0.0
Requête55	0.0
Requête56	0.0
Requête57	12.027091817085296
Requête58	5.376832807734095E-6
Requête59	0.002885298549873349
Requête60	2.4669797036035894
Requête61	1.6617302517079764

Requête62	2.313990707033954
Requête63	1.9165322676125458
Requête64	0.0

Tableau IV.5: Résultat de similarité selon notre méthode hybride

Indice de la requête	Score de la requête candidate du système
Requête1	0.07378048474716901
Requête2	0.0532020531960022
Requête3	0.0
Requête4	0.02301818333499609
Requête5	0.0
Requête6	0.03738936775624452
Requête7	0.020517107994825787
Requête8	0.020024367329902197
Requête9	0.013707382757462428
Requête10	0.0
Requête11	0.0
Requête12	0.010012183664951099
Requête13	0.0
Requête14	0.0
Requête15	0.0
Requête16	0.020984568526157696
Requête17	0.0
Requête18	0.0
Requête19	0.0
Requête20	0.0
Requête21	0.0
Requête22	0.024051002304440502
Requête23	0.0
Requête24	0.0
Requête25	0.014667583953717929
Requête26	0.014667583953717929
Requête27	0.014667583953717929
Requête28	0.017734017732000732
Requête29	0.0
Requête30	0.03565215247987562
Requête31	0.02554837525685347
Requête32	0.05549240279390366
Requête33	0.05949908851060523
Requête34	0.04196913705231539
Requête35	0.0
Requête36	0.0
Requête37	0.05938521036636082
Requête38	0.024679767618669027
Requête39	0.0
Requête40	0.05634032147356381
Requête41	0.05426266489712687

Requête42	0.0
Requête43	0.0
Requête44	0.0
Requête45	0.025820439708599215
Requête46	0.0
Requête47	0.026341351902341967
Requête48	0.0
Requête49	0.012633969144879536
Requête50	0.0
Requête51	0.0
Requête52	0.0
Requête53	0.0
Requête54	0.0
Requête55	0.0
Requête56	0.020984568526157696
Requête57	0.028170160736781906
Requête58	0.057926274764540805
Requête59	0.0
Requête60	0.020024367329902197
Requête61	0.026834092567265554
Requête62	0.0
Requête63	0.0
Requête64	0.010012183664951099

Tableau IV.6: Résultat de similarité selon la méthode de divergence de Kullbak-leiber

Indice de la requête	Score de la requête candidate du système
Requête 1	0.4166666666666667
Requête2	0.25
Requête3	0.0
Requête4	0.06451612903225806
Requête5	0.0
Requête6	0.1875
Requête7	0.08
Requête8	0.0625
Requête9	0.13333333333333333
Requête10	0.0
Requête11	0.0
Requête12	0.1
Requête13	0.0
Requête14	0.0
Requête15	0.0
Requête16	0.058823529411764705
Requête17	0.0
Requête18	0.0
Requête19	0.0
Requête20	0.0
Requête21	0.0

Requête22	0.047619047619047616
Requête23	0.0
Requête24	0.0
Requête25	0.08333333333333333
Requête26	0.08333333333333333
Requête27	0.08333333333333333
Requête28	0.1
Requête29	0.0
Requête30	0.2
Requête31	0.05128205128205128
Requête32	0.07407407407407407
Requête33	0.07142857142857142
Requête34	0.125
Requête35	0.0
Requête36	0.0
Requête37	0.0625
Requête38	0.045454545454545456
Requête39	0.0
Requête40	0.07142857142857142
Requête41	0.10526315789473684
Requête42	0.0
Requête43	0.0
Requête44	0.0
Requête45	0.041666666666666664
Requête46	0.0
Requête47	0.04
Requête48	0.0
Requête49	0.14285714285714285
Requête50	0.0
Requête51	0.0
Requête52	0.0
Requête53	0.0
Requête54	0.0
Requête55	0.0
Requête56	0.058823529411764705
Requête57	0.034482758620689655
Requête58	0.06666666666666667
Requête59	0.0
Requête60	0.09375
Requête61	0.038461538461538464
Requête62	0.0
Requête63	0.0
Requête64	0.1

Tableau IV.7: Résultat de similarité selon l'indice de jaccard

La méthode	L'indice de la requête réponse dans le fichier query.text
Divergence de Kullback-Leibler	R1
Indice de Jaccard	R1
Notre méthode hybride	R1

Tableau IV.8: Comparaison du résultat de similarité

- Les résultats donnés par utilisation de différentes méthodes présentées dans ce travail (y inclus notre méthode) montrent que la première requête candidate R1 de corpus a obtenu un score plus élevé par rapport aux autres requêtes. Ce qui montre que cette requête est similaire par rapport à la requête entrée par l'utilisateur du système. En effet, si nous examinons bien la requête résultante R1 du corpus donnée comme réponse : *“What articles exist which deal with TSS (Time Sharing System), an operating system for IBM computers ?”*, nous nous rendons compte qu'il y a pratiquement une dépendance entre les termes de la requête candidate et la requête système.

Nous constatons d'après les résultats présentés ci-dessus que la mesure hybride que nous proposons pour la mesure de la similarité entre deux requêtes surpasse les différentes méthodes mentionnées. Ce qui prouve que notre proposition est efficace en termes de satisfaction utilisateur si elle est introduite et utilisée par un système de recherche d'information.

IV.4. Fonctionnement du système de recommandation proposé

Pour tester le fonctionnement du système proposé (nommé MX-Search), nous l'avons intégré dans la plate forme e-Learning moodle de la licence professionnelle Java/C++ de la faculté des sciences de Casablanca, Maroc (figure IV.1).

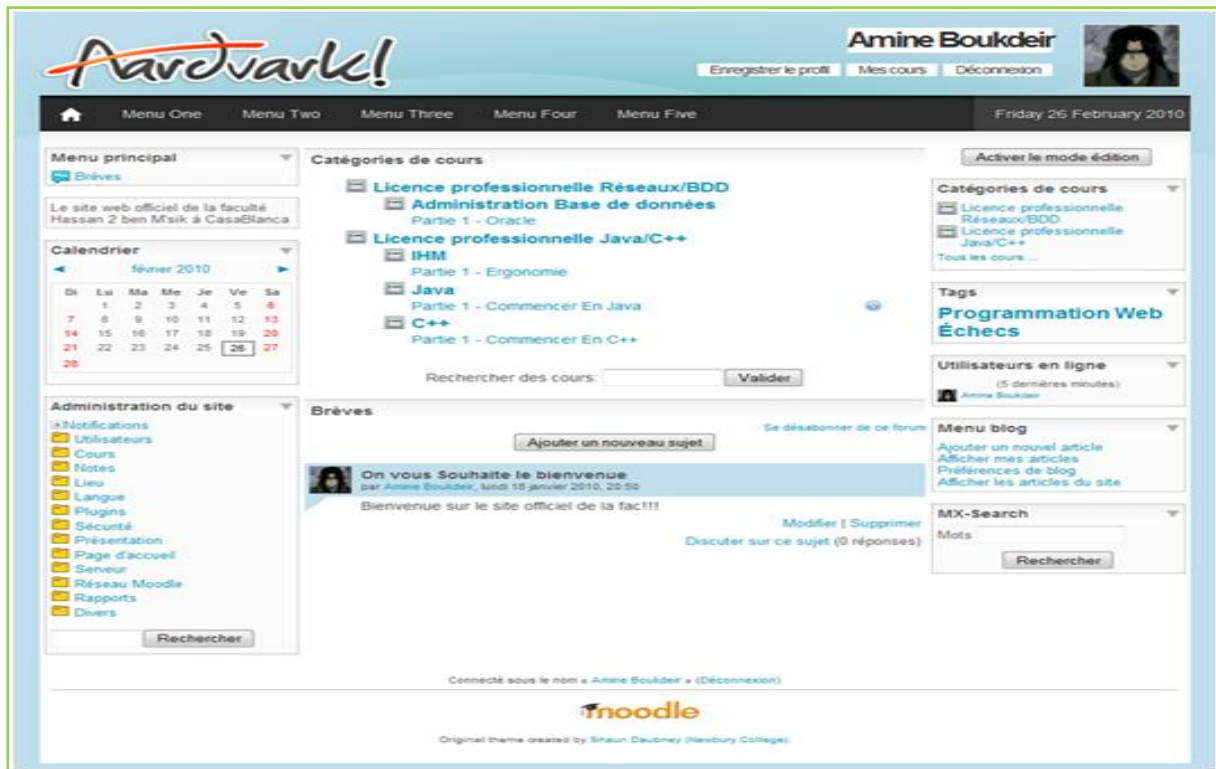


Figure IV.1: Interface de la plate forme e-learning moodle: LP Java/C++

IV.4.1. Assister l'apprenant pour parcourir le résultat de sa recherche sur le web

L'apprenant de la plate forme e-Learning LP Java/C++ peut effectuer des recherches sur le web tout en restant sur la plate forme (figure IV.2).

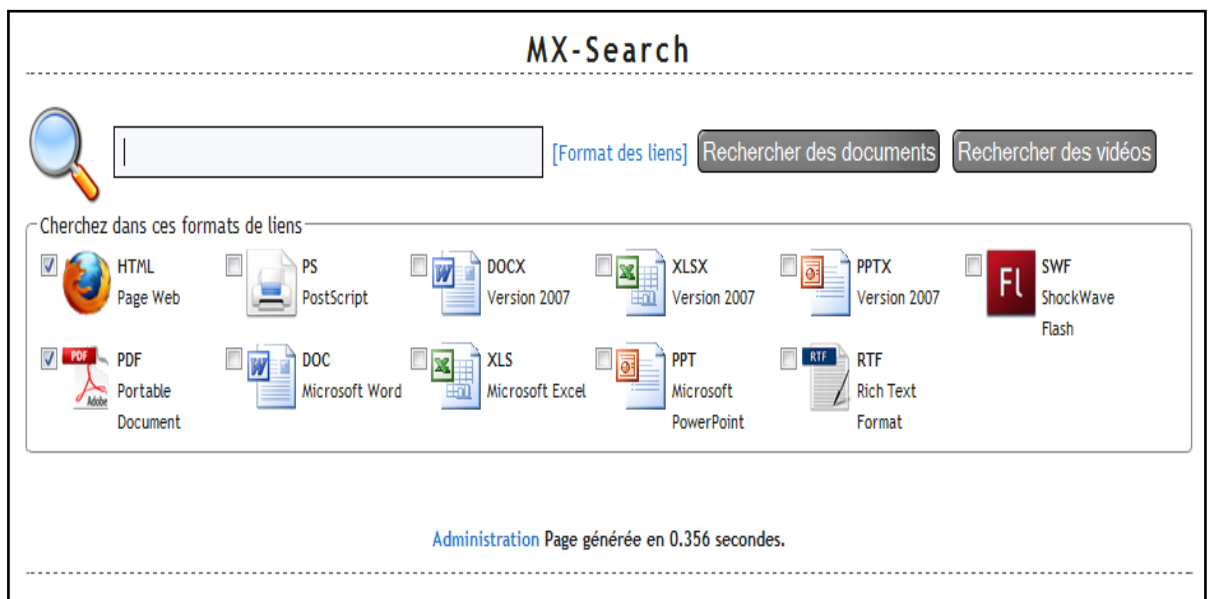


Figure IV.2: Interface pour effectuer des recherches sur le web

Le système MX-Search présente le résultat de la recherche renvoyé par le moteur de recherche avec des statistiques (Nombre de clic, Appréciations des apprenants) sur les liens préalablement visités par d'autres apprenants en vue de faciliter le parcours de ces documents (figure IV.3). Ainsi, l'apprenant visualise le contenu du lien tout en restant sur la plate forme e-Learning avec la possibilité de donner son appréciation (vote) (figure IV.4).

