

# THÈSE

Présentée devant

l'Université Paul Sabatier de Toulouse

en vue de l'obtention du

Doctorat de l'Université Paul Sabatier

Spécialité : **INFORMATIQUE**

Par

**Hamid TEBRI**

---

## Formalisation et spécification d'un système de filtrage incrémental d'information

---

Soutenue le 15 décembre 2004, devant le jury composé de :

M. M. Boughanem	Professeur à l'Université de Toulouse III	Directeur de thèse
M. C. Chrisment	Professeur à l'Université de Toulouse III	Directeur de recherche
M. J.M. Pinon	Professeur à l'INSA de Lyon	Rapporteur
M. J. Martinez	Professeur à l'école Polytechnique de Nantes	Rapporteur
M. M. Bouzeghoub	Professeur à l'Université de Versailles	Examineur
M. L. Lechani	MCF à l'Université de Toulouse III	Examineur

INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE

Centre National de la Recherche Scientifique - Institut National Polytechnique - Université Paul Sabatier

Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse Cedex 04. Tél : 05.61.55.66.11



# Formalisation et spécification d'un système de filtrage incrémental d'information

Hamid TEBRI

15/12/2004

## Remerciements

Je tiens à remercier très sincèrement Messieurs les Professeurs Claude Chrisment et Gilles Zurfluh, responsables de l'équipe SIG, pour m'avoir accueilli dans leur équipe afin de mener à bien cette thèse.

Je tiens à exprimer ma profonde gratitude à Monsieur Mohand Boughanem, Professeur à l'Université de Toulouse III, pour avoir dirigé cette thèse dans la continuité de mon stage de DEA. Son encadrement, ses critiques constructives, ses précieux conseils, ses relectures acharnées de mes travaux m'ont été d'une aide précieuse. Pour tout cela, sa confiance et sa disponibilité du début à la fin de la thèse, je le remercie vivement et qu'il trouve ici l'expression de ma considération profonde.

Je remercie très sincèrement Monsieur Jean-Marie Pinon, Professeur à l'INSA de Lyon, Monsieur José Martinez, Professeur à l'Université de Nantes, pour avoir accepté d'être rapporteurs de ce mémoire, et pour l'honneur qu'ils me font en participant au jury. Merci également à Monsieur Claude Chrisment, Professeur à l'Université de Toulouse III, et Monsieur Mokrane Bouzeghoub, Professeur à l'université Paris IV, d'avoir accepté de juger ce travail et de faire partie du jury.

Je présente mes sincères remerciements à Madame Lynda Lechani-Tamine, Maître de Conférences à l'Université de Toulouse III, pour l'attention qu'elle a portée à la lecture de ce mémoire, pour ses remarques pertinentes et pour l'honneur qu'elle me fait en participant au jury.

Je tiens à souligner les moments passés avec Yannick Loiseau, Faiza Ghazzi, Anis Jedidi, Kais Khrouf, Amélie Shyne, Cédric Teyssie, qui ont parcouru avec moi un bout ou entièrement le chemin de la thèse.

Un grand merci à Youcef Talbi, l'ami de tous les jours, pour ses encouragements renouvelés tout au long de cette thèse, et surtout pour ses remarques fructueuses. Je tiens également à remercier tous les autres amis qui m'ont soutenu, de près ou de loin, pendant ma thèse. La liste est très longue, je me contente par leur dédier cette thèse.

Je remercie profondément les membres de l'équipe SIG, et en particulier Max, Olivier et ainsi que les anciens thésards de l'équipe.

Je tiens à remercier très sincèrement ma famille pour leur soutien constant. Enfin, je remercie tout particulièrement maman et papa pour leur présence, leur confiance et leurs encouragements bienveillants pendant mes études.

# Table des matières

<b>1</b>	<b>Introduction générale</b>	<b>1</b>
1.1	Contexte de travail . . . . .	1
1.2	Problématique . . . . .	2
1.3	Contribution . . . . .	5
1.4	Organisation du mémoire . . . . .	6
<b>I</b>	<b>Collecte active et passive de l'information</b>	<b>9</b>
<b>2</b>	<b>Collecte active : recherche d'information</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	Concepts clés de la RI . . . . .	12
2.3	Statistiques sur le texte . . . . .	15
2.3.1	Lois de Zipf . . . . .	15
2.3.2	Distribution binomiale . . . . .	17
2.3.3	Distribution multinomiale . . . . .	17
2.3.4	Distribution de poisson . . . . .	18
2.4	Pondération des termes d'indexation . . . . .	19
2.4.1	Pondération locale . . . . .	19
2.4.2	Pondération globale . . . . .	20
2.5	Taxonomie des modèles de RI . . . . .	21
2.5.1	Modèles booléens . . . . .	23

2.5.1.1	Modèle booléen de base . . . . .	23
2.5.1.2	Modèle booléen étendu . . . . .	24
2.5.1.3	Modèle booléen basé sur des ensembles flous . . . . .	25
2.5.2	Modèles vectoriels . . . . .	26
2.5.2.1	Modèle vectoriel de base . . . . .	26
2.5.2.2	Modèle connexionniste . . . . .	28
2.5.2.3	Latent Semantic Indexing (LSI). . . . .	31
2.5.3	Modèles probabilistes . . . . .	33
2.5.3.1	Modèle BIR . . . . .	33
2.5.3.2	Modèle du langage . . . . .	37
2.6	Reformulation de requêtes en RI . . . . .	39
2.6.1	Approches basées sur le relevance feedback . . . . .	39
2.6.1.1	Algorithme de Rocchio . . . . .	39
2.6.1.2	Expansion de requête dans Okapi . . . . .	41
2.6.2	Approches basées sur le contexte local/global . . . . .	41
2.6.2.1	Analyse par le contexte local . . . . .	41
2.6.2.2	Thésaurus de similarité . . . . .	42
2.7	Méthodes d'évaluation des SRI . . . . .	44
2.7.1	Mesures de précision et rappel . . . . .	44
2.7.2	Interpolation . . . . .	47
2.7.3	Mesures combinées . . . . .	47
2.7.3.1	Mesure Harmonique . . . . .	48
2.7.3.2	Mesure d'évaluation "E" . . . . .	48
2.8	Conclusion . . . . .	48

<b>3</b>	<b>Collecte passive : filtrage d'information</b>	<b>51</b>
3.1	Introduction . . . . .	51
3.2	Définition du filtrage d'information . . . . .	52
3.3	Terminologie . . . . .	53
3.4	Aperçu historique . . . . .	55
3.5	Grandes familles de filtrage d'information . . . . .	56
3.5.1	Filtrage d'information collaboratif . . . . .	57
3.5.2	Filtrage d'information basé sur le contenu (cognitif) . . . . .	59
3.6	Processus de filtrage d'information . . . . .	60
3.7	Filtrage d'information versus recherche d'information . . . . .	62
3.8	Problématique du filtrage d'information cognitif . . . . .	64
3.9	Evaluation des performances des systèmes de filtrage d'information . . . . .	66
3.9.1	Utilité linéaire . . . . .	66
3.9.2	Mesure orientée précision . . . . .	69
3.10	Présentation de quelques modèles de filtrage adaptatifs . . . . .	70
3.10.1	Modèles de filtrage basés sur la méthode heuristique . . . . .	71
3.10.1.1	Modèle de CAFES . . . . .	71
3.10.1.2	Modèle de Hoashi et al. . . . .	74
3.10.1.3	Modèle de Wu et al. . . . .	77
3.10.2	Modèle de filtrage basé sur la régression logistique . . . . .	80
3.10.3	Modèle de filtrage basé sur la distribution des scores . . . . .	83
3.10.3.1	Optimisation de la fonction d'utilité . . . . .	84
3.10.3.2	Fonction de seuillage par la technique SDS . . . . .	86
3.10.3.3	Discussion sur les modèles basés sur la distribution des scores . . . . .	91
3.11	Conclusion . . . . .	92

<b>II</b>	<b>Contribution aux systèmes de filtrage incrémentaux</b>	<b>93</b>
<b>4</b>	<b>Modèle de filtrage incrémental d'information</b>	<b>95</b>
4.1	Introduction . . . . .	95
4.2	Motivations . . . . .	96
4.3	Modèle de base . . . . .	98
4.3.1	Initialisation du système . . . . .	99
4.3.1.1	Initialisation du profil et des documents . . . . .	99
4.3.1.2	Initialisation du seuil . . . . .	100
4.3.2	Processus de filtrage . . . . .	100
4.4	Apprentissage du profil . . . . .	101
4.4.1	Apprentissage du profil : principe de renforcement . . . . .	101
4.4.2	Principe de renforcement avec normalisation du score . . . . .	105
4.5	Adaptation de la fonction de seuillage . . . . .	107
4.5.1	Construction de la distribution des scores . . . . .	108
4.5.1.1	Conversion des scores en probabilités . . . . .	108
4.5.1.2	Estimation des probabilités par intervalle . . . . .	109
4.5.1.3	Linéarisation de la distribution de probabilités des scores . . . . .	114
4.5.2	Optimisation de la fonction de seuillage . . . . .	118
4.5.2.1	Méthode de seuillage par intervalle . . . . .	120
4.5.2.2	Méthode de seuillage par dérivation . . . . .	122
4.6	Conclusion . . . . .	125
<b>5</b>	<b>Expérimentations du modèle de filtrage</b>	<b>127</b>
5.1	Introduction . . . . .	127
5.2	Programme d'évaluation de TREC-2002 . . . . .	128
5.2.1	Tâche de filtrage adaptatif . . . . .	128
5.2.2	Collection de test -Reuters . . . . .	129
5.2.3	Mesures d'évaluation . . . . .	131

5.3	Notre démarche d'évaluation . . . . .	133
5.4	Expérimentations et résultats . . . . .	134
5.4.1	Apprentissage du profil . . . . .	135
5.4.1.1	Impact du renforcement par normalisation . . . . .	135
5.4.1.2	Sélection du score de renforcement . . . . .	136
5.4.2	Adaptation de la fonction de seuillage . . . . .	137
5.4.2.1	Évaluation des méthodes d'identification des intervalles . . . . .	137
5.4.2.2	Comparaison entre linéarisation et non linéarisation . . . . .	139
5.4.3	Comparaison des méthodes d'adaptation . . . . .	140
5.4.3.1	Apprentissage du profil : Rocchio et Okapi . . . . .	140
5.4.3.2	Adaptation de la fonction de seuillage dans KUN . . . . .	144
5.4.4	Evaluation comparative avec les résultats TREC-2002 . . . . .	145
5.5	Conclusion . . . . .	148
5.6	Conclusion générale et perspectives . . . . .	149

**Bibliographie**



# Table des figures

2.1	Processus de recherche d'information . . . . .	12
2.2	Représentation des termes par ordre décroissant de leurs fréquences . . . . .	16
2.3	Taxonomie des modèles de recherche d'information . . . . .	22
2.4	Partition de la collection pour une requête . . . . .	44
2.5	Courbes de précision-rappel de deux requêtes . . . . .	46
3.1	Principe de filtrage d'information . . . . .	52
3.2	Processus de filtrage social (collaboratif) . . . . .	59
3.3	Processus de filtrage basé sur le contenu (filtrage cognitif) . . . . .	60
3.4	Architecture générale d'un système de filtrage d'information . . . . .	61
3.5	Filtrage <i>vs</i> recherche d'information . . . . .	63
3.6	Processus d'accès à l'information . . . . .	63
3.7	Initialisation du seuil dans <i>CAFES</i> . . . . .	72
3.8	Optimisation du seuil par interpolation . . . . .	73
3.9	Initialisation du profil et du seuil . . . . .	78
4.1	Evolution des scores des documents pertinents (profil 101 de TREC-2002) . . . . .	104
4.2	Distribution de probabilités par intervalle (profil 101 de TREC-2002) . . . . .	111
4.3	Problème de probabilités négatives (profil 102 de TREC-2002) . . . . .	111
4.4	Conversion de probabilités négatives en probabilités positives . . . . .	112
4.5	Densités de probabilités des scores des documents pertinents et non pertinents . . . . .	117
4.6	Estimation de $\int_{\theta}^{+\infty} P_r(x)dx$ en utilisant la surface . . . . .	118

4.7	Détection du seuil par intervalle de scores . . . . .	121
4.8	Identification de la classe $k^*$ via la classe $j^*$ . . . . .	122
4.9	Graphe d'une fonction de seconde degré selon le signe de 'a' . . . . .	124
5.1	Valeur d'utilité pour chaque valeur de $\lambda$ . . . . .	137
5.2	Utilité cumulée par chaque méthode d'identification d'intervalles . . . . .	138
5.3	Comparaison entre la linéarisation et non linéarisation . . . . .	139
5.4	Comparaison des performances des différentes méthodes d'apprentissage . .	143
5.5	Nombre de termes distincts entre deux profils :Renforcement-Rocchio . . .	143
5.6	Différence par rapport à la moyenne dans TREC-2002 . . . . .	147
5.7	Valeur d'utilité (T11SU) par profil . . . . .	147

# Liste des tableaux

2.1	Évaluation de requêtes - modèle booléen classique . . . . .	23
2.2	Évaluation de requêtes - modèle booléen/ensembles flous . . . . .	25
2.3	Table de contingence des occurrences vs. pertinences des termes . . . . .	36
2.4	Exemple de calculs de rappel et précision . . . . .	45
3.1	Types de filtrage d'information . . . . .	65
3.2	Table de contingence . . . . .	67
3.3	Valeurs d'utilité moyennes . . . . .	76
3.4	Echelle de valeurs utilisée dans TREC-8 . . . . .	83
4.1	Résultats de la linéarisation des scores . . . . .	118
5.1	Normalisation par les mesures de Dice et Jaccard . . . . .	136
5.2	Résultats de la linéarisation ou non de la dist. de probabilités . . . . .	139
5.3	Valeurs distinctes de $\alpha$ , $\beta$ et $\gamma$ : Algo. Rocchio . . . . .	142
5.4	Comparaison entre Rocchio, BM25 et renforcement . . . . .	142
5.5	Résultats LDS - KUN . . . . .	144
5.6	Evolution de l'utilité . . . . .	145
5.7	Liste des participants à TREC-2002 : Filtrage adaptatif . . . . .	146



# Chapitre 1

## Introduction générale

### 1.1 Contexte de travail

Le développement sans précédent de l'Internet combiné à la généralisation de l'informatique dans tous les secteurs d'activité ont conduit à la prolifération de sources d'informations distribuées regroupant des volumes considérables d'informations hétérogènes. Le développement d'outils automatisés permettant l'accès efficace à ces informations est une nécessité absolue.

Notre travail se situe dans le contexte de la recherche d'information et plus particulièrement dans le cadre général des systèmes d'accès à l'information. L'objectif principal de ces systèmes est de sélectionner les informations dont le contenu concorde avec un énoncé traduisant un besoin en information d'un utilisateur. Nous nous limitons dans ce mémoire aux informations (documents) textuelles.

Un Système de Recherche d'Information (SRI) capitalise un volume important d'information et offre des outils permettant de localiser les informations pertinentes relatives à un besoin en information d'un utilisateur, exprimé à travers une *requête*. L'utilisateur joue un rôle particulièrement actif dans ces systèmes, du fait que la recherche est réalisée sur la base d'une requête, qu'il définit explicitement et qu'il soumet au SRI. Ce dernier retourne une liste ordonnée de documents susceptibles de répondre à cette requête. Ce type d'approche, que l'on qualifie de *collecte active d'information*, laisse par essence une place importante à l'utilisateur mais qui, en revanche, lui impose des contraintes de compétence (formulation de requête, exploitation des résultats, etc.) et de disponibilité. A cause de ces contraintes, la notion de collecte active s'avère inadaptée à l'ensemble des situations de

recherche d'information auxquelles un utilisateur peut être confronté. En effet, un utilisateur qui possède des besoins en information spécifiques et permanents, ne devrait pas avoir besoin d'effectuer des interrogations répétées en utilisant la même équation de recherche. Il est plus opportun de lui proposer, de façon automatique et continue, des informations pertinentes par rapport à son besoin, sans le contraindre à réécrire sa requête. Ce dernier type d'accès est qualifié de *collecte passive d'information* ou de filtrage d'information. Un Système de Filtrage d'Information (SFI) peut être défini comme un processus qui permet d'extraire à partir d'un flot d'informations (News, e-mail, actualités journalières, etc.), celles qui sont susceptibles d'intéresser un utilisateur ou un groupe d'utilisateurs ayant des besoins en information relativement stables.

Les systèmes de filtrage d'information s'inscrivent dans le cadre plus général des systèmes d'accès personnalisé à l'information. Ces systèmes intègrent l'utilisateur d'une manière implicite, en tant que structure informationnelle dans le processus de sélection de l'information pertinente. Cette structure est souvent représentée à travers la notion de *profil*. Un profil peut comporter différents types d'information sur l'utilisateur, telles que ses préférences, ses centres d'intérêts, ses habitudes de recherche, etc. La nature de ces informations dépend alors de plusieurs paramètres dont le contexte d'utilisation. Celui-ci se décline dans les SFI à travers le mode de filtrage adopté. On en distingue deux principaux : le filtrage cognitif et le filtrage social. Dans le filtrage cognitif, appelé communément filtrage basé sur le contenu, le profil est souvent représenté par des mots clés. Dans le cas du filtrage social, appelé également filtrage collaboratif, le profil est représenté par les annotations et préférences que l'utilisateur a attribué à des documents qu'il a reçu au préalable. Outre le contenu des profils, ces deux modes de filtrage se distinguent par la manière de sélectionner l'information pertinente. La sélection dans le cas du filtrage cognitif est basée sur le contenu des documents, alors que dans le filtrage collaboratif, elle est plutôt basée sur les évaluations des jugements attribués aux documents par les utilisateurs.

Nos travaux s'inscrivent dans le cadre précis des systèmes de filtrage cognitif.

## 1.2 Problématique

Compte tenu de leur mode de sélection de l'information pertinente, les systèmes de filtrage cognitif se basent sur des modèles de recherche d'information augmentés par une fonction de décision. D'une façon générale, à chaque arrivée d'un document, le contenu de celui-ci est comparé à celui du profil. Une fonction de décision permet ensuite d'étiqueter

le document : accepté est donc acheminé à l'utilisateur, car estimé pertinent relativement à son profil ou alors rejeté car estimé non pertinent. Dans le but d'adapter le processus de sélection de l'information pertinente au flot d'informations, un SFI intègre un processus d'apprentissage. Ce processus se base principalement sur les documents préalablement reçus et jugés par l'utilisateur.

La problématique majeure du filtrage cognitif d'information réside dans la difficulté de construire une "bonne" représentation du contenu du profil, et une "bonne" fonction de décision. Ces éléments sont la base de l'acceptation ou de rejet d'un document. En effet, à l'initialisation du processus de filtrage, on ne dispose d'aucune connaissance sur les documents à filtrer pour pouvoir construire ces éléments. Ceci nous amène alors à poser quatre questions principales qui résument la problématique d'un système de filtrage cognitif d'information :

- *la première est comment représenter le profil ?*
- *la seconde, comment construire une fonction de décision ?*
- *la troisième, comment améliorer la représentation du profil ?*
- *et enfin la quatrième, comment adapter la fonction de décision ?*

Concernant les deux premières questions, la majorité des techniques de filtrage représente le profil par une liste de mots clés, éventuellement pondérés, extraits du texte ou des mots clés saisis par l'utilisateur, ou élaborés à partir de documents d'apprentissage, dits aussi d'entraînement. La fonction de décision est souvent une mesure de similarité (probabilité de pertinence) qui attribue un score d'appariement profil-document. Si ce score dépasse un certain seuil alors le document est sélectionné sinon il est rejeté. Ce seuil peut être fixé arbitrairement ou appris à partir d'une collection d'entraînement. En fait, quand on parle de fonction de décision on fait principalement référence à la manière d'identifier ce seuil, on utilise souvent le terme fonction de seuillage.

Une grande partie des travaux effectués dans ce domaine s'est focalisée sur les deux dernières questions, en proposant des techniques permettant l'apprentissage des profils et de la fonction de décision. L'apprentissage peut être effectué d'une façon incrémentale, c'est-à-dire seulement à partir des documents déjà filtrés; ou d'une façon différée en utilisant une collection d'apprentissage. Le filtrage incrémental représente, évidemment, le cas courant de filtrage, car au démarrage du filtrage on ne dispose normalement d'aucune information autre que le profil initial de l'utilisateur. Il est donc difficile de construire un processus de filtrage performant permettant d'identifier effectivement les informations pertinentes pour un profil donné.

La majorité des techniques d'apprentissage de profil proposées dans la littérature, en réponse à la troisième question, est inspirée du principe de reformulation de requêtes. Les techniques utilisées sont principalement basées sur une version incrémentale de l'algorithme de Rocchio ou des techniques basées sur les classifieurs Bayésiens, les réseaux de neurones et des techniques génétiques. Certaines de ces méthodes ont recours à une collection d'entraînement pour apprendre le profil. Ceci pose une première contrainte, donc une *première limite, liée à la disponibilité de ces collections*. Dans le cas où on ne dispose pas de ce type de collection, situation courante, l'apprentissage est réalisé sur des échantillons de documents déjà filtrés. Or, il a été montré que les performances de la majorité des méthodes d'apprentissage de profils dépend de la taille des échantillons considérés. Plus précisément, les méthodes basées sur la régression logistique sont plutôt performantes lorsque le nombre de documents de l'échantillon est conséquent, alors que d'autres, tels que les classifieurs bayésiens ou l'algorithme de Rocchio, fonctionnent mieux lorsque le nombre de documents de l'échantillon est réduit. Autrement dit, *ces méthodes d'apprentissage n'apprennent pas de manière uniforme tout au long du processus de filtrage. Ceci constitue la seconde limite*.

Enfin, l'objet de la quatrième question est de déterminer une valeur optimale du seuil. Les différentes méthodes proposées dans la littérature, tentent de définir un seuil qui permet d'optimiser une fonction d'utilité. Celle-ci permet de mesurer la capacité d'un SFI à ne sélectionner que des documents pertinents. La majorité des techniques de seuillage actuelles se basent sur des méthodes heuristiques, régressions logistiques ou distributions des scores. Les méthodes basées sur la distribution des scores considèrent que les scores des documents pertinents (resp. non pertinents) suivent une loi gaussienne (resp. exponentielle). L'estimation des paramètres des deux distributions se fait soit de manière empirique à partir des collections d'entraînement, ou par une méthode de maximum de vraisemblance. Ces méthodes supposent donc que les scores des documents pertinents et non pertinents suivent certaines lois de probabilité connues à l'avance. L'inconvénient de ces méthodes est que dans un contexte expérimental, *si on admet la forme de la distribution des scores a priori, elle peut ne pas être valable pour des conditions expérimentales particulières, car les scores des documents restent incontrôlables*. De plus, *ces méthodes nécessitent souvent un certain nombre minimum de documents pour avoir des estimations biaisées*.

## 1.3 Contribution

Notre travail rentre dans la catégorie des systèmes de filtrage cognitifs et incrémentaux. Notre contribution se situe à plusieurs niveaux et tente de répondre aux quatre questions soulevées dans la section précédente. Nos propositions concernent plus précisément l'apprentissage des profils et l'adaptation de la fonction de décision. Nous tentons de répondre aux limites des approches mises en évidence dans la section précédente.

1. *Concernant l'apprentissage du profil*, la méthode d'apprentissage du profil que nous proposons, appelée *apprentissage par renforcement*, est purement incrémentale. Elle ne nécessite aucune connaissance autre que le profil initial au démarrage du processus de filtrage. Cette méthode d'apprentissage est déclenchée pour chaque document jugé pertinent par l'utilisateur. Elle consiste tout d'abord à construire un profil temporaire à partir de ce document. Ce profil devrait permettre de sélectionner le document en question, avec un score le plus élevé possible, appelé score de renforcement. Ce profil temporaire est ensuite intégré dans le profil global de l'utilisateur.

La méthode d'apprentissage par renforcement a été tout d'abord présentée et expérimentée dans le modèle initial. Nous qualifions de modèle initial tout ce qui relève des travaux effectués dans le cadre de la thèse de M.Tmar. Nous avons également participé à la mise en oeuvre de certaines solutions, notamment dans notre participation à TREC<sup>1</sup>. Nous y avons apporté des améliorations qui se traduisent par un meilleur contrôle du score de renforcement et une nouvelle manière de construire le profil temporaire. Cette technique d'apprentissage permet d'apprendre les profils de manière uniforme tout au long du processus de filtrage, comparativement aux différentes méthodes proposées dans le domaine.

2. *Au niveau de l'adaptation de la fonction de décision*, nous proposons une méthode d'adaptation du seuil qui s'inscrit dans la catégorie des méthodes basées sur la distribution des scores des documents. Notre méthode suppose que les distributions des scores des documents sont inconnues, mais propose d'estimer les probabilités discrètes des scores des documents, puis de "dessiner" la distribution des scores en utilisant une régression linéaire. Une fois ces distributions sont construites, nous proposons de réécrire la fonction d'utilité en fonction de ces distributions, puis de déduire le score (seuil) qui permet d'optimiser cette fonction. La construction d'une distribution de probabilité des scores des documents consiste tout d'abord à décomposer les scores en plusieurs intervalles, puis à calculer la probabilité des scores dans ces intervalles. Une

---

<sup>1</sup>Text RETrieval Conference

régression linéaire est ensuite utilisée pour convertir la distribution de probabilités discrètes en une distribution de probabilités continue.

Ce travail a été présenté dans un premier temps dans le cadre du modèle initial. Nous y avons apporté tout d'abord des améliorations liées à la manière de construire les distributions de probabilités ; nous avons ensuite proposé une nouvelle formalisation de la fonction d'optimisation du seuil et une méthode déterministe pour la résolution de cette fonction.

Nos différentes propositions ont été évaluées sur une collection standard issue du programme TREC. Dans le but de situer nos travaux par rapport à des travaux similaires dans le domaine, notre démarche d'évaluation a été effectuée selon le canevas TREC. L'objectif étant de confronter nos résultats à ceux présentés notamment à TREC-2002. Le premier résultat important que l'on peut tirer de cette comparaison est que les performances, en terme d'utilité, de notre modèle de filtrage sont meilleures que tous les modèles présentés à l'édition TREC-2002. Un second résultat qui découle du premier concerne les performances de notre modèle vis-à-vis du modèle initial, présenté également à TREC-2002. Nous améliorons les performances du modèle initial d'environ 30%.

## 1.4 Organisation du mémoire

Ce mémoire est organisé en deux parties :

L'objectif de la première partie est de présenter de manière détaillée les deux types de systèmes de collecte d'information : les systèmes de recherche et de filtrage d'information. Deux chapitres distincts, chapitre 2 et 3, sont consacrés à la description de ces deux types de systèmes.

Le chapitre 2 décrit les concepts de base du domaine des systèmes de collecte active d'information, en l'occurrence les systèmes de recherche d'information. Il présente ensuite de manière détaillée les différents modèles de la RI, puis les techniques de reformulation de requêtes.

Nous présentons dans le chapitre 3, le domaine qui nous intéresse particulièrement dans cette thèse : le filtrage d'information. Nous donnons tout d'abord quelques définitions sur les SFI. Nous présentons ensuite les trois grandes familles de filtrage d'information, à savoir le filtrage cognitif, collaboratif et économique. Nous insistons particulièrement sur la problématique du filtrage cognitif. Enfin, nous ferons un tour d'horizon des modèles de

filtrage cognitifs les plus répandus, en insistant plus particulièrement sur leurs méthodes de seuillage.

L'objectif de la deuxième partie est de présenter notre contribution. Nous l'avons subdivisée en deux chapitres.

Dans le chapitre 4, nous décrivons notre travail de formalisation et de spécification d'un système de filtrage incrémental d'information. Nous focalisons notre formalisation sur les méthodes d'apprentissage du profil et d'adaptation du seuil. Pour chacune de ces méthodes, nous présentons les limites et les solutions proposées pour y remédier.

Nous consacrons le chapitre 5 aux expérimentations que nous avons réalisées sur des collections issues du programme TREC. Nous montrons tout d'abord l'impact de nos différentes propositions sur le modèle initial. Nous effectuons ensuite des comparaisons avec des méthodes connues et reconnues pour leur performance dans le domaine. Nous dressons à l'issue de ces expérimentations une étude comparative entre notre modèle et ceux présentés à l'édition 2002 de TREC.

En conclusion, nous dressons un bilan de nos travaux, en mettant en exergue nos propositions. Nous présentons enfin les perspectives de nos travaux.



## Première partie

# Collecte active et passive de l'information



# Chapitre 2

## Collecte active de l'information : recherche d'information

### 2.1 Introduction

La Recherche d'Information (RI) est un domaine de l'informatique qui s'intéresse à l'organisation, au stockage et à la sélection d'informations répondant aux besoins des utilisateurs. Ce domaine manipule différents concepts : la requête, le besoin en information, les documents, la pertinence, etc. L'objectif de ce chapitre est de passer en revue les différents concepts, approches et modèles étudiés dans ce domaine.

Ce chapitre est organisé comme suit : la section 2.2 décrit les concepts clés composant un système de recherche d'information. Plus précisément, nous définissons les notions de collection de documents, de besoin en information, de mécanismes de représentation des informations, d'appariement entre la requête et le document et de reformulation des requêtes. Quelques uns de ces concepts sont détaillés dans les sections ultérieures. La section 2.3 décrit quelques caractéristiques statistiques concernant la distribution des termes dans les documents. Ces caractéristiques sont la clé de voûte des techniques de pondération présentées dans la section 2.4, et de manière générale de la majorité des modèles de RI dont les plus importants sont présentés dans la section 2.5. La section 2.6 détaille un autre concept clé de la RI, la reformulation de requêtes. Quelques méthodes de reformulation de requêtes en rapport avec la problématique traitée dans ce mémoire sont décrites. La section 2.7 est consacrée aux techniques d'évaluation des systèmes de recherche d'information.

## 2.2 Concepts clés de la RI

Le but d'un Système de Recherche d'Information (SRI) est de trouver dans une collection de documents ceux qui sont susceptibles de répondre au besoin en information d'un utilisateur. Un SRI comprend cinq concepts de base (voir figure 2.1) :

- une collection de documents,
- un besoin en information,
- un processus de représentation du contenu des documents et des besoins en information de l'utilisateur,
- un processus d'appariement document-requête,
- un mécanisme de reformulation de requêtes.

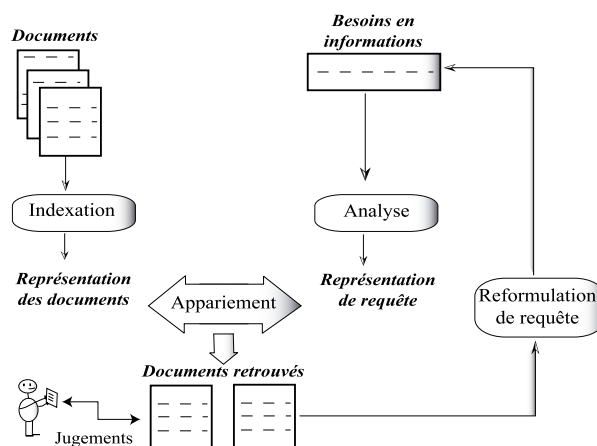


Figure 2.1 – Processus de recherche d'information

- 1. Collection de documents.** La collection de documents constitue l'ensemble des informations exploitables, compréhensibles et accessibles par l'utilisateur. Une collection comporte un ensemble de granules documentaires. Un granule de document peut représenter tout ou une partie d'un document. Il représente l'unité sélectionnée en réponse à une requête de l'utilisateur. Nous nous limitons dans notre étude aux granules de documents textuels. Dans le reste de ce rapport, nous utilisons indifféremment les termes document ou information pour désigner un granule documentaire.
- 2. Besoin en information.** Un besoin en information d'un utilisateur est exprimé à travers une *requête*. Divers types de langages d'interrogation ont été proposés en RI pour formuler une requête. Une requête peut être exprimée en langage naturel ou quasi

naturel [Robertson et al., 1998] [Salton, 1971] (exemple, "*Trouver les universités de France*"), ou dans un format structuré, appelé aussi interrogation en langage booléen [Bourne and Anderson, 1979] (exemple, "*recherche ET filtrage*"), ou la requête est constituée à partir d'une interface graphique [Lelu and François, 1992].

**3. Représentation des documents et des requêtes.** Le processus de représentation permet la traduction de la requête ou du document d'une description brute, souvent en texte libre, vers une description structurée. Ce processus est appelé *indexation*. L'objectif de l'indexation est de trouver les concepts les plus importants dans le document ou la requête. La liste de ces concepts forment ce que l'on appelle le *descripteur du document*.

Cette indexation peut être manuelle, réalisée par un expert humain, ou automatique. Le choix et l'intérêt de l'une par rapport à l'autre dépendent d'un certain nombre de paramètres, dont le plus déterminant est le volume des collections. Il est en effet difficile d'indexer manuellement des collections comportant des milliers de documents. Une étude comparative de ces deux approches a été réalisée récemment par Anderson et Perez-Carballo [Anderson and Pérez-Carballo, 2001]. Le résultat de l'étude montre que les avantages et les inconvénients de chacune des deux approches ont une tendance à s'équilibrer. Autrement dit, le choix de l'une ou de l'autre est en fonction du domaine, de la collection et de l'application considérés.

Le résultat du processus d'indexation est, comme on vient de le mentionner, une liste de concepts. Les concepts peuvent être de formes différentes : des mots simples ou des groupes de mots. Généralement, un concept est assorti d'un poids représentant le degré d'importance du concept dans le document ou la requête. L'idée d'utiliser des termes simples comme des représentants de concepts est assez naturelle. En effet, ce sont des unités linguistiques qui sont les plus souvent faciles à reconnaître, et qu'elles sont assez porteuses de sens. Cependant, utiliser des mots seuls ne donne pas une description toujours précise. Par exemple, le concept de "*filtrage d'information*", une fois représenté par les mots "*filtrage*" et "*information*", perd beaucoup de sens, car ces mots sont très courants en français et de plus sont très imprécis.

De manière générale, l'indexation automatique peut se faire selon deux approches : statistique [Salton and Yang, 1973] [Rijsbergen, 1979] et linguistique [Sheridan and Smeaton, 1992]. L'approche statistique se base sur la distribution statistique des termes dans le document. L'approche linguistique se base sur les tech-

niques de traitement du langage naturel, telles que l'analyse lexicale, syntaxique et sémantique, pour extraire les concepts les plus discriminants dans un document. L'ensemble des termes extraits des documents d'une collection constitue un *langage d'indexation*.

**4. L'appariement document-requête.** Ce processus permet de mesurer la pertinence d'un document vis-à-vis d'une requête. De manière générale, à chaque réception d'une requête, le système crée une représentation similaire à celle des documents, puis calcule un score de correspondance entre la représentation de chaque document et celle de la requête. La correspondance peut être binaire (pertinent ou non pertinent), on parle alors d'appariement exact, ou peut mesurer un degré (similarité, probabilité) de pertinence, on parle alors d'appariement approché. Idéalement, la correspondance entre ces deux représentations, déterminée par le système, doit s'accorder au jugement de pertinence de l'utilisateur. Pour une requête donnée, le système retourne des documents en ordre décroissant du score de pertinence. Les documents seront jugés par l'utilisateur, et son jugement sera utilisé pour améliorer la représentation de la requête ; c'est ce qu'on appelle la reformulation de requêtes dans le contexte de la RI.

**5. Reformulation de requêtes.** La requête initiale est vue en RI comme un moyen permettant d'initialiser le processus de sélection d'informations pertinentes. A ce titre, les SRIs doivent intégrer des fonctionnalités permettant de prendre le relai. Ce relai est souvent effectué par le processus de reformulation de requêtes. Ce processus permet en fait de construire une nouvelle requête en se basant sur des informations/connaissances extraites des documents ou disponibles dans des ressources spécifiques. 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 la pertinence utilisateur [Boughanem, 2000]. Le principe de la reformulation de requêtes consiste à modifier la requête de l'utilisateur en rajoutant des termes significatifs et/ou en réestimant leurs poids associés. Ces termes peuvent provenir :

- de documents jugés par l'utilisateur. On parle alors dans ce cas de la réinjection de pertinence, communément appelée *Relevance feedback* ou *Retour de pertinence*.
- ou des sources construites manuellement (thésaurus), ou automatiquement à partir des documents de la collection. Dans le cas où ces ressources sont construites automatiquement, elles sont souvent représentées sous forme de termes reliés entre eux. Ces liens sont mesurés en se basant sur la co-occurrence entre termes et sont construits, soit à partir des documents retrouvés par le système, on parle alors

de reformulation par *contexte local*, soit à partir d'une collection de documents existante, on parle alors de reformulation par *contexte global*.

La section 2.6 passe en revue les techniques de base utilisées dans la reformulation de requêtes.

## 2.3 Statistiques sur le texte

La majorité des modèles et approches proposés en RI se base sur des considérations et interprétations plus statistiques que linguistiques, pour par exemple identifier les mots importants du document, ou encore mesurer la pertinence d'un document vis-à-vis d'une requête.

Ainsi, dans le cas de l'indexation, les approches statistiques supposent explicitement ou implicitement que les données textuelles des documents suivent une certaine distribution statistique. Ces propriétés sont fondamentales pour, par exemple, assigner une importance à un terme dans le document.

Dans les sous-sections qui suivent, nous présentons quelques distributions utilisées pour représenter les données textuelles dans le contexte de la RI.

### 2.3.1 Lois de Zipf

Les premières études sur la distribution des termes ont été effectuées par George Zipf [Zipf, 1949]. Zipf a réalisé plusieurs études sur l'utilisation des termes dans le contenu des documents d'une collection. Il a observé que les termes suivent plusieurs lois empiriques, représentées par une règle basée sur le principe à moindre effort (*Principle of Least Effort*). L'une des lois utilisant le principe à moindre effort est désignée par la loi de Zipf. Elle décrit la distribution de la fréquence des termes dans une collection. Si on dresse une liste (ou un histogramme, figure 2.2) de l'ensemble des mots différents d'un texte quelconque, classés par ordre de fréquences décroissantes, on constate que la fréquence d'un mot est inversement proportionnelle à son rang de classement dans la liste. Formellement, ceci peut être traduit par la formule suivante :

$$fréquence \times rang = constante$$

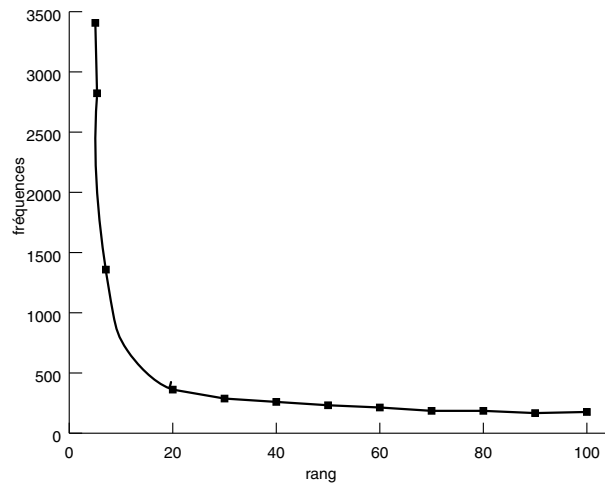


Figure 2.2 – Représentation des termes par ordre décroissant de leurs fréquences

Zipf explique la courbe hyperbolique de la distribution des termes par ce qu’il appelle le principe à moindre effort ; il considère qu’il est plus facile pour un auteur d’un document, de répéter certains termes que d’en utiliser des nouveaux. La relation entre la fréquence et le rang des termes permet de sélectionner les termes représentatifs d’un document [Salton, 1989]. La sélection de termes discriminants par la loi de Zipf consiste à éliminer respectivement les termes de fréquences très élevées car ne sont pas discriminants entre le document et la collection, et les termes de fréquences très faibles (exemple, ceux qui apparaissent dans très peu de documents), car ils sont rarement utilisés dans une requête. En utilisant cette approche, la taille du langage d’indexation d’une collection peut être réduit considérablement.

Une autre loi proposée par Zipf est que le nombre moyen d’occurrences d’un terme est en corrélation avec la racine carrée de sa fréquence. Cette loi indique que les termes moins fréquents sont moins ambigus, et ceux trop fréquents ne sont pas intéressants de les considérer dans le langage d’indexation. Zipf montre aussi que la longueur d’un terme est en relation inverse avec sa fréquence.

Les lois de Zipf sont plutôt intéressantes pour caractériser les termes dans une collection, mais moins intéressantes si on veut distinguer les documents d’une collection. Exemple des SRI qui proposent d’ordonner les documents par rapport à leurs probabilités de pertinence vis-à-vis d’une requête. L’estimation de ces probabilités nécessite, en effet, de caractériser la distribution statistique des termes de la requête dans les documents et leur distribution dans toute la collection.

### 2.3.2 Distribution binomiale

Une distribution binomiale peut être considérée comme la somme d'une série d'épreuves (indépendantes) de Bernoulli [Saporta, 1990], par exemple, une épreuve à deux événements possibles, comme le jet d'une pièce de monnaie. On appelle variable binomiale de paramètres  $n$  et  $p$ , la variable aléatoire définie par le nombre de réalisations, soit  $r$ , d'un événement de probabilité  $p$  au cours de  $n$  épreuves indépendantes. Elle est donnée comme suit :

$$b(r; n, p) = \binom{n}{r} p^r (1 - p)^{n-r} = \frac{n!}{(n-r)!r!} p^r (1 - p)^{n-r}$$

La distribution binomiale peut être utilisée pour modéliser les occurrences des termes dans les textes d'une collection de documents. Les termes dans une collection peuvent être vus comme une séquence de  $n$  épreuves, où  $p$  représente la probabilité qu'un terme soit présent dans la collection, et  $(1 - p)$  sinon. Une propriété intéressante de la distribution binomiale est que, pour des valeurs de  $n$  telle que  $np(1 - p) > 5$ , on peut utiliser une approximation par la loi normale. Cependant, dans l'analyse des textes, les hypothèses exigées pour une approximation d'une loi normale, ne sont pas souvent vérifiées en raison du problème des données rares. On sait, d'après la loi de Zipf, que beaucoup de termes apparaissent rarement dans le texte de la collection. Une alternative d'utiliser une approximation normale et de se baser sur une estimation par une loi multinomiale [Dunning, 1993].

### 2.3.3 Distribution multinomiale

La distribution multinomiale est une extension de la distribution binomiale. Au lieu de supposer qu'une épreuve d'une réalisation soit représentée par deux événements possibles, elle est représentée par  $m$  événements possibles. Le nombre de réalisations de  $m$  événements de fréquence  $f_i$  de probabilité  $p_i$  au cours de  $n$  épreuves est :

$$m(f_1, f_2, \dots, f_m; n, p_1, p_2, \dots, p_m) = \frac{n!}{f_1! f_2! \dots f_m!} p_1^{f_1} p_2^{f_2} \dots p_m^{f_m}$$

où,  $\sum_{i=1}^m p_i = 1$  et  $\sum_{i=1}^m f_i = n$ , l'équation ci-dessus peut être reformulée comme suit :

$$m(S) = \frac{n!}{\prod_{t=1}^m f_t!} \prod_{t=1}^m p_t^{f_t}$$

où  $m(S)$  est la probabilité qu'une phrase  $S$  soit dérivée d'une distribution multinomiale.

La probabilité de réalisation d'une séquence d'événements indépendants (séquence de termes,  $t_1, t_2, \dots, t_n$ ) peut être représentée par le produit de la probabilité de chaque événement :

$$P(t_1, t_2, \dots, t_n) = \prod_{i=1}^n P(t_i)$$

Un exemple de distribution multinomiale est le modèle unigramme des termes. La distribution multinomiale est utilisée dans plusieurs modèles de recherche. L'idée de base est que la probabilité de pertinence d'un document vis-à-vis d'une requête peut être modélisée par la probabilité que la requête soit générée par un modèle unigramme de paramètres à estimer à partir du document. En d'autres termes, pour chaque document on construit un petit modèle de langage statistique et on estime la probabilité que ce modèle génère une requête [Hiemstra, 1998]. Le modèle de langage est défini dans la section 2.5.

### 2.3.4 Distribution de poisson

La distribution de poisson est l'une des distributions probabilistes standard qui est utilisée pour modéliser le nombre d'occurrences d'un certain nombre aléatoire d'événements dans un échantillon de taille fixe. La distribution poissonnienne est décrite par :

$$p(k; \lambda_i) = \frac{e^{-\lambda_i} \lambda_i^k}{k!}$$

où,  $p(k; \lambda_i)$  est la probabilité qu'un événement  $i$  se réalise  $k$  fois dans l'échantillon. La distribution de poisson est caractérisée par les deux propriétés de l'espérance et de variance qui sont égales à  $\lambda_i$ . La distribution de poisson est une limite de la loi binomiale

lorsque le nombre d'événements tend vers l'infini et la probabilité  $p$  tend vers zéro.

La distribution de poisson a été utilisée dans le domaine de la RI pour modéliser la distribution des termes dans les documents [Kraaij, 2004]. Par exemple, calculer la probabilité d'apparition d'un certain terme  $t_i$  de fréquence  $f_k$  dans un nombre aléatoire de documents :  $p_i(f_k) = p(f_k; \lambda_i)$ . Le paramètre  $\lambda_i$  est la fréquence moyenne du terme  $i$  dans une collection. Cette fréquence est égale à la fréquence globale du terme  $gtf_i$  (nombre d'occurrences du terme  $i$  dans la collection) divisé par le nombre de documents. Ce facteur a été utilisé dans les travaux de Manning et Schütze [Manning and Schütze, 1999].

Une étude intéressante sur l'utilisation de ces statistiques dans le texte peut être trouvée dans Kraaij [Kraaij, 2004].

## 2.4 Pondération des termes d'indexation

La pondération est l'une des fonctions fondamentales en RI. Elle est la clé de voûte de la majorité des modèles et approches de RI proposée depuis le début des années 1960. Le poids d'un terme dans un document traduit l'importance de ce terme dans ce document. Cette mesure est souvent calculée en se basant sur les propriétés statistiques présentées précédemment.

De manière générale, la majorité des formules de pondération des termes est construite par combinaison de deux facteurs. Un facteur, de pondération locale, quantifiant la représentativité locale d'un terme dans le document et le second facteur de pondération globale, mesurant la représentativité globale du terme vis-à-vis de la collection des documents.

### 2.4.1 Pondération locale

La pondération locale permet de mesurer la représentativité locale d'un terme. Elle prend en compte les informations locales du terme par rapport à un document donné. Elle indique l'importance du terme dans ce document. Les fonctions de pondération locales les plus utilisées sont les suivantes :

- *fonction brut de  $tf_{ij}$  (term frequency)* : correspond au nombre d'occurrences du terme  $t_i$  dans le document  $D_j$  ;

- *fonction binaire* : elle vaut 1 si la fréquence d'occurrence du terme dans le document est supérieure ou égale à 1, et 0 sinon.
- *fonction logarithmique* : combine  $tf_{ij}$  avec un logarithme, donné par  $\alpha + \log(tf_{ij})$ , où  $\alpha$  est une constante. Cette fonction permet d'atténuer les effets de larges différences entre les fréquences d'occurrence des termes dans le document.
- *fonction normalisée* : permet de réduire les différences entre les valeurs associées aux termes du document, et elle est donnée comme suit :

$$0.5 + 0.5 \times \frac{tf_{ij}}{\max_{t_i \in D_j} tf_{ij}}$$

où  $\max_{t_i \in D_j} tf_{ij}$  est la plus grande valeur de  $tf_{ij}$  des termes du document  $D_j$ .

## 2.4.2 Pondération globale

La pondération globale prend en compte des informations concernant un terme par rapport à la collection de documents. Elle indique la représentativité globale du terme dans l'ensemble des documents de la collection. Un poids plus important doit être donné aux termes qui apparaissent moins fréquemment dans la collection : les termes qui sont utilisés dans de nombreux documents sont moins utiles pour la discrimination que ceux qui apparaissent dans peu de documents. Le facteur de pondération globale qui dépend de la fréquence inverse dans le document a été introduit, comme le facteur *IDF* (pour *Inverted Document Frequency*) donné par plusieurs formules :

$$IDF = \log\left(\frac{N}{n_i}\right)$$

ou

$$IDF = \log\left(\frac{N-n_i}{N}\right)$$

où,  $n_i$  est le nombre de documents où le terme  $t_i$  apparaît dans une collection de documents de taille  $N$ .

De ce fait, par cette double pondération locale et globale, les fonctions de pondérations sont souvent référencées sous le nom de *TF-IDF*.

Les pondérations locales et globales ne tiennent pas compte d'un aspect important du document : sa longueur. En général, les documents les plus longs auront tendance

à utiliser les mêmes termes de façon répétée, ou à utiliser plus de termes pour décrire un sujet. Par conséquent, les fréquences des termes dans les documents seront plus élevées, et les similarités avec la requête seront également plus grandes. En effet, certaines mesures normalisent la formulation de la fonction de pondération en intégrant la taille des documents, ce qu'on appelle facteur de normalisation. Robertson et Spark-Jones [Robertson and Sparck-Jones, 1997] proposent de normaliser la fonction de pondération de la façon suivante :

$$wd_{ij} = \frac{tf_{ij} \cdot (K_1 + 1) \cdot \log\left(\frac{N}{n_i}\right)}{K_1 \cdot ((1-b) + b \cdot \frac{dl_j}{\Delta l}) + tf_{ij}} \quad (2.1)$$

où,  $wd_{ij}$  est le poids du terme  $t_i$  dans le document  $D_j$ ;  $K_1$  contrôle l'influence de la fréquence du terme  $t_i$ , sa valeur optimale dépend de la longueur et de l'hétérogénéité des documents dans la collection de documents (dans TREC,  $K_1 = 2$ );  $b$  est une constante appartenant à l'intervalle  $[0, 1]$  et contrôle l'effet de la longueur du document (dans TREC, elle est fixée à 0.75);  $dl_j$  est la longueur du document  $D_j$ , et  $\Delta l$  est la longueur moyenne des documents dans la collection entière.

Une autre fonction de pondération normalisée utilisée dans le système Inquiry [Callan et al., 1992] est donnée comme suit :

$$wd_{ij} = 0.4 + 0.6 \times \frac{tf_{ij}}{tf_{ij} + 0.5 + 1.5 \cdot \frac{dl_j}{\Delta l}} \cdot \frac{\log\left(\frac{N+0.5}{n_i}\right)}{\log(N+1)} \quad (2.2)$$

## 2.5 Taxonomie des modèles de RI

La première fonction d'un système de recherche d'information est de mesurer la pertinence d'un document vis-à-vis d'une requête. Un modèle de RI a pour rôle de fournir une formalisation du processus de recherche d'information. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de la mesure de pertinence.

La figure 2.3 présente une classification des principaux modèles utilisés en recherche d'information. Les trois principales classes ou modèles de recherche d'information sont :

- i. les modèles booléens

ii. les modèles vectoriels

iii. les modèles probabilistes

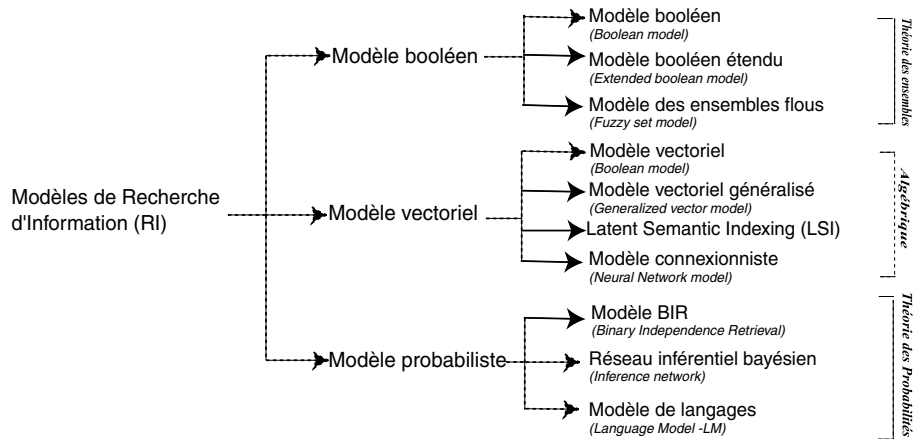


Figure 2.3 – Taxonomie des modèles de recherche d'information

Les modèles booléens se basent sur la théorie des ensembles. En général, la requête est exprimée par une liste de termes et des opérateurs logiques : conjonction (ET), disjonction (OU) et négation (NON). A chaque terme est associé un ensemble de documents où il apparaît. L'opérateur "ET" restreint le résultat de la requête à l'intersection entre deux ensembles, l'opérateur "OU" fournit l'union et l'opérateur "NON" la différence entre les ensembles. D'autres modèles sont dérivés du modèle booléen, tels que le modèle booléen étendu [Salton et al., 1983] et le modèle basé sur les ensembles flous. Les modèles vectoriels reposent sur la théorie algébrique. La pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance (ou similarité) dans un espace vectoriel. Plusieurs modèles s'inspirant du modèle vectoriel ont été proposés dans le domaine de la RI : le modèle vectoriel généralisé [Wong et al., 1985], le modèle connexionniste [Boughanem, 1992] [Mothe, 1994] [Kwok, 1989] et le modèle LSI (*Latent Semantic Indexing*) [Deerwester et al., 1990]. Enfin, les modèles probabilistes se basent sur la théorie des probabilités. La pertinence d'un document vis-à-vis d'une requête est vue comme une probabilité de pertinence document/requête. On distingue le modèle BIR (*Binary Independence Retrieval*) [Rijsbergen and Sparck-Jones, 1973], le modèle inférentiel bayésien [Turtle, 1991] et le modèle de langage [Ponton and Croft, 1998].

Dans ce qui suit, nous décrivons pour chacune des trois classes le modèle de base associé, et quelques modèles dérivés ou inspirés à partir de ces classes de modèles.

## 2.5.1 Modèles booléens

### 2.5.1.1 Modèle booléen de base

Le modèle booléen se base sur la manipulation des ensembles. Une requête est une expression logique composée de termes assemblés par les opérateurs ET ( $\wedge$ ), OU ( $\vee$ ) et NON ( $\neg$ ). Le modèle booléen utilise le mode d'appariement exact, il ne restitue que les documents répondant exactement aux termes de la requête.

*Exemple de requête :*

$$Q_k = (tq_{k1} \wedge tq_{k2}) \vee (tq_{k3} \wedge \neg tq_{k4})$$

où,  $Q_k$  est la kème requête, et  $tq_{ki}$  est le ième terme de la requête  $Q_k$ .

Soumettre ce type de requête à un SRI basé sur un modèle booléen, implique que ce dernier doit retourner un ensemble de documents contenant simultanément les termes  $tq_{k1}$  et  $tq_{k2}$  ou un ensemble de documents contenant le terme  $tq_{k3}$  et non pas le terme  $tq_{k4}$ .

Le processus de recherche mis en oeuvre via ce type de modèle, consiste à effectuer des opérations logiques sur les ensembles de documents, définis par l'occurrence ou l'absence de termes d'indexation, afin de réaliser un appariement exact avec l'équation de la requête. La valeur de pertinence  $rsv(D_j, Q_k)$  (*Retrieval Status Value*) entre la forme de la requête  $Q_k$  et un document  $D_j$  est définie dans le tableau 2.1.

Requête ( $Q_k$ )	$rsv(D_j, Q_k)$	Le résultat de l'évaluation
$tq_{ki}$	$rsv(D_j, tq_{ki})$	= 1 si $tq_{ki} \in D_j$ = 0 sinon
$tq_{k1} \wedge tq_{k2}$	$rsv(D_j, tq_{k1} \wedge tq_{k2})$	= 1 si $rsv(D_j, tq_{k1}) = 1$ et $rsv(D_j, tq_{k2}) = 1$ = 0 sinon
$tq_{k1} \vee tq_{k2}$	$rsv(D_j, tq_{k1} \vee tq_{k2})$	= 1 si $rsv(D_j, tq_{k1}) = 1$ ou $rsv(D_j, tq_{k2}) = 1$ = 0 sinon
$\neg tq_{k1}$	$rsv(D_j, \neg tq_{k1})$	= 1 si $rsv(D_j, tq_{k1}) = 0$ = 0 sinon

Tableau 2.1 – Évaluation de requêtes - modèle booléen classique

Nous remarquons que ce modèle est très simple à mettre en oeuvre. Tout de même, il présente les principaux inconvénients suivants :

- le système retourne un ensemble de documents non ordonnés comme réponse à une requête. Cependant, il n'est pas possible de dire quel document est plus pertinent qu'un autre.
- les termes dans une requête ou dans un document sont pondérés de la même façon (simple 0 ou 1), il est ainsi difficile de distinguer les termes les plus importants.
- les formules de requêtes sont complexes, non accessibles à un large public. Elles nécessitent une maîtrise parfaite des opérateurs booléens, car leur signification est différente de celle qu'ils ont dans la langue naturelle.

Pour remédier aux inconvénients du modèle booléen, des extensions du modèle dites booléen étendu et approche basée sur les ensembles flous sont proposées. Ces deux extensions seront détaillées dans les sous-sections suivantes.

### 2.5.1.2 Modèle booléen étendu

Une alternative pour rendre le modèle booléen plus intéressant est de l'étendre afin de supporter un appariement approché en assignant des poids aux termes de la requête et des documents et en mesurant un score de pertinence. Le principe de base du modèle booléen étendu est de conférer aux termes de recherche des poids, et d'interpréter les opérateurs de l'équation de la requête comme des distances entre requêtes et documents.

Le modèle booléen étendu, appelé aussi modèle *P-Norm*, a été introduit en 1983 par Salton et al. [Salton et al., 1983]. Considérons un ensemble de termes  $t_1, \dots, t_N$ , et soit  $wd_{ij}$  le poids du terme  $t_i$  dans le document  $D_j = (wd_{1j}, \dots, wd_{Nj})$ , avec  $1 \leq i \leq N$  et  $0 \leq wd_{ij} \leq 1$ . La similarité entre le document  $D_j$  et une requête  $Q_k$  décrite sous une forme conjonctive ou disjonctive est donnée comme suit :

$$\text{Opérateur OU : } Sim(D_j, Q_k) = \left( \frac{\sum_{i=1}^N wq_{ik}^p wd_{ij}^p}{\sum_{i=1}^N wq_{ik}^p} \right)^{1/p}$$

$$\text{Opérateur ET : } Sim(D_j, Q_k) = 1 - \left( \frac{\sum_{i=1}^N wq_{ik}^p (1 - wd_{ij}^p)}{\sum_{i=1}^N wq_{ik}^p} \right)^{1/p}$$

où  $p/0 \leq p \leq \infty$  est une constante, et  $wq_{ik}$  le poids du terme  $t_i$  dans la requête  $Q_k$ .

Dans ce modèle booléen étendu, lorsque  $p = 1$ , il n'y a aucune distinction entre les deux connecteurs *ET* et *OU*. Par conséquent, la similarité entre les requêtes et les documents peut être calculée par le produit scalaire entre leurs termes pondérés. Ainsi, on constate que

l'utilisation d'un simple modèle vectoriel (décrit dans la section 2.5.2) est possible lorsque  $p = 1$ .

### 2.5.1.3 Modèle booléen basé sur des ensembles flous

La logique floue a été proposée par Lotfi Zadeh [Zadeh, 1965] dans le milieu des années 1960. À l'inverse de la logique booléenne, la logique floue permet à une condition d'être dans un autre état que vrai ou faux. Elle peut prendre une infinité de valeurs de vérité dans un intervalle  $[0, 1]$ .

En recherche d'information, une extension du modèle booléen basée sur les ensembles flous a été proposée par Salton [Salton, 1989]. Cette extension vise également à tenir compte de la pondération des termes dans les documents. Un poids d'un terme exprime le degré d'appartenance de ce terme à un ensemble. Ainsi, un document peut être représenté par un ensemble de termes pondérés comme suit :

$$D_j = \{(td_{1j}, a_{1j}), \dots, (td_{nj}, a_{nj})\}$$

où  $td_{ij}$  est le  $i$ ème terme du document  $D_j$ , et  $a_{ij}$  est le degré d'appartenance (une valeur comprise entre 0 et 1) du  $i$ ème terme au document  $D_j$ .

L'évaluation d'une requête  $Q_k$  par rapport à un document  $D_j$ , peut prendre plusieurs formes. Un exemple de requêtes floues est présenté dans le tableau 2.2.

Requête ( $Q_k$ )	$rsv(D_j, Q_k)$	Le résultat de l'évaluation
$ tq_{ki}$	$ rsv(D_j, tq_{ki})$	$ = a_{ij}$
$ tq_{k1} \wedge tq_{k2}$	$ rsv(D_j, tq_{k1} \wedge tq_{k2})$	$ = \min(rsv(D_j, tq_{k1}), rsv(D_j, tq_{k2}))$
$ tq_{k1} \vee tq_{k2}$	$ rsv(D_j, tq_{k1} \vee tq_{k2})$	$ = \max(rsv(D_j, tq_{k1}), rsv(D_j, tq_{k2}))$
$ \neg tq_{ki}$	$ rsv(D_j, \neg tq_{ki})$	$ = 1 - a_{ij}$

Tableau 2.2 – Évaluation de requêtes - modèle booléen/ensembles flous

On constate que la fonction d'appariement permet un classement des documents sélectionnés par le système.

L'inconvénient principal de ce modèle est que le calcul de la valeur de similarité est toujours dominé par les petits poids des termes dans le cas des conjonctions et les grands poids des termes dans le cas des disjonctions.

## 2.5.2 Modèles vectoriels

### 2.5.2.1 Modèle vectoriel de base

La première idée de représenter les documents et les requêtes sous forme de vecteurs de termes pondérés a été proposée par Luhn [Luhn, 1957] à la fin des années 1950. Elle est ensuite développée par Gérard Salton et son équipe [Salton, 1971] [Salton, 1983] dans leur projet SMART (*Salton's Magical Automatic Retriever of Text*). L'idée de base du modèle vectoriel est d'utiliser une représentation géométrique pour classer les documents par ordre de pertinence par rapport à une requête.

Dans le cas du modèle vectoriel, les documents et les requêtes sont représentés sous forme de vecteurs dans l'espace vectoriel engendré par les termes du langage d'indexation [Salton, 1971]. La pertinence d'un document vis-à-vis d'une requête est relative aux positions respectives des vecteurs document et requête dans cet espace. Cette position est estimée par une distance ou une similarité définie sur cet espace. Dans ce modèle, les documents et les requêtes sont considérés de la même façon et représentés par des vecteurs de termes pondérés dans le même espace. Chaque poids d'un terme dans un vecteur document (resp. requête) désigne l'importance de ce terme dans ce document (resp. requête). Ces termes pondérés sont utilisés pour calculer le degré de similarité entre chaque document et la requête de l'utilisateur. Les documents retrouvés sont présentés dans un ordre décroissant de leur degré de similarité correspondant.

Formellement, dans un modèle vectoriel, on suppose que le poids  $wd_{ij}$  (resp.  $wq_{ik}$ ) associé au terme  $t_i$  dans le document  $D_j$  (resp. la requête  $Q_k$ ) est positif. Les documents et requêtes sont des vecteurs dans un espace vectoriel de dimension  $N$  et définis comme suit :

$$\begin{aligned} D_j &= (wd_{1j}, \dots, wd_{Nj}) \\ Q_k &= (wq_{1k}, \dots, wq_{Nk}) \end{aligned}$$

Le modèle vectoriel estime le degré de pertinence entre un document et la requête par un degré de corrélation entre leurs vecteurs associés. Cette corrélation peut être spécifiée par le calcul de similarité entre vecteurs, et qui peut être exprimée par le produit scalaire suivant :

$$Sim(D_j, Q_k) = \sum_{i=1}^N (wd_{ij} * wq_{ik}) \quad (2.3)$$

Plusieurs fonctions de similarité ont été proposées dans la littérature. En voici quelques-unes des fonctions les plus répandues : les mesures de Cosinus, Jaccard et Dice.

*Mesure de cosinus (intersection normalisée) :*

$$Sim(D_j, Q_k) = \frac{\sum_{i=1}^N (wd_{ij} * wq_{ik})}{\sqrt{\sum_{i=1}^N wd_{ij}^2 * \sum_{i=1}^N wq_{ik}^2}} \quad (2.4)$$

*Mesure de Jaccard (rapport de l'intersection sur l'union) :*

$$Sim(D_j, Q_k) = \frac{\sum_{i=1}^N (wd_{ij} * wq_{ik})}{\sum_{i=1}^N wd_{ij}^2 + \sum_{i=1}^N wq_{ik}^2 - \sum_{i=1}^N (wd_{ij} * wq_{ik})} \quad (2.5)$$

*Mesure de Dice (rapport de l'intersection sur la moyenne arithmétique) :*

$$Sim(D_j, Q_k) = 2 * \frac{\sum_{i=1}^N (wd_{ij} * wq_{ik})}{\sum_{i=1}^N (wd_{ij}^2 + wq_{ik}^2)} \quad (2.6)$$

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

Théoriquement, le modèle vectoriel présente le principal inconvénient lié à l'indépendance mutuelle des termes d'indexation. Wong et al. [Wong et al., 1985] ont proposé un modèle vectoriel généralisé (*Generalized Vector Space Model*) qui lève l'hypothèse d'indépendance des termes. Dans ce modèle chaque terme est représenté par un vecteur dans un espace vectoriel dont les axes sont orthogonaux par construction : les axes sont en fait les produits logiques des termes d'indexation. Un document est représenté par la moyenne des vecteurs représentant les termes qu'il contient.

### 2.5.2.2 Modèle connexionniste

Ce type de modèle se base sur les fondements des réseaux de neurones biologiques, tant pour la modélisation des documents et des informations descriptives associées (termes, auteurs, mots clés, etc.), que pour la mise en oeuvre du processus de recherche d'information. Ce modèle peut être vu comme un modèle vectoriel récurrent non linéaire, les neurones formels représentent des objets de la recherche d'information. Un réseau de neurone formel est construit à partir des représentations initiales des documents et de la requête. Le mécanisme de recherche d'information est fondé sur le principe de propagation de valeurs depuis les neurones descriptifs de la requête vers ceux des documents, à travers les connexions du réseau. Les résultats sont présentés à l'utilisateur selon le niveau d'activation des neurones documents. Le modèle connexionniste est connu pour sa capacité d'apprentissage, ce qui permet aux SRI de devenir adaptatifs.

Actuellement, plusieurs modèles basés sur le principe des réseaux de neurones sont utilisés en recherche d'information [Boughanem, 1992] [Boughanem and Tamine, 2004] [Crestani, 1994] [Kwok, 1989]. Les modèles RI basés sur les réseaux de neurones sont une solution pour combler les lacunes des modèles vectoriels. La notion de réseau est convenable pour représenter les relations et associations qui existent entre les termes (ex. synonymie, voisinage, ...), entre les documents (ex. similitude, référence, ...), et enfin entre les termes et les documents (exemple, fréquence, poids, ...). Cependant, il n'existe pas de représentation unique d'un réseau de neurones pour la recherche d'information, c'est au constructeur du modèle de le définir, et ce en identifiant les éléments suivants :

- les différentes couches<sup>1</sup> du réseau (couche d'entrée, de sortie, intermédiaires, etc.)
- les neurones de chaque couche,

---

<sup>1</sup>Une couche est un ensemble de neurones formels représentant un concept donné (requête, termes, documents, etc.)

- la fonction d’entrée de chaque neurone,
- la fonction de sortie de chaque neurone,
- les liens entre les neurones et leurs poids associés.

Les systèmes de recherche d’information utilisant l’approche connexionniste peuvent être répartis en deux catégories :

1. ceux qui sont basés sur la carte auto-organisatrice de Kohonen [Kohonen, 1989] (en anglais, SOM pour *Self Organization Map*) . Des modèles d’auto-organisation, inspirés par l’organisation corticale des vertébrés, ont été proposés dès les années 1980, notamment par Kohonen [Kohonen, 1989] et Lelu et François [Lelu and François, 1992]. Les Cartes auto-organisatrices de Kohonen permettent de résoudre des problèmes de classification.

Dans les années 1990, Kohonen, a proposé diverses variantes pour la classification dont les algorithmes de quantification vectorielle à apprentissage [Kohonen et al., 1996]. Le modèle de Kohonen est en général un modèle à deux dimensions. Chaque neurone de la couche d’entrée est relié à chaque neurone de la carte de Kohonen. Le principe d’apprentissage correspond à un apprentissage non supervisé, c’est-à-dire, qu’aucune intervention humaine n’est requise, et consiste en le calcul, à partir d’un vecteur d’entrée, d’une distance avec tous les neurones de la carte. Ensuite, un neurone vainqueur sera sélectionné (neurone dont le vecteur poids est le plus proche du vecteur d’entrée) et les poids du neurone gagnant seront ajustés pour qu’il soit plus proche du neurone d’entrée. Pour plus de détails sur le processus de classification et d’apprentissage du modèle de Kohonen, le lecteur peut se reporter à [Kohonen, 1989] [Kohonen et al., 2000] [Mothe, 2000].

2. ceux basés sur les réseaux à couches [Boughanem, 1992] [Boughanem, 2000] [Kwok, 1989] [Mothe, 1994]. Les modèles à couches sont constitués, au minimum, de deux couches qui correspondent à celle recevant les entrées du milieu extérieur (requête de l’utilisateur) et à celle fournissant les résultats (documents) ; les autres couches, si elles existent, sont appelées couches intermédiaires ou cachées. Les connexions sont orientées de la couche d’entrée vers la couche de sortie, et sont en général pondérées par des valeurs réelles. La recherche d’information est fondée sur un processus d’activation/ propagation, depuis les neurones de la couche d’entrée (les neurones descriptifs de la requête) vers les neurones de la couche de sortie

(neurones représentant les documents de la collection).

Les modèles à couches, les plus performants de ces dernières années, sont ceux proposés par Kwok [Kwok, 1989] dans le système de recherche d'information PIRCS et par Boughanem [Boughanem, 1992] dans le système de recherche d'information MERCURE (*Modèle de Réseau Connexionniste poUr la recherche d'information*) :

Le réseau à couches construit par Kwok utilise trois couches interconnectées dans le sens requête(Q)-termes(T)-documents(D). Les connexions sont bidirectionnelles et de sens asymétriques [Kwok, 1989]. L'approche de Kwok est fondée sur l'idée que les requêtes et documents sont similaires. Sur cette base, elle reprend des éléments du modèle probabiliste et le modèle du langage pour combiner deux valeurs de pertinence avec un paramètre  $\alpha$ . Les deux valeurs de pertinence concernent la pertinence focalisée sur un document  $D_j$  ( $RSV_{D_j}$ ) et la pertinence focalisée sur la requête  $Q_k$  ( $RSV_{Q_k}$ ). La valeur de pertinence  $Sim(D_j, Q_k)$  d'un document  $D_j$  pour la requête  $Q_k$  est :

$$Sim(D_j, Q_k) = \alpha * RSV_{D_j} + (1 - \alpha) * RSV_{Q_k} \quad (2.7)$$

avec :

$$\begin{aligned} RSV_{D_j} &= \sum_k S(qtf_k/L_q) * w_{dk} \\ w_{dk} &= \log [tf_k/(L_d - tf_k) * (N_w - L_d - F_k + tf_k)/(F_k - tf_k)] \\ RSV_{Q_k} &= \sum_k S(tf_k/L_d) * w_{qk} \\ w_{qk} &= \log [qtf_k/(L_q - qtf_k) * (N_w - F_k)/F_k] \end{aligned}$$

$tf_k$ ,  $qtf_k$  sont les fréquences du terme  $t_k$  dans  $D_j$  et dans  $Q_k$  respectivement ;  $L_d = \sum_k tf_k$ ,  $L_q = \sum_k qtf_k$  sont les longueurs du  $D_j$  et de  $Q_k$  ;  $S$  est une fonction sigmoïde ;  $F_k = \sum_{doc\_coll} tf_k$  est la fréquence du terme  $t_k$  dans toute la collection, et  $N_w = \sum_k F_k$  est le nombre de termes de la collection.

L'autre modèle à couches, MERCURE, est caractérisé par un réseau connexionniste à couches interconnectées. Les requêtes, documents et termes sont représentés par des noeuds reliés entre eux par des liens pondérés. L'appariement requête-document est effectué par un processus de propagation de signaux. Le processus est réalisé dans l'ordre suivant :

- La représentation de la requête est de la forme :

$$Q_u^{(t)} = (q_{u1}^{(t)}, \dots, q_{uT}^{(t)})$$

Les poids des termes dans la requête sont affectés aux liens requête-termes.

- Déclenchement de l'évaluation à partir du noeud requête, en envoyant un signal de valeur 1 à travers les liens requête-termes.
- Calcul d'une valeur d'entrée et d'une valeur de sortie à chaque noeud :

$$In(t_i) = q_{ui}^{(t)} \quad Out(t_i) = g(In(t_i))$$

où  $g$  est une fonction sigmoïde.

- Transmission des signaux vers la couche documents. Chaque noeud document calcule une entrée selon la formule :

$$In(d_j) = \sum_{i=1}^T Out(t_i) * d_{ij}$$

puis une valeur d'activation selon la formule :

$$Out(d_j) = g(In(d_j))$$

- Trier les documents répondant à la requête selon l'ordre décroissant de leur valeur d'activation.

MERCURE assure aussi la fonction de reformulation de requête, où deux modes distincts sont proposés : la reformulation directe et indirecte. La reformulation directe consiste à ajouter de nouveaux termes à la requête initiale. Plus précisément, on ajoute les termes actifs, atteints par transfert d'activation à partir des termes de la requête. La reformulation indirecte se base sur l'injection de la pertinence dans le processus de recherche afin d'apprendre les liens requêtes-termes.

### 2.5.2.3 Latent Semantic Indexing (LSI).

Dans ce modèle, les documents sont représentés dans un espace de dimension réduite issu de l'espace initial des termes d'indexation [Deerwester et al., 1990]. Le but de ce modèle est d'aboutir à une représentation conceptuelle des documents où les effets dus à la variation d'usage des termes dans la collection sont nettement atténués. Ainsi, les documents qui partagent des termes co-occurents ont des représentations proches dans

l'espace défini par le modèle. Par conséquent, le système permet de sélectionner des documents même s'ils ne contiennent aucun terme de la requête. Ce modèle se base essentiellement sur la décomposition en valeur singulières, désignée par SVD (*Singular Value Decomposition*) de la matrice représentant, en ligne les termes et en colonne, les documents. Un élément de la matrice représente le poids d'un terme dans un document. La SVD permet d'une part de réduire l'espace des termes d'indexation, et d'autre part, de représenter les documents et les requêtes dans un espace qui ne dépend pas des termes d'indexation mais des concepts contenus dans les documents.

Formellement, la transformation par LSI est présentée comme suit :

1. initialement, les documents sont représentés par des vecteurs de termes,
2. l'ensemble des vecteurs des documents de la collection sont représentés sous forme de matrice  $X$  de dimension  $t \times d$ , où  $t$  est le nombre de termes distincts de la collection, et  $d$  le nombre de documents dans la collection,
3. la SVD permet de décomposer toute matrice rectangulaire en un produit de trois matrices. Ainsi, la matrice  $X$  est transformée par SVD comme suit :

$$X = T_0 \cdot S_0 \cdot D_0^t \quad (2.8)$$

avec :

$T_0$  : une matrice orthogonale de dimension  $t \times m$  ;

$D_0^t$  : une matrice orthogonale de dimension  $m \times d$  ;

$S_0$  : une matrice diagonale de dimension  $m \times m$ , les valeurs sur la diagonale sont les valeurs propres de  $X$  et sont par convention toutes positives et ordonnées par ordre décroissant sur la diagonale ;

$m$  : est le rang de la matrice  $X$  ( $\leq \min(t, d)$ ).

L'intérêt de la SVD est qu'elle permet d'utiliser une stratégie plus simple pour l'optimisation de la matrice  $X$ , en se basant sur des matrices réduites. Ainsi, les  $k$  plus grandes valeurs propres sont supposées suffisantes pour représenter presque toute l'information de la matrice  $X$ . Concrètement, toutes les valeurs propres  $i$  ( $i > k$ ) sont supposées nulles, et l'équation (2.8) est réécrite en utilisant la matrice  $S$  de dimension  $k \times k$ , approximation de  $S_0$  réduite aux  $k$  premières dimensions. Le résultat de la nouvelle transformation est donné par le modèle réduit suivant :

$$X \approx X' = T \cdot S \cdot D^t \quad (2.9)$$

avec :

$T$  : une matrice orthogonale réduite de dimension  $t \times k$  ( $k \leq m$ ),

$D^t$  : une matrice orthogonale réduite de dimension  $k \times d$ ,

$S$  : une matrice diagonale réduite de dimension  $k \times k$ ,

$m$  : est le rang de la matrice  $X$  ( $\leq \min(t, d)$ ).

4. une requête  $X_q$  est, comme tout document, un ensemble de termes. Elle peut être représentée dans le nouveau espace des documents comme suit :

$$Q = X_q.T.S^{-1} \quad (2.10)$$

où  $Q$  est le vecteur des mots de la requête, pondéré par les termes appropriés,  $S^{-1}$  est la matrice inverse de  $S$ .

5. Une valeur de similarité est ensuite calculée entre le vecteur requête  $Q$  et chaque document de la collection, tous deux représentés dans le nouvel espace vectoriel.

Plusieurs applications de ce modèle ont été proposées en recherche d'information, mais également pour le filtrage d'information et la recherche documentaire multilingue [Dumais et al., 1996] [Foltz and Dumais, 1992].

### 2.5.3 Modèles probabilistes

Le premier modèle probabiliste a été proposé par Maron et Kuhns [Maron and Kuhns, 1960] au début des années 1960. Ils proposent de modéliser le processus de sélection des documents dans un SRI en se basant sur la théorie des probabilités. Le principe de base du modèle probabiliste consiste à présenter les résultats de recherche d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête. Robertson [Robertson, 1977] résume ce critère d'ordre par le "principe de classement probabiliste", désigné aussi par PRP (*Probability Ranking Principle*).

Dans les sous-sections suivantes, nous présentons quelques modèles probabilistes les plus répandus, tels que le modèle BIR et le modèle du langage.

### 2.5.3.1 Modèle BIR

Le modèle BIR (*Binary Independence Retrieval*), comme dans tous les modèles probabilistes, cherche à estimer la probabilité qu'un document  $D_j$  soit pertinent pour une requête  $Q_k$ . Pour estimer cette probabilité (notée  $P(R|Q_k, D_j)$ ), une hypothèse sur la distribution des termes dans les documents est considérée. L'hypothèse consiste à considérer que la distribution des termes dans les documents pertinents et non pertinents est différente. Cette hypothèse, appelée aussi "*clustering hypothesis*", est validée expérimentalement par Van Rijsbergen et Sparck-Jones [Rijsbergen and Sparck-Jones, 1973]. L'idée de base du modèle BIR est de représenter les termes des documents par des valeurs binaires (0 ou 1), un terme apparaît dans un document ou non. Ainsi, un document  $D_j$  peut être représenté par un vecteur de termes pondérés  $\omega_i$ ,  $\omega_i \in \{0, 1\}$  et  $i \in \{1, \dots, N\}$ , où  $N$  est le nombre de termes du document.