Numéro	Titre	Nombre de clic	Appréciations
#1	Apprendre Java	0	
#2	Du C/C++ à Java : Table des matières	76	Bien : 5. Moyen : 12. Nul : 3. Pas Mal : 6
#3	Java - page de ressources et cours	81	Bien : 9. Excellent : 2. Moyen : 15. Nul : 2. Pas Mal : 3
#4	Cours Java de Patrick Itey - INRIA Sophia Antipolis	51	
#5	Cours et applets Java	77	Bien : 8. Excellent : 3. Moyen : 8. Pas Mal : 4
#6	Programmation en Java	118	Bien : 10. Excellent : 14. Moyen : 12
#7	Java [Cours]Formation à télécharger .pdf .zip ou .rar	3	
#8	Support de cours JAVA	78	Bien : 1. Moyen : 11. Pas Mal : 3
#9	Cours document langage de programmation Java C C++	76	
#10	Apprentissage de Java (tm) - F. Rossi	75	
#11	Cours POO-JAVA	74	
#12	Cours sur C C++ JAVA en francais	75	Bien : 3. Excellent : 20. Moyen : 6. Nul : 3. Pas Mal : 6
#13	improve-technologies.com - Formation objet : Support de cours, pdf ...	0	
#14	Cours de JavaScript et DHTML [L'éditeur JavaScript]	74	
#15	Livre cours bases de l'informatique, Java et C#	74	

Figure IV.3: Interface pour parcourir le résultat de la recherche



Figure IV.4: Interface pour le contenu du lien web

IV.4.2. Guider l'administrateur de la plate forme à extraire des connaissances

L'administrateur (webmaster ou enseignant) de la plate forme e-Learning moodle de la licence professionnelle Java/C++ peut choisir la filière, le module désiré ainsi que le critère d'extraction de connaissance pour prendre des décisions concernant les documents web à ajouter dans la plate forme e-Learning (figure IV.5).



The screenshot shows a web interface with two dropdown menus at the top, both set to 'Divers'. Below them is a red button labeled 'Proposer moi les meilleurs documents?'. Underneath the button, the text 'Je voudrais savoir :' is followed by a list of six criteria, each preceded by an eye icon.

Choisir une filière

Choisir un module

Proposer moi les meilleurs documents?

Je voudrais savoir :








-  les meilleurs documents qui existent selon le critère : Appréciation ?
-  les meilleurs documents qui existent selon le critère : Ratio ?
-  les meilleurs documents qui existent en combinant les différentes critères ?
-  les meilleurs documents qui existent durant une période ?
-  les meilleurs documents qui existent selon Formule de CHAN ?
-  les meilleurs documents qui existent selon le critère : Appréciation_F.Chan ?
-  les meilleurs documents qui existent selon les formats des liens ?

Figure IV.5: Interface pour choisir les critères d'extraction de connaissances

Après avoir choisi le module java par exemple, le système MX-Search propose à l'administrateur les meilleurs documents web pour ce module d'après le point de vue apprenant et aussi selon les différents critères d'extraction des connaissances (figure IV.6, IV.7, I ,8.VI ,9.VI10.V). Ceci donnera la possibilité aux apprenants d'avoir des ressources supplémentaires pour bien comprendre le module.



Figure IV.6: Résultat obtenu pour le module java selon le critère : Score d'appréciation



Figure IV.7: Résultat obtenu pour le module java selon la combinaison des critères : Score d'appréciation, Ratio

les meilleurs documents qui existent selon le critère : Ratio ?
 les meilleurs documents qui existent en combinant les différentes critères ?
 les meilleurs documents qui existent durant une période ?

Liens classés par Date : Entre 01/01/2010 et 01/07/2010

Nm	Lien	Ratio (sec/ko)	Score Appréciations	Détails Appréciations
#1	http://www-igm.univ-mlv.fr/~dr/C_CPP_index.html	16.6	145 pts	Bien : 3 Excellent : 20 Moyen : 6 Nul : 3 Pas Mal : 6
#2	http://www.lirmm.fr/~ferber/Java/index.html	16.4	99 pts	Bien : 9 Excellent : 2 Moyen : 15 Nul : 2 Pas Mal : 3
#3	http://www.siteduzero.com/tutorial-3-10601-programmation-en-java.html	8.2	146 pts	Bien : 10 Excellent : 14 Moyen : 12
#4	http://www.ducrot.org/java.html	10.2	96 pts	Bien : 2 Moyen : 22 Pas Mal : 6
#5	http://membres.multimania.fr/dancel/java/java.html	13.1	158 pts	Bien : 16 Excellent : 6 Moyen : 16 Pas Mal : 8

[> Résultats Suivants](#)

Aller à la recherche Page générée en 0.455 secondes.

Figure IV.8: Résultat obtenu pour le module java selon le critère : période de recherche

les meilleurs documents qui existent selon le critère : Ratio ?
 les meilleurs documents qui existent en combinant les différentes critères ?
 les meilleurs documents qui existent durant une période ?

Liens classés par format HTML

Nm	Lien	Ratio (sec/ko)	Score Appréciations	Détails Appréciations
#1	http://membres.multimania.fr/dancel/java/java.html	13.1	158 pts	Bien : 16 Excellent : 6 Moyen : 16 Pas Mal : 8
#2	http://www-igm.univ-mlv.fr/~dr/C_CPP_index.html	16.6	145 pts	Bien : 3 Excellent : 20 Moyen : 6 Nul : 3 Pas Mal : 6
#3	http://www.siteduzero.com/tutorial-3-10601-programmation-en-java.html	8.2	146 pts	Bien : 10 Excellent : 14 Moyen : 12
#4	http://www.lirmm.fr/~ferber/Java/index.html	16.4	99 pts	Bien : 9 Excellent : 2 Moyen : 15 Nul : 2 Pas Mal : 3
#5	http://www.ac-creteil.fr/util/programmation/java/welcome.html	10.2	93 pts	Bien : 8 Excellent : 5 Moyen : 10 Pas Mal : 3

[> Résultats Suivants](#)

Figure IV.9: Résultat obtenu pour le module java selon le critère : format du document

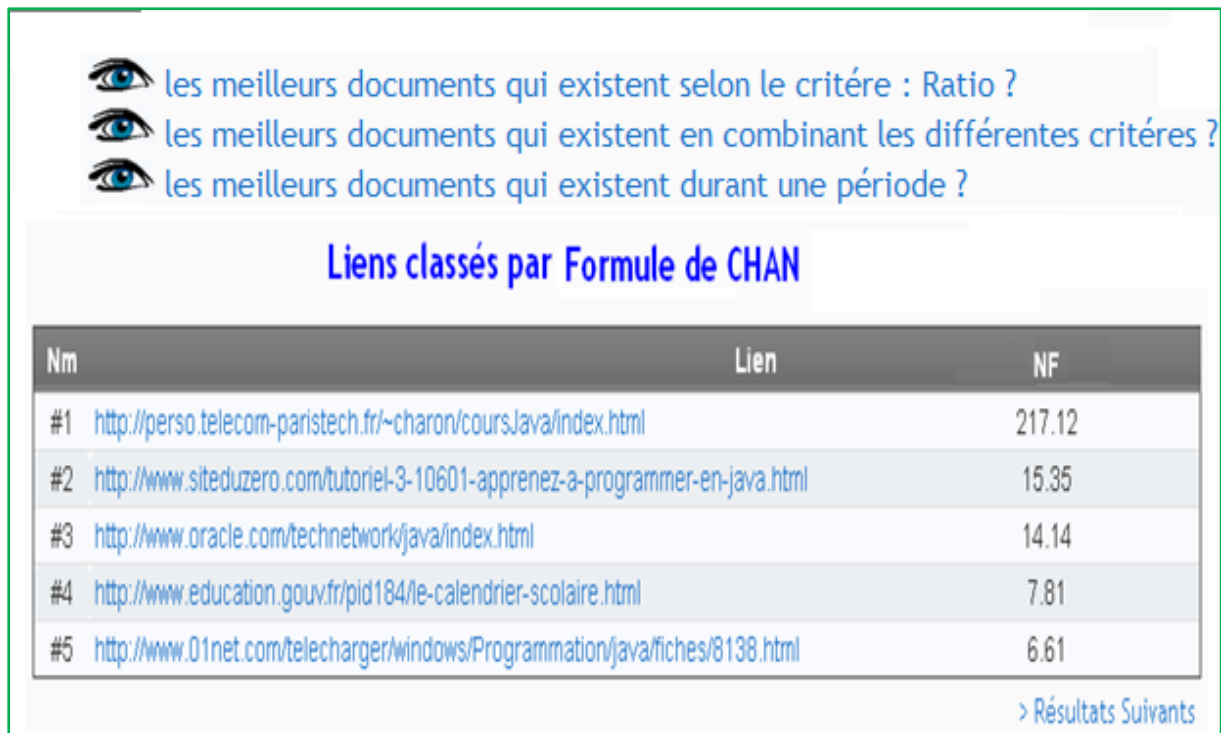


Figure IV.10: Résultat obtenu pour le module java selon le critère : Formule de chan

D'après les résultats obtenus, nous remarquons le feedback des apprenants dans les plates forme e-Learning (soit par vote, selon en utilisant la formule de Chan ou en combinant les deux) nous a permis de proposer une nouvelle méthode pour classer les documents visualisées dans le but d'avoir un système de recommandation capable de suggérer aux apprenants des ressources web facilitant leur apprentissage.