L'estimation de la probabilité  $P(R|Q_k, D_j)$  du document  $D_j$ , revient à calculer le odds<sup>2</sup> de la pertinence d'un document  $D_j$  vis-à-vis de la requête  $Q_k$ . Ceci, peut être obtenu en utilisant le théorème de Bayes comme suit :

$$O(R|Q_k, D_j) = \frac{P(R|Q_k, D_j)}{P(\bar{R}|Q_k, \vec{x})} = \frac{P(R|Q_k)}{P(\bar{R}|Q_k)} \cdot \frac{P(D_j|R, Q_k)}{P(D_j|\bar{R}, Q_k)} \quad (2.11)$$

En se basant sur l'hypothèse de dépendance liée (*Linked dependence assumption*), le rapport entre les probabilités que le vecteur de termes du document  $D_j$  appartienne aux sous ensembles des documents pertinents et non pertinents peut être remplacé par le produit de ce rapport associé à chaque terme. L'équation (2.11) peut être remplacée par :

$$O(R|Q_k, D_j) = O(R|Q_k) \prod_{i=1}^N \frac{P(\omega_i|R, Q_k)}{P(\omega_i|\bar{R}, Q_k)} \quad (2.12)$$

En se basant sur la présence ou l'absence d'un terme dans le document, l'équation (2.12) est réécrite comme suit :

$$O(R|Q_k, D_j) = O(R|Q_k) \prod_{\omega_i=1} \frac{P(\omega_i=1|R, Q_k)}{P(\omega_i=1|\bar{R}, Q_k)} \cdot \prod_{\omega_i=0} \frac{P(\omega_i=0|R, Q_k)}{P(\omega_i=0|\bar{R}, Q_k)} \quad (2.13)$$

---

<sup>2</sup>Le "odds" est une formule statistique souvent utilisée dans le modèle probabiliste, le odds peut être défini par  $O(A) = \frac{P(A)}{P(\bar{A})} = \frac{P(A)}{1-P(A)}$ .

La fonction (2.13) peut être réécrite selon les substitutions suivantes :  $p_{ik} = P(\omega_i = 1|R, Q_k)$  : la probabilité que le terme  $t_i$  soit présent dans le document pertinent et  $p_{ik} = P(\omega_i = 1|\bar{R}, Q_k)$  : la probabilité que le terme  $t_i$  soit présent dans le document non pertinent et en supposant que  $p_{ik} = q_{ik}$ , pour tous les termes  $t_i$  qui n'appartiennent pas à  $Q_k$  :

$$O(R|Q_k, D_j) = O(R|Q_k) \prod_{t_i \in D_j \cap Q_k} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \cdot \prod_{t_i \in Q_k} \frac{1 - p_{ik}}{1 - q_{ik}} \quad (2.14)$$

Dans la pratique, le plus important est comment classer les documents et non pas comment calculer les valeurs exactes des probabilités (ou odds) de pertinence des documents. Partant de cette idée, le second produit de l'équation (2.14) et aussi la valeur de  $O(R|Q_k)$  sont des constantes pour la requête  $Q_k$  ; seulement le premier produit de l'équation sera retenu pour le classement des documents. Si on calcule le logarithme de ce produit, la valeur de pertinence d'un document  $D_j$  pour la requête  $Q_k$  (i.e. la similarité entre  $D_j$  et  $Q_k$ ) est donnée par :

$$Sim(D_j, Q_k) = \sum_{t_i \in D_j \cap Q_k} \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \quad (2.15)$$

Le modèle BIR ne peut être utilisé que si les paramètres  $p_{ik}$  et  $q_{ik}$  sont estimés pour tous les termes de la requête. Robertson et Sparck-Jones proposent quatre principes pour estimer ces paramètres. Deux principes se basent sur les hypothèses concernant l'indépendance des termes et les deux autres concernent l'ordre des documents :

- I1** : la distribution des termes dans les documents pertinents est indépendante et leur distribution dans tous les documents est indépendante.
- I2** : la distribution des termes dans les documents pertinents est indépendante et leur distribution dans les documents non pertinents est indépendante.
- O1** : la probabilité de pertinence est basée seulement sur la présence des termes recherchés dans les documents.
- O2** : la probabilité de pertinence est basée simultanément sur la présence des termes recherchés dans les documents et sur leur absence dans les documents.

Les différentes combinaisons de ces quatre hypothèses ont été testées dans la pratique. La combinaison I2-O2 a permis d'obtenir des résultats très intéressants, et elle est présentée ci-dessous.

Soit une collection de documents de taille  $N$  et une requête  $Q_k$ . Les paramètres  $p_{ik}$  et  $q_{ik}$  peuvent être estimés pour chaque terme de la requête en divisant la collection de documents en quatre parties, présentées par la table de contingence 2.3 suivante :

	Pertinent	Non pertinent	
$\omega_i = 1$	$r$	$n - r$	$n$
$\omega_i = 0$	$R - r$	$(N - n) - (R - r)$	$N - n$
	$R$	$N - R$	$N$

Tableau 2.3 – Table de contingence des occurrences vs. pertinences des termes

Dans le tableau 2.3 les  $N$  documents de la collections peuvent être divisés en deux parties : (i) il y a  $n$  documents qui contiennent le terme de la requête, (ii) il y a  $R$  documents pertinents pour la requête. Le nombre de documents pertinents contenant les termes de la requête est égal à  $r$ .

*Robertson et Sparck-Jones* supposent que la distribution des termes est indépendante dans les deux ensembles de documents pertinents et non pertinents (I2), et supposent aussi que les valeurs de pertinence des documents existent et la probabilité de pertinence est basée sur la présence et l'absence des termes de la requête dans les documents (O2). Ainsi, le paramètre  $p_{ik}$  peut être estimé par  $r/R$  et  $q_{ik}$  par  $(n - r)/(N - R)$ . Si on restreint l'équation (2.15) à tenir compte des termes séparément, alors le poids d'un terme est donné comme suit :

$$\omega_i = \log \frac{\frac{r}{(R-r)}}{\frac{(n-r)}{((N-n)-(R-r))}} \quad (2.16)$$

L'équation (2.16) est toujours référencée dans la littérature par la formule de Robertson/ Sparck-jones. Cependant, l'équation (2.16) devient indéfinie dans le cas où aucune information pertinente n'est disponible ( $R=r=0$ ). Pour éviter ce cas, une petite valeur (0.5) est rajoutée au numérateur et dénominateur de l'équation. L'équation (2.16) est ainsi réécrite comme suit :

$$\omega_i = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/((N-n)-(R-r)+0.5)} \quad (2.17)$$

Le modèle BIR est très dépendant du principe du théorème de Bayes. Si aucune information pertinente n'est disponible, le modèle BIR est moins performant par rapport aux modèles basés sur la formule standard *TF-IDF*.

En outre, le modèle probabiliste est largement répandu dans le domaine de la RI. Quelques systèmes s'inspirent directement des fondements du modèle probabiliste, tel que le modèle 2-poisson développé par Robertson et Walker [Robertson and Walker, 1994], le système Okapi [Robertson et al., 1994] et les modèles basés sur la modélisation du langage [Ponte and Croft, 1998], ou sur son extension, comme le modèle Inquiry basé sur les réseaux inférentiels [Callan et al., 1992] [Turtle, 1991].

### 2.5.3.2 Modèle du langage

Le modèle de langue est emprunté de la linguistique informatique. L'objectif d'un modèle de langue est de capter les régularités linguistiques d'une langue, en observant la distribution des mots, successions de mots, dans la langue donnée. Le modèle de langue désigne une fonction de probabilité qui assigne à chaque séquence de mots une probabilité.

En RI, l'idée de base des modèles de langues est de déterminer la probabilité (notée  $P(Q_j|D_k)$ ) que la requête puisse être générée par le modèle de langue du document.

Pour estimer la probabilité  $P(Q_k|D_j)$ , soient les hypothèses suivantes :

- une requête  $Q_k$  est exprimée comme une suite de termes :  $Q_k = q_{1k}, \dots, q_{nk}$  ;
- les termes de la requête sont indépendants ;
- tous les termes d'indexation (ceux présents dans la requête et ceux absents de la requête) sont considérés lors de l'estimation de  $P(Q_k|D_j)$ , soit  $l$  le nombre de ces termes.

Etant donnée la requête  $Q_k$  composée de l'ensemble des termes  $q_{1k}, \dots, q_{nk}$ , et que les termes  $q_{(n+1)k}, \dots, q_{lk}$  sont absents de la requête. La probabilité  $P(Q_k|D_j)$  peut être exprimée comme suit :

$$Sim(D_j, Q_k) = P(Q_k|D_j) = \prod_{q_{ik} \in Q_k} P(q_{ik}|D_j) \times \prod_{q_{ik} \notin Q_k} (1 - P(q_{ik}|D_j)) \quad (2.18)$$

Ponte et Croft [Ponte and Croft, 1998] ont été les premiers à introduire le modèle de langage en RI. Dans l'approche qu'ils ont proposée, la probabilité d'observer le terme  $q_{ik}$  dans le document  $D_j$  est donnée par :

$$P(q_{ik}|D_j) = \begin{cases} p_{MLE}(q_{ik})^{(1-R_{q_{ik},D_j})} \times p_{avg}(q_{ik})^{R_{q_{ik},D_j}} & \text{si } p_{MLE}(q_{ik}) > 0 \\ p_{corpus}(q_{ik}) & \text{sinon} \end{cases} \quad (2.19)$$

où  $p_{MLE}(q_{ik}) = c(q_{ik})/N_t$ , avec  $c(q_{ik})$  est le nombre de fois où le terme  $q_{ik}$  se trouve dans le corpus (collection de documents), et  $N_t$  le nombre total des termes du corpus;  $p_{corpus}(q_{ik})$  est la probabilité du terme  $q_{ik}$  dans le corpus;  $p_{avg}(q_{ik})$  est la probabilité moyenne du terme  $q_{ik}$  dans les documents qui le contiennent;  $R_{q_{ik},D_j}$ , le risque associé à l'utilisation de  $p_{avg}(q_{ik})$  pour le terme  $q_{ik}$  dans le document  $D_j$ . C'est une fonction de la fréquence du terme  $q_{ik}$  dans le document  $D_j$  et dans le corpus.

L'approche proposée par Ponte et Croft est non paramétrique, c'est à dire, la probabilité  $P(q_{ik}|D_j)$  se base complètement sur les fréquences observées dans le corpus, et il n'est pas nécessaire d'apprendre aucun paramètre spécifique. Ponte et Croft rapportent que cette approche donne de meilleures performances que le modèle vectoriel. D'autres approches ont été proposées peu après, Song et Croft [Song and Croft, 1999] interpole la probabilité du terme  $q_{ik}$  dans le document  $D_j$  avec la probabilité du terme dans le corpus comme suit :

$$P(q_{ik}|D_{j_{interp}}) = \lambda_{D_j} \cdot P(q_{ik}|D_j) + (1 - \lambda_{D_j}) \cdot p_{corpus}(q_{ik}) \quad (2.20)$$

où la valeur optimale du paramètre  $\lambda_{D_j}$  est déterminée empiriquement.

A l'instar de Song et Croft, Hiemstra [Hiemstra, 2002] propose une interpolation entre le modèle du document  $P(q_{ik}|D_j)$  et le modèle du corpus  $p_{corpus}(q_{ik})$ . Plutôt que de déterminer les contributions relatives des modèles de manière empirique, et de manière fixe pour l'ensemble de documents, il introduit un coefficient  $\lambda_i$  qui estime l'importance d'un terme  $q_{ki}$  de la requête. Les coefficients sont appris pour chaque terme d'une requête; la probabilité interpolée est quant à elle exprimée par :

$$P(q_{ik}|D_{j_{interp}}) = \lambda_i \cdot P(q_{ik}|D_j) + (1 - \lambda_i) \cdot p_{corpus}(q_{ik}) \quad (2.21)$$

Le *Relevance feedback* est employé pour créer un ensemble de documents pertinents utilisés par l'algorithme d'entraînement des  $\lambda_i$ . Hiemstra [Hiemstra, 2002] apporte une amélioration en performances par rapport au modèle probabiliste classique qui incorpore le *relevance feedback*.

## 2.6 Reformulation de requêtes en RI

La requête initiale de l'utilisateur est souvent représentée par une liste de termes très réduite. Cette liste manque souvent de termes intéressants pouvant exprimer effectivement le besoin en information de l'utilisateur. Ceci a plusieurs raisons, la plus importante vient de la diversité du vocabulaire de la collection de documents. En effet, l'utilisateur ne connaît pas le vocabulaire utilisé dans les documents qu'il recherche. Pour pallier ces problèmes, les systèmes de recherche d'information proposent des techniques, appelées reformulation de requêtes, pour affiner et améliorer automatiquement la requête initiale de l'utilisateur, en rajoutant de nouveaux termes ou en supprimant des termes inutiles.

La reformulation de requête a été traitée selon deux classes d'approches. La première, appelée la réinjection de la pertinence (*relevance feedback*) est basée sur les documents sélectionnés par le système et jugés par l'utilisateur. La seconde est basée sur l'utilisation des liens sémantiques ou "statistiques" établis entre les termes. Ces liens peuvent être construits manuellement par un expert (exemple, thésaurus *WordNet*) ou automatiquement. Dans ce dernier cas, ces liens peuvent être construits à partir des documents retrouvés par le système, on parle alors de reformulation par contexte local, ou à partir de la collection entière de documents, on parle alors de reformulation par contexte global.

### 2.6.1 Approches basées sur le *relevance feedback*

Le *relevance feedback*, comme a été souligné dans la section 2.2, consiste à construire une nouvelle requête à partir des documents retrouvés et marqués pertinents et non pertinents par l'utilisateur. Plusieurs techniques basées sur le principe *relevance feedback*, ont été proposées dans la littérature [Boughanem et al., 1999b] [Kwok, 1989]

[Robertson and Walker, 2000] [Rocchio, 1971]. Nous décrivons dans les sous-sections suivantes l'algorithme de Rocchio [Rocchio, 1971] et les méthodes d'expansion de requêtes proposées par Robertson et Sparck-Jones dans Okapi [Robertson and Walker, 2000]. Nous nous limitons à ces techniques de reformulation de requêtes, car elles sont réutilisées dans plusieurs modèles de filtrage d'information décrits dans la deuxième partie.

### 2.6.1.1 Algorithme de Rocchio

L'algorithme de reformulation de requêtes développé par Rocchio au milieu des années 60 [Rocchio, 1966] [Rocchio, 1971], est l'un des algorithmes d'expansion de requêtes les plus utilisés dans le domaine de la RI. Il permet de construire une requête performante à partir de la requête initiale et d'un ensemble de documents jugés pertinents et non pertinents. La forme standard de l'algorithme de Rocchio est donnée comme suit :

$$Q_{nlle} = \alpha Q_{init} + \frac{\beta}{|D_r|} \sum_{\forall D_j \in D_r} D_j - \frac{\gamma}{|D_n|} \sum_{\forall D_j \in D_n} D_j \quad (2.22)$$

où :

$|\cdot|$  désigne le cardinal d'un ensemble,

$Q_{nlle}$  : le vecteur de la nouvelle requête,

$Q_{init}$  : le vecteur de la requête initiale,

$D_r$  : l'ensemble des documents restitués et jugés pertinents par l'utilisateur,

$D_n$  : l'ensemble des documents restitués et jugés non pertinents par l'utilisateur,

$D_j$  : le jème document d'un ensemble,

$\alpha$ ,  $\beta$  et  $\gamma$  : des paramètres constants,

Plusieurs autres méthodes de reformulation de requêtes se sont inspirées du principe de l'algorithme de Rocchio. On distingue la :

#### Méthode de régulation de Ide [Ide, 1971] :

$$Q_{nlle} = \alpha Q_{init} + \beta \sum_{\forall D_j \in D_r} D_j - \gamma \sum_{\forall D_j \in D_n} D_j \quad (2.23)$$

#### Méthode *Ide-Dec-Hi* :<sup>3</sup>

$$Q_{nlle} = \alpha Q_{init} + \beta \sum_{\forall D_j \in D_r} D_j - \gamma \max_{non-pertinent}(D_j) \quad (2.24)$$

---

<sup>3</sup>Decrease using Highest ranking non relevant documents

où,  $\max_{non-pertinent}(D_j)$  représente le vecteur de document non pertinent de score le plus élevé.

Le principal avantage de l'utilisation des méthodes de relevance feedback est donné par leur simplicité et leurs performances. La simplicité, car les requêtes sont modifiées à partir des poids des termes calculés directement des documents restitués par le système. De point de vue performances, les expérimentations basées sur ce type de méthodes ont donné de très bons résultats.

### 2.6.1.2 Expansion de requête dans Okapi

La méthode de reformulation de requête dans Okapi se base sur la formule de poids (équation (2.17)) proposée par Robertson et Sparck-Jones. L'idée de base de la méthode consiste à extraire tous les termes des documents restitués et jugés pertinents. Ces termes sont ensuite ordonnés en fonction de leurs valeurs obtenues par des méthodes spécifiques. Une méthode spécifique est basée sur la formule de poids de l'équation (2.17).

Deux méthodes spécifiques ont été proposées par Robertson et Walker [Robertson and Walker, 2000] pour la sélection des termes composant la nouvelle requête : (1) la première méthode consiste à sélectionner un nombre limité de termes (généralement une valeur fixe) dans un ordre relatif à la valeur de sélection du terme (*TSV* pour *Term Selection Value*) (voir section 3.10.2); (2) la deuxième méthode est basée sur les statistiques des nouveaux termes significatifs, et nécessite un seuil absolu pour extraire les termes constituant la nouvelle requête (voir la section 3.10.2). Ces deux méthodes seront détaillées dans le prochain chapitre, où elles sont adaptées pour l'apprentissage des besoins en information des utilisateurs dans le cadre des systèmes de filtrage d'information.

## 2.6.2 Approches basées sur le contexte local/global

Les approches basées sur le contexte local/global consistent à construire automatiquement les liens entre les termes en se basant sur la co-occurrence de ces termes. Ces liens peuvent être construits à partir des documents sélectionnés par le système (contexte local) ou à partir de la collection entière de documents (contexte global). Dans ce qui suit, nous présentons deux méthodes de reformulations de requête, l'analyse du contexte local et une méthode basée sur un thésaurus de similarité.

### 2.6.2.1 Analyse par le contexte local

La méthode d'analyse du contexte local (Local Context Analysis (LCA)) a été développée par Croft et Xu [Croft and Xu, 1995] et utilisée dans leur système de recherche Inquiry. A la différence avec les autres techniques de reformulation, elle utilise la règle de *passage*. Elle consiste à modifier la requête initiale de l'utilisateur à partir des proportions des contenus des meilleurs documents retrouvés. Le principe de base de la méthode est décrit comme suit :

- (1) sélectionner  $n$  passages des  $n$  meilleurs documents retrouvés (un passage est une classe de termes de taille fixe, par exemple 300 termes [Callan, 1995]),
- (2) extraire des concepts (groupes de mots) à partir de ces passages. Ces concepts sont ensuite ordonnés selon l'équation (2.25),
- (3) Les 70 meilleurs termes des concepts ordonnés sont utilisés dans l'expansion de la requête initiale.

$$bel(Q, c) = \prod_{t_i \in Q} (\delta + \log(af_{c,t_i})idf_c / \log(n))^{idf_i} \quad (2.25)$$

où :

$$af_{c,t_i} = \sum_{j=1}^{j=n} ft_{ij}fc_j$$

$$idf_i = \max(1.0, \log 10(N/N_i)/5.0)$$

$$idf_c = \max(1.0, \log 10(N/N_c)/5.0)$$

$c$  est un concept

$ft_{ij}$  est le nombre d'occurrences du terme  $t_i$  dans le passage  $p_j$ ,

$fc_j$  est le nombre d'occurrences du concept  $c$  dans le passage  $p_j$ ,

$N$  est le nombre de passages dans la collection,

$N_i$  est le nombre de passages contenant le terme  $t_i$ ,

$N_c$  est le nombre de passages contenant le concept  $c$ ,

$\delta$  est une constante (égale à 0.1) qui permet d'éviter des valeurs nulles.

### 2.6.2.2 Thésaurus de similarité

Les méthodes d'expansion de requête basées sur un thésaurus de similarité consistent à rajouter à la requête des termes issus d'un thésaurus de similarité. Ceci revient à calculer des valeurs de similarité entre les termes de la requête et ceux d'un thésaurus donné. Les  $k$  meilleurs termes de similarité la plus élevée sont rajoutés à la requête initiale.

Un thésaurus de similarité est une matrice de similarité terme-terme [Qui and Frei, 1993]. Pour construire un thésaurus de similarité, au lieu de représenter un document par un vecteur de termes, on représente chaque terme  $t_i$  par un vecteur de documents dans un espace de vecteurs de documents, par exemple,  $\vec{t}_i = (d_{i1}, \dots, d_{in})$ . Avec,  $d_{ik}$  signifie le poids du document  $D_k$  par rapport au terme  $t_i$  et  $n$  est le nombre de documents dans la collection. La formule de pondération utilisée par Qui et Frei [Qui and Frei, 1993] pour calculer le poids  $d_{ik}$  est la suivante :

$$d_{ik} = \frac{\left(0.5 + 0.5 \frac{ff(d_k, t_i)}{maxff(t_i)}\right) \times iif(d_k)}{\sqrt{\sum_{j=1}^n \left( \left(0.5 + 0.5 \frac{ff(d_j, t_i)}{maxff(t_i)}\right) \times iif(d_j) \right)^2}} \quad (2.26)$$

avec :

$ff(d_k, t_i)$  : la fréquence du terme  $t_i$  dans le document  $D_k$ ,

$iif(d_k) = \log(m/|d_k|)$  : la fréquence inverse des  $D_k$ ; avec  $m$  le nombre de termes dans la collection et  $|d_k|$  est le nombre de termes dans le document  $d_k$ ,

$maxff(t_i)$  : la fréquence maximale du terme  $t_i$  dans toute la collection,

Le thésaurus de similarité est construit en calculant la similarité entre toutes les paires de termes  $(t_i, t_j)$ . La similarité entre deux termes est exprimée par le produit scalaire suivant :

$$Sim(t_i, t_j) = \vec{t}_i \cdot \vec{t}_j = \sum_{k=1}^n d_{ik} \cdot d_{jk} \quad (2.27)$$

Grâce à ce thésaurus, l'expansion d'une requête  $Q$  consiste à calculer une similarité, notée  $Sim_{Qt}(Q, t_j)$ , entre chaque terme du thésaurus et la requête  $Q$ , et non pas avec chaque terme de la requête  $Q$ . La formule utilisée pour calculer cette similarité est la suivante :

$$Sim_{qt}(Q, t_j) = \sum_{t_i \in Q} q_i \cdot Sim(t_i, t_j) \quad (2.28)$$

où,  $q_i$  est le poids du terme  $t_i$  dans la requête  $Q$ .

Ainsi, tous les termes de la matrice du thesaurus de similarité, associés aux termes de la requête, peuvent être ordonnés par ordre décroissant de leur valeur de similarité  $Sim_{qt}$ . Les poids des termes sélectionnés sont donnés par la formule de pondération suivante :

$$wq_i = \frac{Sim_{qt}(Q,t_i)}{\sum_{t_i \in Q} q_i} \quad (2.29)$$

Le nombre de termes à rajouter à la requête est un paramètre important en RI. Plusieurs travaux ont donné des indications sur ce nombre. Des expérimentations effectuées dans [Harman, 1992c] montrent que la performance du système est obtenue lorsque la requête est construite entre 20 et 40 termes. Ces valeurs ont été confirmées dans les travaux de Boughanem [Boughanem et al., 1999a].

## 2.7 Méthodes d'évaluation des SRI

L'évaluation d'un système de recherche d'information peut être appréhendée selon deux aspects : un aspect efficacité et un aspect efficacie. L'aspect efficacité dépend de l'évaluation cognitive de l'utilisateur, tels que la facilité d'utilisation du système, rapidité d'accès, temps de réponse à une requête, présentation des résultats, etc. L'aspect efficacie concerne la capacité du système à sélectionner le maximum de documents pertinents et un minimum de documents non pertinents. Nous nous intéressons dans cette section à présenter l'aspect efficacie, qui est souvent mesuré par deux paramètres Rappel/Précision. Un aperçu bien détaillé sur le deuxième aspect est présenté dans [Kraaij, 2004] [Rijsbergen, 1979]. L'évaluation de l'efficacie d'un SRI repose en général sur les trois principaux éléments suivants :

- une collection de document de test,
- un ensemble de requête,
- une liste de documents pertinents pour chaque requête, produite par des experts.

A partir de ces trois éléments, nous pouvons mesurer les taux de performance des SRI par différentes mesures d'évaluation que nous décrivons ci-dessous.

## 2.7.1 Mesures de précision et rappel

Ce sont les deux métriques les plus utilisées pour évaluer la performance d'un système de recherche d'information. Pour présenter ces deux mesures, la figure 2.4 introduit le partitionnement de l'ensemble  $B$  des documents restitués par le SRI en deux sous ensembles : un sous-ensemble de documents non pertinents et un sous-ensemble de documents pertinents.

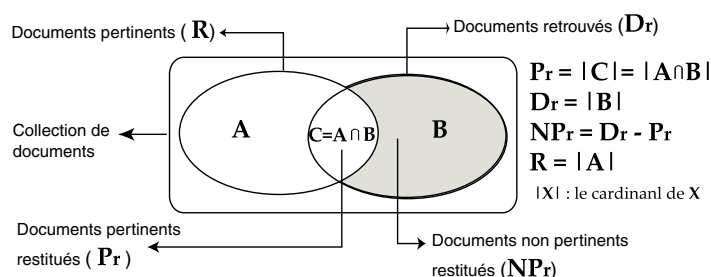


Figure 2.4 – Partition de la collection pour une requête

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

- **Taux de rappel** : le rappel mesure la capacité du système de retrouver tous les documents pertinents répondant à une requête. Autrement dit, il mesure le pourcentage de documents pertinents, selon une liste de référence (ensemble des documents pertinents de la collection), que le système a trouvé :

$$rappel = \frac{P_r}{R} \quad (2.30)$$

- **Taux de précision** : la précision mesure la capacité du système de rejeter tous les documents pertinents à une requête. Autrement dit, elle mesure le pourcentage de documents retrouvés qui sont pertinents :

$$precision = \frac{P_r}{D_r} \quad (2.31)$$

Pour chaque requête soumise à un système de recherche, un tableau de précision-rappel peut être construit. Le tableau 2.4 illustre les calculs de précision et rappel pour les 6 premiers documents trouvés pour une requête donnée, pour laquelle la collection contient 4 documents pertinents. Le premier document retrouvé n'est pas jugé pertinent, et donc la précision et le rappel pour ce document sont nuls. Le deuxième document retrouvé par le système est pertinent. La précision pour les deux premiers documents est alors 1 document

<i>Rang du document restitué</i>	<i>Pertinent</i>	<i>rappel</i>	<i>précision</i>
1	P	0.25	1.00
2	NP	0.25	0.50
3	P	0.50	0.67
4	NP	0.25	0.50
5	P	0.75	0.60
6	NP	0.75	0.50

Tableau 2.4 – Exemple de calculs de rappel et précision

pertinent retrouvé divisé par 2 documents retrouvés. Le rappel est 1 document retrouvé divisé par 4 documents de la collection jugés pertinents.

*P désigne Pertinent, NP désigne non pertinent*

Une façon d'évaluer un système est de tracer une courbe de précision-rappel. Ainsi, si le résultat de recherche dépend d'un certain paramètre, par exemple le rang d'un document restitué, alors pour chaque valeur du paramètre les valeurs de rappel et précision peuvent être calculées. Si  $\lambda$  est ce paramètre, alors  $P_\lambda$  exprime la précision,  $R_\lambda$  le rappel, et la valeur de précision-rappel peut être représentée par le point  $(R_\lambda, P_\lambda)$ . Une liste ordonnée de paires précision-rappel peut être illustrée par une courbe, appelée courbe de précision-rappel (voir figure 2.5).

Le système parfait trouverait seulement les documents pertinents, avec une précision et un rappel de 100%. En pratique, ces deux taux varient en sens inverse, la précision diminue au fur et à mesure que le rappel augmente. La figure 2.5 est un exemple d'une courbe typique de précision-rappel de deux requêtes, où l'indice représente la valeur du paramètre  $\lambda$ .

L'évaluation d'un SRI est effectuée sur la base d'une collection de test. La précision moyenne au taux de rappel  $rp$  est calculé comme suit :

$$P(rp) = \sum_{i=1}^{N_q} \frac{P_i(rp)}{N_q}$$

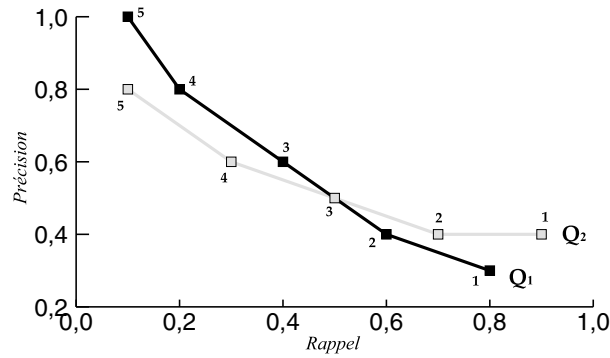


Figure 2.5 – Courbes de précision-rappel de deux requêtes

où,  $N_q$  est le nombre de requêtes et  $P_i(rp)$  la précision de la  $i$ ème requête au niveau de rappel  $rp$ .

Comme les niveaux de rappel ne sont pas unifiés pour l'ensemble des requêtes, on retient dans la littérature, 11 points de rappel standards 0.00 à 1.00 à pas de 0.1 [Tamine, 2000].

## 2.7.2 Interpolation

Pour deux points de rappel,  $i$  et  $j$ ,  $i < j$ , si la précision au point  $i$  est inférieure à celle au point  $j$ , on dit que la précision interpolée à  $i$  égale la précision à  $j$ . Formellement :

$$p'_i = \max(p_i, p_j) \quad \forall i < j \quad (2.32)$$

où  $p'_i$  est la précision interpolée au point de rappel  $i$ , et  $p_i$  est la vraie précision au point de rappel  $i$ . Cette interpolation est encore discutable, mais présente un intérêt dans l'évaluation de systèmes de recherche d'information. Elle permet entre autre de construire des courbes décroissantes plus simple à comparer [Salton, 1983].

## 2.7.3 Mesures combinées

Comme le rappel et la précision, en dépit de leur popularité, ne sont pas toujours les mesures les plus appropriées pour évaluer un système de recherche d'information. Les

chercheurs ont été amenés à s'investir dans d'autres mesures, comme les mesures combinées ou composées. Celles ci se basent toujours sur le principe des mesures de rappel et de précision, mais en les combinant de telle manière à en sortir une valeur représentative. Cependant, ces mesures sont des fonctions ad-hoc et ne peuvent pas être justifiées d'une façon raisonnable [Rijsbergen, 1981]. L'exemple le plus simple de ce type de mesures est la somme de la valeur de rappel et de précision :

$$S(j) = R(j) + P(j) \quad (2.33)$$

où :

$R(j)$  : valeur de rappel au  $j^{eme}$  document restitué,

$P(j)$  : valeur de précision au  $j^{eme}$  document restitué.

D'autres encore plus compliquées [Rijsbergen, 1981] sont données comme suit :

$$V(j) = 1 - \frac{1}{2 \cdot \left(\frac{1}{P(j)}\right) + 2 \cdot \left(\frac{1}{R(j)}\right) - 3} \quad (2.34)$$

$$Q(j) = \frac{R(j) - F(j)}{R(j) + F(j) - 2R(j) \cdot F(j)} \quad (2.35)$$

avec,  $F = \frac{NP_r}{NP}$ , où  $NP$  est le nombre de documents non pertinents de la collection.

En restant dans cette optique de combinaison de rappel et de précision, deux autres mesures ont été proposées : la mesure Harmonique et une mesure d'évaluation appelée  $E$ .

### 2.7.3.1 Mesure Harmonique

Une mesure Harmonique  $H$  est une fonction de paire de valeurs de précision et de rappel :

$$H(j) = \frac{2}{\frac{1}{R(j)} + \frac{1}{P(j)}} \quad (2.36)$$

La mesure Harmonique  $H$  tient ses valeurs dans un intervalle fermé  $[0, 1]$ . Elle est égale à 0 lorsque aucun document pertinent n'est restitué et 1 lorsque tous les documents restitués sont pertinents. La valeur de  $H$  est élevée quand les valeurs de rappel et précision sont élevées. Ainsi, la mesure Harmonique garantit le compromis entre les deux mesures de rappel et précision.

### 2.7.3.2 Mesure d'évaluation "E"

Une autre mesure permettant de combiner le rappel et la précision est proposée par Van Rijsbergen [Rijsbergen, 1981] et elle est appelée mesure d'évaluation E. Le but est de permettre à l'utilisateur de spécifier laquelle des valeurs de précision et de rappel est plus intéressante. La mesure d'évaluation E est définie par :

$$E(j) = 1 - \frac{1+b^2}{\frac{b^2}{R(j)} + \frac{1}{P(j)}} \quad (2.37)$$

où :

$b$  est un paramètre de l'utilisateur, lui permettant de spécifier l'importance du rappel et précision. Si  $b = 1$ , alors  $E(j)$  est le complément de la mesure Harmonique  $H(j)$ . Plus les valeurs de  $b$  sont supérieures à 1, plus la précision est plus intéressante que le rappel, l'inverse est vrai.

## 2.8 Conclusion

Ce premier chapitre a porté essentiellement sur l'étude des systèmes de recherche d'information de manière générale et les modèles de recherche et de représentation d'information de manière particulière. Chacun de ces modèles ou stratégies contribue à la résolution des problèmes inhérents à la recherche d'information.

Nous avons présenté trois classes de modèles de recherche d'information. Le coût d'application d'une théorie de recherche ou de représentation plaide souvent pour le passage du niveau théorique au niveau pratique. En revanche, l'expérimentation de ces stratégies par l'utilisation de bases de référence améliore remarquablement leur représentation, et permet de fournir des compléments expérimentaux pertinents.

Parallèlement à la recherche d'information d'une façon volontaire dans des corpus de documents répertoriés et connus, on assiste de plus en plus au développement de processus permettant la sélection, d'une façon involontaire, d'informations pertinentes dans des flots d'informations provenant de sources différentes. Ce processus dual à la recherche d'information est appelé filtrage d'information.

Nous présentons dans le chapitre suivant les principaux concepts du filtrage d'information ainsi que les quelques travaux réalisés dans ce domaine.



# Chapitre 3

## Collecte passive de l'information : filtrage d'information

### 3.1 Introduction

L'accès à une information pertinente, adaptée aux besoins de l'utilisateur, est un enjeu capital dans le contexte actuel caractérisé par une prolifération massive de ressources hétérogènes. L'accès à ces informations peut se faire de manière délibérée et instantanée via une requête exprimée par l'utilisateur, mais quand ce besoin est permanent, il est inutile de demander à l'utilisateur de reprendre à chaque fois sa requête. Le processus qui permet de répondre à cette attente est le filtrage d'information.

Un Système de Filtrage d'Information (SFI) vise à ramener à l'utilisateur des informations susceptibles de répondre à ses besoins définis au préalable de manière permanente.

Ce chapitre a pour objectif de présenter les systèmes de filtrage d'information d'une façon générale. Pour cela, nous définissons tout d'abord, dans la section 3.2, les concepts de base du filtrage d'information. Nous présentons ensuite la terminologie, ainsi qu'un aperçu historique de ce domaine, respectivement dans les sections section 3.3 et section 3.4. La section 3.5 décrit les grandes familles de filtrage d'information : le filtrage collaboratif et le filtrage basé sur le contenu. La section 3.6 présente le principe de base d'un processus de filtrage cognitif. La section 3.7 liste les principales différences entre le filtrage et la recherche d'information. Dans la section 3.8, nous présentons la problématique du filtrage

d'information cognitif. La section 3.9 décrit les méthodes d'évaluation des systèmes de filtrage d'information. Enfin, dans la section 3.10, nous ferons un tour d'horizon des modèles de filtrage cognitifs les plus répandus. Nous insisterons plus particulièrement sur leurs problèmes d'adaptation.

## 3.2 Définition du filtrage d'information

La RI correspond à la sélection de documents, en réponse à une requête au sein de collections connues. Le Filtrage d'Information (FI) quant à lui, sélectionne des documents provenant de sources, souvent non connues *à priori*.

Le filtrage d'information est un processus dual à la recherche d'information comme le montre N. Belkin dans [Belkin and Croft, 1992]. Il traite des documents provenant de sources dynamiques (News, Email, etc.) et décide à la volée, si le document correspond ou pas aux besoins en information des utilisateurs, besoins exprimés au travers du concept de *profils* utilisateurs. Dans les deux cas, l'objectif est de sélectionner les informations répondant aux besoins en information des utilisateurs.

Un SFI peut donc être vu comme un assistant personnel qui permet à des utilisateurs, ayant défini préalablement leur(s) centre(s) d'intérêt, de recevoir des documents pertinents provenant de sources dynamiques.

En général, les informations dites dynamiques proviennent de sources différentes et sont sujettes à des modifications au cours du temps. Elles peuvent être collectées passivement (News), activement (WWW) ou les deux à la fois.

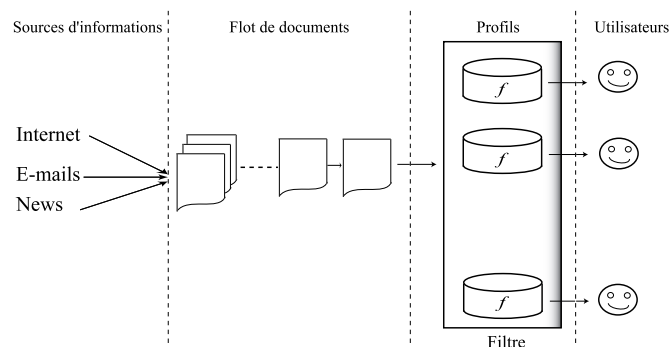


Figure 3.1 – Principe de filtrage d'information

La figure 3.1 illustre un système de filtrage composé d'utilisateurs ayant défini leurs profils et un flot d'informations comportant des documents provenant du *Web*, *Emails* et *News*. A chaque arrivée d'un document, celui-ci est comparé à chaque profil. L'acceptation et donc l'acheminement d'un document vers un utilisateur ou son rejet sont assurés par une fonction de décision, notée  $f$ , propre à chaque profil. A travers cette définition, on peut distinguer les différents concepts clés du filtrage d'information suivants :

- 1. Flot d'informations.** Contrairement à un SRI qui gère une collection statique de documents, un SFI reçoit des documents suivant un flot. En effet, la collection de documents dans le cas du filtrage est représentée par des flots d'informations provenant de sources diverses.
- 2. Profil.** Un profil peut être modélisé par différents types d'information permettant de caractériser un utilisateur ou un groupe d'utilisateurs. Ces types d'informations sont définis selon le contexte dans lequel le profil est utilisé. On peut trouver par exemple des informations sur ces centres d'intérêts, des préférences, des connaissances sur l'utilisateur, etc. Différentes formes de construction d'un profil ont été proposées dans la littérature. Un profil peut être construit de façon explicite à travers une liste de termes pondérés établie par l'utilisateur, de façon supervisée par le système en recueillant les jugements de l'utilisateur sur les documents déjà reçus ou d'une façon implicite par observation du comportement de l'utilisateur lors de ses interactions avec le système. Un état de l'art sur la notion de profil a été présenté par [Bouzeghoub et al., 2004]. Dans notre étude, un profil est représenté, de manière explicite, par une liste de termes pondérés définissant le besoin en information de l'utilisateur. Tout au long du processus de filtrage, le profil est constamment mis à jour à l'aide de méthodes d'apprentissage.
- 3. Fonction de décision.** La notion de fonction de décision n'existe pas dans le domaine de la recherche d'information, elle est spécifique aux systèmes de filtrage d'information et elle est souvent de type seuil. On parle aussi de fonction de seuillage. L'objectif de cette fonction est de décider d'accepter ou de rejeter un document.