IV.5. Conclusion

Dans ce chapitre, nous avons justifié que les méthodes et les outils proposés permettent de mieux répondre à la problématique citée dans cette thèse. En effet :

- Nous avons présenté dans un premier lieu, une mesure hybride pour calculer la similarité entre deux requêtes dans un système de recherche d'information. Cette mesure se base sur deux aspects : l'un est statistique qui calcule le score de dépendance positive entre la requête usagée Q et la requête candidate R_k en se basant sur la méthode de chir, quant à l'autre aspect, il est sémantique ; il cherche les requêtes candidates similaires à la requête utilisateur par le biais de l'information mutuelle. Cette mesure hybride, montre sa performance en termes

de satisfaction usager par rapport aux différentes méthodes et mesures mentionnées auparavant ;

- Dans un second lieu, nous avons prouvé comment les techniques du web usage mining et celles du filtrage de l'information contribuent à produire des recommandations pour l'amélioration des services offerts par les plates formes e-Learning afin de guider et faciliter l'apprentissage des apprenants. La solution proposée MX-Search est basée sur un système de recherche d'information Web, sur les concepts du web usage mining et celle du filtrage de l'information. Son but est de garder la trace de navigation des apprenants de la plate forme e-Learning moodle pendant les recherches effectuées sur le web tout en les faisant participer : (vote, clic ...). Ceci permettra d'avoir une base d'usage .Celle-ci sera utilisée par le système pour aider l'administrateur (Webmaster ou enseignant) à prendre des décisions au sujet des meilleurs documents (selon le point de vue apprenant bien sûr) existant sur le web. L'objectif de cette opération est de les ajouter dans la plate forme comme ressources supplémentaires pour que ces apprenants ou d'autres promotions puissent en profiter.

Chapitre V. **CONCLUSION ET PERSPECTIVES**

En guise de conclusion, nous rappellerons le contexte de la thèse, puis, nous présenterons la contribution de ce travail. Enfin, nous dévoilerons les perspectives et les nouveaux thèmes de développement et de recherche abordés actuellement dans notre équipe.

V.1. Rappel du contexte de la thèse

Le système de recherche d'information fournit souvent en réponse à une requête des milliers de documents en un temps inférieur à la seconde. Ce sont là précisément les points forts et les points faibles de ce type de méthode : l'utilisateur est submergé par des milliers de réponses présentées "en vrac", dont une partie ne correspond pas à la requête, une autre correspond à des pages qui n'existent plus, sans parler de nombreux types de documents qui ne sont pas pris en compte dans la mémoire du moteur. Dans bien des cas, l'utilisateur passe un temps assez long à analyser les résultats du moteur, sans garantie de résultats.

Le cadre général de nos travaux s'appuie donc, sur la satisfaction des utilisateurs qui cherchent de l'information sur le Web. C'est dans cette optique qu'ont été élaborées de nouvelles méthodes pour être introduites dans un système de recherche d'information (SRI) en vue de présenter à l'utilisateur les documents qui sont susceptibles de répondre à leur besoin exprimé par une requête. Dans un premier temps, nous avons présenté une nouvelle démarche pour calculer la similarité entre deux requêtes dans un SRI. Dans un second temps, nous avons exploité le feedback utilisateur pour produire un nouveau système de recommandation qui participe à l'amélioration de la recherche d'information dans les plates formes e-Learning.

V.2. Apport du présent travail

L'apport principal de nos travaux réside dans la proposition d'une démarche qui tente de répondre au besoin de l'utilisateur exprimé par une simple requête. Nous pouvons examiner l'apport du travail effectué selon plusieurs axes :

- **Nouvelle mesure hybride pour calculer la similarité entre deux requêtes dans un système de recherche d'information :**

Notre contribution consiste à proposer une nouvelle mesure pour calculer la similarité entre deux requêtes dans un SRI, cette similarité est mesurée entre, une nouvelle requête qu'un SRI vient de recevoir et qui exprime bien sûr un besoin usager, et, une requête candidate dont le système a mémorisé les documents jugés pertinents lors des recherches préalables. Ce calcul de similarité passe par trois phases : Dans un premier temps, nous avons présenté une statistique plus précise axée sur la version étendue de la statistique X2, dite la méthode de Chir pour sélectionner les requêtes positivement dépendantes par rapport à la requête donnée. Dans un second temps, nous avons utilisé l'information mutuelle pour mesurer la similarité sémantique entre la requête usagée et la requête candidate du système. Finalement, nous avons combiné ces deux mesures, statistique et sémantique à l'aide de notre méthode dite d'alpha pour prédire la requête candidate la plus proche à la requête donnée en terme de similarité.

Cette mesure de similarité permet en outre de présenter à l'utilisateur d'un système de recherche d'information les documents qui répondent mieux à leurs besoins. De surcroît, la méthode proposée trouve des solutions aux problèmes liés à la recherche d'information à savoir : le bruit et le silence.

- **Système de recommandation pour améliorer le service de recherche d'information dans les plates formes E-Learning :**

Notre contribution consiste à proposer une méthode de classement des documents web en se basant sur l'appréciation de l'utilisateur du système de recherche d'information. Cette méthode a été insérée au sein d'un système de recommandation en profitant de différentes méthodes, outils et techniques qui entrent en jeu à savoir : la recherche d'information, le filtrage d'information et le web usage mining. Le système proposé est appliqué dans le domaine e-Learning pour tester son impact sur le processus de la recherche. En effet, chaque apprenant de la plate forme e-Learning effectue nécessairement des recherches sur le Web

pour avoir des ressources qui peuvent l'aider à bien saisir et comprendre les cours dont il est inscrit. Or, la tâche de trouver des ressources pertinentes dans un temps minimal est fastidieuse, d'où la nécessité d'avoir un système qui profite des recherches préalablement effectuées pour produire des recommandations afin d'aider les apprenants durant leur apprentissage. La méthode que nous suggérons exploite justement le feedback des apprenants exprimé sous la forme d'une appréciation (ou vote) à l'égard des ressources consultées auparavant, ainsi, que leurs interactions avec le système de la recherche d'information.

V.3. Perspectives de la thèse

La démarche présentée dans cette thèse, se focalise sur la satisfaction de l'utilisateur dans le domaine de la recherche d'information. Néanmoins, elle présente certaines limites et peut faire l'objet d'éventuelles recherches sur plusieurs voies :

- Le test de notre méthode de calcul de la similarité suppose que le système de recherche d'information possède pour chaque requête un ensemble de documents jugés pertinents. Or, en général ce n'est pas toujours le cas pour un nouveau SRI. D'où la nécessité de :
 1. Faire une catégorisation des documents web pour grouper les documents de même sujets : travail en cours de publication au sein de notre équipe de recherche ;
 2. À l'arrivée d'une nouvelle requête utilisateur, au lieu de filtrer tous les documents web, nous proposerons dans nos futurs travaux inchaallah de calculer la similarité entre la requête usager et les clusters des documents. Cela nous permettrait d'avoir un corpus qui sera utilisé par le système de recherche d'information pour le calcul de la similarité entre une nouvelle requête et les requêtes de la base grâce à la méthode proposée dans cette thèse ;
 3. Nous proposerons également une expansion de la requête de l'utilisateur pour réduire le bruit et le silence dont les SRI souffrent et par conséquent d'en augmenter la précision et le rappel.

- Nous avons remarqué également que le système de recommandation proposé, basé sur l'appréciation de l'utilisateur et leur comportement tirés à partir de la formule de Chan, est difficile à implémenter sur un système de recherche d'information web. Nous suggérons de faire appel à des outils et techniques de génie logiciel qui pourraient faciliter son implémentation.

V.4. Publications

Cette section présente chronologiquement les publications que nous avons réalisées pendant cette thèse :

- H.Moutachaouik, A.Marzak, H.Behja, H.Douzi and B.Ouhbi. Système de Recommandation pour améliorer le service de recherche d'information dans les plates formes Elearning:Application sur la plate forme E-learning Moodle. . In Proc. of the second edition of the International Conference on Next Generation Networks and Services (NGNS'10)., Marrakesh, Morocco, 8-10 July, 2010.
- H.Moutachaouik, A.Marzak, H.Behja, H.Douzi and B.Ouhbi. Recommendation system to improve service to search for information in e-learning platforms:Application on Elearning platform Moodle. In Proc. of the 2nd International Conference on Multimedia Computing and Systems (ICMCS'11), Ouarzazate, Morocco, 7-9 April, 2011.
- H.Moutachaouik, A.Marzak,H.Behja,H.Douziand B.Ouhbi. نظام توصية لتسهيل تعلم متعلمي منهاج التعلم الإلكتروني التطبيق على منصة التعلم عن بعد: مودل . In Proc. of The International Conference on Computing Science, in Arabic (ICCA11). Riad, Arabie saoudi, Mai 31-Juin 2, 2011.
- H.Moutachaouik, A.Marzak,H.Behja,H.Douziand B.Ouhbi. Recommendation Plugin to facilitate the learning of learners in e-learning platforms . In Proc. of The workshop on Information Technologies and Communication (WOTIC'11), Casablanca, Morocco, 13-15 Octobre, 2011.
- H.Moutachaouik, A.Marzak,H.Behja,H.Douziand B.Ouhbi. Recommendation Plugin to facilitate the learning of learners in e-learning platforms. . In Proc. of the third edition of the International Conference on Next Generation Networks and Services (NGNS'11), Hammamet, Tunisie, 18-20 Decembre 2011.
- H.Moutachaouik, A.Marzak,H.Behja,H.Douziand B.Ouhbi. Plugin of Recommendation based on a hybrid method for the ranking of documents in the e-learning platforms . In Proc. of the 5th International Conference on Image and Signal Processing (ICISP'12), Agadir, 28-30 Juin, 2012.
- H.Moutachaouik, A.Marzak,H.Behja,H.Douziand B.Ouhbi. Plugin of Recommendation based on a hybrid method for the ranking of documents in the e-learning platforms . In Springer Lecture Notes in Computer Science series (Springer LNCS), Volume 7340, 2012, DOI: 10.1007/978-3-642-31254-0,pp.587-595, © Springer-Verlag Berlin-Heidelberg 2012.

H.Moutachaouik, B.Ouhbi, H.Behja, B.frikh, A.Marzak and H.Douzi. New hybrid measure for the similarity between two queries in an information retrieval system. . In Proc. of the fourth edition of the International Conference on Next Generation Networks and Services (NGNS'12), Algarve, Portugal, 2-4 Decembre 2012.

H.Moutachaouik, B.Ouhbi, H.Behja, B.frikh, A.Marzak and H.Douzi. Hybrid Method for Information Retrieval Based on the Similarity between Queries. In International journal Review on Computers and Software (IRECOS), Vol. 7 N. 6, ISSN 1828-6003, November 2012.