### 3.3 Terminologie

Dans un domaine aussi riche que le filtrage d'information, on peut difficilement échapper à une multitude de terminologies différentes. Parfois c'est simplement le résultat des perspectives différentes, où des fois une nouvelle terminologie est nécessaire pour évoquer

subtilement différentes significations. Par exemple, la recherche d'information est parfois utilisée largement pour inclure le filtrage d'information. Le filtrage d'information est alternativement défini par plusieurs noms, tels que routage, recommandation et "diffusion sélective de l'information" (ou "SDI pour *Selective Dissemination of new Information*"). Le routage est utilisé pour indiquer qu'une liste de documents est acheminée à un ou plusieurs utilisateurs. Le filtrage d'information est parfois associé à la collecte passive de l'information. La recommandation consiste à exploiter les préférences d'une communauté d'utilisateurs pour prédire la préférence d'un utilisateur donné. La SDI est utilisée pour insinuer que les profils décrivant des besoins en information sont construits manuellement, et permettent d'identifier les domaines de spécialisations des utilisateurs. On utilise de plus en plus le terme accès personnalisé à l'information. Ce terme englobe toutes les formes d'accès à l'information qui mettent l'utilisateur au centre du processus de sélection d'information. Toutes ces interprétations ont une base historique, mais il n'est pas rare de trouver quelques unes de ces terminologies employées pour décrire des systèmes de manière différentes par rapport à leurs historiques. Pour éviter ce problème, nous utilisons l'expression "*filtrage d'information*" pour décrire toute forme de sélection d'informations, intéressant un utilisateur, dans un flux d'informations.

La seconde notion qui nécessite quelques éclaircissements est la notion de *besoin en information*. Différents chercheurs ont proposé différentes définitions et interprétations pour cette notion. Mizzaro [Mizzaro, 1997] considère que le besoin en information passe par plusieurs phases. La phase initiale, c'est-à-dire, l'état anormal de la connaissance, appelé aussi *problème*. Un problème se transforme ensuite en un besoin en information lorsque la personne se rend compte de ce qu'elle désire comme information. L'expression du besoin en information se transforme en une requête. Le besoin en information est souvent désigné par "*centre d'intérêt*". Il est parfois identifié sous le nom "*topic*" (particulièrement dans la campagne TREC). On retrouve également les termes profil et requête qui peuvent désigner l'expression d'un besoin selon le contexte dans lequel ils sont invoqués.

Pour notre part, nous considérons une requête comme la représentation (ou l'expression) d'un besoin instantané. On exprime un besoin en information à travers une requête et grâce à un langage d'interrogation (peut être des mots clés). La requête est soumise au SRI. Quand un besoin en information est permanent, cas des systèmes de filtrage d'information, on parle de profil. Donc, le profil dans notre cas, correspond à la représentation (ou expression) d'un besoin en information dans les SFI. Afin d'éviter ces confusions, nous

employons seulement le terme "profil" lorsqu'il s'agit de décrire un besoin en information dans le contexte du filtrage d'information.

### 3.4 Aperçu historique

La notion du filtrage d'information est au centre des préoccupations de nombreux chercheurs depuis des décennies. Elle a été traitée pour la première fois par les travaux de Luhn dans "Business Intelligent System" en 1958 [Luhn, 1958]. Plus précisément, dans ce système, les employés d'une bibliothèque créent des profils pour des utilisateurs différents. Ces profils sont ensuite utilisés dans un système de sélection d'information pour fournir à chaque utilisateur une liste de nouvelles références bibliographiques. Les commentaires (connotations|jugements) sur des références spécifiques peuvent être enregistrés et utilisés pour faire évoluer les profils des utilisateurs. Luhn constate qu'il est possible d'utiliser les profils pour identifier les domaines de spécialisation des utilisateurs. En décrivant un système de sélection d'information comme un "*Selective Dissemination of new Information*" (Diffusion Sélective d'une nouvelle information), il a ainsi inventé ce terme qui est resté presque un quart de siècle avant qu'il soit rejoint par le terme filtrage d'information.

Une décennie plus tard, l'intérêt porté à la SDI a permis la création du groupe SIG pour "Special Interest on SDI" (SIG-SDI) de American Society of Information Science. Houseman en 1969 a effectué une étude sur ce groupe et il a identifié soixante systèmes opérationnels, dont neuf ont été utilisés par plus de 1000 utilisateurs chacun [Housman, 1969]. Ces systèmes se basent en général sur le modèle de Luhn. Quatre des soixante systèmes apprennent les profils de manière automatique, et le reste maintient les profils manuellement par des experts ou par les utilisateurs eux-mêmes. La raison principale qui a poussé ce groupe à s'investir autant dans la SDI est la disponibilité de l'information sous format électronique. Ce facteur combiné avec la distribution des résumés scientifiques, dans des supports magnétiques et réseaux d'ordinateurs, ont motivé l'intérêt porté aujourd'hui au filtrage d'information.

Denning [Denning, 1982] a inventé le terme "filtrage d'information" (*information filtering*) dans un papier publié dans les "Communication ACM" (CACM) en Mars 1982. L'objectif de Denning était d'élargir une discussion traditionnellement concentrée sur la

génération de l'information pour inclure aussi bien sa réception. Il a décrit le besoin de filtrer l'information arrivée par courrier électronique afin de séparer les messages urgents de ceux routiniers, et de limiter l'affichage des messages routiniers de façon à tenir compte des aspects mentaux des utilisateurs.

Pendant la décennie suivante, plusieurs articles sur le filtrage d'informations sont apparus dans la littérature. Tandis que le courrier électronique était au centre d'étude de Denning, d'autres domaines ont été testés, comme les articles newswire, des articles de "News" sur Internet, etc. L'article le plus influent de cette période est proposé par Malone dans une communication ACM de 1987 [Malone et al., 1987], où il introduit trois paradigmes pour la sélection d'information : *cognitif, économique et social* (aujourd'hui appelé collaboratif) en se basant sur le système "*Information Lens*" [Mackay et al., 1989].

En 1989, *United States Advanced Research Projects Agency* des Etat-Unis (DARPA) sponsorise Message Understanding Conference (MUC) [Hirschman, 1991], dont l'objectif était de développer des techniques d'extraction d'information pour la sélection de messages. En 1990, DARPA a lancé le projet Tapestry pour réunir les efforts de recherche des différents participants à MUC [Harman, 1992a]. Tapestry a ajouté des techniques statistiques de présélection de messages pouvant être soumis à des processus de traitement automatique du langage naturel plus sophistiqués [Ram, 1992].

En 1992, National Institut of Standards and Technology (NIST) sponsorise avec DARPA le projet international Text REtrieval Conference (TREC) qui s'intéresse à l'évaluation des systèmes de recherche et de filtrage d'information [Harman, 1992b]. En décembre de la même année, 9 articles traitant le filtrage d'information sont apparus dans un numéro spécial de communication ACM [Baclace, 1992] [Belkin and Croft, 1992] [Bowen et al., 1992] [Foltz and Dumais, 1992] [Goldberg et al., 1992] [Loeb, 1992] [Ram, 1992] [Stadnyk and Kass, 1992] [Stevens, 1992]. Suite au lancement du projet TREC, différentes tâches de filtrage ont été proposées, tels que le filtrage adaptatif, filtrage différé et routage [Harman, 2000]. Les tâches de filtrage ont cessé d'apparaître dans le programme d'évaluation TREC depuis l'année 2002 [Voorhees, 2002].

## 3.5 Grandes familles de filtrage d'information

Malone [Malone et al., 1987] a identifié trois grandes familles de filtrage : le filtrage cognitif, social et économique. Ces types de filtrage se différencient principalement sur les "indices" qu'ils utilisent pour décider de sélectionner ou de rejeter un document. Dans le filtrage cognitif, souvent appelé filtrage basé sur le contenu, le filtrage tient compte seulement des contenus du document et du profil. Dans le filtrage social, également appelé filtrage collaboratif, la décision de filtrage se base sur les annotations et commentaires attribuées par les utilisateurs aux documents. Enfin, le filtrage économique se base sur des incitations additionnelles, tels que des crédits attachés au document par son créateur. Le filtrage économique a été développé dans le cadre des applications du courrier électronique. L'idée de base est d'assigner certains coûts pour lire un message du récepteur aux expéditeurs dans le but d'éviter de l'imprimer à des fins publicitaires. Le filtrage économique se base sur plusieurs critères d'évaluation de coûts et de rendements, et des mécanismes d'évaluation explicites et implicites [Malone et al., 1987]. Les concepts du filtrage économique font souvent partie intégrante des filtres basés sur le contenu et le filtrage collaboratif. Ce sont ces deux filtres qui présentent des caractéristiques spécifiques. Nous décrivons brièvement ces deux techniques dans les sections suivantes.

### 3.5.1 Filtrage d'information collaboratif

Le filtrage collaboratif se base sur l'hypothèse que les personnes à la recherche d'information devraient pouvoir se servir de ce que d'autres ont déjà trouvé et évalué.

Le principe de base du filtrage collaboratif [Goldberg et al., 1992] est d'automatiser les processus sociaux, tel que le langage parlé. Dans la vie quotidienne, les personnes communiquent par des recommandations entre elles : des mots, des lettres de recommandation, des films et des livres, des journaux, etc. Les systèmes de filtrage collaboratif permettent d'automatiser ceci en facilitant la prise de décision sur les informations qu'elles reçoivent.

Différents termes ont été utilisés pour décrire le filtrage collaboratif. Initialement, il est désigné par le filtrage social [Malone et al., 1987] [Shardanand, 1994]. Certains auteurs montrent que dans un système automatisé les recommandations ne peuvent pas nécessairement collaborer avec les destinataires et les recommandations peuvent être inconnues entre elles, ainsi d'autres auteurs préfèrent utiliser le terme système de recommandation. Mais, souvent, ce terme n'est pas toujours employé avec une signification identique. Quelques

auteurs l'utilisent pour se référer seulement aux systèmes de filtrage collaboratif, d'autres incluent aussi des techniques basées sur le contenu ou en mélangeant les deux approches.

Les systèmes de filtrage collaboratif fonctionnent en enregistrant les réactions des utilisateurs sur les différents objets. Ces réactions sont appelées *annotations* [Goldberg et al., 1992]. Le système agrège ensuite ces annotations et les achemine aux destinataires appropriés. Dans certains cas, la transformation primaire est dans l'agrégation ; dans d'autres, la valeur du système se situe dans sa capacité de mieux comparer ces recommandations (annotations) et les recommandations recherchées.

Pour permettre une agrégation efficace, les annotations que les utilisateurs doivent assigner aux documents sont souvent limitées à des indices. Comme ces indices sont affectés manuellement, le filtrage collaboratif peut adopter n'importe quel type de mesure de pertinence. Souvent, des notations sont utilisées, comme "*le meilleur*", ..., "*je déteste ça*" [Shardanand and Maes, 1995] ou des valeurs d'une échelle numérique ([1-5]) [Miller et al., 1997]. La dimension de la pertinence donnée par ces notations peut être définie explicitement (ex. "*la qualité de l'écriture*", "*la convenance du besoin avec les news-group*") ou laissée à l'utilisateur.

Les indices peuvent être fournis explicitement, par exemple, en les sélectionnant à partir d'un menu déroulant ou en cliquant sur un bouton numérique, ou de manière implicite en observant les activités de l'utilisateur, par exemple, le temps de lire un message [Morita and Shinoda, 1994], la sauvegarde d'un objet pour une lecture ultérieure [Nichols, 1997] [Balabanovic, 1998].

Le filtrage collaboratif, d'après Goldberg et al. [Goldberg et al., 2000], décrit les techniques d'un groupe d'utilisateurs pour prédire la préférence d'un nouvel utilisateur ; les recommandations pour le nouvel utilisateur sont basées sur ces prédictions. Une préférence d'un utilisateur est décrite par un profil, qui est défini par un vecteur de dimension fini (correspondant au nombre de documents disponibles). Chaque valeur du vecteur représente l'évaluation que l'utilisateur a attribué au document.

La figure 3.2 présente un cas de processus de filtrage d'information collaboratif, où la prédiction de l'opinion (ici, représenté par  $v$  et  $x$ ) qu'un utilisateur  $u_0$  sur un document donné, est calculée en rapprochant les évaluations passées de l'utilisateur des évaluations que d'autres utilisateurs ( $u_1, u_2, u_5$  et  $u_7$ ) de la communauté ont donné par le passé sur les

mêmes documents [Berrut and Denos, 2003].

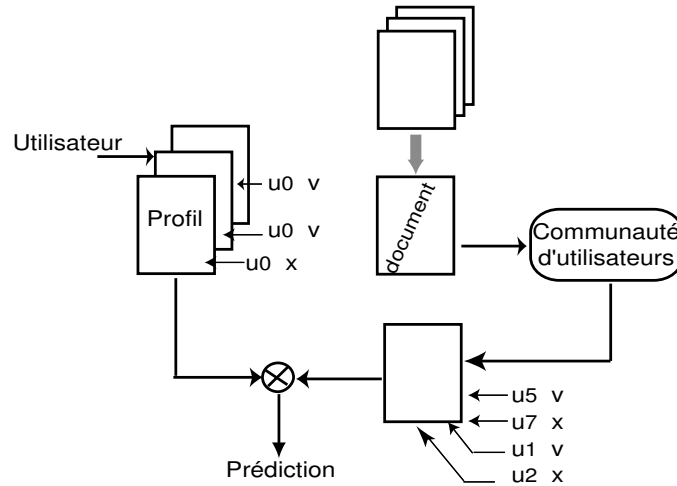


Figure 3.2 – Processus de filtrage social (collaboratif)

En se référant au système de recherche d’information, dans un système de filtrage collaboratif les représentations utilisées dans le processus de comparaison sont les annotations. Cependant, un système de filtrage basé sur le contenu considère deux objets sont similaires si leurs représentations sont identiques ou presque, deux objets sont considérés similaires en filtrage collaboratif s’ils sont annotés par les mêmes utilisateurs.

### 3.5.2 Filtrage d’information basé sur le contenu (cognitif)

Le filtrage basé sur le contenu est un type de filtrage dont la décision de sélection ou non d’un document se base uniquement sur le contenu de ce document.

Les techniques de filtrage basées sur le contenu fonctionnent par la caractérisation du contenu de l’information (document) à filtrer [Malone et al., 1987]. Les représentations des documents et des profils dans ce type de filtrage exploitent seulement les informations qui peuvent être dérivées de leur thème respectif [Oard and Marchionini, 1996]. Autrement dit, la sélection de documents se base sur une comparaison des thèmes abordés dans les documents par rapport aux thèmes intéressant l’utilisateur.

La figure 3.3 présente un processus de filtrage d'information basé sur le contenu, où la décision de sélection d'un document donné, est calculée en rapprochant les thèmes énoncés par l'utilisateur comme constituant son profil, et les thèmes extraits des documents par un processus d'indexation [Berrut and Denos, 2003].

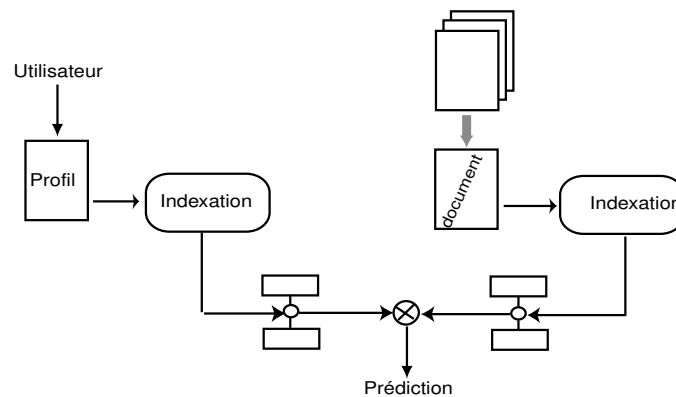


Figure 3.3 – Processus de filtrage basé sur le contenu (filtrage cognitif)

La technique de filtrage basée sur le contenu tient ses racines dans la communauté de la RI et elle réutilise plusieurs de ses modèles. L'exemple le plus saillant du filtrage basé sur le contenu est le filtrage d'objets textuels (par exemple, les mails ou les pages WEB) basé sur les mots contenus dans leurs représentations textuelles. A chaque objet, ici le texte du document, est assigné un ou plusieurs termes d'index extraits pour caractériser son contenu. L'extraction des index est réalisée par un processus d'indexation (figure 3.3). Ce processus d'indexation est appliqué également pour représenter le profil utilisateur. La sélection d'un document est basée sur le degré de ressemblance entre les index du profil et ceux du document. Le principe de base de ce type de filtrage est que les représentations des documents et des profils peuvent être exprimées par des mots et/ou des phrases plus spécifiques.

Le processus général de ce type de filtrage est présenté dans la section suivante.

## 3.6 Processus de filtrage d'information

Habituellement, on considère qu'un système de recherche d'information a pour fonction "d'amener à l'utilisateur les documents qui satisfont son besoin en information". Un sys-

tème de filtrage d'information "achemine des documents qui se présentent vers des groupes de personnes, en se basant sur leurs profils à long terme". En effet, compte tenu de la dualité entre le FI et la RI, l'architecture générale d'un système de filtrage d'information est semblable à celle d'un système de recherche d'information (voir figure 3.4).

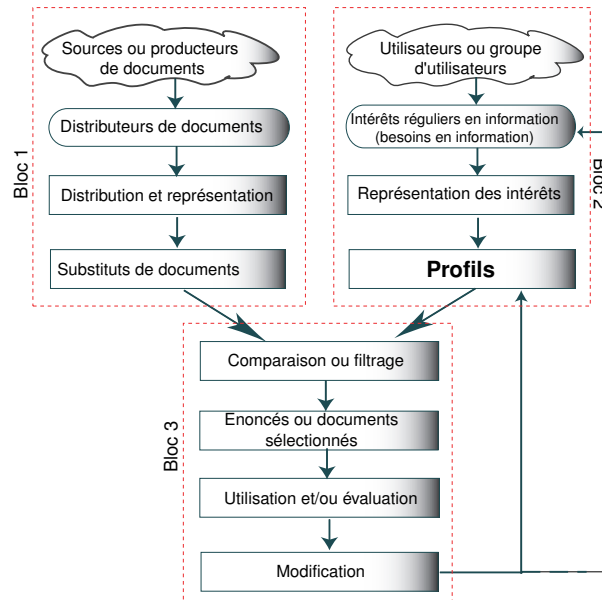


Figure 3.4 – Architecture générale d'un système de filtrage d'information

La figure 3.4 illustre l'architecture de base des systèmes de filtrage d'information, telle qu'elle a été présentée par Belkin et Croft [Belkin and Croft, 1992]. La figure est caractérisée par trois blocs, et chacun est décrit par quatre étapes. Les trois blocs représentent la création de substituts (représentation) de documents (Bloc 1), création de profils (Bloc 2), et le processus de comparaison et de filtrage (Bloc 3).

Le processus de filtrage d'information (voir figure 3.4) commence tout d'abord par des individus ou groupes d'individus qui définissent leurs centres d'intérêts, besoins en information qui peuvent évoluer au cours du temps au fur et à mesure que des informations sont filtrées. De tels intérêts engagent les utilisateurs dans un processus relativement passif de recherche d'information. Ce processus est caractérisé par la représentation des besoins en information de l'utilisateur par des *profils*. En parallèle, les producteurs de documents entreprennent de distribuer leurs produits dès qu'ils sont générés. Pour accomplir cette tâche on associe aux documents une représentation de leur contenu, qui est ensuite comparée aux profils. Cette évaluation permet de décider si un document est pertinent ou non pour

un profil donné, et peut mener, dans la plupart des cas, à l'amélioration des profils et des domaines d'intérêt.

### **3.7 Filtrage d'information versus recherche d'information**

Comme nous venons de le mentionner dans les sections précédentes, le filtrage d'information est une fonction duale de la recherche d'information. Cette dualité est traduite par les faits suivants :

- un système de recherche d'information suppose l'existence d'une collection de documents, alors que le filtrage d'information maintient un ensemble de profils ;
- en recherche d'information, l'interaction de l'utilisateur avec le document durant une session de recherche est unique, alors qu'en filtrage d'information, l'utilisateur pourra effectuer des changements à long terme à travers une série de session de recherche ;
- collecter et organiser les documents est une des fonctionnalités des systèmes de recherche d'information, diffuser (ou distribuer) des documents à des utilisateurs ou à des groupes d'utilisateurs demeure la priorité fonctionnelle des systèmes de filtrage ;
- un système de recherche d'information gère une collection statique de documents, alors qu'un système de filtrage d'information reçoit des documents d'un flot d'informations dynamiques ;
- un système de filtrage d'information permet de sélectionner ou d'éliminer des documents à partir d'un flux d'informations. En contrepartie, un système de recherche d'information sélectionne des documents à partir d'une base statique ;
- un système de filtrage d'information simule un processus peu ou pas interactif, puisque les utilisateurs consultent les documents filtrés d'une façon périodique dans le temps. Au contraire, un système de recherche d'information interagit avec l'utilisateur en consultant les résultats de recherche, en jugeant les résultats, etc.
- un système de filtrage d'information donne une décision sur la pertinence ou non d'un document, alors qu'un système de recherche d'information propose une liste de documents classée par ordre de pertinence ;
- l'architecture d'un système de recherche d'information est symétrique à celle d'un système de filtrage d'information, car, l'entrée de l'un est la sortie de l'autre, et l'inverse est vrai. Un système de recherche d'information compare une requête avec

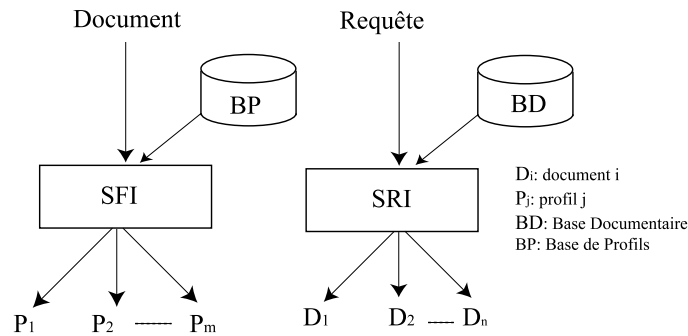


Figure 3.5 – Filtrage *vs* recherche d'information

une liste de documents et sélectionne ceux qui sont pertinents pour la requête, alors qu'un système de filtrage d'information compare un document avec une liste de profils et achemine le document aux profils adéquats (voir figure 3.5).

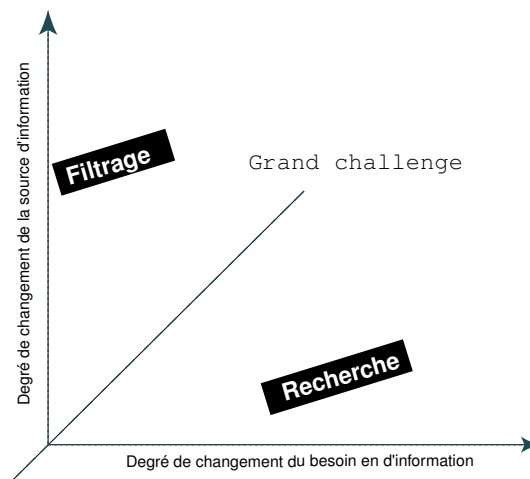


Figure 3.6 – Processus d'accès à l'information

- on identifie les systèmes de recherche et de filtrage d'information par rapport au degré de changement des sources d'informations et des besoins en informations (voir figure 3.6). Plus le degré de changement des sources d'informations croît (réciproquement décroît) et plus le degré de changement des besoins en information décroît (réciproquement croît) et on a tendance à utiliser un SFI (réciproquement un SRI). Le type de système à considérer se complique lorsque le degré de changement des sources et des besoins en information évolue constamment, ce qu'on appelle le "*grand challenge*" dans l'accès à l'information [Oard and Marchionini, 1996]. Nous utilisons l'expression "*accès à l'information*" par référence aux FI et RI. Par exemple, l'étude du marché

des actions, où le marché de la bourse et les intérêts des boursiers changent régulièrement. Ces types de systèmes sont caractérisés par des accès fréquents à l'information, et ils sont appelés système de filtrage d'information dynamique.

### 3.8 Problématique du filtrage d'information cognitif

Compte tenu de la dualité RI/FI, bon nombre de modèles de filtrage sont basés sur des modèles de recherche d'information augmentés par une fonction de décision. Le plus souvent, les documents et les profils sont représentés par des listes de mots pondérés. L'appariement document-profil consiste à mesurer un score de similarité. La décision quant à l'acceptation ou le rejet d'un document est assurée par une fonction de décision souvent de type seuil. Si le score est supérieur au seuil le document est accepté sinon il est rejeté.

Or, en l'absence de base de référence, la détermination de ce seuil et les pondérations adéquates associées aux profils et aux documents sont les problèmes majeurs rencontrés dans ce domaine. En effet, dans un système de recherche d'information, les techniques de pondération et de reformulation automatique de requêtes, basées sur la collection entière de documents, se sont avérées efficaces. Or, dans un système de filtrage d'information, à l'initialisation du processus de filtrage, on ne dispose ni d'une connaissance sur les documents à filtrer pour pouvoir construire une fonction de décision, ni pour bien pondérer les profils et les documents entrants.

Ceci nous ramène alors à poser quatre questions qui résument la problématique de ce type de système de filtrage :

- *la première est comment représenter le profil ?*
- *la seconde, comment construire une fonction de décision ?*
- *la troisième, comment améliorer la représentation du profil ?*
- *et enfin la quatrième, comment adapter la fonction de décision ?*

Ainsi, concernant les deux premières questions, dans la majorité des techniques de filtrage, le profil est représenté par une liste de mots clés, éventuellement pondérés, extraits du texte du profil ou élaborés à partir de données d'apprentissage, et la fonction de décision est un seuil fixé arbitrairement ou appris sur une collection d'entraînement<sup>1</sup>. En fait, quand on parle de la fonction de décision on fait principalement référence à la manière d'identifier le seuil de décision, on parle souvent de fonction de seuillage.

---

<sup>1</sup>Une collection de documents déjà jugés pertinents ou non pertinents pour chaque profil

Une grande partie des travaux effectués dans ce domaine s’est plutôt focalisée sur l’amélioration des profils et de la fonction de décision. Ceci est réalisé grâce à un processus d’apprentissage basé sur les éléments déjà filtrés. Cet apprentissage peut se faire de manière incrémental, dans ce cas, le processus est déclenché à chaque réception d’un document (pertinent et/non pertinent), ou bien de manière différé, c’est-à-dire sur des ensembles de documents déjà filtrés.

Outre ces éléments liés à l’apprentissage, les systèmes de filtrage peuvent se distinguer par rapport au mode de sélection d’information. En effet, cette sélection peut se faire de manière synchrone (ou adaptatif), c’est-à-dire déclenché à chaque arrivée d’un document ou de manière asynchrone (ou différé), c’est-à-dire effectué selon soit des périodes de temps spécifiques ou sur la base de quantité de documents déjà reçus.

La réponse aux différentes questions précédentes dépend en fait du mode opératoire du système de filtrage. Ces modes se différencient par la collection d’apprentissage (le choix de l’état initial), le mode d’apprentissage (continu ou non), la sélection de documents et le résultat de filtrage est une la liste ordonnée ou non. Chaque combinaison donne lieu à un type de filtrage particulier.

Comme nos travaux ont été évalués et expérimentés selon le canevas TREC, le tableau 3.1 dresse les différents types de filtrage proposés dans TREC [Voorhees, 2002]. Par exemple, le filtrage différé utilise une collection d’apprentissage pour initialiser et apprendre les différents paramètres du système, aucun apprentissage n’est effectué tout au long du filtrage de documents et plusieurs documents non ordonnés peuvent être acheminés à l’utilisateur. Dans le cas du filtrage adaptatif, aucune collection d’apprentissage n’est utilisée, l’apprentissage du système se fait à chaque sélection d’un document et le résultat du filtrage est toujours un seul document.

Nom selon TREC ( <i>terme anglais</i> )	Collection d’apprentissage	Apprentissage	Sélection de documents	Résultats : liste ordonnée
<b>Adaptatif</b> ( <i>adaptive</i> )	non	oui	un seul	non
<b>Différé</b> ( <i>batch</i> )	oui	non	plusieurs	non
<b>Différé-adaptatif</b> ( <i>batch-adaptive</i> )	oui	oui	plusieurs	non
<b>Routage</b> ( <i>routing</i> )	oui	non	plusieurs	oui

Tableau 3.1 – Types de filtrage d’information

Le filtrage adaptatif représente évidemment le cas de filtrage le plus courant. Il est plus difficile à entreprendre en raison des difficultés associées au bon démarrage du processus, à l’apprentissage permanent du profil et l’adaptation de la fonction de seuillage.

Nous allons dans la section 3.10 tenter de répondre aux questions posées ci dessus à travers quelques travaux du domaine. Avant de décrire ces travaux, nous consacrons la section suivante à présenter les mesures d'évaluation des systèmes filtrage d'information. Dans certains travaux, les fonctions d'évaluation, notamment la fonction d'utilité, sont importantes dans le processus d'adaptation de la fonction de seuillage.

## 3.9 Evaluation des performances des systèmes de filtrage d'information

Plusieurs mesures standards utilisées dans l'évaluation des systèmes de recherche d'information (par exemple, les mesures de précision et rappel) ne sont pas applicables dans le cas de filtrage. Par exemple, dans le cas d'un système qui filtre, quotidiennement, des médias (images, vidéo) sur Internet, il est quasiment impossible de calculer le rappel, car l'utilisateur ne dispose pas des informations pertinentes non sélectionnées.

Un autre exemple plus explicite, considérons deux systèmes de filtrage différents pour un même flux de documents : le premier système sélectionne une liste de 100 documents non pertinents et zéro document pertinent, et le deuxième système sélectionne un document non pertinent et aucun document pertinent. Si nous calculons les valeurs de précision et de rappel, on peut constater qu'elles sont nulles ; néanmoins, dans la pratique, le deuxième système est plus performant que le premier, puisque l'utilisateur ne perd pas de temps à lire des documents qui ne l'intéressent pas. Ceci rend les mesures de précision et de rappel inadéquates, car elles ne permettent pas de différencier entre les systèmes.

Pour remédier aux insuffisances des mesures utilisées en RI, des fonctions d'utilité ont été introduites lors de la campagne d'évaluation de TREC dans le cadre de la tâche de filtrage d'information. Ainsi, la performance d'un système de filtrage est souvent mesurée en terme d'utilité. Nous présentons dans les sections suivantes les différentes mesures d'évaluations utilisées dans le programme **TREC**.

### 3.9.1 Utilité linéaire

Une fonction d'utilité est représentée par la table 3.2 de contingence ci-dessous, où un coût positif ou négatif est assigné à chaque élément de la table :

Documents	Pertinents	Non pertinents
Sélectionnés	$R_+/\lambda_1$	$S_+/\lambda_2$
Non sélectionnés	$R_-/\lambda_3$	$S_-/\lambda_4$

Tableau 3.2 – Table de contingence

$$U_{(\lambda_1, \lambda_2, \lambda_3, \lambda_4)} = \lambda_1 R_+ + \lambda_2 S_+ + \lambda_3 R_- + \lambda_4 S_- \quad (3.1)$$

Avec  $R_+$  ( $S_+$ ) est le nombre de documents pertinents (non pertinents) sélectionnés, et  $R_-$  ( $S_-$ ) est le nombre de documents pertinents (non pertinents) non sélectionnés. ( $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$ ,  $\lambda_4$ ) représentent le gain ou coût associé à chaque document dans sa catégorie de correspondance.

Plus la valeur d'utilité est grande, plus le système de filtrage d'information est performant pour un profil donné. Dans TREC-6, deux fonctions d'utilité linéaires ont été proposées :

$$\begin{aligned} U_{(3,-2,0,0)} &= F1 = 3 * R_+ - 2 * S_+ \\ U_{(3,-1,-1,0)} &= F2 = 3 * R_+ - S_+ - R_- \end{aligned} \quad (3.2)$$

Dans la fonction  $F2$ , le paramètre  $\lambda_3$  est négatif (égal à  $-1$ ). Cette fonction a été remplacée, dans TREC-7, par la fonction  $F_3$ , car il est difficile d'estimer le nombre exact de documents pertinents non retrouvés, et surtout la majorité des systèmes obtiennent souvent une valeur négative de  $F2$ . La fonction d'utilité  $F_3$  est représentée par les paramètres  $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (4, -1, 0, 0)$ .

La mesure d'utilité peut être ramenée à l'estimation de la probabilité de pertinence. Il existe une formule générale pour convertir une fonction d'utilité en un seuil de probabilité. Une dérivation de cette formule existe dans la théorie de la décision. Pour maximiser théoriquement la valeur d'utilité, le système doit sélectionner un document si et seulement si :

$$p(\text{pertinence}) \geq \frac{-\lambda_2}{\lambda_1 - \lambda_2} \quad (3.3)$$

Pour plus d'explications sur le principe de conversion de la fonction d'utilité en un seuil de probabilité, le lecteur peut se référer à [Lewis, 1995].

Quand l'évaluation d'un système de filtrage est basée seulement sur l'utilité linéaire, il est difficile de comparer la performance du système dans le cas d'une collection de test avec plusieurs profils. Par conséquent, une mesure d'utilité normalisée a été proposée. L'intérêt de normaliser la fonction d'utilité est d'atténuer l'effet de très grandes utilités sur la moyenne de l'ensemble des utilités. Cette mesure est donnée comme suit :

$$u_s^*(S, T) = \frac{\max(u_s(S, T), U(s)) - U(s)}{\text{Max}U(T) - U(s)} \quad (3.4)$$

où :

$u_s^*(S, T)$  : la valeur d'utilité normalisée du système  $S$  pour le profil  $T$ ,

$u_s(S, T)$  : la valeur d'utilité originale du système  $S$  pour le profil  $T$ ,

$U(s)$  : l'utilité quand  $s$  documents non pertinents sont sélectionnés,

$\text{Max}U(T)$  : utilité maximale possible pour le profil  $T$ .

De nouvelles fonctions d'utilité ont été aussi testées. Dans TREC-8, en plus des deux fonctions d'utilité linéaires ci-dessus ( $F_1$  et  $F_3$ , avec  $\lambda_1$  dans  $F_3$  fixé à 3), deux autres fonctions d'utilité non linéaires ont été proposées :

$$\begin{aligned} NF1 &= 6 * (R_+)^{0.5} - S_+ \\ NF3 &= 6 * (R_+)^{0.8} - S_+ \end{aligned} \quad (3.5)$$

A partir de TREC-9, une nouvelle mesure d'utilité est définie. Cette mesure est réutilisée dans toutes les évaluations postérieures à TREC-9. Les paramètres de la fonction d'utilité (3.1) sont ainsi fixés respectivement à  $(2, -1, 0, 0)$  :

$$U = 2 * R_+ - S_+ \quad (3.6)$$

Dans l'évaluation de TREC-9, la normalisation de la fonction d'utilité est simplement réduite à remplacer  $U(s)$  de la fonction (3.4) par  $\text{Min}U$ . Ainsi,  $\text{Min}U$  est fixé soit à  $-100$

ou à  $-400$ , selon les collections de documents utilisées. Dans TREC-10 la fonction d'utilité de TREC-9 a été réutilisée avec l'initialisation de  $MaxU$ , pour chaque profil, à  $2 * R$ , où  $R$  est le nombre de documents pertinents total ( $R = R_+ + R_-$ ).

A partir de TREC-11, la normalisation de la fonction d'utilité utilisée est donnée comme suit :

$$U = \frac{\max(T11NU, \min NU) - \min NU}{1 - \min NU} \quad (3.7)$$

avec :

$$\begin{aligned} T11NU &= \frac{T11U}{MaxNU} \\ T11U &= 2R_+ - S_+ \\ MinNU &= -0.5 \\ MaxNU &= 2 * R \end{aligned}$$

### 3.9.2 Mesure orientée précision

Proposée dans le cadre des évaluations de TREC-9, l'idée de base de cette mesure est de fixer une borne  $L$ , qui est le nombre de documents à retrouver durant une période de simulation. La mesure est essentiellement une fonction de précision (T9U), mais avec une pénalité pour ne pas atteindre la valeur fixée à l'avance :

$$\begin{aligned} T9U &= \frac{R_+}{\max(L, N)} \\ L &= 50 \text{ documents} \end{aligned} \quad (3.8)$$

avec,  $N$  le nombre de documents retrouvés ( $N = R_+ + S_+$ ).

Une autre fonction d'utilité  $F - beta$ , définie par Van Rijsbergen, est proposée dans les versions dernières évaluations de TREC. Cette fonction permet de réutiliser les mesures de précision et de rappel de la RI. La forme de la fonction  $F - beta$  est la suivante :

$$F - beta = \frac{1.25 * R_+}{R_+ + S_+ + 0.25 * R} \quad (3.9)$$

$F - beta$  est égale à zéro lorsque aucun document n'est sélectionné.

## 3.10 Présentation de quelques modèles de filtrage adaptatifs

Comme nous l'avons souligné dans la section (3.8), les modèles de filtrage d'information sont le plus souvent basés sur des modèles de RI. Il n'existe donc pas d'approche mise en place spécialement pour le filtrage d'information. La spécificité des travaux dans ce domaine réside principalement dans leur manière d'apprendre le profil et d'adapter la fonction de seuillage.

La majorité des techniques d'apprentissage du profil proposée dans la littérature est inspirée du principe de reformulation de requêtes [Salton, 1989]. Les techniques utilisées sont principalement basées sur une version incrémentale de l'algorithme de Rocchio [Rocchio, 1971], on y trouve notamment les travaux de [Allan, 1996] [Ault and Yang, 2000] [Callan, 1998] [Schapire et al., 1998] [Hoashi et al., 2000] ou des techniques d'apprentissage basées sur les classifieurs bayesiens [Kim et al., 2000], régression logistique [Zhang, 2004], les réseaux de neurones [Kwok et al., 2000] et techniques génétiques [Boughanem et al., 1999b].

Concernant la fonction de seuillage, les méthodes proposées tentent de définir un seuil (un score) qui permet d'optimiser une fonction d'utilité. La majorité des techniques de seuillage actuelles se basent sur des méthodes heuristiques [Zhai et al., 1998], régression logistique [Robertson and S.Walker, 2000] ou distribution des scores [Arampatzis and Hameren, 2001] [Zhang and Callan, 2001] [Boughanem et al., 2004c].

Nous décrivons dans les sections suivantes quelques modèles de filtrage d'information adaptatifs les plus répandus actuellement. Les modèles seront classés par rapport au type de méthodes d'adaptation de la fonction de seuillage adoptées. Nous décrivons trois sortes de modèles, les modèles basés sur la méthode heuristique, la régression logistique et la distribution des scores. Nous nous focalisons particulièrement sur leurs approches d'apprentissage du profil et de l'adaptation de la fonction de seuillage. L'apprentissage du profil, dans la plupart des modèles basés sur la méthode heuristique, est inspiré du principe de l'algorithme de Rocchio. Dans certains modèles, nous détaillons le principe d'initialisation du processus de filtrage, à savoir la représentation des profils, des documents et les mesures de similarité utilisées.

### 3.10.1 Modèles de filtrage basés sur la méthode heuristique

Les modèles utilisant les méthodes heuristiques sont souvent basés sur une approche vectorielle [Hoashi et al., 1999] [Zhai et al., 1998] [Wang et al., 2001] [Wu et al., 2001]. Les profils et les documents sont représentés par une liste de termes pondérés par la formule standard *TF-IDF*. Le score de similarité entre les représentations du profil et du document est calculé par une mesure cosinus ou produit scalaire. Dans la majorité des modèles présentés dans cette section, l'apprentissage du profil se fait selon Rocchio ou une version modifiée de Rocchio.

Nous décrivons dans cette section, trois modèles de filtrage, chacun a sa spécificité au niveau de l'initialisation et de l'apprentissage du profil et de l'adaptation de la fonction de seuillage.

#### 3.10.1.1 Modèle de CAFES

Le système de filtrage adaptatif de *CLARIT* (*CAFES* : *Clarit Adaptive Filtering Evaluation System*) est développé au sein de CLARITECH Corporation. *CAFES* est une extension du système de recherche d'information *CLARIT* [Milic-Frayling et al., 1997] [Zhai et al., 1998].

Le processus d'apprentissage du profil et de l'adaptation du seuil sont déclenchés après avoir accumulé un certain nombre de documents pertinents (2 ou 4, selon des expérimentations). Plus précisément, l'apprentissage du profil est basé sur une version de l'algorithme de Rocchio, où seulement les documents pertinents sont considérés dans l'apprentissage du profil. Quant à l'adaptation de la fonction de seuillage, elle est effectuée grâce à une méthode de régulation dite *Béta-gamma*. Nous détaillons dans ce qui suit les différentes parties du modèle *CAFES* :

*a. Initialisation du seuil.* Le seuil initial est estimé par une méthode appelée taux de livraison (delivery ratio). L'idée de base est d'initialiser le seuil, à partir d'une collection d'entraînement, à une valeur permettant de délivrer une proportion bien déterminée de documents. Ce seuil est déterminé de la façon suivante : supposons qu'un utilisateur souhaite recevoir un taux "*r*" de documents, par exemple 20% de documents délivrés (voir figure 3.7), on considère alors une collection d'entraînement de *N* documents et on calcule un score de similarité entre le profil de l'utilisateur et chaque

document de la collection. Les scores des documents sont ensuite ordonnés par ordre décroissant. Le seuil initial, donné par un taux de livraison, est égal au score du  $k$ -ème document ( $k = r * N$ ).

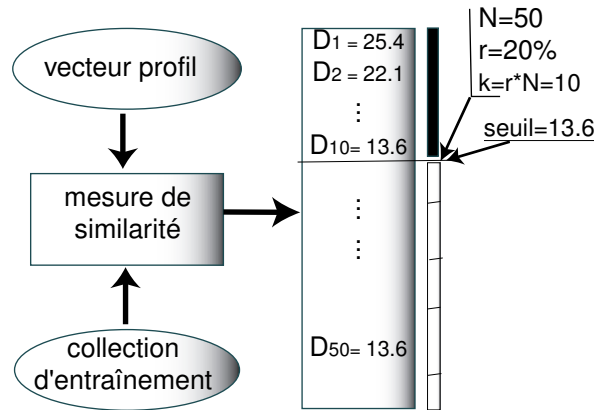


Figure 3.7 – Initialisation du seuil dans *CAFES*

Dans TREC 7 et 8, la collection d'entraînement FT91 (Financial Times 1991) et un taux égal à 0.0005 (i.e. délivrer un document parmi 2000 documents) sont utilisés pour fixer le seuil initial.

*b. Apprentissage du profil.* Une version incrémentale de l'algorithme de Rocchio est utilisée pour apprendre le profil de l'utilisateur. Cependant, seuls les documents sélectionnés pertinents sont exploités lors de l'apprentissage du profil. Le nouveau profil est calculé comme suit :

$$P_{nouv} = P_{org} + \delta.P_{rel} \quad (3.10)$$

où

- $P_{nouv}$  est le nouveau vecteur profil,
- $P_{org}$  le vecteur original du profil,
- $P_{rel}$  est le vecteur centroïde des documents pertinents sélectionnés (accumulés),
- $\delta$  un coefficient qui peut être défini de deux façons différentes : (1) constant pour l'ensemble des profils ou (2) dynamique, c'est-à-dire, il dépendra du nombre de documents de la collection à partir duquel les termes sont extraits. La formule dynamique utilisée pour initialiser  $\delta$  est définie comme suit :

$$\delta = 0.1 * e^{-0.1N} + 0.2 * (1 - e^{-0.1N}) \quad (3.11)$$

où N est le nombre de documents pertinents de la collection.

d. *Adaptation de la fonction de seuillage.* L'adaptation de la fonction de seuillage est basée sur la méthode de régulation adaptative *Béta-gamma*. Cette méthode sélectionne un seuil  $\theta$  par interpolation entre un seuil optimal  $\theta_{opt}$  et un seuil zéro  $\theta_{zero}$  (voir figure 3.8). Ces seuils sont estimés à partir d'un ensemble de documents déjà filtrés pour un profil. Le seuil optimal ( $\theta_{opt}$ ) est un seuil qui permet d'obtenir une utilité optimale (ou maximale) sur cet ensemble de documents. Le seuil zéro ( $\theta_{zero}$ ) est le plus grand seuil, inférieur au seuil optimal, qui donne une utilité négative sur l'ensemble des documents et en supposant que l'ensemble des documents non sélectionnés sont non pertinents.

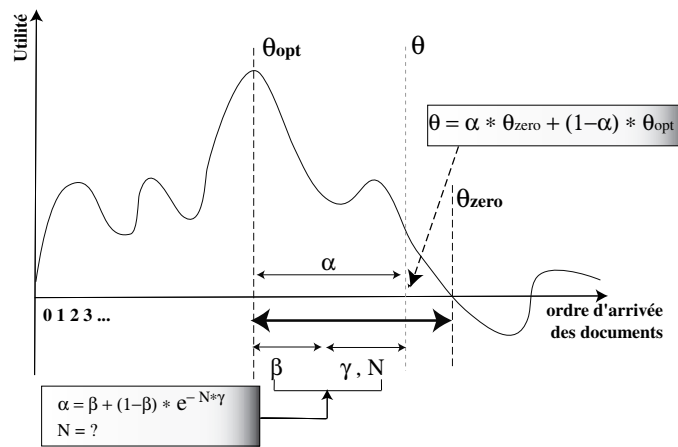


Figure 3.8 – Optimisation du seuil par interpolation

Le facteur d'interpolation est sensible au nombre de documents examinés pour calculer le seuil. Dans les premières expérimentations, l'interpolation utilisée est linéaire, cela en utilisant un paramètre constant  $\alpha$ , ainsi la méthode est appelée "régulation alpha", avec  $\alpha$  est donné comme suit :

$$\theta = \alpha\theta_{zero} + (1 - \alpha).\theta_{opt} \quad (3.12)$$

Après plusieurs expérimentations, et quelques études sur le comportement de la méthode, le paramètre  $\alpha$  est donné en fonction de deux paramètres  $\beta$  et  $\gamma$ , d'où l'appellation de la méthode "*Béta-gamma*". Ainsi,  $\alpha$  est redéfini comme suit :

$$\alpha = \beta + (1 - \beta).e^{-\gamma*N} \quad (3.13)$$

Où  $\beta$  est un facteur de correction biaisé des scores des documents les plus élevés et  $\gamma$  exprime un degré de confiance dans l'estimation du seuil optimal proche de la

valeur optimale exacte. Lors des expérimentations, les valeurs de  $\beta$  et  $\gamma$  sont fixées respectivement à 0.1 et 0.5.

La figure 3.8 illustre l'expression graphique des différentes formules ci-dessus. En considérant la liste des documents sélectionnés dans un ordre croissant par rapport à leurs scores, à leurs pertinences et à leurs valeurs d'utilité associées, on peut tracer une courbe correspondant à la valeur d'utilité dans chaque position de la liste. Chaque point de la courbe correspond à un score d'un document. La figure 3.8 montre comment le choix de  $\alpha$  détermine un seuil entre l'utilité optimale et l'utilité zéro, et les paramètres  $\beta$  et  $\gamma$  permettent d'ajuster  $\alpha$  dynamiquement et en fonction du nombre de documents jugés par le système.

L'apprentissage du profil et de l'adaptation de la fonction de seuillage est déclenchée à chaque sélection de  $n$  documents. Plusieurs valeurs de  $n$  ont été testées, soit à 1 ou 2 ou 4. Les expérimentations effectuées montrent que le système obtient des résultats satisfaisant lorsque  $n = 2$  ou  $n = 4$ , et moins satisfaisant lorsque  $n = 1$ .

### 3.10.1.2 Modèle de Hoashi et al.

Hoashi et al. [Hoashi et al., 1999] [Hoashi et al., 2000] ont adapté le système de recherche d'information, développé au sein du laboratoire de recherche KDD R&D, au filtrage d'information adaptatif. L'apprentissage du profil est basé sur une analyse de la contribution de chaque terme dans un document sélectionné, et la fonction de seuillage est adaptée par une méthode heuristique.

*a. Apprentissage du profil.* Hoashi et al. [Hoashi et al., 1999] ont proposé une méthode basée sur la contribution des termes des documents sélectionnés pour apprendre le profil, où ce qu'on appellera ici "*l'expansion de profil*".

*a.1. Définition de la contribution d'un terme :* la contribution d'un terme est une mesure qui exprime l'influence d'un terme dans la similarité entre le profil et le document. Elle est définie par la formule suivante :

$$Cont(tp_i, P, D_j) = Sim(P, D_j) - Sim(P'(tp_i), D'_j(tp_i)) \quad (3.14)$$

Avec  $Cont(tp_i, P, D_j)$  est la contribution du terme  $tp_i$  dans la similarité entre le profil  $P$  et le document  $D_j$ ,  $Sim(P, D_j)$  est la similarité entre  $P$  et  $D_j$ ,  $P'(tp_i)$  est le profil  $P$  sans le terme  $tp_i$  et  $D'_j(tp_i)$  est le document  $D_j$  sans le terme  $tp_i$ . En d'autres termes, la contribution d'un terme est la différence entre la similarité entre le document et le profil, et la similarité entre le document et le profil, tout deux privés de ce terme. Un terme peut avoir une contribution positive ou négative, si la contribution du terme est positive il augmente la similarité, si elle est négative il diminue la similarité.

Hoashi et al. [Hoashi et al., 2000] ont remarqué que les termes ayant une contribution positive sont ceux qui co-occurrent dans le document et le profil. Ces termes peuvent être considérés comme des éléments informatifs du document pertinent pour le profil. Par contre, les termes ayant une contribution négative sont des termes discriminants mais qui n'occurrent pas dans le profil. Les termes ayant une contribution négative sont alors appropriés pour l'apprentissage du profil.

a.2. *Expansion du profil* : en se basant sur le principe de contribution des termes décrit ci-dessus, la méthode d'expansion de profil suivante est proposée :

On calcule tout d'abord la contribution de chacun des termes du profil et des documents pertinents. Soit  $D_{rel} = \{D_1, \dots, D_N\}$  l'ensemble des  $N$  documents pertinents pour le profil  $P$ , puis, pour chaque document, on extrait les  $K$  termes de faible contribution. Ensuite, pour chaque terme  $tp_i$  extrait, on calcule son score par la formule suivante :

$$Score(tp_i) = wgt * \sum_{D_j \in D_{rel}} Cont(tp_i, P, D_j) \quad (3.15)$$

où  $wgt$  est un paramètre négatif (car la contribution de chaque terme extrait est toujours négative). Le score de chaque terme sera ensuite multiplié par son facteur  $IDF$  pour obtenir son poids final. Les termes et leurs poids sont rajoutés au profil initial.

Cette expansion de profil basée sur un schéma à la Rocchio peut également intégrer les documents non pertinents. Le schéma d'expansion proposé est le suivant :

$$\begin{aligned}
P_{now} = & P_{org} + wgt_{relR} * \sum_{D_j \in D_{rel}} Cont(tp_i, P_{org}, D_j) + \\
& wgt_{nrelR} * \sum_{D_j \in D_{nrel}} Cont(tp_i, P_{org}, D_j)
\end{aligned} \tag{3.16}$$

où,  $D_{nrel}$  est un ensemble de documents non pertinents sélectionnés et les valeurs de  $wgt_{relR}$  et  $wgt_{nrelR}$  sont obtenues expérimentalement. La table 3.3 illustre les valeurs testées dans l'une de leurs expérimentations [Hoashi et al., 2000], et les valeurs moyennes de la fonction d'utilité  $F1$  (équation (3.2)). Le tableau 3.3 montre que  $\{wgt_{relR}, wgt_{nrelR}\} = \{-200, -800\}$  permettent d'obtenir une meilleure utilité.

$wgt_{relR}$	$wgt_{nrelR}$			
	-100	-200	-400	-800
-200	0.4558	0.4840	0.5091	0.5257
-400	0.4172	0.4777	0.5197	0.5184
-800	0.3815	0.4349	0.4842	0.5100

Tableau 3.3 – Valeurs d'utilité moyennes

Dans le but de réduire le nombre de documents non pertinents sélectionnés, Hoashi et al. [Hoashi et al., 1999] proposent d'utiliser, en plus du profil initial de l'utilisateur, un autre profil négatif. Ce profil permet en fait de rejeter les documents non pertinents. Le profil négatif est construit à partir des documents sélectionnés non pertinents. Le processus de filtrage consiste alors à sélectionner un document, si et seulement si, la similarité entre le document et le profil initial (le profil construit à partir des documents pertinents) est supérieure à un certain seuil (seuil des documents pertinents), et à rejeter le document si sa valeur de similarité avec le profil négatif est supérieure à un autre seuil (seuil des documents non pertinents).

*b. Adaptation de la fonction de seuillage.* Le processus d'adaptation du seuil se déclenche d'une façon régulière. Le nouveau seuil  $\theta_{now}$  est calculé en fonction du seuil original  $\theta_{org}$  et de deux fractions  $A1$  et  $A2$ . Il est donné comme suit :

$$\theta_{now} = \theta_{org} * (1 + A_1.A_2) \tag{3.17}$$

$$\begin{aligned}
A_1 &= \frac{e^{\alpha(x-\beta)} - 1}{e^{\alpha(x-\beta)} + 1} \\
A_2 &= \begin{cases} 4(y - 0.5)^2 & \text{si } y < 0.5 \\ 0 & \text{si } y > 0.5 \end{cases}
\end{aligned} \tag{3.18}$$

avec :

$$x = \frac{S_+}{R_+ + S_+} \quad \text{et} \quad y = \frac{R_+}{N}$$

où  $R_+$  et  $S_+$  sont définis dans la section 3.9.1,  $N$  est le nombre de documents examinés, et  $\alpha$  et  $\beta$  sont deux paramètres constants.

La valeur initiale du seuil et des paramètres  $\alpha$  et  $\beta$  sont fixés d'une façon arbitraire (dans TREC-7, le seuil initial est fixé à 0.1, et plusieurs valeurs sont testées pour les deux paramètres  $\alpha$  et  $\beta$ , dont  $\{1, 10\}$  pour  $\alpha$  et  $\{-0.005, 0, 0.00001, 0.001, 0.01, 0.05\}$  pour  $\beta$ ).

### 3.10.1.3 Modèle de Wu et al.

Dans le système de filtrage d'information, développé par Wu et al. [Wu et al., 2001], l'initialisation et l'apprentissage du profil se basent sur une fonction logarithmique, et l'adaptation de la fonction de seuillage sur l'optimisation d'une fonction de précision.

*a. Initialisation du profil.* Tout d'abord, des vecteurs de composants sont extraits à partir d'une collection de documents d'entraînement<sup>2</sup>. Cette dernière est constituée d'un échantillon de documents pertinents et d'un échantillon de documents pseudo-pertinents. Un document pseudo-pertinent pour un profil donné, est un document dont le score de similarité avec le profil est très élevé et jugé non pertinent dans la collection d'entraînement. Pour sélectionner seulement les termes les plus importants dans un vecteur de composants, une valeur logarithmique, appelée *logarithm Mutual Information (LMI)*, pour chaque terme est calculée. Les termes dont la valeur LMI est supérieure à 3.0 et apparaissant plus d'une fois dans les documents pertinents sont sélectionnés. Le *LMI* d'un terme est calculé comme suit :

---

<sup>2</sup>Les documents pertinents utilisés à l'initialisation rentre dans la tâche de filtrage adaptatif de TREC.

$$\log MI(tp_i, P_j) = \log\left(\frac{P(tp_i|P_j)}{P(tp_i)}\right) \quad (3.19)$$

où  $tp_i$  est le  $i$ ème terme du vecteur de composants et  $P_j$  est le  $j$ ème profil. Plus cette mesure est élevée, plus  $tp_i$  et  $P_j$  sont pertinents.  $P(tp_i|P_j)$  et  $P(tp_i)$  sont estimés par une méthode de maximum de vraisemblance.

Il faut noter également que la LMI est non seulement utilisée comme un critère de sélection de termes pour un profil donné, mais aussi pour assigner des poids aux termes.

Enfin, le profil initial est donné donc en combinant les vecteurs de composants des deux échantillons de documents pertinents et pseudo-pertinents. La combinaison représente la somme pondérée des différents vecteurs de composants (voir figure 3.9).

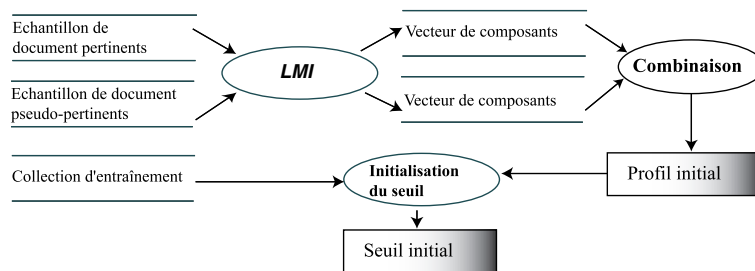


Figure 3.9 – Initialisation du profil et du seuil

- b. *Initialisation du seuil.* Le seuil est initialisé en se basant sur les scores de similarité entre le profil initial et les documents d'une collection d'entraînement (figure 3.9). Ces scores sont calculés par la mesure cosinus. En se basant sur ces scores, le seuil initial pour un profil est fixé à une valeur permettant de maximiser une fonction d'utilité (exemple, T10U de TREC-10).
- c. *Adaptation du profil et du seuil.* Une procédure adaptative est utilisée pour améliorer le profil initial et la valeur du seuil tout au long du filtrage des documents.
  - c.1. *Apprentissage du profil :* le processus d'apprentissage du profil est inspiré de l'algorithme de Rocchio [Wu et al., 2001]. Il est déclenché à chaque sélection de  $L$  documents pertinents (dans TREC-10,  $L$  est égal à 5).
  - c.2. *Adaptation de la fonction de seuillage :* le calibrage du seuil est effectué à chaque sélection d'un document pertinent. Soit :

- $t$  : représente le nombre de documents d'une séquence, puisque les documents sont traités par ordre chronologique,  $t$  peut également être considérée comme une valeur temporelle.
- $N^{(t)}$  : nombre de documents examinés jusqu'à l'instant  $t$ ,
- $R_+^{(t)}$  : nombre de documents pertinents sélectionnés jusqu'à l'instant  $t$ ,
- $S_+^{(t)}$  : nombre de documents non pertinents sélectionnés jusqu'à l'instant  $t$ ,
- $\theta^{(t)}$  : le seuil à l'instant  $t$ ,
- $S(t_k, t_{k+1})$  : la similarité moyenne des documents rejetés dans l'intervalle  $[t_k, t_{k+1}]$ ,
- $P(t_k, t_{k+1})$  : la précision du système dans l'intervalle  $[t_k, t_{k+1}]$  donnée par :

$$P(t, t+1) = \frac{R_+^{(t+1)} - R_+^{(t)}}{N^{(t+1)} - N^{(t)}} \quad (3.20)$$

Intuitivement, le seuil augmente si la précision est faible et diminue si très peu de documents pertinents sont sélectionnés. Les mesures  $P(t_k, t_{k+1})$  et  $S(t_k, t_{k+1})$  peuvent alors être utilisées pour décider d'augmenter ou de diminuer le seuil. La fonction de seuillage est donnée par cet algorithme :

- *si*  $P(t_k, t_{k+1}) \leq EP^{(t_{k+1})}$  *alors*  
 $\theta^{(t_{k+1})} = \theta^{(t_k)} + \alpha^{(t_{k+1})} * (1 - \theta^{(t_k)})$
- *sinon*,  
*si*  $S(t_k, t_{k+1}) < \theta^{(t_{k+1})} * D$  *alors*  
 $\theta^{(t_{k+1})} = \theta^{(t_k)} * A + S(t_k, t_{k+1}) * (1 - A)$   
*sinon*  $\theta^{(t_{k+1})} = (1 - \beta^{(t_{k+1})}) * \theta^{(t_k)}$

où  $\alpha^{(t_k)}$  (resp.  $\beta^{(t_k)}$ ) est un coefficient pour augmenter (resp. diminuer) le seuil. Les deux coefficients peuvent être considérés comme une fonction de  $R_+^{(t_k)}$  :

$$\alpha^{(t_k)} = \begin{cases} \alpha_0 \frac{\mu - R_+^{(t_k)}}{\mu} & \text{si } R_+^{(t_k)} \leq \mu \\ 0 & \text{sinon} \end{cases} \quad (3.21)$$

$$\beta^{(t_k)} = \begin{cases} \beta_0 \frac{\mu - R_+^{(t_k)}}{\mu} & \text{si } R_+^{(t_k)} \leq \mu \\ 0 & \text{sinon} \end{cases} \quad (3.22)$$

où  $\alpha^{(0)}$  (= 0.02) et  $\beta^{(0)}$  (= 0.1) sont les paramètres initiaux.  $\mu$  (= 300) représente le nombre maximal de documents pertinents qui doit être utilisé

pour ajuster le seuil et modifier le profil.

Les paramètres  $A$  et  $D$  valent respectivement 0.8 et 0.1. L'introduction du paramètre  $D$  vise à augmenter le rappel. Quand la similarité moyenne entre le profil et les documents rejetés est très faible, le seuil doit diminuer.

$EP^{(t_k)}$  est la précision désirée. Quand cette valeur est fixe, les résultats ne sont pas satisfaisants. Le système est supposé incapable, au début du processus, d'offrir une précision élevée. L'adaptation de cette valeur est effectuée à l'aide d'une fonction d'ascendance progressive donnée par :

$$EP^{(t_{k+1})} = \begin{cases} P^{(0)} + (P^{(final)} - P^{(0)}) \frac{R_+^{(t_{k+1})}}{\mu} & \text{si } R_+^{(t_k)} \leq \mu \\ 0 & \text{sinon} \end{cases} \quad (3.23)$$

où  $P^{(0)}$  (= 0.2) et  $P^{(final)}$  (= 0.6) sont la précision initiale et la précision finale désirées.

### 3.10.2 Modèle de filtrage basé sur la régression logistique

Le modèle le plus intéressant qui se base sur la régression logistique est proposé par Robertson et Walker [Robertson and Walker, 1999]. Ce modèle de filtrage, développé par le laboratoire de recherche Microsoft de Cambridge, est basé sur le modèle probabiliste OKAPI et implanté dans le système Keenbow [Robertson and Walker, 1999] [Robertson et al., 2002].

Le système de filtrage utilise la fonction de pondération BM25 [Robertson and Walker, 1994] pour calculer le score de similarité d'un document vis-à-vis d'un profil :

$$\sum_{t_i \in P} \omega_i^{(1)} \frac{(k_1 + 1).tf_i}{K + tf_i} \frac{(k_3 + 1).qtf_i}{k_3 + qtf_i} \quad (3.24)$$

avec :

$P$  : un profil contenant les termes  $t_i$ ,

$tf_i$  : fréquence d'apparition du terme  $t_i$  dans le document  $D$ ,

$qtf_i$  : fréquence d'apparition du terme  $t_i$  dans le profil  $P$ ,

$K = ((1 - b) + b.dl/avdl)$ ,

$k_1, b$  et  $k_3$  : paramètres dépendant de la nature des profils et du corpus (les valeurs suivantes

ont été utilisées dans les expérimentations effectuées dans TREC,  $k1 = 1.2$ ,  $b = 0.75$ ,  $k3 \rightarrow 7$  ou 1000 dans un profil long),

$dl$  : la longueur du document  $D$ ,

$avdl$  : la longueur moyenne des documents,

$\omega_i^{(1)}$  : poids de Robertson-Sparck Jones du terme  $T$  dans le profil  $P$  :

$$\omega_i^{(1)} = \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \quad (3.25)$$

$r_i$  : nombre de documents pertinents accumulés contenant le terme  $t_i$ ,

$R$  : nombre total de documents pertinents accumulés pour le profil  $P$ ,

$n_i$  : nombre de documents contenant le terme  $t_i$ ,

$N$  : nombre total de documents accumulés.

Le profil initial est construit à partir de la description initiale du besoin en information de l'utilisateur, et la pondération des termes associés sont déduits à partir des statistiques des documents d'une collection quelconque et en utilisant l'équation (3.25).

*a. Apprentissage du profil.* Deux méthodes d'apprentissage du profil ont été proposées :

1. la première méthode, proposée par Robertson [Robertson, 1990], consiste à extraire les termes du document sélectionné et jugé pertinent, et de les ranger dans un ordre décroissant de la valeur *TSV* (*Term Selection Value*), définie comme suit :

$$TSV = r_i \cdot \omega_i^{(1)} \quad (3.26)$$

Les  $m$  meilleurs termes sont rajoutés au profil initial (Dans TREC-11,  $m$  est fixé à 25). L'équation (3.26) a été améliorée pour tenir compte des statistiques des documents non pertinents sélectionnés [Robertson et al., 1998] :

$$TSV = (r_i/R - \alpha \cdot s_i/S) \cdot \omega_i^{(1)} \quad (3.27)$$

avec,  $\alpha \in [0, 1]$ ,  $s_i$  est le nombre de documents non pertinents contenant le terme  $t_i$ , et  $S$  le nombre de documents non pertinents sélectionnés pour le profil.

2. la deuxième méthode, proposée à partir de TREC-8 [Robertson and Walker, 1999], se base sur les statistiques des nouveaux termes contenus dans le document pertinent sélectionné. Le principe de la méthode consiste à calculer pour chaque terme une valeur *NTSV*, et de

sélectionner seulement ceux dont la valeur  $NTSV$  est supérieure à un certain seuil ("c"). La formule  $NTSV$  est donnée comme suit :

$$NTSV = r_i \cdot \log \left( \frac{N}{n_i} \right) - \log \binom{R}{r_i} - \log(V) \quad (3.28)$$

avec  $V$  la taille du vocabulaire utilisé (nombre de termes distincts dans la collection de documents accumulés); le facteur  $\binom{R}{r_i} = \frac{R!}{r_i!(R-r_i)!}$ .

b. *Calibrage et adaptation de la fonction de seuillage.* L'initialisation du seuil sera expliquée dans cette section. Le processus de filtrage dans Keenbow consiste à sélectionner un document si sa probabilité de pertinence est supérieure à un certain seuil. La probabilité de pertinence d'un document est donnée par un modèle de calibrage du score du document. Le modèle de base de calibrage de score est le suivant :

$$\log \frac{p_d}{1-p_d} = \beta + \gamma \frac{s_d}{ast1} \quad (3.29)$$

où  $p_d$  est la probabilité de pertinence du document  $d$ ,  $s_d$  est le score du document  $d$ , et  $ast1$  la moyenne des scores de 1% meilleur documents sélectionnés. Les valeurs initiales de  $\alpha$ ,  $\gamma$  et  $ast1$  sont estimés par la régression logistique en utilisant une collection d'entraînement [Robertson et al., 1998].

A partir du score d'un document et la valeur estimée de  $ast1$ , l'équation (3.29) est utilisée pour estimer la valeur de pertinence du document. Le score calibré  $c_d$  peut être converti en une probabilité  $p_d$  :

$$\begin{aligned} c_d &= \beta + \gamma \frac{s_d}{ast1} \\ p_d &= \frac{e^{c_d}}{1+e^{c_d}} \end{aligned} \quad (3.30)$$

Au fur et à mesure que les jugements de pertinence sont obtenus,  $ast1$  est ré-estimé et  $\beta$  est corrigé ( $\gamma$  reste inchangée). Soit un ensemble de documents jugés, où  $r$  documents sont jugés pertinents. La ré-estimation de  $\beta$  est basée sur un argument Bayésien. Afin d'éviter qu'elle prenne des valeurs inattendues à cause du peu de données, la probabilité Bayésienne à priori est estimée sur  $m$  documents, sachant que  $m$  est tel que, la probabilité à priori d'avoir des estimations correctes est de 0.5. La ré-estimation de  $\beta^{(n)}$ , avec  $c_d^{(n)}$  et  $p_d^{(n)}$  les valeurs correspondantes, se fait d'une façon itérative, et donnée par la formule de descente du gradient suivante :

$$\beta^{(n+1)} = \beta^{(n)} + \frac{r - \sum_{d \in D_{rel}} p_d^{(n)} + m \frac{1 - \exp(\beta^{(n)} - \beta^{(0)})}{2(1 + \exp(\beta^{(n)} - \beta^{(0)}))}}{\sum_{d \in D_{rel}} p_d^{(n)} (1 - p_d^{(n)}) + m \frac{\exp(\beta^{(n)} - \beta^{(0)})}{(1 + \exp(\beta^{(n)} - \beta^{(0)}))^2}} \quad (3.31)$$

$\beta^{(0)}$  est l'estimation initiale obtenue par l'application de la régression de l'équation (3.29) sur la collection d'entraînement. La valeur de  $m$  est fixée à 3 dans TREC-9.  $D_{rel}$  est un ensemble de documents sélectionnés, dont  $r$  documents sont pertinents. Le processus de calcul de  $\beta$  est réitéré jusqu'à ce que la variation entre deux itérations soit inférieure à une constante  $\epsilon = 0.01$ .

La pertinence d'un document est exprimée en terme de probabilités. Au démarrage du processus de filtrage, un ensemble de documents d'entraînement est utilisé pour initialiser le seuil et le profil de l'utilisateur. L'initialisation du seuil est obtenue à partir d'une liste de valeurs, appelée "échelle de valeurs". Une valeur dans l'échelle représente un seuil permettant d'optimiser la fonction d'utilité. Les valeurs de l'échelle et du seuil initial sont données de façon arbitraire. Aucun moyen théorique ne permet de les déterminer correctement. Cependant, le seuil initial est fixé à une valeur basse dans l'échelle qui permet de sélectionner un certain nombre de documents. Le tableau 3.4 est un exemple d'échelle utilisée dans TREC-8 :

$p_d$	$\log \frac{p_d}{1-p_d}$	
0.5	0	
0.4	-0.4	
0.25	-1.1	
0.18	-1.5	
0.13	-1.9	
0.1	-2.2	
0.07	-2.6	
0.05	-2.9	
0.04	-3.2	valeur initiale du seuil

Tableau 3.4 – Echelle de valeurs utilisée dans TREC-8

Dans TREC-11, les valeurs des différents paramètres utilisés sont initialisées comme suit :

- les paramètres de la fonction BM25 :  $k_1 = 1.3$  et  $b = 0.55$  ;
- le calibration des scores : La valeur initiale de  $\beta = -0.66$ ,  $m = 3$  et  $\gamma = 2.9$  ;
- apprentissage du profil : le nombre de documents pertinents utilisé est égal à 20, le nombre de termes maximum (resp. minimum) considéré à chaque apprentissage est de 25 (resp. 3), le critère de sélection, en utilisant la formule *NTSV*, est égal à 2.

### 3.10.3 Modèle de filtrage basé sur la distribution des scores

Une autre catégorie de systèmes de filtrage se base sur les distributions des scores des documents (pertinents et non pertinents) pour l'adaptation de la fonction de seuillage. L'idée de base est d'estimer les distributions des scores des documents pertinents et non pertinents, puis optimiser la fonction d'utilité en se basant sur ces distributions. Deux techniques de formalisation de la distribution des scores des documents sont recensées pour ce type de modèles. La première technique, SDS (Simple Distribution des Scores), suppose que la distribution de probabilités des scores des documents pertinents (resp. non pertinents) suit une loi normale (resp. une loi exponentielle) [Arampatzis et al., 2000] [Zhang and Callan, 2001]. La seconde technique, LDS (Linéarisation de la Distribution des Scores), qui fait l'objet de notre travail, suppose que la forme de la distribution de probabilités des scores des documents pertinents et non pertinents est inconnue, et tente de la "dessiner" ou de la construire en utilisant la méthode de régression linéaire [Boughanem et al., 2001] [Boughanem et al., 2004c].

Dans ce qui suit, nous nous limitons à présenter le processus d'adaptation de la fonction de seuillage basé sur l'optimisation de la fonction d'utilité. L'apprentissage des profils peut être effectué par n'importe quelle technique de reformulation ou d'apprentissage. Avant de décrire les travaux ci-dessus, nous consacrons la section suivante à la formalisation du problème d'optimisation de la fonction d'utilité.

#### 3.10.3.1 Optimisation de la fonction d'utilité

L'idée sous jacente à ces techniques consiste, tout d'abord, à représenter la fonction d'utilité en fonction des distributions des scores des documents pertinents et non pertinents, puis déduire le score qui permet d'optimiser (de maximiser) cette fonction.

Plus précisément, supposons que les scores des documents pertinents sont représentés par la densité de probabilités  $P_r(x)$  où  $r$  est le nombre de documents pertinents total parmi les documents filtrés (documents pertinents sélectionnés ou pas). La quantité  $r.P_r(x)$  représente le nombre de documents pertinents dont le score appartient à l'intervalle  $[x, x + dx]$ . On déduit alors que le nombre de documents pertinents ayant un score supérieur au seuil  $\theta$  est donné par :

$$R_+(\theta) = r \int_{\theta}^{+\infty} P_r(x) dx \quad (3.32)$$

Similairement, le nombre de documents non pertinents ayant un score supérieur à  $\theta$  est :

$$S_+(\theta) = (n - r) \int_{\theta}^{+\infty} P_{nr}(x) dx \quad (3.33)$$

Où  $P_{nr}(x)$  est la densité de probabilités des scores des documents non pertinents et  $n$  le nombre total de documents filtrés.

Ainsi, à partir des équations (3.32) et (3.33), on peut déduire que le nombre de documents pertinents non retrouvés  $R_-(\theta)$  (resp. non pertinents non retrouvés  $S_-(\theta)$ ) pour le seuil  $\theta$  est :

$$R_-(\theta) = r \int_{-\infty}^{\theta} P_r(x) dx \quad (3.34)$$

$$S_-(\theta) = (n - r) \int_{-\infty}^{\theta} P_{nr}(x) dx$$

En s'inspirant des quatre dernières équations, la fonction d'utilité  $U$  (3.1) peut être réécrite en fonction de  $\theta$  :

$$U_{(\lambda_1, \lambda_2, \lambda_3, \lambda_4)}(\theta) = U(R_+(\theta), S_+(\theta), R_-(\theta), S_-(\theta)) \quad (3.35)$$

Optimiser  $U$ , revient donc à maximiser ou minimiser  $U$ . Le seuil optimal est alors obtenu en résolvant l'équation suivante :

$$\frac{dU(R_+(\theta), S_+(\theta), R_-(\theta), S_-(\theta))}{d\theta} = 0 \quad (3.36)$$

Dans la plupart des cas, l'équation (3.36) n'a pas de solutions à cause des intégrales dans la formule, mais elle peut être résolue numériquement. Pour toute fonction d'utilité, la dérivée de la fonction (3.1) donne :

$$\frac{dU_{(\lambda_1, \lambda_2, \lambda_3, \lambda_4)}(\theta)}{d\theta} = -\lambda_1 r P_r(\theta) - \lambda_2 (n - r) P_{nr}(\theta) + \lambda_3 r P_r(\theta) + \lambda_4 (n - r) P_{nr}(\theta) \quad (3.37)$$

Par un simple calcul et en tenant compte des équations (3.36) et (3.37), on obtient :

$$\lambda\rho P_r(\theta) = P_{nr}(\theta) \quad (3.38)$$

où  $\lambda$  et  $\rho$  sont calculés comme suit :

$$\lambda = \frac{\lambda_3 - \lambda_1}{\lambda_2 - \lambda_4} \quad \text{et} \quad \rho = \frac{r}{n - r}$$

$\rho$  est la densité relative des documents pertinents sur les documents non pertinents, est le nombre de documents filtrés.

L'équation (3.38) permet de retrouver un seuil optimal, ceci en remplaçant  $P_r(x)$  ( $P_{nr}(x)$ ) par une distribution de probabilités des scores des documents pertinents (non pertinents) associée.

Nous présentons dans ce qui suit les techniques d'adaptation de la fonction de seuillage basées sur l'approches SDS.

### 3.10.3.2 Fonction de seuillage par la technique SDS

Le seuil optimal dans le cas de le SDS est calculé en supposant que la distribution des scores des documents pertinents suit une loi normale et celle des documents non pertinents une loi exponentielle. Pour déterminer ce seuil, il suffit de remplacer  $P_r(\theta)$  et  $P_{nr}(\theta)$  dans l'équation (3.38) par leurs densités correspondantes, puis rechercher le seuil  $\theta$  permettant de maximiser l'équation.

La densité de probabilités de la distribution des scores des documents pertinents est donc donnée comme suit :

$$P_r(\theta) = \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(-\frac{(\theta - \mu_r)^2}{2\sigma_r^2}\right) \quad (3.39)$$

où  $\sigma_r$  est la moyenne des scores, et  $\mu_r$  est l'écart type des scores des documents pertinents.

La densité de probabilités de la distribution des scores des documents non pertinents est :

$$P_{nr}(\theta) = c_1 \exp(-c_2\theta) \quad (3.40)$$

où  $c_1$  et  $c_2$  sont des paramètres à calculer.

En remplaçant les équations (3.39) et (3.40) dans l'équation (3.38), on obtient :

$$\lambda\rho\frac{1}{\sqrt{2\pi\sigma_r^2}}\exp\left(-\frac{(\theta-\mu_r)^2}{2\sigma_r^2}\right) = c_1 \exp(-c_2\theta) \quad (3.41)$$

En appliquant un logarithme des deux cotés et en déplaçant tous les éléments dans la partie gauche de l'équation, nous obtenons :

$$\ln\left(\lambda\rho\frac{1}{c_1\sqrt{2\pi\sigma_r^2}}\right) - \frac{1}{2\sigma_r^2}(\theta - \mu_r)^2 + c_2\theta = 0 \quad (3.42)$$

Après quelques transformations, nous obtenons le polynôme de  $2^{nd}$  degré suivant :

$$\frac{1}{2}a\theta^2 - b\theta + \frac{1}{2}c = 0 \quad (3.43)$$

où :

$$a = \frac{1}{\sigma_r^2}$$

$$b = \frac{\mu_r}{\sigma_r^2} + c_2$$

$$c = \frac{\mu_r^2}{\sigma_r^2} + 2\alpha m - 2\ln\left(\lambda\rho\frac{1}{c_1\sqrt{2\pi\sigma_r^2}}\right)$$

Le discriminant est  $\Delta = b^2 - ac$ , et le seuil optimal est donné par :

$$\theta = \begin{cases} (b - \sqrt{\Delta})/a, & \text{si } \Delta \geq 0 \\ +\infty, & \text{si } \Delta < 0 \end{cases} \quad (3.44)$$

En fait, l'estimation de ces deux densités nécessite deux échantillons distincts. Le seul échantillon dont on dispose pendant le processus de filtrage est celui composé des documents effectivement filtrés (ceux ayant un score supérieur au seuil). Or, cet échantillon de

documents pourrait être biaisé, car il existe des documents pertinents (resp. non pertinents) n'ayant pas été sélectionnés (puisque leurs scores sont inférieurs au seuil). Nous décrivons dans ce qui suit deux façons d'estimer ces distributions : une solution biaisée proposée dans [Arampatzis et al., 2000] et une autre non biaisée dans [Zhang and Callan, 2001] [Bennett, 2003].

**Estimation biaisée.** Dans cette approche proposée par Arampatzis [Arampatzis et al., 2000], les paramètres  $(\mu, \sigma, c_1, c_2)$  sont estimés de manière incrémentale et à partir d'un échantillon de documents ayant leurs scores supérieurs au seuil. Ces paramètres sont estimés de la manière suivante : Considérons  $P$  un profil représenté par un vecteur  $P = [\omega_1, \dots, \omega_m]$ , où  $\omega_i$  est le poids du terme  $i$  dans le profil  $P$ , et un document  $D_j$  représenté similairement par  $D_j = [d_{1j}, \dots, d_{mj}]$ , où  $d_{kj}$  est le poids du  $k^{eme}$  terme dans le document  $D_j$ . Le score de similarité entre le document  $D_j$  et le profil  $P$  est donné par :

$$\langle P, D_j \rangle = \sum_i \omega_i d_{ij} \quad (3.45)$$

On considère qu'au fur et à mesure que le système filtre les documents, l'utilisateur juge ces documents. Ainsi, à partir de l'échantillon de scores des documents pertinents sélectionnés, l'estimation de la moyenne  $\mu_r$  est donnée par la formule suivante :

$$\mu_r = \frac{1}{r} \sum_{j=1}^r \langle P, D_j \rangle = \frac{1}{r} \langle P, \sum_{j=1}^r D_j \rangle \quad (3.46)$$

En effet, la somme des documents pertinents est quantifiable et pourra être mise à jour incrémentalement. Cette manière de calculer la moyenne a été déjà présentée par Callan [Callan, 1998].