BIBLIOGRAPHIE

- (Aiken et al., 1998) Aiken, R., Leng, P., Muhlbacher, J., Schauer, H., and Shave, M. (1998). Interactive Seminars Using the Web: An International Experience. In *Teleteaching, Distance Learning, Training and Education, Proceedings of the XV. IFIP World Computer Congress*, pages 869-875.
- (Al-Ajlan et al., 2008) A. Al-Ajlan, H.Zedan, "Why Moodle," *ftdcs*, pp.58-64, 2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, 2008.
- (Buell et al., 1981) Buell D., Kraft D., « Threshold values and Boolean retrieval systems », *Information Processing & Management*, vol. 17, p. 127-136, 1981.
- (Buell, 1982) Buell D., « An analysis of some fuzzy subset applications to information retrieval systems », *Fuzzy Sets & Systems*, vol. 7, p. 35-42, 1982.
- (Burke al., 1996) R. Burke, K. Hammond, and E. Cooper. Knowledge based navigation of complex information spaces. In *Proc. of the 13th National Conference on Artificial Intelligence (AAAI'96)*, pages 462–468, Menlo Park, Canada, 1996.
- (Brin et Page, 1998) Sergey Brin et Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7) :107–117, 1998.
- (Brun et al., 2002) A. Brun, K. Smaili, and J.-P. Haton (2002), WSIM: une méthode de détection de thèmes fondée sur la similarité entre mots, 9ème conf. fran. TALN'2002, Nancy, France.
- (Benayache, 2005) Benayache A., (2005). Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning: Le projet MEMORAE. thèse PHD, Université de technologie de Compiègne . Disponible sur <http://www.hds.utc.fr/%7Eabenayac/PhD/PhD-Ahcene.pdf>
- (Brini, 2005) BRINI.A, A Model for Information Retrieval based on Possibilistic Networks , *Proc. of the symposium on String Processing and Information REtrieval (SPIRE 2005)*, LNCS, Springer, 2005.
- (Brini, 2005) BRINI.A, Un Modèle de Recherche d'Information basé sur les Réseaux Possibilistes. Thèse de doctorat, Université Paul Sabatier, 07 Décembre 2005.
- (Berry, 2005) M. Berry, An investigation of the effectiveness of Moodle in primary education, in Deputy Head. 2005, Haslemere.

- (Ben Lahmer et al., 2006) Ben lahmar el habib, Abd Elaziz sdigui doukkali, " la recherche sur Internet : nouveau concept- nouveaux outils" in "The 4th ACS/IEEE International Conference on Computer Systems and Applications"(AICCSA- 06), Mars 2006, Dubai/Sharah, UAE.
- (Brun et al., 2010) A. Brun , A. Hamad, O. Buffet and A. Boyer . Vers l'utilisation de relations de préférence pour le filtrage collaboratif, Actes du dixseptième congrés francophone AFRIF-AFIA sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'10), Caen, France,2010.
- (Crof, 1987) CROF, a new approach to the design of document retrieval systems, University of Massachusetts Amherst, MA, USA, 1987.
- (Callan et al, 1992) Callan, J., Croft, W., and Harding, S. The INQUERY retrieval system. In Proc. of International Conference on Database and Expert Systems Applications (DEXA) (1992), pp. 78–83.
- (Calvert et al., 1997) Kenneth L. Calvert, Matthew B. Doar, et EllenW. Zegura. Modeling internet topology. IEEE Communications Magazine, 35(6) :160–163, June 1997.
- (Chavan et al, 2004) A. Chavan and S. Pavri, Open Source Learning Management in Moodle. linux journal, 2004, 1(2): p. 78-97.
- (Cheng-chao, 2005) Cheng-chao. Su. An Open Source Platform for Educators, in Proceedings of the Fifth IEEE Advanced Learning Technologies. 2005: IEEE Computer Society.
- (Cole et al, 2007) Cole, J. and H. Foster, Using Moodle: Teaching with the Popular Open Source Course Management System.2 ed. 2007: O'Reilly.
- (Caprio, 2010) Enrichissement de requêtes et visualisation semantique dans une coopération de systèmes d'information : méthodes et outils d'aide à la recherche d'information. Thèse de doctorat, Université de Bourgogne, Décembre 2010.
- (Dubois et Prade, 1988) Dubois D., and Prade H., "Possibility Theory", Plenum, New York (USA), 1988.
- (Dubois et Prade, 1998) Dubois D., and Prade H., "Possibility theory : qualitative and quantitative aspects", Dans : Quantified Representation of Uncertainty and Imprecision. Dov M. Gabbay, Philippe Smets (Eds.), KLUWER ACADEMIC PUBLISHERS, The Netherlands, p. 169-226, Vol. 1, Handbook of Defeasible Reasoning and Uncertainty Management Systems, 1998.
- (Dagan et al., 1999) I. Dagan, L. Lee, and F. C. N. Pereira (1999), Similarity-Based Models of Word cooccurrence Probabilities, Machine Learning, Vol 34, 43–69.
- (Dherent, 2002) Dherent C., (2002). Les Archives électroniques. Manuel pratique. Paris, Direction des Archives de France (2002) 104 p.
- (De Campos et al., 2002) De Campos L., Fernandez-Luna J., et Huete J., "A layered bayesian network model for document retrieval", In Proc. of the 24th BCS-IRSG European Colloquium on IR Research: Advances in Information Retrieval, pp. 169 – 182, 2002.
- (De Campos et al., 2003) De Campos L. M., Fernandez-Luna J. M., et Huete J. F., "The BNR Model: foundations and performance of Bayesian Network-based retrieval model", JASIST, 54(4): 302-313, 2003.

- (Desjardins, 2006) desjardins g., (2006). Modélisation connexionniste du repérage de l'information. These presentee comme exigence partielle du doctorat en informatique cognitive, universite du quebec a montreal, aout 2006. Disponible sur http://www.dic.dinfo.uqam.ca/etudiants/diplomes/desjardins_these
- (Dougiamas, 2004) M. Dougiamas, Moodle: Virtual learning environment for the rest of us. TESL-EJ, 2004. 8(2): p. 1-8.
- (Dougiamas, 2008) M. Dougiamas. Moodle. 2008, www.Moodle.org.
- (Djaanfar, 2011) A. S. Djaanfar, B. Frikh, and B. Ouhbi, (2012), A Hybrid Method for Domain Ontology Construction from the Web, KEOD 2011, pp. 285-292, Paris-France, 26-29 Octobre, 2011.
- (Elayeb, 2009) Elayeb B. , “ SARIPOD: Système multi-Agent de Recherche Intelligente POSSibiliste de Documents Web”, Thèse de doctorat en informatique, Université de Toulouse,Toulouse (France), 2009.
- (Fabiani, 1996) Fabiani P., “Représentation Dynamique de l’Incertain et stratégie de Prise d’Information pour un Système Autonome en Environnement Evolutif”, Thèse de Doctorat en Automatique et Informatique Industrielle, Ecole Nationale Supérieure de l’Aéronautique et de l’Espace, Toulouse, 1996.
- (frikh et al., 2011) B. Frikh, A.S. Djaanfar and B. Ouhbi (2011), A New Methodology For Domain Ontology Construction From The Web. International Journal on Artificial Intelligence Tools Vol. 20, No. 6 (2011) 1157–1170.
- (Goldberg et al., 1992) D. Goldberg, D. Nichols, B.M. Oki, and D. Terry.Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12) :61–70, 1992.
- (Gaul et Schmidt-Thieme, 2002) Gaul, W., Schmidt-Thieme, L. (2002), Recommender Systems Based on User Navigational Behavior in the Internet, Behaviormetrika, 29, 1-22.
- (Gandon, 2006) http://interstices.info/jcms/c_17672/ontologies-informatiques, 22/05/06, par Fabien Gandon
- (Galichet, 2007) Galichet, F. (2007). Concepts de base / ou l’enseignement en ligne. Vers une approche constructiviste de la formation à distance. In l’Enseignement en ligne, A l’université dans la formation professionnelle, Pourquoi ? Comment ? Manderscheid, J.-C. Jeunesse, C.,Collection Perspectives en éducation & formation, éd. De Boeck, 356p.
- (Hartley, 1928) Hartley, R.V.L., "Transmission of Information", Bell System Technical Journal, Volume 7, Number 3, pp. 535–563, (July 1928).
- (Hersh et al., 1994) Hersh, W. R., Buckley, C., Leone, T. J., and Hickam, D. H.OHSUMED: An interactive retrieval evaluation and newlarge test collection for research. Proceedings of the 17th Annual ACM SIGIR Conference, pages 192-201, 1994.
- (Hersh et al., 2001) Hersh WR, Hickam DH, Use of a multi-application computer workstation in a clinical setting, Bulletin of the Medical Library Association, 1994, 82: 382-389.
- (Itmazi, 2005) J. Itmazi, Flexible Learning Management System To Support Learning In The Traditional And Open Universities, 2005, Granada University, Spain.

- (Jaccard, 1901) Jaccard, P. (1901). Bulletin de la société vaudoise des sciences naturelles. 37, 241–272.
- Levenstein, A. (1966). Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10, 707–710.
- (Jon M. Kleinberg , 1999) J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5) : pages 604–632, September 1999.
- (Joye et al., 2003) Joye, F., Deschryver, N., & Peraya, D. (2003). Comment développer un campus virtuel ? Dans Charlier, B. & Peraya, D. (Ed.) Technologie et innovation en pédagogie : dispositifs innovants de formation pour l'enseignement supérieur, Perspectives en éducation et formation (p. 93-102), Bruxelles: De Boeck Université.
- (Luhn, 1957) Luhn H.P., (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. IBM Journal of Research and Development, 1, pp. 309-317. Disponible sur www.research.ibm.com/journal/rd/014/ibmrd0104D.pdf
- (Lukaswicz, 1963) Lukasiewicz, 1963 J. Lukasiewicz. Elements of Mathematical Logic. Pergamon Press, 1963.
- (Lackinger et al., 1984) Lackinger, H. and Muhlbacher, J. (1984). Lernen mit Bild-schirmtext: dezentral und zeitlich ungebunden. In Tagungsband "Mikroelektronik für den Menschen", page 731.
- (Lafferty et al, 2001) Lafferty J., Zhai C., «Document language models, query models, and risk minimization for information retrieval », Proceedings of the ACM SIGIR '01 conference, 2001, p.111-119.
- (Lewandowski, 2003) Lewandowski, J.-C. (2003). Regards croisés sur les nouvelles façons de former, le e-learning, enjeux et outils, Éditions d'Organisation, 28 p.
- (Langville et al., 2006) Langville A.N., Meyer C.D., (2006). Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, chapter 1. Disponible sur <http://press.princeton.edu/chapters/s8216.pdf>.
- (Lebrun, 2006) Lebrun, M. (2006). E-learning, Les TICE valeur ajoutée et métamorphose de la pédagogie. Séminaire de l'Institut de pédagogie universitaire et des multimédias (IPM), Université Catholique de Louvain, Belgique [en ligne] <http://eductice.inrp.fr/EducTice/projets/scenario/INRP-2009-Seminaire-lebrun.pdf>.
- (Li et al., 2008) Y. Li, C. Luo, and S. M. Chung (2008), "Text Clustering with Feature Selection by Using Statistical Data" Knowledge and Data Engineering, IEEE Transactions on Know and Data Eng., Vol. 20(5), 641–651.
- (Mooers, 1948) Mooers, C.N., Application of Random Codes to the Gathering of Statistical Information, MIT Master's Thesis, 1948.
- (Muhlbacher, 1998) Muhlbacher, J. (1998). Das Ende des Präsenzunterrichts? Österreichische Zeitschrift für Berufsbildung, pages 8-11.
- (Manning et Schütze, 1999) Christopher D. Manning and Hinrich Schütze. Foundations of statistical natural language processing. MIT Press, Cambridge, MA, USA, 1999.
- (Muhlbacher et al., 2002) Muhlbacher, J., Muhlbacher, S. C., and Reisinger, S. (2002). Learning Arrangements and Settings for Distance Teaching / Coaching /

Learning: Best Practice Report. In Hofer, C. and Chroust, G., editors, IDIMT - 2002: 10th Interdisciplinary Information Management Talks, pages 243-253. Universitätsverlag Rudolf Trauner.