La variance  $\sigma_r^2$  est calculée par  $\sigma_r^2 = \mu_r^{(2)} - \mu_r^2$ , où la somme de la moyenne des carrées des scores est :

$$\mu_r^{(2)} = \frac{1}{r} \sum_{j=1}^r \langle P, D_j \rangle^2 = \frac{1}{r} \sum_{jk} \omega_i \left( \sum_{j=1}^r d_{ij} d_{kj} \right) \omega_k \quad (3.47)$$

Où  $d_{ij}$  est le poids du  $i^{eme}$  composant du document  $D_j$ .

Cette façon d'estimer les paramètres des deux distributions est donc biaisée, car les échantillons sont composés seulement de documents ayant des scores supérieurs au seuil. Tous les documents pertinents non sélectionnés ne sont pas pris en compte. Ces échantillons pourraient donc induire des erreurs dans le calcul de la variance et de la moyenne [Zhang and Callan, 2001]. Une expérimentation réalisée dans le cadre de TREC, sur le profil 3 de TREC9, a montré que la moyenne et la variance de la distribution gaussienne sont de (0.4343, 0.0169) quand on considère tous les documents pertinents dans la collection TREC9, alors que, si l'échantillon de documents pertinents est restreint aux documents ayant des scores supérieurs au seuil, la moyenne et la variance sont de (0.4551, 0.007).

Dans le but d'obtenir une estimation non biaisée, Zhang et Callan [Zhang and Callan, 2001] proposent une méthode d'estimation basée sur le maximum de vraisemblance. Cette méthode est présentée dans la section suivante.

**Estimation non biaisée.** La méthode proposée par Zhang et Callan [Zhang and Callan, 2001] tient compte de l'évolution progressive du seuil et s'inspire du théorème de Bayes pour estimer les paramètres des deux distributions. La forme de la distribution proposée pour des documents pertinents est la suivante :

$$P_r(score/R = rel) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(score - \mu)^2}{2\sigma^2}\right) \quad (3.48)$$

et celle des documents non pertinents est donnée comme suit :

$$P_{nr}(score/R = nrel) = \lambda \exp(-\lambda(score - c)) \quad (3.49)$$

Avec :

$P_r(score/R = rel)$  ( $P_{nr}(score/R = nrel)$ ) est la probabilité qu'un document ait un score sachant qu'il est pertinent (non pertinent),

$\mu$  : moyenne de la loi gaussienne,

$\sigma$  : variance de la loi gaussienne,

$\lambda$  : variance de la loi exponentielle,

$c$  : score minimum des documents non pertinents sélectionnés.

En considérant qu'à un instant donné du processus de filtrage,  $n$  documents sont délivrés à l'utilisateur et des jugements de pertinence sont fournis par ce dernier pour chaque document. Ces documents peuvent être traités comme étant un échantillon de documents d'entraînement. Chaque jugement de pertinence d'un document  $D_i$  est représenté par le triplet  $(R_i, score_i, \theta_i)$ , où  $R_i$  est le jugement de pertinence du  $i^{eme}$  document :

$$R_i = \begin{cases} rel & \text{pour un document pertinent} \\ nrel & \text{pour un document non pertinent} \end{cases}$$

où :

$score_i$  : le score du document  $D_i$ ,

$\theta_i$  : le seuil du profil donné, quand le document  $D_i$  est délivré.

Afin de représenter les distributions de probabilités des scores des documents, quatre paramètres sont nécessaires à estimer  $(\mu, \sigma, \lambda, p)$  où  $p$  est la proportion de documents pertinents.

Soit  $D$  l'ensemble de documents sélectionnés par le système à un instant donné du filtrage. En appliquant la formule de Bayes, la valeur optimale de  $H = (\mu, \sigma, \lambda, p)$  est donnée par :

$$H^* = \underset{H}{\operatorname{argmax}} p(H/D) = \underset{H}{\operatorname{argmax}} \frac{p(D/H)p(H)}{p(D)} \quad (3.50)$$

Pour des raisons de simplification, on suppose qu'aucune information n'est connue sur la distribution de  $H$  et que la probabilité à priori  $p(H)$  est uniforme,  $p(D)$  est une constante indépendante de  $H$ . Ainsi, la solution la plus probable de  $H$  est celle qui maximise la probabilité de l'échantillon de documents pertinents :

$$H^* = \underset{(\mu, \sigma, \lambda, p)}{\operatorname{argmax}} \sum_{i=1}^N \log(p(score = score_i, R_i/H, score > \theta_i)) \quad (3.51)$$

où :

$$p(score = score_i, R_i/H, score > \theta_i) = \frac{p(score = score_i/R_i, H)p(R_i/H)}{p(score/\theta_i/H)} \quad (3.52)$$

Soit  $f_1(\mu, \sigma, \theta_i)$  la probabilité qu'un document ayant un score supérieur au seuil  $\theta_i$  soit pertinent et  $f_2(\lambda, \theta_i)$  la probabilité qu'un document ayant un score supérieur au seuil  $\theta_i$  soit non pertinent.  $f_1$  et  $f_2$  sont définies comme suit :

$$\begin{aligned} f_1(\mu, \sigma, \theta_i) &= \int_{\theta_i}^{+\infty} p(\text{score} = x/R = \text{rel})dx \\ &= \int_{\theta_i}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \end{aligned} \quad (3.53)$$

$$\begin{aligned} f_2(\lambda, \theta_i) &= \int_{\theta_i}^{+\infty} p(\text{score} = x/R = \text{nrel})dx \\ &= \int_{\theta_i}^{+\infty} \lambda \exp(-\lambda(x-c))dx \\ &= \exp(-\lambda(\theta_i - c)) \end{aligned} \quad (3.54)$$

Si nous utilisons  $g(\mu, \sigma, \lambda, p, \theta_i)$  pour représenter la probabilité qu'un document donné ait un score supérieur au seuil  $\theta_i$ , on obtient ceci :

$$\begin{aligned} g(\mu, \sigma, \lambda, p, \theta_i) &= p(\text{score} > \theta_i/H) \\ &= p.f_1(\mu, \sigma, \theta_i) + (1-p).f_2(\lambda, \theta_i) \end{aligned} \quad (3.55)$$

Si nous supposons que la somme des aires formées par les deux distributions gaussienne et exponentielle est égale 1, alors  $p.f_1(\mu, \sigma, \theta_i)$  correspond à l'aire définie par la distribution gaussienne à droite du seuil  $\theta_i$ , et  $(1-p).f_2(\lambda, \theta_i)$  à l'aire définie par la distribution exponentielle à droite de  $\theta_i$ .

A partir de ces hypothèses et des équations (3.52) et (3.55), la fonction logarithmique de l'équation (3.51), peut être remplacée par la formule  $LP$ , d'où :

$$(\mu^*, \sigma^*, \lambda^*, p^*) = \underset{(\mu, \sigma, \lambda, p)}{\operatorname{argmax}} \sum_{i=1}^N LP_i \quad (3.56)$$

où, pour les documents pertinents :

$$LP_i = \frac{(\text{score}_i - \mu)^2}{2\sigma^2} + \ln(p/(\sigma.g(\mu, \sigma, \lambda, p, \theta_i))) \quad (3.57)$$

et pour les documents non pertinents :

$$LP_i = -\lambda(\text{score}_i - c) + \ln((1-p)\lambda/g(\mu, \sigma, \lambda, p, \theta_i)) \quad (3.58)$$

Nous remarquons que les équations (3.57) et (3.58) n'admettent pas de solutions uniques, des méthodes numériques peuvent être utilisées. Zhang et Callan [Zhang and Callan, 2001] ont utilisé la méthode itérative de la descente de gradient conjuguée (abréviation de CG de Conjugate Gradient), pour retrouver une solution rapprochée.

### 3.10.3.3 Discussion sur les modèles basés sur la distribution des scores

Les deux méthodes présentées ci-dessus supposent que les scores des documents pertinents et non pertinents suivent des distributions connues. Même si cette idée est acceptable, elle présente tout de même les limites et les inconvénients suivants :

- tout d’abord dans un contexte expérimental, si on admet la forme de la distribution *a priori*, elle peut ne pas être valable pour des conditions expérimentales particulières, car les scores des documents restent incontrôlables,
- ensuite, on doit disposer d’un nombre minimum de documents dans les échantillons pour avoir des estimations non biaisées.

Pour pallier ces problèmes, nous avons proposé [Boughanem et al., 2002], une approche d’adaptation de la fonction de seuillage LDS (Linéarisation de la Distribution des Scores) qui rentre dans cette catégorie de modèles. L’approche LDS suppose que les distributions de probabilités des scores des documents pertinents et non pertinents sont inconnues, mais propose d’estimer les probabilités discrètes des scores des documents, puis à ”dessiner” la distribution des scores en utilisant une régression linéaire [Boughanem et al., 2004c]. La régression linéaire permet de transformer une distribution de probabilités discrète en une densité de probabilités continue. Une formalisation plus détaillée de cette approche est présentée dans la deuxième partie.

## 3.11 Conclusion

Ce chapitre a porté essentiellement sur l’étude des systèmes de filtrage d’information, plus particulièrement sur le filtrage cognitif, dit aussi basé sur le contenu. La majorité des approches proposées dans la littérature se basent principalement sur des modèles de recherche d’information augmentés par une fonction de décision. Les travaux dans le domaine de filtrage s’intéressent principalement à l’apprentissage du profil et l’adaptation de la fonction de décision.

Le problème majeur de l’adaptation vient de la nature extrêmement dynamique du processus de filtrage. Il est donc difficile d’arriver à un point de stabilité, car d’une part on ne connaît pas les informations qui vont arriver et d’autre part, on ne maîtrise pas les jugements de l’utilisateur. Le système doit continuellement tenir compte des informations qu’il reçoit et doit exploiter ces informations pour tenter d’être toujours plus performant. Une performance qui est souvent mesurée par une fonction d’utilité, où le système doit optimiser cette fonction.

## Deuxième partie

# Contribution aux systèmes de filtrage incrémentaux



# Chapitre 4

## Modèle de filtrage incrémental d'information

### 4.1 Introduction

Nous avons étudié dans le troisième chapitre, quelques travaux sur les systèmes de filtrage d'information basés sur le contenu. Ces systèmes sont essentiellement basés sur des modèles de recherche d'information auxquels sont rajoutées des fonctions d'adaptation et une fonction de décision le plus souvent de type seuil. Une grande partie des ces travaux s'est focalisée sur l'apprentissage des profils et l'adaptation de la fonction de décision, on parle aussi de fonction de seuillage. L'apprentissage peut être effectué d'une façon incrémentale (appelé *adaptive* dans la terminologie TREC [Voorhees, 2002]), où les différents facteurs sont déduits à partir des documents déjà filtrés, ou d'une façon différée (appelé *Batch* dans la terminologie TREC), où les différents paramètres du système sont déduits à partir d'une collection de documents d'apprentissage, dite d'entraînement.

Le filtrage incrémental représente évidemment le cas réel de filtrage. En effet, au démarrage du processus de filtrage on ne dispose d'aucune connaissance sur les documents à filtrer. Ce cas de filtrage est plus difficile à entreprendre en raison des difficultés associées au bon démarrage du processus et à l'apprentissage permanent du profil, l'adaptation de la fonction de seuillage et les autres facteurs de filtrage. Notre travail s'inscrit dans cette catégorie de système. Nous proposons précisément des méthodes d'apprentissage du profil et d'adaptation de la fonction de décision permettant de faire évoluer ces éléments au fur

et à mesure que des documents pertinents sont sélectionnés. Il faut noter que, quand on parle de la fonction de décision on fait principalement référence à la manière d'identifier ce seuil, on parle souvent de fonction de seuillage.

Ce chapitre est organisé comme suit : la section 4.2 présente nos motivations. Nous dressons tout d'abord dans cette section quelques limites des modèles de filtrage présentés précédemment, puis nous proposons des solutions permettant de pallier ces limites. Ces solutions concernent l'apprentissage du profil et l'adaptation de la fonction de seuillage. Dans la section 4.3, nous présentons notre modèle de base du système de filtrage d'information, à savoir la représentation des documents et des profils, leur initialisation et le processus de filtrage de documents. La section 4.4 présente la méthode d'apprentissage des profils basée sur le principe de renforcement. Enfin, la section 4.5 décrit la méthode d'adaptation du seuil basée sur la linéarisation de la distribution des scores des documents.

## 4.2 Motivations

Nous rappelons que notre travail s'inscrit dans le cadre du filtrage incrémental. Dans ce contexte, les principales limites des travaux présentés dans le chapitre précédent se résument comme suit :

1. tout d'abord, au niveau de l'apprentissage des profils, de nombreux modèles de filtrage, tels que le modèle CAFES [Zhai et al., 1998], Wu et al. [Wu et al., 2001] et Wang et al. [Wang et al., 2001] ont recours à une collection d'entraînement. Ceci pose une première limite, car dans une situation courante de filtrage incrémental d'information, nous ne disposons d'aucune connaissance sur les informations à filtrer.

De plus, la plupart des méthodes d'apprentissage de profils se basant sur des échantillons de documents se voient dépendantes de la taille de l'échantillon. Il a été montré notamment dans les travaux de Zhang et al. [Zhang, 2004] que les méthodes basées sur la régression logistique sont plutôt performantes lorsque le nombre de documents de l'échantillon est conséquent. En revanche, d'autres, tels que les classifieurs bayésiens ou Rocchio, fonctionnent mieux lorsque le nombre de documents de l'échantillon est réduit.

Pour pallier ces deux inconvénients, nous proposons une méthode d'apprentissage du profil purement incrémentale, qui ne nécessite aucune connaissance, autre que le profil

initial de l'utilisateur, au démarrage du processus de filtrage. Elle permet également d'apprendre les profils d'une manière uniforme tout au long du processus de filtrage, comparativement aux méthodes basées sur Rocchio ou sur la régression logistique. Cette méthode consiste tout d'abord à construire un profil *temporaire* à partir d'un document sélectionné et jugé pertinent par l'utilisateur. Ce profil *temporaire* permet de sélectionner le document en question avec un score le plus élevé possible. Ce profil *temporaire* est ensuite intégré dans le profil global de l'utilisateur.

Cette méthode permet de mettre à jour le profil de l'utilisateur à chaque réception d'un document pertinent.

La méthode d'apprentissage par renforcement a été présentée et expérimentée dans le cadre des travaux<sup>1</sup> de recherches de Tmar [Boughanem et al., 2002] [Tmar, 2002] [Tmar et al., 2002]. Nous avons apporté des améliorations qui permettent notamment de mieux contrôler le score de renforcement (ou le seuil maximum désiré) et de mieux construire le profil *temporaire*. Cette nouvelle modélisation de la méthode d'apprentissage a permis d'améliorer considérablement les performances de notre modèle de filtrage d'information.

2. La seconde limite que l'on peut relever dans les travaux précédents concerne l'adaptation de la fonction de seuillage. Nous nous limitons aux modèles basés sur la distribution des scores des documents pertinents et non pertinents. Nous rappelons que ces méthodes supposent que les scores des documents pertinents et non pertinents suivent certaines lois de probabilité connues à l'avance. Même si cette idée est acceptable, elle présente tout de même les inconvénients suivants :
  - Tout d'abord dans un contexte expérimental, si on admet la forme de la distribution des scores *à priori*, elle peut ne pas être valable pour des conditions expérimentales particulières, car les scores des documents restent incontrôlables. Ceci a été notamment mis en évidence dans les travaux de Zhang [Zhang and Callan, 2001] et présentés dans la section (3.10.3) du chapitre 3 ;
  - on doit disposer d'un nombre minimum de documents dans les échantillons pour avoir des estimations non biaisées.

L'approche que nous proposons, pour notre part, suppose que les distributions des scores des documents sont inconnues *à priori*, mais propose d'estimer les probabilités discrètes des scores des documents, puis de "dessiner" la distribution des scores en

---

<sup>1</sup>Les résultats de ces travaux sont présentés lors de notre participation au programme d'évaluation de TREC-2002

utilisant une régression linéaire [Boughanem et al., 2004c]. Une fois ces distributions construites, nous proposons de réécrire la fonction d'utilité en fonction de ces distributions, puis de déduire le score (seuil) qui permet d'optimiser cette fonction. La construction d'une distribution de probabilités continue des scores des documents, consiste tout d'abord à décomposer les scores en plusieurs intervalles, puis calculer la probabilité qu'un score soit un intervalle. Une régression linéaire est ensuite utilisée pour convertir la distribution de probabilités discrète en une distribution continue. Ce travail a été réalisé dans un premier temps dans le cadre des travaux de Tmar<sup>2</sup>, auxquels nous avons apporté des améliorations liées à la manière de construire des intervalles et de linéariser les probabilités. Nous avons proposé également une nouvelle modélisation et méthode de résolution de l'équation d'optimisation de la fonction d'utilité.

Avant de présenter les processus d'apprentissage du profil et d'adaptation de la fonction de seuillage, nous décrivons le modèle de base du filtrage d'information que nous avons utilisé.

### 4.3 Modèle de base

Le modèle de base que nous avons utilisé est basé sur une approche vectorielle. Les documents et les profils sont représentés sous forme de listes de termes pondérés.

Un profil  $P^{(t)}$  (resp. un document  $D_j^{(t)}$ ) est représenté par un ensemble de termes sans les mots vides. Il est donné sous une forme vectorielle, où à chaque terme  $t_i$  est associé un poids  $w_i^{(t)}$  (resp.  $d_{ij}^{(t)}$ ), où  $t$  représente l'instant où le système reçoit un document. Le profil  $P^{(t)}$  est appelé profil global de l'utilisateur.

La représentation des profils et des documents est construite à l'aide d'un processus d'indexation automatique. Il consiste à extraire les termes les plus représentatifs du contenu d'un document (resp. profil). Les différentes étapes du processus d'indexation sont les suivantes :

1. Identifier les mots du texte du document (resp. profil). Cette étape permet d'éliminer tous les éléments du contenu du document (resp. profil) qui n'ont aucune représentation sémantique (caractères spéciaux, ponctuations, balises, etc.) ;

---

<sup>2</sup>Nous qualifions tout ce qui relève des travaux effectués initialement avec M. Tmar de modèle initial

2. Eliminer les mots vides. En s'inspirant d'une liste de mots vides, tous les mots du document (resp. profil) figurant dans la liste sont ôtés.
3. Normaliser tous les mots significatifs (radicalisation). Un radical d'un terme est obtenu selon le type de la langue du document (resp. profil) par une méthode spécifique :
  - l'algorithme de Porter [Porter, 1980] pour les documents en anglais,
  - la troncature pour les autres langues (français, allemand, etc.).

### 4.3.1 Initialisation du système

#### 4.3.1.1 Initialisation du profil et des documents

Initialement, les termes du profil peuvent être saisis par un utilisateur ou extraits à partir d'un ensemble de documents représentant le centre d'intérêt de l'utilisateur. Le poids du terme dans le profil à l'étape initiale est calculé comme suit :

$$w_i^{(0)} = \frac{t f p_i}{\max_j (t f p_j)} \quad (4.1)$$

où,  $t f p_i$  est la fréquence du terme  $t_i$  dans le profil. Ce poids sera ajusté par apprentissage chaque fois qu'un document est sélectionné et jugé pertinent.

A chaque arrivée d'un document, soit  $D_j^{(t)}$ , celui-ci est indexé. Le résultat de cette opération est une liste de termes pondérés. Le poids  $d_{ij}^{(t)}$  de chaque terme  $t_i$  dans le document  $D_j^{(t)}$  est calculé par une fonction de pondération utilisée dans le système de recherche d'information **Mercure** [Boughanem, 2000] :

$$d_{ij}^{(t)} = \frac{t f_i^{(t)}}{h_3 + h_4 \frac{d l^{(t)}}{\Delta l^{(t)}} + t f_i^{(t)}} \log\left(\frac{N^{(t)}}{n_i^{(t)}} + 1\right) \quad (4.2)$$

Où :

$t f_i^{(t)}$  : fréquence du terme  $t_i$  dans le document  $D_j^{(t)}$ , à l'instant  $t$ ,

$h_3, h_4$  : paramètres constants, dans les expérimentations  $h_3 = 0.2$  et  $h_4 = 0.7$ ,

$d l^{(t)}$  : nombre de termes dans le document  $D_j^{(t)}$ ,

$\Delta l^{(t)}$  : longueur moyenne des documents,

$N^{(t)}$  : nombre de documents examinés jusqu'à l'instant  $t$ ,

$n_i^{(t)}$  : nombre de documents parmi  $N^{(t)}$  contenant le terme  $t_i$ .

Les différents paramètres du système sont mis à jour au fur et à mesure que des documents pertinents sont sélectionnés. Ces paramètres concernent  $\Delta l^{(t)}$ ,  $N^{(t)}$  et  $n_i^{(t)}$ .

#### 4.3.1.2 Initialisation du seuil

Le seuil initial est la première valeur du seuil  $\theta$  utilisée pour filtrer le premier document arrivé. Dans notre cas, la valeur du seuil initial dépend de l'état initial du système, si un échantillon de documents d'entraînement est utilisé ou non :

- Dans le cas, où aucun document pertinent n'est disponible au démarrage, alors le seuil initial est fixé à une valeur arbitraire (zéro par exemple),
- Dans le cas, où un échantillon de documents d'entraînement est utilisé, alors le seuil initial est estimé par la méthode d'adaptation de la fonction seuillage.

#### 4.3.2 Processus de filtrage

Le processus de filtrage d'information consiste à mesurer un score, noté  $rsv(D_j^{(t)}, P^{(t)})$  (équation (4.3)), entre le document et le profil. Ce score est défini par le produit scalaire entre le document  $D_j^{(t)}$  et le profil  $P^{(t)}$ .

$$rsv(D_j^{(t)}, P^{(t)}) = \sum_{i=1}^n d_{ij}^{(t)} \cdot w_i^{(t)} \quad (4.3)$$

Ce score est ensuite comparé à un seuil ( $\theta^{(t)}$ ) de filtrage, pour décider si le document est accepté ou non :

$$\begin{cases} \text{si } rsv(D_j^{(t)}, P^{(t)}) > \theta^{(t)} & \text{accepter le document } D_j^{(t)} \\ \text{sinon} & \text{rejeter le document } D_j^{(t)} \end{cases} \quad (4.4)$$

Le profil, le seuil et les statistiques liés à la pondération des termes des documents sont appris à chaque sélection d'un document pertinent. Le processus d'apprentissage du profil que nous utilisons est basé sur le principe de renforcement et l'adaptation de la fonction de seuillage est réalisée par une méthode de linéarisation de distribution des scores des documents pertinents et non pertinents.

## 4.4 Apprentissage du profil

La méthode d'apprentissage du profil que nous utilisons est purement incrémentale. Contrairement aux travaux cités dans le chapitre précédent, qui rappelons-le, utilisent le plus souvent des collections d'entraînement ou effectuent l'apprentissage sur un lot de documents (pertinents et/ou non pertinents), nous proposons une méthode qui ne nécessite aucune information supplémentaire, à l'exception du profil initial de l'utilisateur ainsi que les documents filtrés et jugés pertinents par l'utilisateur. La méthode d'apprentissage peut être déclenchée à chaque arrivée d'un document.

L'idée de base de notre méthode d'apprentissage est basée sur le principe de renforcement du profil. A cet effet, nous considérons que quand un document pertinent pour un profil donné est disponible, on cherche à construire un profil *temporaire* permettant de retrouver (ou de sélectionner) ce même document avec un score "fort". Ce profil *temporaire* est ensuite introduit dans le profil global de l'utilisateur. Ceci implique l'expansion et la modification des termes du profil global, simultanément. Le détail de la méthode est présenté dans la section suivante.

### 4.4.1 Apprentissage du profil : principe de renforcement

L'apprentissage par renforcement consiste donc à trouver un profil *temporaire*  $P_x^{(t)}$  qui permet de sélectionner un document  $D_j^{(t)}$  pertinent avec un score fort, soit  $\lambda$  ce score de renforcement. Formellement, soient  $d_{ij}^{(t)}$ , avec  $i = 1 \dots n$ , les poids des termes du document  $D_j^{(t)}$ , et  $pw_i^{(t)}$  les poids (initialement inconnus) des termes dans le profil  $P_x^{(t)}$ , l'objectif de la méthode de renforcement est double :

1. chercher les  $pw_i^{(t)}$  qui satisfont l'équation suivante :

$$\sum_{k=1}^n d_{kj}^{(t)} \cdot pw_k^{(t)} = \lambda \quad (4.5)$$

2. intégrer le profil *temporaire* dans le profil global de l'utilisateur via une formule de distribution de gradient,  $h$ , suivante :

$$\forall t_i \in P_x^{(t)}, \quad w_i^{(t+1)} = h(w_i^{(t)}, pw_i^{(t)}) \quad (4.6)$$

On remarque que l'équation (4.5) admet une infinité de solutions, c'est une équation à plusieurs inconnus  $pw_i^{(t)}$ . Pour pallier ce problème, nous proposons d'ajouter une contrainte pour pouvoir réduire le nombre de solutions et donc arriver à une solution unique.

Avant de donner cette contrainte, nous précisons la notion du profil idéal et du poids idéal.

**Définition 1** *Nous appelons profil idéal à l'instant  $t$ , le profil qui permet de sélectionner tous les documents pertinents et que les documents pertinents.*

**Définition 2** *Le poids idéal d'un terme est son poids dans le profil idéal.*

La contrainte que nous proposons de rajouter est la suivante : compte tenu de l'aspect incrémental de l'apprentissage, nous considérons que les termes du profil, solution de l'équation (4.5), doivent contribuer, à chaque instant, de manière proportionnelle à leur importance réelle dans le profil idéal. Ceci, peut se traduire formellement comme suit : supposons que le poids idéal d'un terme  $t_i$  dans un profil idéal est donné par  $f_i^{(t)}$ , la contrainte ci-dessus peut alors s'écrire :  $pw_i^{(t)} / f_i^{(t)}$  est une constante. Le système d'équations à résoudre devient alors :

$$\left\{ \begin{array}{l} \sum_{i=1}^n d_{ij}^{(t)} \cdot pw_i^{(t)} = \lambda \\ \forall (t_i, t_j) \in D_j^{(t)} \times D_j^{(t)}, \frac{pw_i^{(t)}}{f_i^{(t)}} = \frac{pw_j^{(t)}}{f_j^{(t)}} = Cste \end{array} \right. \quad (4.7)$$

La deuxième équation du système (4.7) peut être développée comme suit :  $\forall i \in \{1 \dots n\}$

$$\left\{ \begin{array}{l} \frac{pw_1^{(t)}}{f_1^{(t)}} = \frac{pw_i^{(t)}}{f_i^{(t)}} \Leftrightarrow pw_1^{(t)} d_{1j}^{(t)} = f_1^{(t)} d_{1j}^{(t)} \frac{pw_i^{(t)}}{f_i^{(t)}} \\ \vdots \\ \frac{pw_n^{(t)}}{f_n^{(t)}} = \frac{pw_i^{(t)}}{f_i^{(t)}} \Leftrightarrow pw_n^{(t)} d_{nj}^{(t)} = f_n^{(t)} d_{nj}^{(t)} \frac{pw_i^{(t)}}{f_i^{(t)}} \end{array} \right. \quad (4.8)$$

En additionnant le premier opérande de chaque équation du système (4.8), on obtient l'unique équation suivante :  $\forall i \in \{1 \dots n\}$

$$\sum_{k=1}^n d_{kj}^{(t)} \cdot pw_k^{(t)} = \frac{pw_i^{(t)}}{f_i^{(t)}} \cdot \sum_{k=1}^n f_k^{(t)} d_{kj}^{(t)} = \lambda \quad (4.9)$$

A partir de l'équation (4.9), on peut ainsi calculer les poids des termes du profil *temporaire* qui satisfont l'équation (4.5). Le poids  $pw_i^{(t)}$  d'un terme  $t_i$  dans le profil *temporaire* est donné comme suit :  $\forall i \in \{1 \dots n\}$

$$pw_i^{(t)} = \frac{\lambda f_i^{(t)}}{\sum_{k=1}^n f_k^{(t)} d_{kj}^{(t)}} \quad (4.10)$$

Concernant le poids idéal d'un terme, il est clair que nous n'avons aucun moyen théorique permettant de donner une mesure de ce poids. Notre but est de trouver une fonction  $f$  qui permet de l'estimer. Ainsi, sur la base de ce qui se fait dans le domaine de la recherche d'informations, on peut considérer que la fonction  $f$  dépend de plusieurs paramètres comme : la fréquence d'apparition du terme dans le document, le nombre de documents pertinents et non pertinents contenant ce terme, le nombre total de documents sélectionnés, etc. Nous avons expérimenté plusieurs fonctions ([Boughanem et al., 2004a]) et avons opté pour une fonction dérivée de la formule de Robertson et Sparck-Jones [Robertson and Sparck-Jones, 1976], donnée comme suit :

$$f(d_{ij}^{(t)}, r_i^{(t)}, s_i^{(t)}) = f_i^{(t)} = d_{ij}^{(t)} * \log\left(1 + \frac{r_i^{(t)} \cdot (S^{(t)} - s_i^{(t)})}{(s_i^{(t)} + 1)(R^{(t)} - r_i^{(t)} + 1)}\right) \quad (4.11)$$

avec :

$R^{(t)}$  : nombre de documents pertinents sélectionnés par le système à l'instant  $t$  ;

$S^{(t)}$  : nombre de documents non pertinents sélectionnés par le système ;

$r_i^{(t)}$  : nombre de documents pertinents sélectionnés et contenant le terme  $t_i$  ;

$s_i^{(t)}$  : nombre de documents non pertinents sélectionnés et contenant le terme  $t_i$  ;

Il faut noter que  $P_x^{(t)}$  est la solution pour sélectionner le document  $D_j^{(t)}$ . Cependant, il faut maintenant intégrer cette solution dans le profil global. La fonction  $h$  que nous utilisons est la formule de distribution de gradient donnée comme suit :

$$\begin{aligned}
\forall t_i \in P_x^{(t)} \quad h : R \times R &\longrightarrow R \\
(w_i^{(t)}, pw_i^{(t)}) &\longrightarrow w_i^{(t+1)} = h(w_i^{(t)}, pw_i^{(t)}) \\
&= w_i^{(t)} + 0.1 * \log(1 + pw_i^{(t)})
\end{aligned} \tag{4.12}$$

La fonction logarithmique utilisée dans la fonction (4.12) permet de réduire l'effet des grandes valeurs des poids  $pw_i^{(t)}$ . Autrement dit, cela permet d'éviter d'orienter le profil global de l'utilisateur vers quelques termes de poids importants.

Des expérimentations ont été effectuées pour tester l'apport de la méthode d'apprentissage par renforcement [Boughanem et al., 2004b]. Les résultats obtenus montrent qu'elle réalise effectivement une bonne amélioration du profil, tout au long du processus de filtrage. Cependant, l'amélioration des poids des termes dans le profil global de l'utilisateur, entraîne automatiquement une évolution croissante des scores des documents pertinents sélectionnés. La figure 4.1 illustre l'évolution des scores des documents pertinents dans le cas du profil 101 de la tâche TREC-2002.

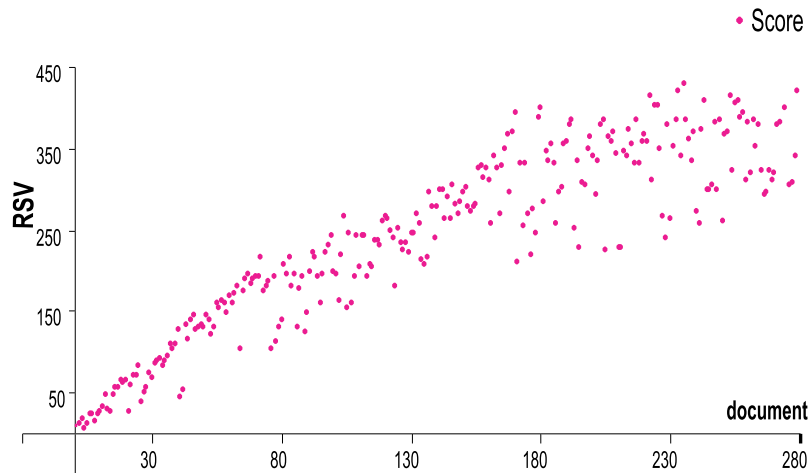


Figure 4.1 – Evolution des scores des documents pertinents (profil 101 de TREC-2002)

Cette façon d'apprendre le profil de l'utilisateur, proposée dans le modèle initial [Tmar, 2002], présente un problème concernant la valeur du score de renforcement  $\lambda$ . En fait, compte tenu de l'accroissement continu des scores, il est difficile de contrôler la valeur du score de renforcement  $\lambda$ . Autrement dit, la valeur de  $\lambda$  doit évoluer avec l'évolution des

scores des documents pertinents et elle tend vers l'infini. A cet effet, pour mieux contrôler la valeur de  $\lambda$ , nous proposons de normaliser la fonction de calcul du score d'un document pertinent par rapport au profil *temporaire* (équation, (4.5)) [Tebri et al., 2005]. Ainsi, le choix de la valeur de  $\lambda$  est limité aux valeurs de l'intervalle  $[0, 1]$ .

On pourrait penser à juste titre qu'il suffit de normaliser la fonction de similarité entre le profil global de l'utilisateur et un document (équation 4.3) pour pallier cet inconvénient. En fait, ceci pose un second problème. Nous rappelons que l'objectif de notre méthode d'apprentissage consiste à renforcer le score de chaque document pertinent filtré. Or, la normalisation de la fonction de similarité peut entraîner la non croissance des scores. C'est à dire, si  $\lambda_1$  est le score normalisé à l'instant  $t$ , entre le profil global  $P^{(t)}$  et le document  $D_j^{(t)}$ , et si nous recalculons le score du même document par rapport au profil global amélioré  $P^{(t+1)}$ , on n'est pas certain d'améliorer le score  $\lambda_1$ . Ceci vient du fait qu'une fonction de similarité normalisée n'est pas croissante (au sens où nous l'entendons). Contrairement à la fonction de similarité basée sur le produit scalaire, qui est par définition une fonction croissante.

Des expérimentations ont été effectuées pour comparer les résultats obtenus par les deux types de fonction de similarité, à savoir des fonctions normalisées en l'occurrence, Dice et Jaccard, dans le cas de notre expérimentation, et puis la mesure de produit scalaire. Les résultats sont loin d'être satisfaisants lorsque la fonction de similarité est normalisée (voir la section 5.4.1).

Dans ce qui suit, nous décrivons la méthode d'apprentissage par renforcement avec normalisation de la fonction de calcul du score de renforcement pour un profil *temporaire*.

#### 4.4.2 Principe de renforcement avec normalisation du score

Plusieurs mesures de similarité permettant de normaliser la fonction de score ont été proposées dans la littérature, à savoir les mesures de *Dice*, *Jaccard* et *Cosinus* (cf. 2.5.2). Dans notre cas, nous utilisons la mesure de *Dice* pour trouver le profil *temporaire*  $P_x^{(t)}$ . Une étude comparative empirique de ces différentes mesures a été réalisée dans le chapitre suivant (voir la section 5.4.1).

Ainsi, en remplaçant l'équation (4.5) par la mesure de Dice, le système d'équations (4.7) est alors réécrit comme suit :  $\forall i \in \{1 \dots n\}$

$$\left\{ \begin{array}{l} \frac{2 \cdot \sum_{k=1}^n d_{kj}^{(t)} \cdot pw_k^{(t)}}{\sum_{k=1}^n (d_{kj}^{(t)})^2 + \sum_{k=1}^n (pw_k^{(t)})^2} = \lambda \\ \forall (t_i, t_j) \in D_j^{(t)} \times D_j^{(t)}, \frac{pw_i^{(t)}}{f_i^{(t)}} = \frac{pw_j^{(t)}}{f_j^{(t)}} \end{array} \right. \quad (4.13)$$

A partir du système (4.8), en additionnant au carré les premiers opérandes des équations de la première partie du système, et en additionnant simplement les premiers opérandes de la deuxième partie du système, on obtient les deux équations suivantes :  $\forall i \in \{1 \dots n\}$

$$\sum_{k=1}^n (pw_k^{(t)})^2 = \left( \frac{pw_i^{(t)}}{f_i^{(t)}} \right)^2 \cdot \sum_{k=1}^n (f_k^{(t)})^2 \quad (4.14)$$

$$\sum_{k=1}^n d_{kj}^{(t)} \cdot pw_k^{(t)} = \frac{pw_i^{(t)}}{f_i^{(t)}} \cdot \sum_{k=1}^n f_k^{(t)} d_{kj}^{(t)} \quad (4.15)$$

En remplaçant les équations (4.14) et (4.15) dans la première équation du système (4.13), nous obtenons le polynôme de seconde degré suivant :

$$\frac{1}{2}a \cdot \left( \frac{pw_i^{(t)}}{f_i^{(t)}} \right)^2 - b \cdot \left( \frac{pw_i^{(t)}}{f_i^{(t)}} \right) + \frac{1}{2}c = 0 \quad (4.16)$$

avec :

$$\begin{aligned} a &= \lambda \cdot \sum_{k=1}^n (f_k^{(t)})^2 \\ b &= \sum_{k=1}^n f_k^{(t)} d_{kj}^{(t)} \\ c &= \lambda \cdot \sum_{k=1}^n (d_{kj}^{(t)})^2 \end{aligned}$$

Le discriminant de l'équation (4.16) est  $\Delta = b^2 - ac$ . Pour chaque terme  $t_i$  du profil temporaire  $P_x^{(t)}$ , le poids  $pw_i^{(t)}$  solution du système (4.13) est donné par :  $\forall i \in \{1 \dots n\}$

$$pw_i^{(t)} = \begin{cases} \frac{f_i^{(t)} \cdot (b \pm \sqrt{\Delta})}{a}, & \text{si } \Delta \geq 0 \\ \infty, & \text{si } \Delta < 0 \end{cases} \quad (4.17)$$

Des expérimentations ont montré que  $(b + \sqrt{\Delta})$  obtient des résultats meilleurs par rapport à  $(b - \sqrt{\Delta})$ .

Les poids  $pw_i^{(t)}$  du profil *temporaire*  $P_x^{(t)}$  sont ensuite introduits dans le profil global de l'utilisateur via la formule de distribution de gradient de l'équation (4.12).

Récapitulons le principe de la méthode de renforcement : à chaque arrivée d'un document  $D_j^{(t)}$  à l'instant  $t$ , on calcule son score de similarité  $\lambda'$  avec le profil global de l'utilisateur  $P^{(t)}$ . Si le score  $\lambda'$  est supérieur au seuil  $\theta^{(t)}$ , le processus de renforcement est déclenché. Le principe de renforcement consiste à trouver un profil *temporaire*  $P_x^{(t)}$  permettant de re-sélectionner le même document  $D_j^{(t)}$  avec un score  $\lambda$ , le plus élevé possible. Les profils *temporaire* et global sont ensuite combinés, par une formule de distribution de gradient, pour construire le nouveau profil global de l'utilisateur  $P^{(t+1)}$ .

## 4.5 Adaptation de la fonction de seuillage

Notre technique de seuillage rentre dans la classe des modèles de seuillage basés sur la distribution des scores des documents pertinents et non pertinents, présentés dans le second chapitre. L'idée de base de notre technique de seuillage consiste, tout d'abord, à représenter la fonction d'utilité en se basant sur les distributions des scores des documents pertinents et non pertinents, puis à déduire le score (ou le seuil) qui permet d'optimiser (de maximiser) cette fonction.

Cette technique de seuillage, présentée à l'origine dans le modèle initial [Tmar, 2002], consiste à dessiner (ou linéariser) la densité de probabilité d'un échantillon de scores des documents pertinents et (resp. non pertinents). De manière générale, la méthode s'effectue en deux principales étapes :

1. la première étape permet de construire des distributions de probabilité des scores des documents pertinents et non pertinents,
2. la seconde étape, consiste à réécrire la fonction d'utilité en se basant sur les distributions de probabilité des documents, puis calculer le seuil optimal permettant de maximiser cette fonction.

Notre contribution dans cette technique de seuillage porte sur plusieurs aspects et intervient pratiquement dans les différents niveaux de la technique d'adaptation de la fonction de seuillage. Principalement, nous avons proposé une autre méthode d'estimation des intervalles, de linéarisation des probabilités discrètes ainsi qu'une nouvelle modélisation de la fonction d'optimisation du seuil. Nous décrivons dans ce qui suit les différentes étapes de la technique d'adaptation de la fonction de seuillage ainsi que nos apports par rapport au modèle initial.

### 4.5.1 Construction de la distribution des scores

La méthode d'adaptation de la fonction de seuillage que nous utilisons se base sur la modélisation de la distribution de probabilité des scores de documents. L'idée de base est d'estimer les probabilités discrètes des scores des documents, puis de dessiner la distribution des scores en utilisant une régression linéaire. Dans notre cas, la régression linéaire nous permet de transformer une distribution de probabilité discrète en une densité de probabilité continue. La démarche globale de cette approche peut être résumée comme suit :

1. Considérons deux échantillons de documents filtrés à un instant  $t$  à l'aide du processus de filtrage : un échantillon de documents pertinents et un autre de documents non pertinents ;
2. Pour chaque échantillon :
  - (a) pour chaque document, calculer la probabilité que son score appartient à un intervalle donné ; on obtient un couple (score, probabilité),
  - (b) relier, par régression linéaire, les probabilités associées aux différents intervalles,
  - (c) transformer la courbe obtenue par linéarisation en une distribution de probabilités continue.

Cette technique de linéarisation est appliquée à un échantillon de documents pertinents et à un échantillon de documents non pertinents. Comme c'est les mêmes étapes qui sont appliquées à chacun des deux échantillons, nous nous limitons dans notre démarche à expliquer le processus de linéarisation d'un échantillon de documents pertinents. Les différentes étapes de notre démarche sont détaillées dans les sections ci-dessous.

#### 4.5.1.1 Conversion des scores en probabilités

Les scores d'un échantillon de documents peuvent être convertis en probabilités comme suit : considérons un échantillon de documents déjà filtrés, la probabilité qu'un document  $D_i^{(t)}$ , tiré aléatoirement de cet échantillon, ait un score donné ( $score_x$ ) est par définition égale au nombre de documents ayant ce score divisé par le nombre de documents total dans l'échantillon. Formellement, cette probabilité est donnée comme suit :

Soient,  $E_r^{(t)} = \{D_1^{(t)}, \dots, D_n^{(t)}\}$ , un échantillon de documents pertinents pour un profil  $P^{(t)}$ , à l'instant  $t$ , et une variable aléatoire  $X$  représentant les scores des documents de l'échantillon  $E_r^{(t)}$ .  $\forall score_x \in [score_{min}, score_{max}]$  :

$$p(X = score_x) = \frac{|\{D_i^{(t)} | rsv(D_i^{(t)}, P^{(t)}) = score_x\}|}{|\{E_r^{(t)}\}|} \quad (4.18)$$

avec,

$|\cdot|$  : est le cardinal d'un ensemble fini ;

$score_{min}$  : le score minimum des documents dans  $E_r^{(t)}$ ,

$score_{max}$  : le score maximum des documents dans  $E_r^{(t)}$ ,

Comme les valeurs des scores sont très variées, elles ont tendance à être équiprobables, c'est à dire  $|\{D_i^{(t)} | rsv(D_i^{(t)}, P^{(t)}) = score_x = score_i\}| = 1$  ou 0. En effet, dans un échantillon donné, il est difficile de trouver deux ou plusieurs documents ayant exactement le même score. Une solution possible à ce problème est d'utiliser la méthode d'estimation de probabilités par intervalle.

#### 4.5.1.2 Estimation des probabilités par intervalle

L'estimation des probabilités des scores des documents par intervalle consiste à calculer pour chaque document la probabilité que son score appartienne à un intervalle donné. Le découpage de la distribution des scores en plusieurs intervalles, peut se faire de différentes manières :

- les intervalles sont d'amplitudes égales établies à partir de l'étendue. L'étendue d'une distribution est égale à la différence entre la plus grande et la plus petite valeur des scores de la distribution ;
- Les intervalles sont obtenus à partir de l'écart-type et de la moyenne des scores de la distribution.

### i. Intervalles d'amplitudes égales établies à partir de l'étendue :

Pour calculer la probabilité qu'un score d'un document appartienne à un intervalle donné, on doit tout d'abord définir ces intervalles. Ces intervalles doivent être assez réduits pour que les scores des documents appartenant au même intervalle soient réellement presque égaux. Pour établir cette partition, nous proposons d'utiliser une décomposition en intervalles égaux basée sur l'étendue des scores des documents de l'échantillon. Le principe est le suivant :

On fixe *à priori* le nombre d'intervalles, soit  $m$ , que l'on souhaite obtenir, puis on divise la distribution en  $m$  intervalles dont l'amplitude ou rayon est le même pour tous. Soient,  $I_1, \dots, I_m$  les  $m$  intervalles à définir, où :

$$\begin{aligned}
 I_i &= [score_{i-1}, score_i] \\
 score_0 &= score_{min} \\
 score_i &= score_{i-1} + \varepsilon \\
 \varepsilon &= \frac{score_{max} - score_{min}}{m} \\
 score_{min} &= \min_{D_j^{(t)} \in E_r^{(t)}} rsv(D_j^{(t)}, P^{(t)}) \\
 score_{max} &= \max_{D_i^{(t)} \in E_r^{(t)}} rsv(D_i^{(t)}, P^{(t)})
 \end{aligned} \tag{4.19}$$

Le nombre d'intervalles est proportionnel à la taille de l'échantillon, car plus la taille de l'échantillon augmente, plus le domaine de définition des scores des documents s'élargit. On choisit donc  $m$  comme la moitié de la taille de l'échantillon :  $m = |E_r^{(t)}|/2$ .

Ainsi, la probabilité  $p_i$  d'appartenance d'un score donné à un intervalle est définie par :

$$p(score_{i-1} \leq X < score_i) = \frac{|\{D_j^{(t)} | rsv(D_j^{(t)}, P^{(t)}) \in I_i\}|}{|\{E_r^{(t)}\}|} \tag{4.20}$$

La répartition des scores sur des intervalles d'amplitudes égales, telle qu'elle est utilisée dans le modèle initial [Tmar, 2002], peut conduire le processus de linéarisation des probabilités à manipuler des probabilités de valeurs inattendues (négatives). La valeur centrale (la moyenne des scores) peut se trouver dans n'importe quel intervalle et n'intervient pas dans la détermination de ceux-ci. On peut donc avoir des surprises désagréables sous la forme d'intervalles vides successifs, comme le cas de la figure 4.2. En effet, la figure 4.2 illustre, dans le cas du profil 101 de TREC-2002, la répartition des scores des documents pertinents en plusieurs intervalles, et les probabilités

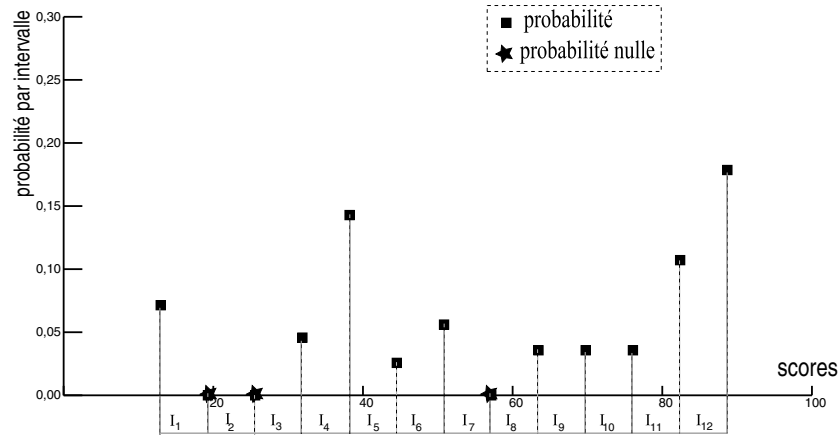


Figure 4.2 – Distribution de probabilités par intervalle (profil 101 de TREC-2002)

associées pour chaque intervalle. Comme on peut le constater, l'existence des probabilités nulles n'est que le fait qu'aucun document n'a un score appartenant à ces intervalles.

Les deux principaux inconvénients rencontrés par l'utilisation d'intervalles vides lors de la linéarisation de la distribution de probabilités des scores sont les suivants :

1. *probabilités dispersées* : la concentration des scores dans quelques intervalles entraîne une grande variation entre les probabilités calculées pour chaque intervalle. Dans la figure 4.2, on constate qu'il n'y a aucun document dans les intervalles  $I_2$ ,  $I_3$  et  $I_8$ , car les probabilités associées à ces intervalles sont nulles. Par contre, les intervalles  $I_5$ ,  $I_{13}$  et  $I_{14}$  possèdent plus de scores que le reste des intervalles, en conséquence, leurs probabilités (resp. 0.14, 0.18 et 0.29) sont très importantes comparées aux autres ;

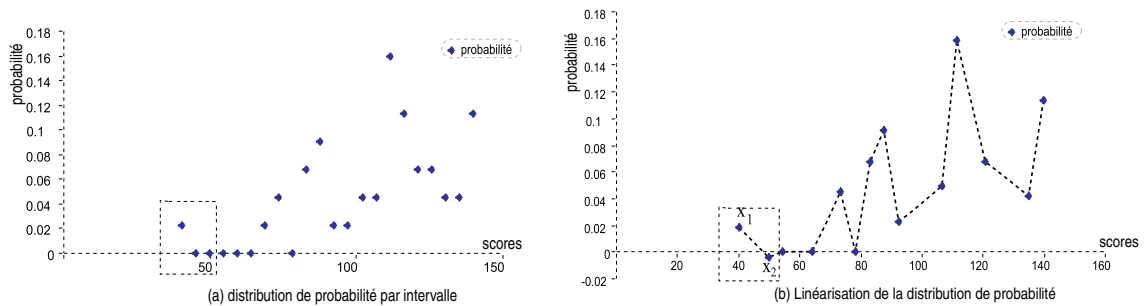


Figure 4.3 – Problème de probabilités négatives (profil 102 de TREC-2002)

2. *probabilités négatives lors de la linéarisation* : ce cas non souhaité parvient lors de la linéarisation des probabilités des scores par la méthode de régression linéaire (détaillée dans la section 4.5.1.3). La figure 4.3 illustre un exemple où il est possible d’avoir des probabilités négatives lors de la linéarisation. La figure 4.3.a présente une distribution de probabilités par intervalle d’un échantillon de documents pertinents, cas du profil 102 de TREC-2002. La linéarisation de cette distribution est présentée par la figure 4.3.b. On remarque que les trois premiers points dans la figure 4.3.a sont remplacés, dans la figure 4.3.b, par une droite qui passe le plus près d’eux. Cependant, cette droite est représentée, dans la figure 4.3.b, par les deux points  $x_1 = (40.274, 0.019)$  et  $x_2 = (49.740, -0.003)$ . Le deuxième point a comme ordonnée, représentant une probabilité donnée, une valeur négative. Ce cas est intolérable dans la théorie des probabilités.

Pour contourner ces deux inconvénients, nous proposons deux solutions possibles :

1. La première consiste à continuer d’utiliser la répartition des scores des documents sur des intervalles d’amplitudes égales, sauf que les probabilités négatives générées par la linéarisation seront convertis en probabilités positives comme suit :

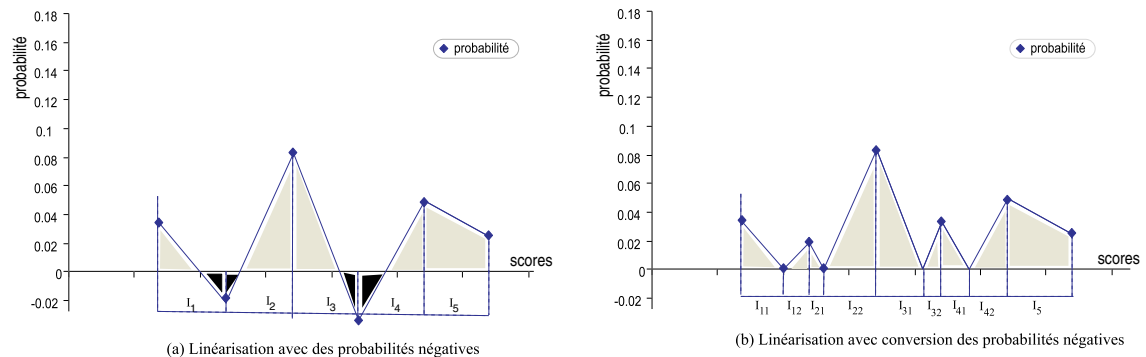


Figure 4.4 – Conversion de probabilités négatives en probabilités positives

- (a) remplacer les probabilités négatives par des zéro. Cette solution est à écarter, car on risque de perdre des informations sur l’échantillon considéré ;
- (b) prendre les valeurs absolues des probabilités négatives. En fait, compte tenu de notre technique d’optimisation du seuil, technique basée sur le calcul de la surface d’une distribution, ces valeurs absolues nous permettent de tenir compte des

surfaces générées par les probabilités négatives. La figure 4.4 illustre la conversion des probabilités négatives (figure 4.4.a) en probabilités positives (figure 4.4.b) et la création de nouveaux intervalles. Les intervalles ainsi créés sont  $I_{11}$ ,  $I_{12}$ ,  $I_{21}$ ,  $I_{22}$ ,  $I_{31}$ ,  $I_{32}$ ,  $I_{41}$ ,  $I_{42}$  et  $I_5$ . On remarque que nous conservons les mêmes surfaces par rapport à la figure initiale.

2. la deuxième solution que nous proposons permet d'effectuer un découpage de l'échantillon des scores des documents en fonction de l'écart-type de la distribution des scores considérés. Autrement dit, l'étendue des différents intervalles est déduite à partir de l'écart-type de la distribution.

## ii. Intervalles d'amplitudes égales établies à partir de l'écart type :

Contrairement à l'étendue, l'écart type est la mesure de dispersion la plus couramment utilisée en statistique pour calculer une tendance centrale. L'écart type mesure donc la dispersion des données autour de la moyenne. Pour notre part, nous proposons de fixer l'amplitude des intervalles à un écart type. Ainsi, les  $m$  intervalles à définir,  $I_1, \dots, I_m$ , sont donnés de la façon suivante :

$$\begin{aligned}
 I_i &= [score_{i-1}, score_i[ \\
 score_0 &= score_{min} \\
 I_m &= [score_{m-1}, score_{max}] \\
 score_i &= score_{i-1} + \sigma \\
 score_{min} &= \min_{D_j^{(t)} \in E_r^{(t)}} rsv(D_j^{(t)}, P^{(t)}) \\
 score_{max} &= \max_{D_j^{(t)} \in E_r^{(t)}} rsv(D_j^{(t)}, P^{(t)})
 \end{aligned} \tag{4.21}$$

où,  $\sigma$  est l'écart type des scores de l'échantillon. Il est donné par l'équation suivante :

$$\left\{ \begin{array}{l}
 \sigma_x = \sqrt{\frac{1}{n} \sum_{i=1}^n (rsv(D_i^{(t)}, P^{(t)}) - \bar{X})^2} \\
 \bar{X} = \frac{1}{n} \sum_{j=1}^n rsv(D_j^{(t)}, P^{(t)}), \text{ avec } D_j^{(t)} \in E_r^{(t)} \text{ et } n = |E_r^{(t)}|
 \end{array} \right.$$

comme,  $(score_i = score_{i-1} + \sigma)$  est une suite arithmétique de raison  $\sigma$ , alors on peut déduire le nombre d'intervalles  $m$  comme suit :

$$\left\{ \begin{array}{l}
 score_{m-1} = score_0 + (m-1)\sigma \\
 score_{max} - score_{m-1} \geq \sigma \\
 \Rightarrow m \simeq E\left(\frac{score_{max} - score_{min}}{\sigma}\right)
 \end{array} \right.$$

avec  $E(Y)$  désigne la partie entière de  $Y$ .

### 4.5.1.3 Linéarisation de la distribution de probabilités des scores

L'objectif de notre méthode de seuillage est donc de ne pas supposer l'existence d'une loi de distribution de probabilités donnée et d'estimer ses différents paramètres, mais plutôt de tenter de dessiner (ou de tracer), pour un échantillon donné, la forme de la distribution de probabilités correspondante.

L'idée de base de notre méthode de linéarisation de la distribution des scores (LDS) consiste à diviser le domaine de définition (l'ensemble des probabilités associées aux intervalles) en plusieurs classes, tels que la courbe reliant les points représentant les probabilités discrètes dans chaque classe puisse être assimilée à une courbe linéaire.

Le processus de détection des classes linéaires consiste à chercher le maximum de points adjacents tels que la courbe reliant ces points est linéaire. Pour mesurer la linéarité d'un ensemble de points, nous utilisons la méthode des moindres carrés [Saporta, 1990]. Nous allons tout d'abord définir la notion de classe linéaire :

**Définition 3** Une classe linéaire  $C^c$  est définie par deux scores  $[score_x, score_y]$ , avec  $score_x < score_y$ , et tous les points  $(s_i, p_i)$  de la classe forment une ligne droite, où  $s_i$  est le score du  $i$ ème point dans la classe et  $p_i$  la probabilité de  $s_i$  donnée par l'équation (4.20).

Le processus de création de classes linéaires se base sur la méthode des moindres carrée, utilisée pour la régression linéaire, et l'évaluation de l'écart quadratique entre les points et la courbe linéaire estimée. Un point est ajouté à une classe si seulement si l'erreur mesurée par l'écart quadratique entre ce point et la courbe linéaire (reliant les autres points de la classe) est inférieure à un seuil donné. Dans le cas contraire, ce point sera repris pour créer une nouvelle classe avec d'autres points. Le mécanisme ou l'algorithme de construction de ces différentes classes linéaires est donnée comme suit :

1.  $c = 1$  ( $c$  est un indice d'une classe linéaire  $C$ )
2.  $P = \emptyset$ ,
3.  $seuil\_erreur$ , (le seuil de l'erreur quadratique)
4.  $m$  : le nombre de points (score, probabilité)  $= (s_i, p_i)$  (on considère que les points sont ordonnés par ordre croissant de leurs scores)
5. pour  $i \in \{0 \dots m - 1\}$ ,

- (a)  $P \leftarrow P \cup \{i\}$ ,
- (b) déterminer l'équation de la droite  $D_c : y(x) = a + bx$  par la régression linéaire sur tous les points de coordonnées  $(s_j, p_j) \forall j \in P$ ,
- (c) calculer l'erreur représentée par l'écart quadratique (EQ) des points  $(s_j, p_j) \forall j \in P$  et la droite  $D_c$  :

$$EQ = \sum_{j \in P} d^2((s_j, p_j), D_c)$$

$$d^2((s_j, p_j), D_c) = \left( \frac{a + b \cdot s_j - p_j}{\sqrt{a^2 + 1}} \right)^2$$

- (d) si  $EQ > \text{seuil\_erreur}$  alors

- i. une classe de points est formée :  $C^c = (s_{inf}^c, s_{sup}^c, a_c, b_c)$  où,  $s_{inf}^c = \min(s_j)$ ,  $s_{sup}^c = \max(s_j) \forall j \in P$ ,  $a_c$  et  $b_c$  sont les coefficients de l'équation de la droite  $y = a_c + b_c x$  par la régression linéaire sur tous les points de coordonnées  $(s_j, p_j)$  où  $j \in P \setminus \{i\}$ ,
- ii.  $P = \emptyset$  (réinitialiser  $P$ )
- iii.  $P \leftarrow \{i\}$ ,
- iv.  $C \leftarrow C \cup \{C^c\}$ ,
- v.  $c \leftarrow c + 1$ .

(e) fin si

(f) fin pour

Ce processus (algorithme) permet seulement de représenter la distribution de probabilités des scores des documents sous forme de plusieurs segments de droite. Cependant, il est nécessaire de transformer cette représentation linéaire pour former une distribution de probabilités continue. Pour cela il faut :

1. tout d'abord, relier les deux extrémités de chaque paire de classes adjacentes. Cette liaison s'effectue comme suit : pour deux classes linéaires adjacentes  $C^c$  et  $C^{(c+1)}$ , relier  $s_{sup}^c$  et  $s_{inf}^{(c+1)}$  par une droite  $y = \alpha_c + \beta_c x$ . Cette droite doit passer par les points  $(s_{sup}^c, a_c + b_c \cdot s_{sup}^c)$  et  $(s_{inf}^{(c+1)}, a_{(c+1)} + b_{(c+1)} \cdot s_{inf}^{(c+1)})$ , avec :

$$\alpha_c = \frac{a_c + b_c \cdot s_{sup}^c - a_{(c+1)} + b_{(c+1)} \cdot s_{inf}^{(c+1)}}{s_{sup}^c - s_{inf}^{(c+1)}}$$

$$\beta_c = a_c + b_c \cdot s_{sup}^c - \frac{a_c + b_c \cdot s_{sup}^c - a_{(c+1)} + b_{(c+1)} \cdot s_{inf}^{(c+1)}}{s_{sup}^c - s_{inf}^{(c+1)}} \cdot s_{sup}^c \quad (4.22)$$

Ainsi, un sous ensemble  $C_{int}$  de classes intermédiaires est créé à partir des classes de l'ensemble  $C$ . Ces classes intermédiaires sont définies comme suit :

- (a)  $C_{int} = \emptyset$ ,
- (b) pour  $c \in \{0 \dots |C| - 1\}$ 
  - i.  $C_{int}^{(c,c+1)} = (s_{sup}^c, s_{inf}^{(c+1)}, \alpha_c, \beta_c)$ , avec  $\alpha_c$  et  $\beta_c$  sont donnés par l'équation (4.22) et en considérant les classes  $C^c$  et  $C^{(c+1)}$  de l'ensemble  $C$ ,
  - ii.  $C_{int} \leftarrow C_{int} \cup \{C^{c,c+1}\}$

Soit  $\Phi$  la fonction définie par :

$$\Phi : [score_{min}, score_{max}] \rightarrow R$$

$$x \mapsto \begin{cases} a_c + b_c x & \text{si } \exists c, x \in C^c / s_{inf}^c \leq x \leq s_{sup}^c \\ \alpha_c + \beta_c x & \text{si } x \in C^{(c,c+1)} \end{cases}$$

avec,  $s_{inf}^c = \max_{i, s_{inf}^i \leq x} (s_{inf}^i)$

2. puis normaliser les coefficients  $a_c$ ,  $b_c$ ,  $\alpha_c$  et  $\beta_c$  pour que :

$$\int_{score_{min}}^{score_{max}} \Phi(x) dx = 1 \quad (4.23)$$

Puisque  $\int_{score_{min}}^{score_{max}} \Phi(x) dx$  représente la surface de l'aire formée par la représentation graphique de  $\Phi$  et l'axe des abscisses, il suffit alors de diviser les coefficients  $a_c$ ,  $b_c$ ,  $\alpha_c$  et  $\beta_c$  par cette valeur. L'aire est calculée comme la somme des aires de chaque surface d'une classe linéaire.

Considérons  $C_{total}$ , un ensemble défini par la fusion des deux ensembles  $C$  et  $C_{int}$  ( $C_{total} = C \cup C_{int}$ ), et  $c_c$  le nombre de classes de l'ensemble  $C_{total}$ . L'aire de la surface (figure 4.5) formée par l'ensemble des classes de  $C_{total}$  est calculée comme suit :

$$\int_{score_{min}}^{score_{max}} \Phi(x) dx = \sum_{i=1}^{i=c_c} (\alpha_{i1} + \alpha_{i2}) \quad (4.24)$$

$$= \frac{1}{2} \sum_{i=1}^{i=c_c} (s_{sup}^i - s_{inf}^i) (2a_i + b_i (s_{sup} + s_{inf}))$$

Cette technique de linéarisation est appliquée respectivement aux deux échantillons de documents (pertinents et non pertinents). La figure 4.5 illustre une linéarisation effectuée

sur l'ensemble des documents pertinents et non pertinents de la base *Reuters* pour le profil 101 de la tâche TREC-2002. Elle montre que la linéarisation des probabilités de scores tend à avoir une allure exponentielle pour les documents non pertinents et une allure gaussienne pour les documents pertinents.

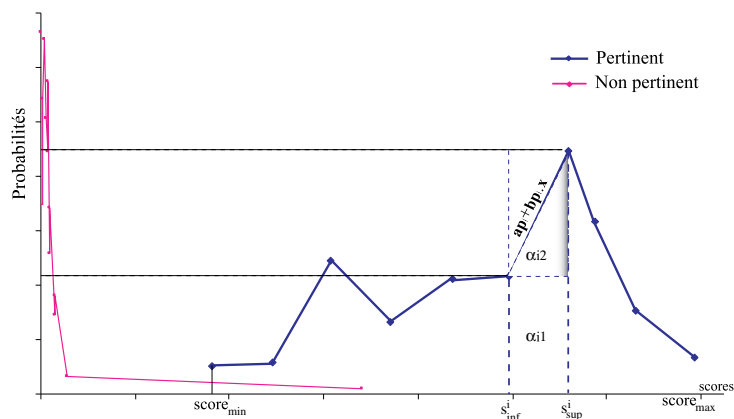


Figure 4.5 – Densités de probabilités des scores des documents pertinents et non pertinents

Comme nous pouvons le constater, on peut se poser la question suivante : est-il possible de se passer de la méthode de linéarisation des probabilités des scores des documents ? Autrement dit, est-il possible de relier directement les probabilités associées aux différents intervalles sans appliquer la technique de linéarisation. Pratiquement, cela est possible dans le cas d'un échantillon de scores de documents pertinents ou non pertinents de taille réduite. Plus la taille de l'échantillon est petite plus les points représentant les différentes probabilités ont tendance à se disperser dans l'espace, donc les différents points ne forment pas de classes linéaires. Le tableau 4.1 présente les valeurs d'utilité des profils 101, 102 et 103 de TREC-2002, avec linéarisation et sans linéarisation des scores des documents.

On remarque que, plus la taille de l'échantillon augmente, plus il est nécessaire de recourir à la linéarisation. Le profil 101, avec linéarisation des probabilités, présente une amélioration de la valeur d'utilité d'ordre de 4.55% par rapport au cas où la linéarisation n'est pas appliquée. Contrairement aux autres profils, on peut constater qu'il n'y a pas une importante amélioration dans le cas de la linéarisation ou non des probabilités des scores des documents. Dans le chapitre expérimentation, nous présentons les résultats obtenus, pour les 50 profils de TREC-2002, dans le cas de la linéarisation des différents points et dans le cas de la liaison directe de ces points.

profil	Linéarisation des scores		sans linéarisation des scores	
	$ E_r^{(t)} $	utilité	$ E_r^{(t)} $	utilité
101	288	0.9161	288	0.8763
102	144	0.7352	150	0.7124
103	27	0.4674	27	0.4756

Tableau 4.1 – Résultats de la linéarisation des scores

## 4.5.2 Optimisation de la fonction de seuillage

L'adaptation de la fonction de seuillage se base sur l'optimisation de la fonction d'utilité  $U_{(\lambda_1, \lambda_2, \lambda_3, \lambda_4)}$ . Ceci revient à trouver la valeur du seuil  $\theta^*$  tel que :

$$\theta^* = \operatorname{argmax}_{\theta} [ \lambda_1 R_+(\theta) + \lambda_2 S_+(\theta) + \lambda_3 R_-(\theta) + \lambda_4 S_-(\theta) ] \quad (4.25)$$

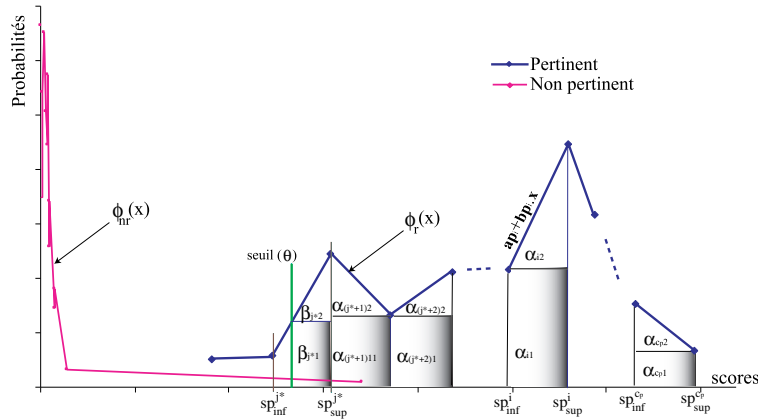


Figure 4.6 – Estimation de  $\int_{\theta}^{+\infty} P_r(x)dx$  en utilisant la surface

Avec  $R_+(\theta)$ ,  $S_+(\theta)$ ,  $R_-(\theta)$  et  $S_-(\theta)$  sont obtenus à partir des équations (3.32), (3.33) et (3.34).

Le facteur  $\int_{\theta}^{+\infty} P_r(x)dx$  de l'équation de  $R_+(\theta)$  (resp.  $\int_{\theta}^{+\infty} P_{nr}(x)dx$  de  $S_+(\theta)$ ) est défini par l'aire de la surface formée par la courbe correspondant à la densité de probabilité des scores des documents pertinents (resp. non pertinents) à partir du seuil  $\theta$ . Ainsi, en se référant à la figure 4.6, on peut écrire :

$$\begin{aligned}\int_{\theta}^{+\infty} P_r(x)dx &= \int_{\theta}^{+\infty} \Phi_r(x)dx \\ &= \beta_{j^*1} + \beta_{j^*2} + \alpha_{(j^*+1)1} + \alpha_{(j^*+1)2} + \dots + \alpha_{i1} + \alpha_{i2} + \dots + \alpha_{c_p1} + \alpha_{c_p2}\end{aligned}$$

Soient  $c_p$  le nombre d'intervalles ayant des équations linéaires différentes sur les scores des documents pertinents et  $Cp^i(sp_{inf}^i, sp_{sup}^i, ap_i, bp_i)$  (resp.  $Cp^{j^*}(\theta, sp_{sup}^{j^*}, ap_{j^*}, bp_{j^*})$ ) une classe  $i$  (resp.  $j$ ), où  $i \in \{(j^* + 1)..c_p\}$  and  $j^* \geq 1$ , alors :

$$\begin{aligned}\beta_{j^*1} &= (sp_{sup}^{j^*} - \theta)(ap_{j^*} + bp_{j^*}.\theta) \\ \beta_{j^*2} &= \frac{1}{2}(sp_{sup}^{j^*} - \theta)((ap_{j^*} + bp_{j^*}.sp_{sup}^{j^*}) - (ap_{j^*} + bp_{j^*}.\theta)) \\ \alpha_{i1} &= (sp_{sup}^i - sp_{inf}^i)(ap_i + bp_i.sp_{inf}^i) \\ \alpha_{i2} &= \frac{1}{2}(sp_{sup}^i - sp_{inf}^i)((ap_i + bp_i.sp_{sup}^i) - (ap_i + bp_i.sp_{inf}^i))\end{aligned}$$

$\int_{\theta}^{+\infty} P_r(x)dx$  peut être réécrit comme suit :

$$\begin{aligned}\int_{\theta}^{+\infty} P_r(x)dx &= \frac{1}{2}(\sum_{i, sp_{inf}^i > \theta} (sp_{sup}^i - sp_{inf}^i)(2ap_i + bp_i(sp_{sup}^i + sp_{inf}^i)) \\ &\quad + (sp_{sup}^{j^*} - \theta)(2ap_{j^*} + bp_{j^*}(sp_{sup}^{j^*} + \theta)))\end{aligned} \quad (4.26)$$

où  $j^*$  est tel que  $\theta \in [sp_{inf}^{j^*}, sp_{sup}^{j^*}]$ .

De la même manière, si on considère  $c_n$  le nombre d'intervalles ayant des équations linéaires différentes sur les scores des documents non pertinents et  $Cn^i$  ( $i \in \{1..c_n\}$ ) une classe linéaire  $[sn_{inf}^i, sn_{sup}^i]$ ,  $S_+(\theta)$  sera calculé comme suit :

$$\begin{aligned}\int_{\theta}^{+\infty} P_{nr}(x)dx &= \int_{\theta}^{+\infty} \Phi_{nr}(x)dx \\ &= \frac{1}{2}(\sum_{i, sn_{inf}^i > \theta} (sn_{sup}^i - sn_{inf}^i)(2an_i + bn_i(sn_{sup}^i + sn_{inf}^i)) \\ &\quad + (sn_{sup}^{k^*} - \theta)(2an_{k^*} + bn_{k^*}(sn_{sup}^{k^*} + \theta)))\end{aligned} \quad (4.27)$$

où  $k^*$  est tel que  $\theta \in [sn_{inf}^{k^*}, sn_{sup}^{k^*}]$ .

A noter que,  $Cp^{j^*}$  and  $Cn^{k^*}$  sont deux classes superposées. Ainsi, si  $\theta \in Cp^{j^*}$  alors  $\theta \in Cn^{k^*}$

Cependant,  $\int_{-\infty}^{\theta} P_r(x)dx$  et  $\int_{-\infty}^{\theta} P_{nr}(x)dx$  peuvent être déduits à partir des équations (4.26) et (4.27), comme suit :

$$\begin{aligned}\int_{-\infty}^{\theta} P_r(x)dx &= 1 - \int_{\theta}^{+\infty} P_r(x)dx \\ \int_{-\infty}^{\theta} P_{nr}(x)dx &= 1 - \int_{\theta}^{+\infty} P_{nr}(x)dx\end{aligned}$$

Ainsi, l'équation (4.25) peut être remplacée par :

$$\begin{aligned}\theta^* &= \operatorname{argmax}_{\theta} [(\lambda_3 r + \lambda_4(n-r)) + (\lambda_1 - \lambda_3)r \int_{\theta}^{+\infty} P_r(x)dx \\ &\quad + (\lambda_2 - \lambda_4)(n-r) \int_{\theta}^{+\infty} P_{nr}(x)dx]\end{aligned}\quad (4.28)$$

comme  $(\lambda_3 r + \lambda_4(n-r))$  est une constante indépendante de  $\theta$ , alors l'optimisation de  $U_{(\lambda_1, \lambda_2, \lambda_3, \lambda_4)}$  revient à trouver le seuil  $\theta^*$  qui maximise la fonction suivante :

$$\theta^* = \operatorname{argmax}_{\theta} [(\lambda_1 - \lambda_3)\rho \int_{\theta}^{+\infty} P_r(x)dx + (\lambda_2 - \lambda_4) \int_{\theta}^{+\infty} P_{nr}(x)dx] \quad (4.29)$$

avec,  $\rho = \frac{r}{n-r}$

Enfin, en remplaçant les intégrales par leurs formules respectives, l'équation finale à optimiser est donnée comme suit :

$$\begin{aligned}\theta^* &= \operatorname{argmax}_{\theta} [\frac{1}{2}(\lambda_1 - \lambda_3)\rho(\sum_{i, sp_{inf}^i > \theta} (sp_{sup}^i - sp_{inf}^i)(2ap_i + bp_i(sp_{sup}^i + sp_{inf}^i)) + \\ &\quad (sp_{sup}^{j*} - \theta)(2ap_{j*} + bp_{j*}(sp_{sup}^{j*} + \theta))) \\ &\quad + \frac{1}{2}(\lambda_2 - \lambda_4)(\sum_{i, sn_{inf}^i > \theta} (sn_{sup}^i - sn_{inf}^i)(2an_i + bn_i(sn_{sup}^i + sn_{inf}^i)) + \\ &\quad (sn_{sup}^{k*} - \theta)(2an_{k*} + bn_{k*}(sn_{sup}^{k*} + \theta)))]\end{aligned}\quad (4.30)$$

Pour calculer ce seuil  $\theta^*$ , deux méthodes sont proposées : la méthode de détection du seuil dans un intervalle de valeurs (méthode de seuillage par intervalle) et la méthode de détection du seuil optimal en résolvant l'équation (4.29) (méthode de seuillage par dérivation).

#### 4.5.2.1 Méthode de seuillage par intervalle

Cette méthode a été utilisée lors de nos premières expérimentations. Le principe de la méthode consiste à dégager une valeur de seuil optimal dans un intervalle de scores  $I_{opt}$  (voir figure 4.7). Autrement dit, on fait varier  $\theta$  dans un intervalle de scores et pour

chaque valeur de  $\theta$  on calcule l'utilité correspondante (équation (4.30)). Le seuil optimal  $\theta^*$  est donné par la valeur de  $\theta$  qui maximise la fonction d'utilité. Les bornes de l'intervalle correspondent respectivement à la valeur minimale ( $sp_{inf}^0$ ) et maximale ( $sn_{sup}^{c_n}$ ) des scores des documents pertinents et non pertinents. Cependant, si  $sp_{inf}^0$  est supérieure à  $sn_{sup}^{c_n}$  alors le seuil optimal sera représenté par le centre de l'intervalle  $I_{opt} = [sn_{sup}^{c_n}, sp_{inf}^0]$ , sinon l'algorithme suivant est utilisé pour trouver une valeur de seuil optimale :

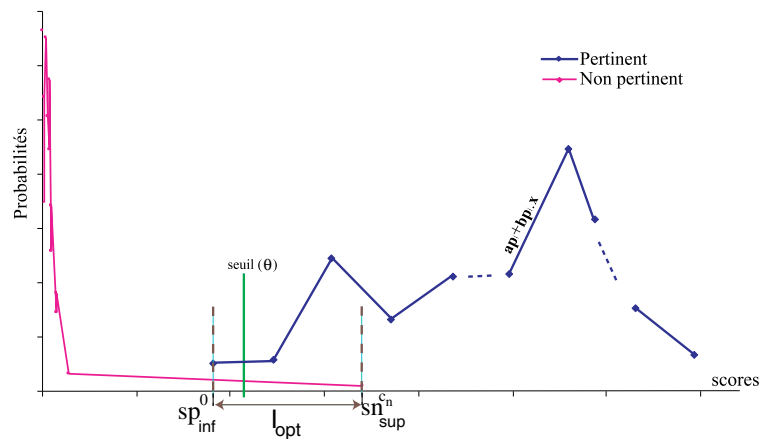


Figure 4.7 – Détection du seuil par intervalle de scores

1.  $\theta^* = \theta = sp_{inf}^0$  //le seuil de départ, qui est la borne inférieure de l'intervalle des scores
2. //calculer l'utilité associée au seuil  $\theta$ , et on suppose qu'elle est optimale, soit  $U^*$   
 $U^* = U(\theta)$
3. tant que ( $\theta \leq sn_{sup}^{c_n}$ ) faire
  - calculer  $U(\theta)$  // modifier la valeur de l'utilité optimale
  - si ( $U(\theta) > U^*$ ) alors
    - $U^* = U(\theta)$
    - $\theta^* = \theta$
  - $\theta = \theta + \Delta\theta$  //incrémenter  $\theta$
4. fin tantque.

L'inconvénient de cette méthode est que les valeurs du seuil sont générées de manière aléatoire dans un intervalle et elles ne permettent pas de trouver la valeur réelle du seuil qui maximise la fonction d'utilité. Pour pallier cet inconvénient, nous proposons une autre

méthode, appelée méthode de seuillage par dérivation, basée sur la résolution de l'équation d'optimisation du seuil.