- (Manning et al., 2008) C. D. Manning, P. Raghavan, and H. Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.
- (Maedche, 2002) A. Maedche, V. Pekar, and S. Staab (2002), Ontology learning part one-on discovering taxonomic relations from the web. In N. Zhong, J. Liu, and Y. Yao, (eds.), Web Intelligence (Springer Verlag).
- (Middleton et al., 2004) Middleton S. E., Shadbolt N. R., De Roure D. C., « Ontological user profiling in recommender systems », ACM Transactions on Information Systems (TOIS), vol. 22, n° 1, ACM Press, 2004, p. 54-88.
- (Moreau, 2006) Moreau F., (2006). Revisiter le couplage traitement automatique des langues et recherche d'information. Thèse de doctorat, l'université de Rennes 1, décembre 2006. Disponible sur <http://www.irisa.fr/texmex/people/moreau/publications/these.pdf>
- (Nallapati et al., 2004) Nallapati, R. Discriminative models for information retrieval. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, July 25-29, 2004, Sheffield, United Kingdom.
- (Nie, 2007) Nie J.Y., (2007). Le domaine de recherche d'information – Un survol d'une longue histoire. Support de cours Recherche d'Information, Département d'informatique et recherche opérationnelle, Université de Montréal, Hiver 2007. Disponible sur <http://www.iro.umontreal.ca/%7Enie/IFT6255/historique-RI.pdf>.
- (Porter, 1980) Porter M. F., “An algorithm for suffix stripping”, Program, 14(13): 130-137, 1980.
- (Pearl, 1988) Pearl J., “Probabilistic reasoning in intelligent systems : Networks of plausible Inference”, Morgan Kaufman Publishers, Inc., San Mateo, CA, 2nd Edition, 1988.
- (Paris et al., 2001) Paris, C. and Gespass, S., Examining the Mismatch Between Learner-Centered Teaching and Teacher-Centered Supervision. Journal of Teacher Education, 52(5):398-412.
- (Piwowarski, 2003) Piwowarski B., Techniques d'apprentissage pour le traitement d'informations structurées : application à la recherche d'information. Thèse de doctorat, Université Paris 6, 17 Juillet 2003. Disponible sur www.connex.lip6.fr/download_article/695.pdf.
- (Picarougne, 2004) Picarougne F., (2004). Recherche d'information sur Internet par algorithmes évolutionnaires. Thèse de doctorat, Université François Rabelais Tours, novembre 2004. Disponible sur www.antsearch.univ-tours.fr/publi/picarougne04these.pdf
- (Pazzani et al., 2007) M. Pazzani and D. Billsus. The Adaptive Web, chapter Content-Based Recommendation Systems, pages 325–341. Springer Berlin Heidelberg, 2007.
- (Kendal et al., 2006). Kendal S.L., Creen M., (2006). An Introduction to Knowledge Engineering, Springer, 1 edition, ISBN: 1846284759, 290 pages.

- (Kim et al., 1987) Kim J. H., et Pearl J., “CONVINCE: A Conversational Inference Consolidation Engine”, In IEEE Trans. on Systems, Man and Cybernetics, vol. 17, pp. 120-132, 1987.
- (Quinlan, 1986) J. R. Quinlan, “Induction of Decision Trees,” Machine Learning, vol. 1, pp. 81–106, 1986.
- (Qin et al., 2007) Qin, T., Liu, T. Y., Lai, W., Zhang, X. D., Wang, D. S., and Li, H. Ranking with Multiple Hyperplanes. Proceedings of the 30th Annual International ACM SIGIR Conference, 2007. to appear.
- (Rocchio, 1965) J. J. Rocchio, ‘Relevance Feedback in Information Retrieval’, Harvard University, ISR-9, 1965.
- (Robertson , 1976) Robertson, S., and Jones, K. S. Relevance weighting of search terms. Journal of the American Society for Information Science (JASIS) 27, 3(1976), 129–146.
- (Rijsbergen, 1979) C. J. van Rijsbergen, Information Retrieval, 2Rev Ed. Butterworth-Heinemann Ltd, 1979.
- (Robertson , 1994) Robertson, S., and Walker, S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In Proc.of the International ACM-SIGIR Conference (1994), pp. 232–241.
- (Ribeiro-Neto et al., 1996) Ribeiro-Neto B., Silva I., et Muntz R., “A Belief Network Model for IR”, Proc. of the 19th ACM-SIGIR Conf. on Research and Development in Information Retrieval, 253-260, 1996.
- (Resnik, 1999) P. Resnik (1999), Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research, 11(1):95–130
- (Razan, 2004) RAZAN TAHER, Recherche d’Information Collaborative, Communication de congrès (Toulouse-France), Vol 2, 2004.
- (Shannon, 1948) Claude E. Shannon: A Mathematical Theory of Communication, Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, 1948.
- (Switzer et al., 1963) P. Switzer, ‘Vector Images in Document Retrieval’, Harvard University, ISR-4, 01 1963.
- (Stevens, 1964) M. E. Stevens, V. E. Giuliano, and L. B. Heilprin, Statistical association methods for mechanized documentation: symposium proceedings. Washington, DC: G.P.O., 1964.
- (Salton, 1968) G. Salton, Automatic Information Organization and Retrieval. McGraw Hill Text, 1968.
- (Salton, 1971) G. SALTON, The SMART Retrieval System : Experiments in Automatic Document Processing, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1971.
- (Salton, 1975) G. Salton, A. Wong, and C. S. Yang, ‘A vector space model for automatic indexing’, Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.

- (Schutz et al., 1973) Schutz, A. and Luckmann, T. Structures of the Life World. Northwestern University Press, Evanston, Ill., Ed. ACM, New York, Sept. 1990, p. 45-61, 1973.
- (Salton et al., 1983) Gerard Salton, Edward A. Fox, et Harry Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11) :1022–1036, 1983.
- (Salton, 1990) G. SALTON, On the application of syntactic methodologies in automatic text analysis, *Information Processing and Management*, Pergamon Press, Inc. Tarrytown, NY, USA , Vol 26 , N° 1, 1990.
- (Savoy, 1994) SAVOY J, A learning scheme for information retrieval in Hypertext. *Information Processing & Management*, CAT.INIST, vol. 30, no4, 1994.
- (S. Brin et al., 1998) S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*,30 :107-117, 1998.
- (Silva et al., 2000) Silva I., Ribeiro-Neto B., Calado P., Moura E., et Ziviani N., “Link-Based and Content-Based Evidential Information in a Belief Network Model”, *ACM/SIGIR 23rd Int. Conference on Information Retrieval*, pp. 96–103, 2000.
- (Strehl, 2002) Strehl, A. (2002). Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining. PhD thesis, University of Texas at Austin.
- (Shearer, 2003) S. Shearer, *Open Source Software in Education*. 2003, The Compton School: London.
- (Sanchez et al., 2004) D. Sanchez and A. Moreno (2004), *Creating ontologies from Web documents*, *Recent Advances in Artificial Intelligence Research and Development*, IOS Press Vol. 113,11–18.
- (Turtle et al., 1990) Turtle H. R., et Croft W. B., “Inference networks for document retrieval”, In *Proc. 13th International Conference on Research and Development in Information Retrieval*, 1–24, 1990.
- (Turtle, 1991) Turtle H. R., “Inference networks for document retrieval”, Ph.D. thesis, University of Massachusetts, USA, 1991.
- (Turtle et al., 1991) Turtle H. R., et Croft W. B., “Evaluation of an inference network-based retrieval model”, In *ACM Transaction on Information System*, 9(3) : 187–222, 1991.
- (Tezenas du Montcel, 1997) Tézenas du Montcel, H. (1997). *Investissement immatériel*. *Encyclopédie de gestion*, Economica, Coordonnée par Yves Simon et Patrick Joffre, 2ème édition, 1721-1725.
- (Tortora , 2002) G. Tortora, et al., A multilevel learning management system, in *Proceedings of the 14th international conference on Software and knowledge engineering*.2002, ACM: Ischia, Italy.
- (Van Rijsbergen, 1979) C. J. Van Rijsbergen. *Information Retrieval*, 2nd edition. Dept.of Computer Science, University of Glasgow, 1979.
- (Voorhees, 2003) Voorhees E.M., (2003). Overview of TREC 2002. NIST Special Publication: SP 500-251, The Eleventh Text Retrieval Conference (TREC 2002), February 2003. Disponible sur http://trec.nist.gov/pubs/trec11/t11_proceedings.html.

- (Waller et al., 1979) Waller W., Kraft D., « A mathematical model of a weighted Boolean retrieval system », *Information Processing & Management*, vol. 15, p. 235-245, 1979.
- (Weimer, 2002) Weimer, M. (2002). *Learner-Centered Teaching. Five Key Changes to Practice*. Jossey-Bass, John Wiley & Sons.
- (Williams et al., 2005) B. Williams and M. Dougiamas, *Moodle for Teachers, Trainers and Administrators of Remote-Learner.net.2005*, Moodle.org
- (Zadeh, 1978) Zadeh L. A., "Fuzzy Sets as a basis for a theory of Possibility", *Fuzzy Sets and Systems*, Vol. 1, pp. 3-28, 1978.
- (Zhai et al, 2001) Zhai C., Lafferty J., A study of smoothing methods for language models applied to ad hoc information retrieval, *Proceedings of the ACM SIGIR'01 conference*,2001,p.334-342.
- (Ziegler, 2005) Ziegler, Cai-Nicolas. 2005. «Towards Decentralized Recommender Systems». Albel1Ludwigs-Universitat Freiburg -Fakultat fur Angewandte Wissenschaften, Institut fur Informatik.
- (Zenha-Rela et al, 2006) M. Zenha-Rela and R. Carvalho. *Work in Progress: Self Evaluation Through Monitored Peer Review Using the Moodle Platform*. in *Frontiers in Education Conference, 36th Annual*. 2006. San Diego, CA: IEEE.
- (Zaier, 2010) Z. zaier, these : modèle multi-agents pour le filtrage collaboratif de l'information,JANVIER 2010.

LES ANNEXES :

Annexe 1 : Calcul de la similarité hybride

```
-----  
-- Auteur hicham Moutachaouik  
-- Plate forme Eclipse Platform  
--  
-----  
  
-----  
-- La classe de la similarité basée sur le chir statistique  
-----  
package util;  
import java.io.* ;  
import java.util.* ;  
import lecture.Clavier;  
  
public class Chire  
{  
    private ArrayList <Double> ere = new ArrayList <Double> () ;  
    private ArrayList <ArrayList <Integer>> calc = new ArrayList  
<ArrayList <Integer>> () ;  
    private static ArrayList <Integer> nv = new ArrayList <Integer> ()  
;  
    private String req ;  
  
    static  
    {  
        try  
        {  
            BufferedReader entree;  
            BufferedReader entreeC ;  
            entree = new BufferedReader(new FileReader ("basReq.txt"));  
            int cpt = 0 ;  
            while (entree.readLine() != null)  
            {  
                cpt++ ;  
                entreeC = new BufferedReader(new  
FileReader("Clusters_avec\\C_" + cpt + "_avec.txt")) ;  
                int cpt1 = 0 ;
```

```

        while (entreeC.readLine() != null)
        {
            cpt1++ ;
        }
        nv.add(cpt1) ;
        entreeC.close() ;
    }
    entree.close() ;
}
catch (Exception e)
{
    System.out.println("") ;
}
}

public Chire (String req) throws IOException
{
    this.req = req ;
    BufferedReader entree = new BufferedReader(new FileReader
("basReq.txt")) ;
    int cpt = 0 ;
    StringTokenizer tok ;
    while (entree.readLine() != null)
    {
        cpt++ ;
        calc.add(new ArrayList<Integer>()) ;
        tok = new StringTokenizer (req , " ") ;
        while (tok.hasMoreTokens())
        {
            calc.get(cpt-1).add(new Integer (compt (tok.nextToken() ,
cpt))) ;
        }

    }

    entree.close() ;
}

public int compt (String mot , int num) throws IOException
{
    BufferedReader entree = new BufferedReader ( new FileReader
("Clusters_avec\\C_" + num + "_avec.txt")) ;
    String lg ;
    int cpt = 0 ;
    do
    {
        lg = entree.readLine() ;
        if (lg != null)

```

```

        {
            if (lg.contains(mot))
            {
                cpt++ ;
                continue ;
            }
        }
    } while (lg != null) ;

    entree.close() ;
    return cpt ;
}

public static int nv()
{
    int snv=0;
    for (int i=0 ; i < nv.size() ; i++)
        snv += nv.get(i);
    return snv;
}

// ***** somme des colonnes *****

public int som(int j)
{
    int som=0;
    for (int i=0 ; i < nv.size() ; i++)
        som += calc.get(i).get(j);
    return som;
}

// ***** l'essentiel *****

public double sere (int rang_clus)
{
    double sere=0;

    StringTokenizer tokReq = new StringTokenizer (req , " ") ;

    for(int j=0 ; tokReq.hasMoreTokens() ; j++)
    {

        double ehhe=0 ;
        double tet=0 ;
        int teta [][]= new int [2][] ;
        teta [0] = new int [2] ;
    }
}

```

```

        teta [1] = new int [2] ;
        teta [0][0] = ((calc.get(rang_clus)).get(j)) ; //
nv.get(rang_clus) ;
        teta [0][1] = (som(j)-(calc.get(rang_clus)).get(j)); //
(nv()-nv.get(rang_clus)) ;
        teta [1][0] = nv.get(rang_clus) - teta [0][0] ;
        teta [1][1] = nv() - nv.get(rang_clus) - teta[0][1] ;

        int total = teta [0][0] +teta [0][1] +teta [1][0] +teta
[1][1];
        //System.out.println(total);

        tet = teta [0][0];
        ehhe = (double)(teta [0][0] + teta [1][0]) * (teta [0][0] +
teta [0][1])/total;

        ere.add( tet / ehhe) ;
        if (ere.get(j) > 1)
            sere += ere.get(j) ;

        tokReq.nextToken() ;

    }
    //System.out.println(ere);

return sere;
}

    public double chir(int rang_clus)
{
    double r=0;

    double somR = sere (rang_clus) ;

    StringTokenizer tokReq1 = new StringTokenizer (req , " " ) ;

    for(int j=0 ; tokReq1.hasMoreTokens() ; j++)
    {
        if (ere.get(j) > 1)
        {
            double x2=0;
            double p=0;

            int teta [][] = new int [2][2] ;
            teta [0] = new int [2] ;

```

```

        teta [1] = new int [2] ;
        //System.out.println("rang_clus = " + rang_clus + "
nv.get(rang_clus) = " + nv.get(rang_clus) +
"(calc.get(rang_clus)).get(j) = " + (calc.get(rang_clus)).get(j) + "
mot = " + mot) ;
        teta [0][0] = ((calc.get(rang_clus)).get(j)) ; //
nv.get(rang_clus) ;
        teta [0][1] = (som(j)-(calc.get(rang_clus)).get(j)); //
(nv()-nv.get(rang_clus)) ;
        teta [1][0] = nv.get(rang_clus) - teta [0][0] ;
        teta [1][1] = nv() - nv.get(rang_clus) - teta[0][1] ;

        int total = teta [0][0] +teta [0][1] +teta [1][0] +teta
[1][1];

        double ehee [][] = new double [2][2];
        ehee [0] = new double [2] ;
        ehee [1] = new double [2] ;
        ehee [0][0] = (double)(teta [0][0] + teta [1][0]) * (teta
[0][0] + teta [0][1])/total;
        ehee [0][1] = (double)(teta [0][1] + teta [1][1]) * (teta
[0][0] + teta [0][1])/total;
        ehee [1][0] = (double)(teta [0][0] + teta [1][0]) * (teta
[1][0] + teta [1][1])/total;
        ehee [1][1] = (double)(teta [1][0] + teta [1][1]) * (teta
[1][1] + teta [0][1])/total;

//System.out.println(ehee [0][0]+" "+ehee [0][1]+" "+ehee [1][0]+"
"+ehee [1][1]);

        double exxe [][] = new double [2][2];
        exxe [0] = new double [2] ;
        exxe [1] = new double [2] ;
        exxe [0][0] = (double)( ( (teta [0][0]-ehee [0][0]) *
(teta [0][0]-ehee [0][0]) ) / ehee [0][0] ) ;
        exxe [0][1] = (double)( ( (teta [0][1]-ehee [0][1]) *
(teta [0][1]-ehee [0][1]) ) / ehee [0][1] ) ;
        exxe [1][0] = (double)( ( (teta [1][0]-ehee [1][0]) *
(teta [1][0]-ehee [1][0]) ) / ehee [1][0] ) ;
        exxe [1][1] = (double)( ( (teta [1][1]-ehee [1][1]) *
(teta [1][1]-ehee [1][1]) ) / ehee [1][1] ) ;

        x2 = exxe [0][0] + exxe [0][1] + exxe [1][0] + exxe
[1][1] ;
        p = ere.get(j) / somR ;
        r += p * x2 ;

    }

```

```

        tokReq1.nextToken() ;
    }

    ere.clear();

    return r;
}
/*
public static void main (String args[])throws IOException
{
    System.out.println(new Date());
    //simQ() ;
    System.out.println("Veuillez saisir votre requête : ") ;
    String requete = Clavier.lireString() ;
    System.out.println("requete = " + requete) ;
    Chire A = new Chire(requete) ;
    BufferedReader entree = new BufferedReader ( new FileReader
("basReq.txt")) ;
    PrintWriter sortie = new PrintWriter(new
FileWriter("reponse_chire.txt")) ;
    String lg ;
    int cpt = 0 ;
    double res ;
    do
    {
        lg = entree.readLine() ;
        if (lg != null)
        {
            res = A.chir(cpt) ;
            System.out.println("chir " + (cpt+1) + " = " + res) ;
            sortie.println((cpt+1) + " " + res);
            cpt++;
        }
    } while (lg != null) ;
    System.out.println(new Date());
    entree.close() ;
    sortie.close() ;
}
*/
}

-- =====
-- fin de la classe de la similarité basée sur le chir statistique
-- =====

```



```

-- =====
-- la classe de la similarité semantique basé sur IM
-- =====

package util;
import java.io.*;
import java.util.*;
import lecture.* ;

public class Sim
{
    ArrayList<String> tab = new ArrayList<String>() ;
    ArrayList<String> tabd = new ArrayList<String>() ;
    public ArrayList<String> t = new ArrayList<String>() ;
    public ArrayList<String> tt = new ArrayList<String>() ;
    //private static long nbMots = 0; // nombre total de mots
    private static int d = 4 ; // fenêtre
    private double som2SimQ ;
    private String req ;
    HashMap <Long, Double> m = new HashMap<Long, Double>() ;
    private HashMap <String, Double> p = new HashMap<String, Double>()
;
    private boolean premFois = true ;
    static int t_lgt ;
    static int tt_lgt ;

    public Sim (String req) throws IOException
    {
        this.req = req ;
        double som = 0. ;
        double x ;
        long cpt = 0 ;

        StringTokenizer tokReq = new StringTokenizer(req , " ") ;
        while (tokReq.hasMoreTokens())
        {
            tab.add(tokReq.nextToken()) ;
        }
        BufferedReader entreeBas = new BufferedReader (new FileReader
("basReq.txt")) ;
        String lgBas ;
        int i ;
        do
        {
            lgBas = entreeBas.readLine() ;
            if (lgBas != null)
            {
                //***** Remplissage des tableaux t, tt et p *****

```

```

        BufferedReader entree = null;
        BufferedReader entre = null;
        entree = new BufferedReader (new
FileReader("Clusters_avec\\C_"+ (cpt + 1) + "_avec.txt"));
        entre = new BufferedReader (new
FileReader("Clusters_sans\\C_"+ (cpt + 1) + "_sans.txt"));
        String lg = null ;
        //int i = 0 ;

do
{
    lg = entree.readLine() ;
    if ( lg != null)
    {
        StringTokenizer tok = new StringTokenizer(lg , " ") ;
        while (tok.hasMoreTokens())
        {
            t.add(tok.nextToken()) ;
        }
    }

} while (lg != null) ;

entree.close() ;

/*for (int j = 0 ; j < t.size() ; j++)
{
    System.out.println("t["+ j +"]=" + t.get(j));

}*/
t_lgt = t.size();
//System.out.println(t_lgt);

do
{
    lg = entre.readLine() ;
    if ( lg != null)
    {
        StringTokenizer tok = new StringTokenizer(lg , " ") ;
        while (tok.hasMoreTokens())
        {
            tt.add(tok.nextToken()) ;
        }
    }
} while (lg != null) ;
entre.close() ;

```

```

        /*for (int j = 0 ; j < tt_lgt ; j++)
        {
            System.out.println("tt["+ j +"]="+ tt.get(j));

        }*/
        tt_lgt = tt.size();
        //System.out.println(tt_lgt);

        for (int j = 0 ; j < tt_lgt ; j++)
        {
            p.put(tt.get(j), pr(tt.get(j))) ;

        }

        // ***** Fin Remplissage ***** \\

premfOis = true ;
        for (i = 0, som = 0. ; i < tab.size() ; i++)
        {
            x = simQ(tab.get(i) , lgBas) ;
            som += x ;
        }
        m.put(cpt, som) ;
        cpt++ ;
    }
    } while (lgBas != null) ;
    entreeBas.close() ;
    System.out.println("fin constructeur") ;

}

public double pr (String m) throws IOException // p(m)
{
    int cpt=0, i ;
    for (i = 0; i < t_lgt ; i++)
    {
        if (t.get(i).equalsIgnoreCase(m))
        {
            cpt++ ;
        }
    }
    return ((double)cpt / t_lgt) ;

}

public double pD (String zi , String q)

```

```

{
    int k = 0 ;
    int f = 0 ;
    for (k=0 ; k < t_lgt ; k++)
    {
        if ( k < t_lgt-4)
        {
            if ( t.get(k).equalsIgnoreCase(zi))
            {
                if ( t.get(k+1).equalsIgnoreCase(q) ||
t.get(k+2).equalsIgnoreCase(q) || t.get(k+3).equalsIgnoreCase(q) ||
t.get(k+4).equalsIgnoreCase(q))
                {
                    f++;
                }
            }
            continue;
        }

        if ( k == t_lgt-4)
        {
            if ( t.get(k).equalsIgnoreCase(zi) )
            {
                if ( t.get(k+1).equalsIgnoreCase(q) ||
t.get(k+2).equalsIgnoreCase(q) || t.get(k+3).equalsIgnoreCase(q))
                {
                    f++;
                }
            }
            continue;
        }

        if ( k == t_lgt-3)
        {
            if ( t.get(k).equalsIgnoreCase(zi) )
            {
                if (t.get(k+1).equalsIgnoreCase(q) ||
t.get(k+2).equalsIgnoreCase(q) )
                {
                    f++;
                }
            }
            continue;
        }

        if ( k == t_lgt-2)
        {
            if ( t.get(k).equalsIgnoreCase(zi) &&
t.get(k+1).equalsIgnoreCase(q) )

```

```

        {
            f++;
        }

    }
    break;

}
return ((double)f / tt_lgt) ;
}

public int nbMotsDoc (String doc)
{
    int cpt=0 ;
    StringTokenizer tok = new StringTokenizer(doc, " ") ;
    cpt = tok.countTokens() ;

    return cpt;
}

//****
public double i(String zi , String q) throws IOException // I(zi, q)
{
    double res ;
    double pd = pD(zi , q) ;
    if (pd == 0.)
        res = 0. ;
    else
        res = pd * Math.log10 (pd / (d*d * p.get(zi) * p.get(q))) ;
    return res ;
}

// **** public double similarite (String q , String yk) throws
IOException
{
    ss_Simiarité0 A = new ss_Simiarité0(q , yk , 0, tt_lgt / 4,
this) ;
    ss_Simiarité0 B = new ss_Simiarité0(q , yk , tt_lgt / 4, 2 *
tt_lgt / 4, this) ;
    ss_Simiarité0 C = new ss_Simiarité0(q , yk , 2 * tt_lgt / 4, 3 *
tt_lgt / 4, this) ;
    ss_Simiarité0 D = new ss_Simiarité0(q , yk , 3 * tt_lgt / 4,
tt_lgt, this) ;
    A.start() ;
    B.start() ;
    C.start() ;
    D.start() ;
    try {

```

```

        Thread.sleep(200) ;
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    double som1 = A.getSom() ;
    double som2 = B.getSom() ;
    double som3 = C.getSom() ;
    double som4 = D.getSom() ;
    double som = (som1 + som2 + som3 + som4) / (2 * tt_lgt) ;

    return som ;
}

//*****

public double pbQdsD ( String q , String doc) // prob (q /d)
{
    double res ;
    int cpt =0 ;
    StringTokenizer tok = new StringTokenizer(doc, " ") ;
    String mot ;
    while (tok.hasMoreTokens())
    {
        mot = tok.nextToken() ;
        if (mot.equalsIgnoreCase(q))
            cpt++ ;
    }
    res = ((double)cpt / nbMotsDoc(doc)) ;
    return res ;
}

public double simQ (String q, String doc) throws IOException //
sim (q , d)
{
    double x ;
    //PrintWriter sortie = new PrintWriter ( new FileWriter ("sim
bas.txt")) ;
    double somme = 0. ;

    double som = 0. , pd;
    pd = pbQdsD (q , doc) ;
    //System.out.println("pd = " + pd) ;
    if (pd == 0.)
    {
        return 0. ;
    }
    else

```

```

{
StringTokenizer tok = new StringTokenizer(doc, " ") ;
String yk ;
while (tok.hasMoreTokens())
{
    yk = tok.nextToken() ;
    som += similarite(q, yk) ;

}

if (som == 0.)
{
    return 0. ;
}
else
{
    if (premFois)
    {

        int i, j ;
        for (i = 0 ; i < tt_lgt ; i++)
        {
            for (j =i ; j < tt_lgt ; j++)
            {
                x = similarite(tt.get(i), tt.get(j)) ;
                somme += 2 * x ;
                //System.out.println("sim (" + (i+1) + " , " + (j+1)
+ ") = " + x ) ;
                //System.out.println(new Date());
            }
            somme -= similarite(tt.get(i), tt.get(i)) ;
        }
        //System.out.println("somme bas" + somme);
        //sortie.print(somme) ;
        //sortie.close() ;
        som2SimQ = somme ;
        premFois = false ;
    }
    if (som2SimQ == 0.)
    {
        return 0. ;
    }
    else
    {
        return (pd * som / som2SimQ) ;
    }
}
}

```

```

    }
}

//*****

public int compt (String rep)
{
    int abc;
    int j;
    tabd.clear();
    int cpt = 0;
    StringTokenizer tokRep = new StringTokenizer(rep, " ") ;
    while (tokRep.hasMoreTokens())
    {
        tabd.add(tokRep.nextToken());
    }
    for(abc=0 ; abc < tab.size() ; abc++)
    {for(j=0 ; j < tabd.size() ; j++){

        if (tab.get(abc).equalsIgnoreCase(tabd.get(j)))
        {
            cpt++ ;
            break ;
        }
    }
    }

    return cpt ;
}

//*****

public double simD (String doc , long rang_doc ) throws
IOException
{
    double res ;
    double s;
    int somSigmaD ;
    premFois = true ;
    som2SimQ = 0. ;

    if ((s = m.get(rang_doc)) == 0. || (somSigmaD = compt(doc)) == 0
)
    {
        res = 0. ;
    }
    else

```



```

    {
        res = (s * somSigmaD ) ;
    }

    return res ;
}
/*
public static void main (String args [])throws IOException
{
    System.out.println(new Date());
    //simQ() ;
    System.out.println("Veuillez saisir votre requête : ") ;
    String requete = Clavier.lireString() ;
    Sim A = new Sim(requete) ;
    BufferedReader entree = new BufferedReader ( new FileReader
("basReq.txt")) ;
    PrintWriter sortie = new PrintWriter(new
FileWriter("reponse_sim.txt")) ;
    String lg ;
    int cpt = 0 ;
    double res ;
    do
    {
        lg = entree.readLine() ;
        if (lg != null)
        {
            res = A.simD(lg, cpt) ;
            System.out.println("sim " + (cpt+1) + " = " + res) ;
            sortie.println((cpt+1) + " " + res);
            cpt++ ;
        }
    } while (lg != null) ;
    System.out.println(new Date());
    entree.close() ;
    sortie.close() ;
}
*/
}

```

```

class ss_Simiarité0 extends Thread
{
    private String q ;
    private String yk ;
    private int debut ;
    private int fin ;
    private Sim objet ;
    public double som = 0. ;
}

```

```

public ss_Simiarité0 (String q , String yk , int debut , int fin,
Sim objet)
{
    this.q = q ;
    this.yk = yk ;
    this.debut = debut ;
    this.fin = fin ;
    this.objet = objet ;
}
public void run ()
{
    synchronized (this)
    {

double mi1 = 0 , mi2 = 0 , ma1 = 0 , ma2 = 0 ;
double i1 , i2 , i3 , i4 ;

String mot ;
int ii ;
for (ii = debut ; ii < fin ; ii++)
{
    mot = objet.tt.get(ii) ;
    try {
        if ((i1 = objet.i(mot , q)) == 0. || (i2 = objet.i(mot ,
yk)) == 0. )
        {
            mi1 = 0. ;
            ma1 = 1. ;
        }
        else
        {
            mi1 = Math.min (i1 , i2 ) ;
            ma1 = Math.max (i1 , i2 ) ;
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        if ((i3 = objet.i(q , mot)) == 0. || (i4 = objet.i(yk ,
mot)) == 0. )
        {
            mi2 = 0. ;
            ma2 = 1. ;
        }
    }
}
}

```

```

        else
        {
            mi2 = Math.min (i3 , i4);
            ma2 = Math.max (i3 , i4);
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    som += ( (mi1 / ma1) + (mi2 / ma2) );

}

}

}

public synchronized double getSom ()
{
    return som ;
}
}

-- =====
-- fin la classe de la similarité semantique basé sur IM
-- =====

-- =====
-- La classe qui calcul la similarité hybride
-- =====

package util;
import java.io.* ;
import java.util.* ;
import lecture.Clavier;

public class TstSimChir
{
    public static void main(String[] args) throws IOException
    {
        System.out.println(new Date());
        //simQ() ;
        System.out.println("Veuillez saisir votre requête : ") ;
        String requete = Clavier.lireString() ;
        Sim A = new Sim(requete) ;
        Chire B = new Chire(requete) ;
        BufferedReader entree = new BufferedReader ( new FileReader
("basReq.txt")) ;
        PrintWriter sortie = new PrintWriter(new
FileWriter("reponse.txt")) ;

```

```

String lg ;
int cpt = 0 ;
double resSim ;
double resChir ;
do
{
    lg = entree.readLine() ;
    if (lg != null)
    {
        resSim = A.simD(lg, cpt) ;
        resChir = B.chir(cpt) ;
        System.out.println((cpt+1) + " sim = " + resSim + " ; chire
= " + resChir + " ; hybride = " + (resSim * 0.5 + resChir * 0.5)) ;
        sortie.println((cpt+1) + " sim = " + resSim + " ; chire = "
+ resChir + " ; hybride = " + (resSim * 0.5 + resChir * 0.5));
        cpt++ ;
    }
} while (lg != null) ;
System.out.println(new Date());
entree.close() ;
sortie.close() ;
}

}

-- =====
-- fin de la classe qui calcul la similarité hybride
-- =====

```

Annexe 2 : La similarité selon la divergence de Kullbak-leiber

```

-- =====
-- La classe qui calcul la divergence de Kullbak-leiber
-- =====
import java.util.*;
import java.io.*;
public class dive
{
    private static int compt = 1 ;
    private static final int N = 64;
    private ArrayList <ArrayList <Double>> tab = new ArrayList
<ArrayList <Double>> () ;
    private ArrayList <ArrayList <Double>> calc = new ArrayList
<ArrayList <Double>> () ;
    int nv [] = new int [64];
    public int nn;
    public dive (String lgReq) throws IOException
    {String newReq ="" ;
    StringTokenizer tokReq ;

```

```

tokReq = new StringTokenizer(lgReq, " ") ;
String mot, mot2 ;
while (tokReq.hasMoreTokens())
{
    mot = tokReq.nextToken() ;
    if (newReq.indexOf(mot) == -1)
    {
        newReq = newReq + mot + " " ;
    }
}
int i = 0 , cpt =0;
BufferedReader entree2 ;
entree2 = new BufferedReader (new FileReader ("bas.txt")) ;
String lgDoc = null ;
do
{
    lgDoc = entree2.readLine() ;

    if (lgDoc != null)
    {

        calc.add(new ArrayList <Double> ()) ;
        tokReq = new StringTokenizer (newReq, " ") ;
        while (tokReq.hasMoreTokens())
        {
            cpt = 0 ;
            String motReq = tokReq.nextToken() ;
            StringTokenizer tokDoc = new StringTokenizer(lgDoc , " ") ;
            nv[i]=tokDoc.countTokens();
            while (tokDoc.hasMoreTokens())
            {
                if (motReq.equalsIgnoreCase(tokDoc.nextToken()))
                {
                    cpt++ ;
                }
            }
            (calc.get(i)).add(new Double ((double)cpt/nv[i])) ;

        }
        i++ ;
    }
} while (lgDoc != null) ;

calc.add(new ArrayList<Double> ()) ;
StringTokenizer tokNewReq = new StringTokenizer(newReq, " ") ;
int nbmo = tokNewReq.countTokens() ;
nn = nbmo;
while (tokNewReq.hasMoreTokens())

```

```

{
    cpt = 0 ;
    mot = tokNewReq.nextToken() ;
    tokReq = new StringTokenizer(lgReq, " ") ;

    while (tokReq.hasMoreTokens())
    {
        mot2 = tokReq.nextToken() ;
        if (mot.equalsIgnoreCase(mot2))
        {
            cpt++ ;
        }
    }

    (calc.get(i)).add(new Double ((double) cpt/nbmo)) ;

}
System.out.println(calc.get(64));

entree2.close() ;

    for(int ii=0;ii<N;ii++){
        tab.add(new ArrayList<Double>());
        for(int j=0;j<nbmo;j++){
            tab.get(ii).add(calcWfinal((calc.get(ii)).get(j),j));
        }
        tab.add(new ArrayList<Double>());
        for(int iii=0;iii<nbmo;iii++){
            tab.get(N).add((new Double (calc.get(N).get(iii))));
        }

        //System.out.println(tab.size()) ;
    }

public int calcni(int j){
    int cpt=0;
    for(int i=0 ; i<N ; i++){
        if ((calc.get(i)).get(j) != 0)
            cpt++;}
    return cpt;}

public double calcQ2 (){
    double somQ=0.;
    for(int i=0;i<nn;i++){
        somQ += ((tab.get(N).get(i))*(tab.get(N).get(i)));
    }
    return somQ;}

```

```

public double calcW2 (int a){
    double somD=0;
    for(int j=0;j<nn;j++){
        somD += ((tab.get(a).get(j))*(tab.get(a).get(j)));
    }
return somD;}

public void vect () throws IOException
{

    PrintWriter s =new PrintWriter(new FileWriter("dive_req_" +
compt + ".txt")) ;

    double b;
    int i ;

    for( i = 0 ; i < N ; i++)

    {b = 0.;
    for(int j=0;j<nn;j++)
    {
        b +=(tab.get(N).get(j))*(tab.get(i).get(j));
    }

    s.println((i+1) +" "+ b) ;

    }

    compt++ ;
s.close();}

public double calcndfi(int j){
double c;
int cc=calcni(j);
if (cc==0) return 0;
else
c=Math.log10(N/cc);
return c;
}

public double calcWfinal(double tf , int j){
if (tf==0) return 0;
else
return ((Math.log(tf)));
}

```

```

    public static void main (String [] args) throws IOException
    {

        dive A ;
        BufferedReader entree = new BufferedReader(new FileReader
("reqnew.txt")) ;
        String lg ;

        do
        {
            lg = entree.readLine() ;
            if ( lg != null)
            {
                A = new dive (lg) ;
                A.vect() ;
            }
        } while (lg != null) ;

        entree.close() ;
    }
}
-- =====
-- fin de la classe qui calcul la divergence de Kullbak-leiber
-- =====

```

Annexe 3 : La similarité selon la méthode de jaccard

```

-- =====
-- La classe qui calcul la similarité de jaccard
-- =====
import java.util.*;
import java.io.*;
import java.lang.*;

public class jacar {

public static void main (String args[])throws IOException{
    PrintWriter S = new PrintWriter (new FileWriter ("jacar.txt"));
    BufferedReader B = new BufferedReader (new FileReader("bas.txt"));
    String lB = B.readLine();

    while(lB != null){

        int cpt=0;
        BufferedReader A = new BufferedReader (new FileReader
("reqnew.txt"));
        String lA = A.readLine();
        StringTokenizer mA = new StringTokenizer (lA , " ");
        int AA=mA.countTokens();

```



```

StringTokenizer mB=null;
    String MotB;
String MotA;
int BB=0;

while ( mA.hasMoreTokens() ){
    mB = new StringTokenizer (lB , " ");
    BB=mB.countTokens();
    MotA = mA.nextToken();

    while (mB.hasMoreTokens()){
        MotB = mB.nextToken();

        if (MotA.equalsIgnoreCase(MotB))
            {
                cpt++;
            }
    }
}

double res= (double)cpt/(AA+BB-cpt);
S.println(res);
lB=B.readLine();
A.close();

}
B.close();
S.close();
}
}
-- =====
-- fin de la classe qui calcul la similarité de jaccard
-- =====

```

RÉSUMÉ

La recherche d'information sur Internet devient un acte quotidien impérativement indispensable pour tous les acteurs économiques et sociaux. Ainsi, il est de plus en plus attrayant d'extraire les données pertinents de ce source et de les rendre disponibles pour les utilisateurs finaux ou pour les programmes et les applications.

Cette thèse s'inscrit dans le cadre d'une contribution pour la résolution et l'amélioration des systèmes de recherche d'information (SRI). Elle a comme objectif d'apporter une solution pour les problèmes majeurs de la recherche d'information sur le web à savoir, le bruit et le silence et de proposer ainsi des méthodes et outils qu'on peut introduire dans un SRI pour guider, faciliter et également satisfaire l'utilisateur de ces systèmes en proposant les documents susceptibles de répondre à ses besoins traduits par une simple requête.

Partant de ces problèmes majeurs de la recherche d'information que nous proposons :

- **Une nouvelle mesure hybride visant à calculer la similarité entre deux requêtes dans un système de recherche d'information notamment** : la mesure de cette similarité concerne : une requête nouvellement reçue par le système exprimant le besoin de l'utilisateur et des requêtes candidates dont le système mémorise les documents pertinents.

Ce calcul de similarité passe par trois phases :

- Dans un premier temps, nous présentons une statistique plus précise basée sur la version étendue de la statistique khi2, appelée la méthode statistique de CHIR pour sélectionner les requêtes positivement dépendantes par rapport à la requête donnée ;
- Dans un second temps, nous utilisons l'information mutuelle pour mesurer la similarité sémantique entre la requête de l'utilisateur et la requête candidate du système ;
- Finalement, nous combinons ces deux mesures, statistique et sémantique par le biais de notre méthode dite d'alpha pour prédire la requête candidate la plus proche à la requête donnée en termes de similarité.

- **Système de recommandation pour améliorer le service de recherche d'information dans les plates-formes e-Learning** : cette contribution consiste à proposer une méthode de classement des documents web par une insistance sur l'appréciation de l'utilisateur du système de recherche d'information. Cette méthode a été insérée au sein d'un système de recommandation en s'inspirant de différentes méthodes, outils et techniques qui entrent en jeu à savoir : la recherche d'information, le filtrage d'information et le « Web usage mining ». Le système proposé est appliqué dans le domaine e-Learning pour tester son impact sur le processus de la recherche.

Mots clés : recherche d'information, la méthode de Chir, information mutuelle, système de recommandation, feedback, formule de Chan.

RESUME

The information retrieval on the Internet becomes a daily act which is absolutely essential for all economic and social actors. Thus, it is increasingly attractive to extract relevant data from this source and make it available to end users or to programs and applications.

This thesis is part of a contribution to the resolution and improvement of information retrieval systems (IRS). Its primary objective is to provide a solution to the major problems of information retrieval on the web i.e, the noise and the silence and thus propose methods and tools that can be introduced in a IRS to guide, facilitate and also satisfy the user of these systems by suggesting documents that may meet their needs expressed by a simple query.

Based on these major problems in information retrieval that we will propose:

- **A new hybrid measure to calculate the similarity between two queries in a information retrieval system:** the measure of this similarity concerns: a new request received by the system expressing the need of the user and the candidate requests that the system stores the relevant documents.

This similarity calculation goes through three phases:

- At first, we present a more accurate statistic based on the extended version of the statistical khi2, called the statistical method of CHIR to select queries positively dependent in relation to the given query ;
- In a second step, we use the mutual information to measure the semantic similarity between the user query and the query candidate of the system ;
- Finally, we combine these two statistical and semantic measures, using our method called alpha to predict the nearest candidate query to the request given in terms of similarity.

- **Recommendation system to improve service of information search in e-Learning platforms:** this contribution aims at proposing a method for ranking the web documents by an emphasis on the assessment of the user of information retrieval system. This method has been included in a recommendation system getting inspired on different methods, tools and techniques that interact namely: information retrieval, the information filtering and Web usage mining. The proposed system is applied in the e-Learning field to test its impact on the research process.

Keywords: information retrieval, Chir method, mutual information, recommendation system, feedback, Chan formula.