#### 4.5.2.2 Méthode de seuillage par dérivation

Cette méthode consiste à résoudre l'équation d'optimisation du seuil (4.30). Cette dernière, peut être réécrite comme suit :

$$\begin{aligned} \theta^* = & \operatorname{argmax}_{\theta} \left[ \frac{1}{2}(\lambda_1 - \lambda_3) \rho(\sum_{i, sp_{inf}^i > \theta} (sp_{sup}^i - sp_{inf}^i)(2ap_i + bp_i(sp_{sup}^i + sp_{inf}^i))) + \right. \\ & \left. \frac{1}{2}(\lambda_2 - \lambda_4) (\sum_{i, sn_{inf}^i > \theta} (sn_{sup}^i - sn_{inf}^i)(2an_i + bn_i(sn_{sup}^i + sn_{inf}^i))) \right] \\ & \frac{1}{2}(\lambda_1 - \lambda_3) \rho(sp_{sup}^{j^*} - \theta)(2ap_{j^*} + bp_{j^*}(sp_{sup}^{j^*} + \theta)) + \\ & \left. \frac{1}{2}(\lambda_2 - \lambda_4) (sn_{sup}^{k^*} - \theta)(2an_{k^*} + bn_{k^*}(sn_{sup}^{k^*} + \theta)) \right] \end{aligned} \quad (4.31)$$

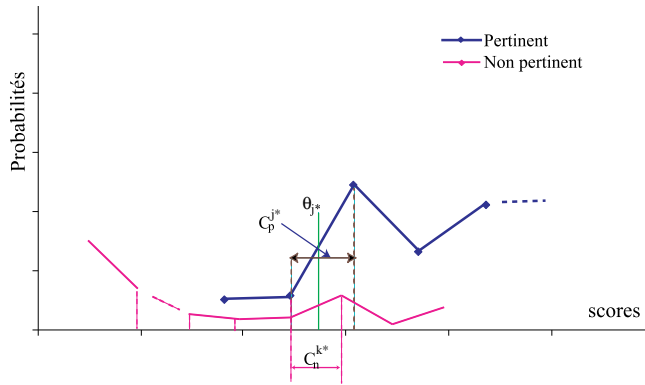


Figure 4.8 – Identification de la classe  $k^*$  via la classe  $j^*$

La calcul du seuil optimal revient à résoudre l'équation (4.31) de la manière suivante : si nous supposons que le seuil optimal se trouve dans une classe  $Cp^{j^*}$ , qui se superpose avec la classe  $Cn^{k^*}$ , alors l'argument  $sp_{inf}^i > \theta$  (resp.  $sn_{inf}^i > \theta$ ) correspond à la première classe qui suit la classe  $Cp^{j^*}$  (resp.  $Cn^{k^*}$ ). Ainsi, pour une classe donnée  $Cp^{j^*}$  (resp.  $Cn^{k^*}$ ), on peut remplacer l'argument  $sp_{inf}^i > \theta$  (resp.  $sn_{inf}^i > \theta$ ) par l'indice de la classe ( $j^* + 1$ ) (resp. ( $k^* + 1$ )). À noter que, à partir de l'indice  $j^*$  d'une classe de documents pertinents, il est possible d'identifier la classe  $k^*$  des documents non pertinents associée. Cette dernière, correspond à l'indice  $k^*$  de la classe qui se superpose avec la classe de l'indice  $j^*$ . En tenant compte de ces notations, l'équation (4.31) peut être réécrite de la façon suivante :

$$\theta^* = \operatorname{argmax}_{\theta} U(\theta, j^*) = \operatorname{argmax}_{\theta} [F(j^*) + G(\theta)] \quad (4.32)$$

avec :

$$F(j^*) = \frac{1}{2}(\lambda_1 - \lambda_3)\rho \sum_{l=(j^*+1)}^{l=c_p} (sp_{sup}^l - sp_{inf}^l)(2ap_l + bp_l(sp_{sup}^l + sp_{inf}^l)) + \frac{1}{2}(\lambda_2 - \lambda_4) \sum_{l=(k^*+1)}^{l=c_n} (sn_{sup}^l - sn_{inf}^l)(2an_l + bn_l(sn_{sup}^l + sn_{inf}^l))$$

$$G(\theta) = \frac{1}{2}(\lambda_1 - \lambda_3)\rho(sp_{sup}^{j^*} - \theta)(2ap_{j^*} + bp_{j^*}(sp_{sup}^{j^*} + \theta)) + \frac{1}{2}(\lambda_2 - \lambda_4)(sn_{sup}^{k^*} - \theta)(2an_{k^*} + bn_{k^*}(sn_{sup}^{k^*} + \theta))$$

La résolution de l'équation (4.32) peut se faire de manière itérative. Pour cela, il suffit de parcourir toutes les classes des documents pertinents. Comme le nombre de classe est fini cela ne pose donc aucun problème. Ainsi, si une classe est fixée, soit  $j^*$ , la fonction  $F$  devient une constante, il suffit alors de chercher le seuil  $\theta_{j^*}^*$  qui maximise  $G$ , puis on calcule l'utilité associée  $U(\theta_{j^*}^*, j^*)$ . On refait ce calcul pour toutes les classes. Le seuil optimal  $\theta^*$  correspond à  $\theta_{j^*}^*$  qui maximise  $U(\theta_{j^*}^*, j^*)$ ,  $\forall j^* \in \{1 \cdot c_p\}$ . L'algorithme suivant peut être utilisé :

1. pour  $j^* = 1$  à  $c_p$  faire
  - calculer  $\theta_{j^*}^* / \theta_{j^*}^* = \operatorname{argmax}_{\theta^* \in [sp_{inf}^{j^*}, sp_{sup}^{j^*}]} U(\theta^*, j^*)$
2.  $\theta^* = \operatorname{argmax}_{j^*} U(\theta_{j^*}^*, j^*) // \theta^*$  est le seuil optimal recherché

Comme nous venons de le souligner, le calcul du seuil optimal  $\theta_{j^*}^*$ , ceci revient à trouver  $\theta$  qui maximise la fonction  $G$  comme suit :

$$\theta_{j^*}^* = \operatorname{argmax}_{\theta} G(\theta) / \theta \in [sp_{inf}^{j^*}, sp_{sup}^{j^*}] \quad (4.33)$$

Pour résoudre l'équation (4.33), la fonction  $G$  peut être réécrite comme suit :

$$G(\theta) = \frac{1}{2}.a.\theta^2 - b.\theta + \frac{1}{2}.c \quad (4.34)$$

avec :

$$a = -\rho.(\lambda_1 - \lambda_3).bp_{j^*} - (\lambda_2 - \lambda_4).bn_{k^*}$$

$$b = \rho.(\lambda_1 - \lambda_3).ap_{j^*} + (\lambda_2 - \lambda_4).an_{k^*}$$

$$c = \rho.(\lambda_1 - \lambda_3)(2sp_{sup}^{j^*}.ap_{j^*} + bp_{j^*}.sp_{sup}^{j^* 2}) + (\lambda_2 - \lambda_4)(2sn_{sup}^{k^*}.an_{k^*} + bn_{k^*}.sn_{sup}^{k^* 2})$$

La valeur de  $\theta$  qui maximise la fonction  $G$  dépend du signe du paramètre "a" :

- si "a" est négatif et la fonction  $G$  est non monotone dans l'intervalle  $[sp_{inf}^{j*}, sp_{sup}^{j*}]$  alors le seuil  $\theta$  qui maximise la fonction  $G$  est donné par la valeur de  $\theta$  qui annule sa première dérivée ;
- si "a" est négatif et la fonction  $G$  est monotone dans l'intervalle  $[sp_{inf}^{j*}, sp_{sup}^{j*}]$  alors le seuil  $\theta$  qui maximise la fonction  $G$  est donné soit par  $sp_{sup}^{j*}$  si  $G(sp_{sup}^{j*}) > G(sp_{inf}^{j*})$ , sinon par  $sp_{inf}^{j*}$  ;
- si "a" est positif alors le seuil  $\theta$  est égal soit à  $sp_{inf}^{j*}$  ou à  $sp_{sup}^{j*}$  qui maximise la fonction  $G$ .

La figure 4.9 illustre les conséquences de la variation de "a" sur la forme de la parabole représentant une fonction de second degré. D'après le signe de "a", on calcule la valeur de  $\theta_{j*}^*$  comme suit :

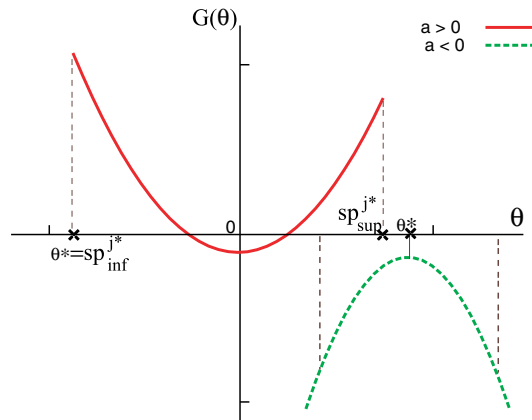


Figure 4.9 – Graphe d'une fonction de seconde degré selon le signe de 'a'

1. si  $a < 0$  alors

$$\theta_{j*}^* / G'(\theta_{j*}^*) = a \cdot \theta_{j*}^* - b = 0 \Rightarrow$$

$$\theta_{j*}^* = \frac{b}{a} = -\frac{\rho \cdot (\lambda_1 - \lambda_3) \cdot ap_{j*} + (\lambda_2 - \lambda_4) \cdot an_{k*}}{\rho \cdot (\lambda_1 - \lambda_3) \cdot bp_{j*} + (\lambda_2 - \lambda_4) \cdot bn_{k*}}$$

– si  $(\theta_{j*}^* \notin [sp_{inf}^{j*}, sp_{sup}^{j*}])$  alors

$$\theta_{j*}^* = \arg \max_{\theta \in \{sp_{inf}^{j*}, sp_{sup}^{j*}\}} (G(\theta))$$

2. sinon

$$\theta_{j*}^* = \arg \max_{\theta \in \{sp_{inf}^{j*}, sp_{sup}^{j*}\}} (G(\theta))$$

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté un modèle de filtrage d'information adaptatif. Un modèle de filtrage adaptatif est caractérisé par deux fonctions principales, l'initialisation et l'adaptation. L'initialisation concerne la représentation du profil et la valeur du seuil à l'état initial du processus de filtrage. L'adaptation implique l'amélioration de la représentation du profil et de la fonction de décision tout au long du processus de filtrage.

Nous nous sommes intéressés plus particulièrement aux problèmes d'apprentissage du profil et d'adaptation de la fonction de seuillage. Les méthodes d'adaptation se déclenchent d'une façon incrémentale, c'est-à-dire à chaque réception d'un document pertinent.

La méthode d'apprentissage du profil que nous avons proposée se base sur le principe de renforcement. Elle consiste à construire un profil *temporaire* permettant de re-sélectionner un document pertinent, déjà sélectionné et jugé par l'utilisateur, avec un score le plus élevé possible. Ce profil *temporaire* est constitué des termes de ce document pertinent, mais avec des poids différents. La mesure de similarité utilisée pour sélectionner un document est le produit scalaire entre la représentation du profil *temporaire* et le document pertinent. Cependant, il s'est avéré que l'amélioration du profil entraîne systématiquement une croissance des scores des documents pertinents à sélectionner, ce qui rend le score de similarité entre le profil *temporaire* et un document pertinent incontrôlable au fur et à mesure que les documents pertinents soient sélectionnés. Dans ce contexte, nous avons apporté des améliorations en normalisant la fonction d'appariement entre un profil *temporaire* et un document donné. Ceci nous a conduit à proposer une nouvelle approche de construction du profil *temporaire*. Cette nouvelle approche a permis d'améliorer considérablement les performances de notre système de filtrage. Les résultats obtenus par cette nouvelle approche sont plus intéressants par rapport à ceux obtenus par la méthode initiale d'apprentissage.

Concernant le seuillage, nous avons proposé une méthode basée sur la distribution des scores des documents pertinents et non pertinents. L'idée de base consiste à réécrire la fonction d'utilité en se basant sur les distributions de probabilités des scores des documents pertinents et non pertinents. Une distribution de probabilités est déduite par une technique de régression linéaire basée sur l'estimation des intervalles. Le premier problème que nous avons résolu concerne l'estimation des intervalles. Nous avons constaté que la répartition des scores des documents sur des intervalles basée sur l'étendue, entraîne souvent une succession d'intervalles vides. L'utilisation d'intervalles vides lors de la linéarisation génère souvent

des probabilités indésirables. Pour résoudre ce problème, nous avons proposé deux solutions possibles : convertir les probabilités négatives en probabilités positives, ou bien introduire les valeurs d'écart type dans l'estimation des intervalles. Nous avons également présenté l'intérêt d'utiliser la technique de linéarisation des scores ou de relier directement les scores associés aux différents intervalles. Enfin, nous avons proposé une nouvelle modélisation de la fonction d'utilité et une méthode de calcul de seuil optimal basée sur le principe de dérivation. Les résultats obtenus par cette nouvelle modélisation du seuil sont présentés dans le chapitre suivant. Ils montrent que l'ensemble de nos propositions permet d'améliorer considérablement la performance de notre système.

# Chapitre 5

## Expérimentations du modèle de filtrage incrémental

### 5.1 Introduction

Dans ce chapitre nous présentons les expérimentations effectuées pour évaluer l'apport des différentes améliorations et techniques proposées. Les améliorations concernent les différentes modifications opérées sur le modèle initial. Nous avons organisé nos expérimentations de la manière suivante :

- nous présentons tout d'abord l'impact de la nouvelle formalisation sur les performances de notre modèle de filtrage adaptatif. A cet effet, nous effectuons des expérimentations pour déterminer les paramètres concernant le choix adéquat du score de renforcement et la normalisation ou non des scores. D'autres expérimentations sont ensuite réalisées, d'une part, pour comparer les techniques d'identification des intervalles proposées, et d'autre part, pour montrer l'intérêt de la linéarisation de la distribution des scores.
- nous comparons ensuite la méthode d'apprentissage du profil avec deux méthodes connues et reconnues pour leurs performances en RI, puis la méthode d'adaptation de la fonction de seuillage avec celle utilisée dans KUN ;
- enfin, nous dressons à l'issue de ces expérimentations une étude comparative entre notre modèle de filtrage et les autres modèles présentés dans TREC-2002, y compris notre modèle initial.

Ce présent chapitre est organisé comme suit : la section 5.2 décrit le programme d'évaluation de TREC-2002. Nous mettons notamment en évidence la tâche de filtrage adaptatif proposée, la collection de test et les mesures d'évaluation que nous avons utilisées pour évaluer les résultats de nos différentes expérimentations. La section 5.3 décrit la méthodologie adoptée pour mettre en oeuvre notre modèle de filtrage adaptatif. La section 5.4 présente les différentes expérimentations énumérées par les trois points ci-dessus.

## 5.2 Programme d'évaluation de TREC-2002

TREC (Text REtrival Conference) est un programme international initié au début des années 90 par le NIST (*National Institute of Standards and Technology*) et co-sponsorisé par le NIST et DARPA/ITO (*Defense Advanced Research Projects Agency - Information Technology Office*), dans le but de proposer des moyens homogènes d'évaluation de systèmes de recherche d'informations sur des bases de documents volumineuses. L'objectif de TREC est d'encourager les travaux de recherche en informatique documentaire permettant l'accès à des bases volumineuses en leurs fournissant :

- une collection de test importante, constituée d'un ensemble de documents et de requêtes (appelés aussi topics dans la terminologie TREC, ou profils dans la tâche de filtrage) ;
- la liste de documents pertinents pour chaque topic ;
- des procédures d'évaluation des résultats de recherche.

Différentes tâches sont proposées chaque année dans le cadre du programme d'évaluation de TREC : tâche multilingue (Cross language track), tâche de filtrage (adaptatif, différé et routage), tâche nouveauté (Novelty track), etc. Ces tâches évoluent d'une année à une autre, et elles tiennent compte des résultats obtenus, de l'évolution des attentes des utilisateurs réels et des collections de documents disponibles. Notre travail de recherche dans le cadre du programme TREC concerne la tâche de filtrage adaptatif.

### 5.2.1 Tâche de filtrage adaptatif

Le processus de filtrage adaptatif démarre seulement avec la description du profil initial (appelé topic dans la terminologie TREC) et un lot très petit de documents d'entraînement. Ces derniers sont utilisés pour initialiser les différents paramètres d'un

système de filtrage donné. Ces paramètres concernent le profil initial, la fonction de décision, les statistiques des termes, etc. Dans TREC-2002 (désignée aussi par TREC11, 11ème version de TREC), trois documents pertinents par profil sont proposés pour démarrer le processus de filtrage adaptatif; aucun autre document pertinent ne peut être utilisé. Le processus de filtrage adaptatif doit s'adapter au fur et à mesure que des documents sont sélectionnés à partir d'une collection de test.

## 5.2.2 Collection de test -Reuters

La collection de test que nous avons utilisée est celle de TREC-2002. Elle est composée de :

- documents issus du corpus de test de Reuters entre 20 août 1996 et 19 août 1997. Ce corpus est constitué d'environ 810.000 documents. Chaque document de la collection est identifié par un numéro unique, auquel une date de création du document est associée;
- un ensemble de 100 ( 101-200) topics pour filtrer les documents de la collection de test. Nous utilisons les 50 premiers topics, créés par les assesseurs du NIST, dans nos différents tests pour évaluer les performances de notre système. Un topic représente une description initiale d'un profil;
- liste de documents pertinents par profil.

### Exemple d'un document TREC :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<newsitem itemid="304704" id="root" date="1997-01-14" xml:lang="en">
<title>CANADA : Great Lakes Power to sell Brascan shares. [CORRECTED 23 :40
GMT]</title>
<headline>Great Lakes Power to sell Brascan shares. [CORRECTED 23 :40
GMT]</headline>
<dateline>TORONTO 1997-01-14</dateline>
<text><p>(Corrects headline and text throughout to make clear Great Lakes Power Inc
intends to sell and not buy Brascan shares.)</p> <p>Great Lakes Power Inc said on
Tuesday that it agreed to sell 6.38 million class A common shares of Brascan Ltd for
C31.25ashare. </p >< p > TheBrascadealwasvaluedatC199.4 million, Great Lakes
said. </p><p>Great Lakes said it planned to sell 4.5 million of the class A shares to the
```

public. An affiliate of Edper Group Ltd agreed to buy 1.8 million class A shares.</p>
<p>Great Lakes said it would file a preliminary short-form prospectus for the offering within two business days. The offering was expected to close on or about February 4.</p>
<p>((Reuters Toronto Bureau (416) 941-8100))</p></text>
<copyright>(c) Reuters Limited 1997</copyright>
<metadata>
<codes class="bip :countries :1.0">
<code code="CAN">
<editdetail attribution="Reuters BIP Coding Group" action="confirmed" date="1997-01-14"/>
</code>
</codes>
<codes class="bip :industries :1.0">
<code code="I83100"> <editdetail attribution="Reuters BIP Coding Group" action="confirmed" date="1997-01-14"/> </code>
<code code="I83960"> ... </code>
</codes>
<codes class="bip :topics :1.0">
<code code="C18"> <editdetail attribution="Reuters BIP Coding Group" action="confirmed" date="1997-01-14"/> </code>
<code code="C181"> ... </code>
</codes>
</metadata>
</newsitem>

### Exemple de topic TREC :

<top>
<num> Number : R101
<Title> Economic espionage
<desc> Description :
What is being done to counter economic espionage internationally?
<narr> Narrative :
Documents which identify economic espionage cases and provide action(s) taken to reprimand offenders or terminate their behavior are relevant. Economic espionage would encompass commercial, technical, industrial or corporate types of espionage. Documents

*about military or political espionage would be irrelevant.*

</top>

<top>

<num> Number : R101 <title> Economic espionage

<desc> Description : What is being done to counter economic espionage internationally?

<narr> Narrative : Documents which identify economic espionage cases and provide action(s) taken to reprimand offenders or terminate their behavior are relevant. Economic espionage would encompass commercial, technical, industrial or corporate types of espionage. Documents about military or political espionage would be irrelevant.

</top>

### 5.2.3 Mesures d'évaluation

Plusieurs mesures d'évaluation sont proposées dans le cadre de la tâche de filtrage de TREC-2002 [Voorhees, 2002]. L'évaluation de notre système est basée sur les deux principales mesures suivantes :

**Mesure d'utilité T11SU** : La fonction d'utilité que nous utilisons pour évaluer notre modèle est la fonction T11SU proposée dans TREC-2002. Cette fonction est présentée dans le chapitre 3 du présent mémoire. Elle consiste à créditer le système de 2 pour un document pertinent sélectionné, et de le débiliter de 1 pour un document non pertinent sélectionné :

$$T11U = U_{(2,-1,0,0)} = 2.R_+ - S_+$$

où,  $R_+$  ( $S_+$ ) est le nombre de documents pertinents (non pertinents) sélectionnés par le système.

En utilisant une telle mesure, il sera très difficile de comparer la performance d'un système par rapport à la totalité des profils utilisés. L'utilité moyenne sur l'ensemble

des profils est toujours dominée par les profils ayant beaucoup de documents pertinents. Ainsi, pour donner une même importance à chaque profil, une normalisation de la fonction  $T11U$  est proposée comme suit :

$$T11SU = \frac{\max(T11NU, MinNU) - MinNU}{1 - MinNU} \quad (5.1)$$

avec :

$$\begin{aligned} T11NU &= \frac{T11U}{MaxNU} \\ MinNU &= -0.5 \\ MaxNU &= 2 * R \end{aligned}$$

où,  $R$  est le nombre de documents effectivement pertinents.

**Mesure T11F :** La mesure T11F, désignée aussi par **F-Beta**, est proposée pour évaluer la performance des systèmes dans TREC-2002. Cette mesure est définie de la façon suivante :

$$T11F = \frac{1.25 * R_+}{R_+ + S_+ + 0.25 * R} \quad (5.2)$$

Dans la majorité des expérimentations présentées dans ce chapitre, les résultats sont toujours représentés par une valeur moyenne sur l'ensemble des profils utilisés. Une valeur moyenne est calculée en fonction de la mesure d'évaluation adoptée. Elle est calculée comme suit :

$$\begin{aligned} T11SU : \quad T11SU_{avg} &= \sum_{i=1}^n \frac{T11SU(P_i)}{n} \\ T11F : \quad T11F_{avg} &= \sum_{i=1}^n \frac{T11F(P_i)}{n} \end{aligned} \quad (5.3)$$

où,  $T11SU(P_i)$  (resp.  $T11F(P_i)$ ) est la valeur d'utilité  $T11SU$  (resp. de  $T11F$ ) du  $i$ ème profil,  $n$  est le nombre de profils utilisés ( $n = 50$ ).

### 5.3 Notre démarche d'évaluation

Au démarrage du processus de filtrage, le profil initial est représenté par les termes les plus significatifs des champs "Title", "Description" et "Narrative" du topic. Les profils et les documents sont représentés par un ensemble de termes pondérés. L'extraction des termes les plus significatifs est réalisée par le module d'indexation de Mercure [Boughanem et al., 1999a]. Les documents sont filtrés dans l'ordre chronologique de leur création, donné par le champ "DOCNO" du document TREC. Le processus de filtrage consiste à comparer chaque document arrivé avec le profil, en calculant un score de similarité basé sur le produit scalaire. Un document est sélectionné si seulement si le score calculé est supérieur au seuil, sinon il est rejeté. Chaque fois qu'un document est sélectionné et jugé pertinent, les processus d'apprentissage du profil et d'adaptation de la fonction de seuillage sont activés.

Pour pouvoir comparer nos résultats à ceux obtenus par les autres participants dans le cadre de TREC-2002, nous avons utilisé en phase initiale les trois documents pertinents proposés dans la tâche de filtrage adaptatif. Ces trois documents sont utilisés par notre processus de filtrage dans le but d'initialiser le profil et le seuil de décision. Nous avons utilisé les trois documents pertinents pour représenter l'échantillon des documents pertinents et un lot de 1000 documents non pertinents issus d'une collection externe, pour représenter l'échantillon des documents non pertinents. Il faut souligner également que les scores des documents dans les deux échantillons sont réajustés à chaque sélection d'un document pertinent. Nous insistons sur le fait que la phase d'initialisation n'est pas obligatoire dans notre approche. Nous l'avons adoptée simplement pour être dans le canevas de l'évaluation TREC.

Enfin, nous utilisons les mesures d'évaluation présentées ci-dessus, pour évaluer les performances de notre système. Nous calculons pour chaque profil, une valeur d'utilité associée. Pour mesurer les performances globales de notre système, nous calculons une valeur d'utilité moyenne des profils selon l'équation (5.3).

Cette démarche d'évaluation est décrite de manière algorithmique comme suit :

*Pour un profil donné,  $P^{(t)}$  :*

–  $E_r^{(0)} = \emptyset$ , représente l'ensemble des documents pertinents, à l'instant  $t = 0$

- $E_{nr}^{(0)} = \{d_1, \dots, d_n\}$ , un ensemble de  $n = 1000$  documents non pertinents,
- $P^{(0)}$ , correspond au profil initial
- **Phase d'initialisation.**
- pour chaque document pertinent,  $D_j^{(t)}$ 
  1. mettre à jour les statistiques des termes
  2. apprendre le profil initial
  3.  $E_r^{(t)} = E_r^{(t-1)} \cup \{D_j^{(t)}\}$
  4. passer au document suivant
- Adapter de la fonction de seuillage, donc le seuil  $\theta^{(0)}$
- **Phase de filtrage.**
- pour chaque document,  $D_j^{(t)}$ , de la collection de test
  1. calculer  $rsv(D_j^{(t)}, P^{(t)})$
  2. si  $rsv(D_j^{(t)}, P^{(t)}) > \theta^{(t)}$  alors
    - (a) si  $D_j^{(t)}$  est pertinent alors
      - mettre à jour les statistiques des termes
      - apprendre le profil
      - $E_r^{(t)} = E_r^{(t-1)} \cup \{D_j^{(t)}\}$  //échantillon de doc. pertinents
      - adapter le seuil  $\theta^{(t)}$
      - passer au document suivant
    - (b) sinon
      - mettre à jour les statistiques des termes
      - $E_{nr}^{(t)} = E_{nr}^{(t-1)} \cup \{D_j^{(t)}\}$  ////échantillon de doc. non pert.
      - passer au document suivant
- Evaluation du système ( $T11SU_{avg}$ ,  $T11F_{avg}$ )

## 5.4 Expérimentations et résultats

L'objectif de nos expérimentations est d'évaluer nos apports par rapport tout d'abord au modèle initial, puis à des travaux, reconnues dans le domaine de la RI, en liaison avec notre travail. Nous avons à cet effet organisé cette section de la manière suivante :

1. nous présentons tout d'abord les expérimentations effectuées pour déterminer les meilleurs paramètres à sélectionner au niveau de l'apprentissage du profil, en l'occurrence la valeur de  $\lambda$  et la normalisation ou non des scores (sous-section 5.4.1). Au

niveau de l'adaptation de la fonction de seuillage, ces choix concernent la méthode d'identification des intervalles et l'intérêt de la linéarisation (sous-section 5.4.2). Une fois ces choix effectués, ils sont retenus pour le reste de l'évaluation ;

2. nous comparons ensuite la méthode d'apprentissage du profil avec deux méthodes connues et reconnues pour leurs performances en RI, puis la méthode d'adaptation de la fonction de seuillage avec celle utilisé dans KUN ;
3. enfin, nous dressons à la fin de ces expérimentations une étude comparative entre notre modèle de filtrage et les autres modèles présentés dans TREC-2002.

## 5.4.1 Apprentissage du profil

### 5.4.1.1 Impact du renforcement par normalisation

Notre première réflexion, pour améliorer le modèle de filtrage adaptatif existant, concerne le processus d'apprentissage du profil. Nous avons constaté qu'il est difficile de fixer le score de renforcement  $\lambda$  entre le profil temporaire et un document pertinent sélectionné. La solution que nous avons proposée consiste donc à normaliser la fonction d'appariement entre le profil temporaire et le document pertinent. La première question que nous nous sommes alors posée concerne l'intérêt de normaliser la fonction d'appariement au niveau du calcul du profil temporaire par rapport à la normalisation directe de la mesure de score document-profil global. Comme nous l'avons souligné dans le chapitre 4, nous avons soulevé l'inconvénient de normaliser la fonction de similarité directement dans ce processus de filtrage. Afin d'étayer ce point, nous avons réalisé des expérimentations en considérant deux mesures d'appariement normalisées : la mesure de Dice et la mesure de Jaccard.

Le tableau 5.1 présente les résultats obtenus en considérant les différents cas possibles. Dans la première colonne du tableau, nous représentons la mesure d'appariement entre un profil temporaire et un document donné par  $rsv(P_x, D_j)$ , et entre un profil global et un document donné  $rsv(P, D_j)$ . La deuxième colonne du tableau présente les cas où on normalise ou non ces mesures. Les deux dernières colonnes présentent les valeurs d'utilité moyennes obtenues ( $T11SU$ ), selon les différents cas, par les deux mesures d'appariement normalisées de Jaccard et de Dice. Le premier constat clair que l'on peut tirer est que la normalisation de la fonction d'appariement au niveau du processus de filtrage ( $rsv(P, D_j)$ ) réduit considérablement les performances du système. Deuxièmement, on remarque bien que la

	Normalisa.	Mesure de Jaccard	Mesure de Dice
$rsv(P_x, D_j)$	oui	0.4535	0.4745
$rsv(P, D_j)$	non		
$rsv(P_x, D_j)$	oui	0.2670	0.2438
$rsv(P, D_j)$	oui		

Tableau 5.1 – Normalisation par les mesures de Dice et Jaccard

mesure de Dice permet d'obtenir de meilleurs résultats par rapport à celle de Jaccard. Il faut noter que la non normalisation des deux fonctions de similarité n'a pas été traitée, car nous rappelons que notre but est de ramener la valeur  $\lambda$  à un intervalle contrôlable.

Dans la suite de nos expérimentations, nous utilisons la mesure d'appariement de Dice dans la construction du profil temporaire.

#### 5.4.1.2 Sélection du score de renforcement

Dans le but de retenir la valeur du score avec lequel on souhaite construire le profil temporaire, nous avons testé plusieurs valeurs de  $\lambda$ . La figure 5.1 illustre les valeurs d'utilité obtenues pour chaque valeur de  $\lambda$ . On constate tout d'abord qu'il n'y pas une valeur unique de score qui décroche les autres, mais la valeur optimale est plutôt située dans un intervalle optimal  $\lambda \in [0.3..0.9]$ . Ce résultat est intéressant car, comme il n'y a pas une valeur unique et remarquable, le risque de faire le mauvais choix de la valeur de  $\lambda$  est minimisé. Ce risque est plus important, comme nous l'avons souligné, dans le cas de la non normalisation du score de renforcement tel qu'il est réalisé dans le modèle initial.

La figure 5.1 montre également, qu'au delà de la valeur 0.9, les profils tendent à se personnaliser par rapport au document d'apprentissage. Autrement dit, une valeur très grande de  $\lambda$  implique une caractérisation du profil global par le contenu du document pertinent en cours de traitement. En deçà de 0.3, l'amélioration du profil reste faible comparativement aux autres valeurs de  $\lambda$ . Pour le reste de nos expérimentation, nous fixons la valeur de  $\lambda$  à 0.85, valeur pour laquelle nous avons obtenu la meilleure utilité.

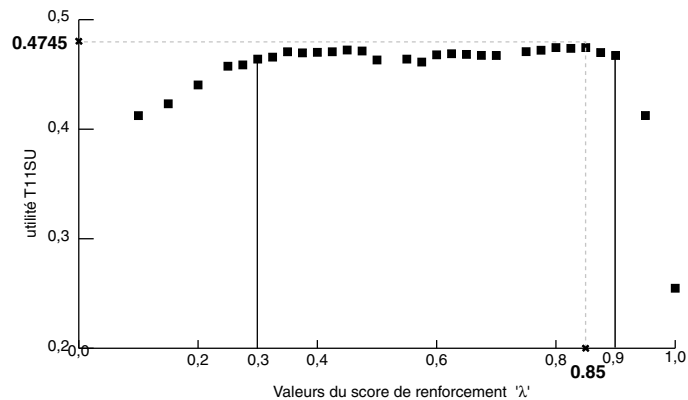


Figure 5.1 – Valeur d'utilité pour chaque valeur de  $\lambda$

## 5.4.2 Adaptation de la fonction de seuillage

Concernant l'adaptation de la fonction de seuillage, les expérimentations que nous avons effectuées permettent d'évaluer les impacts de l'identification des intervalles et l'utilisation de la régression linéaire ou non dans la construction de la distribution de probabilités. Ces différentes expérimentations sont réalisées en utilisant la méthode d'optimisation du seuil par dérivation que nous avons proposée. L'adaptation du profil est réalisée par notre méthode de renforcement par normalisation des scores.

### 5.4.2.1 Évaluation des méthodes d'identification des intervalles

L'identification de "bons" intervalles pour la mesure de probabilités des scores est une étape importante de notre approche. Le but de cette expérimentation est de comparer les méthodes que nous proposons vis-à-vis de celle présentée dans le modèle initial.

Les trois méthodes d'identification des intervalles que nous avons expérimentées sont les suivantes :

- la méthode d'identification des intervalles par l'étendue sans rectification du problème des probabilités négatives, représentée par *Etendu.Prob.Négat.* dans la figure 5.2 ;
- la méthode d'identification des intervalles par l'étendue avec rectification du problème des probabilités négatives, représentée par *Etendu.Prob.Posit.* dans la figure 5.2 ;
- la méthode d'identification des intervalles par l'écart type, représentée par *Ecart-type* dans la figure 5.2.

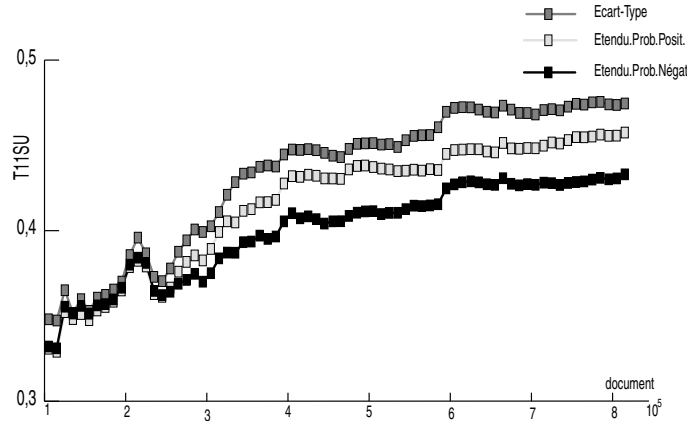


Figure 5.2 – Utilité cumulée par chaque méthode d’identification d’intervalles

Pour comparer ces trois méthodes, nous avons calculé pour chaque méthode les valeurs d’utilité moyenne cumulées pour l’ensemble des profils. A cet effet, à chaque passage d’une quantité bien définie de documents (par exemple,  $10^5$ ), nous calculons la valeur moyenne d’utilité  $T11SU_{avg}$  de tous les profils.

La figure 5.2 illustre, pour chaque méthode, les valeurs d’utilité moyenne cumulée dans le temps pour les 50 profils. On remarque que l’évolution de l’utilité dans chaque méthode est uniforme dans le temps. Au début du processus de filtrage il n’y a aucune différence d’utilité entre les trois méthodes, car, comme le nombre de documents pertinents sélectionnés est faible (une moyenne de 5) les intervalles obtenus sont souvent similaires. On constate également que, le fait de considérer les probabilités négatives (*Etendu.Prob.Négat.*) lors de l’optimisation du seuil provoque des anomalies au niveau du calcul de surface entre les différentes classes d’une distribution donnée. L’amélioration enregistrée par la deuxième méthode (*Etendu.Prob.Posit.*) pourrait être expliquée par les surfaces récupérées par le remplacement des probabilités négatives par leurs valeurs absolues (voir section (4.5.1.2)). Enfin, la performance de la méthode d’identification des intervalles par l’écart type (*Ecart-type*) a permis de pallier les différentes contraintes des deux autres méthodes. Par conséquent, nous retenons la méthode d’identification des intervalles par l’écart type pour le reste de nos expérimentations.

Dist. de Prob.	Avec linéarisation	Sans linéarisation
$T11SU_{avg}$	0.4745	0.4528
$T11F_{avg}$	0.4624	0.4482

Tableau 5.2 – Résultats de la linéarisation ou non de la dist. de probabilités

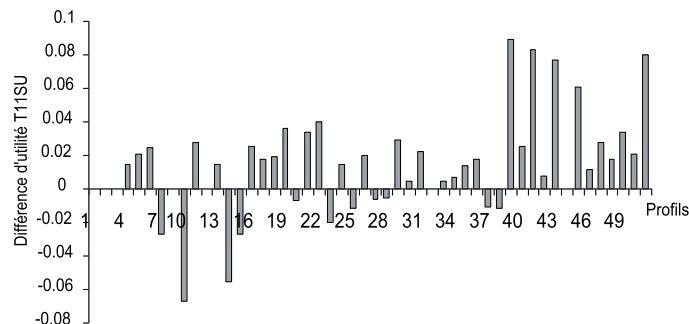


Figure 5.3 – Comparaison entre la linéarisation et non linéarisation

#### 5.4.2.2 Comparaison entre linéarisation et non linéarisation

La question posée dans cette expérimentation concerne l'intérêt de la linéarisation des scores. Nous avons pour cela comparé les deux techniques de construction de distribution de probabilités présentées dans le chapitre 4, celle basée sur la régression linéaire et celle qui consiste juste à relier les scores des documents.

Un premier résultat que l'on peut constater (voir tableau 5.2) est que la linéarisation permet d'obtenir de meilleurs résultats.

Pour mieux analyser l'impact réel de la linéarisation sur les profils, nous avons tout d'abord ordonné ces derniers par rapport à la taille de l'échantillon des documents pertinents associé. Nous calculons ensuite pour chaque profil la différence entre l'utilité obtenue par la technique de linéarisation et l'utilité obtenue par la liaison directe des scores des échantillons.

La figure 5.3 illustre les résultats obtenus. On remarque que plus la taille de l'échantillon est élevée plus l'utilité obtenue par la technique de linéarisation est plus intéressante par rapport à celle obtenue par la liaison directe des scores.

### 5.4.3 Comparaison des méthodes d'adaptation

Cette section présente les expérimentations effectuées pour comparer nos méthodes d'apprentissage du profil et d'adaptation de la fonction de seuillage avec d'autres méthodes connues et reconnues pour leur performance dans le domaine de la RI. Ainsi, pour évaluer notre méthode d'apprentissage par renforcement avec normalisation, nous l'avons comparée avec deux méthodes d'apprentissage : une version incrémentale de l'algorithme de Rocchio et les méthodes d'apprentissage utilisées par le modèle de filtrage Okapi [Robertson and Walker, 1999]. Pour évaluer les performances de notre méthode d'adaptation de la fonction de seuillage, nous l'avons comparée à la méthode développée dans KUN par Arampatzis et détaillée dans le chapitre 3.

#### 5.4.3.1 Apprentissage du profil : Rocchio et Okapi

Afin d'évaluer notre méthode d'apprentissage du profil avec une version incrémentale de Rocchio et d'Okapi, nous avons remplacé dans notre modèle de filtrage le processus d'apprentissage du profil par renforcement par chacune de ces méthodes. Autrement dit, nous utilisons notre méthode d'adaptation de la fonction de seuillage pour chaque méthode testée.

Avant de présenter les résultats des différentes méthodes, nous donnons un bref aperçu sur les méthodes d'apprentissage de Rocchio et d'Okapi. La mise en oeuvre de ces méthodes d'apprentissage est donnée par l'algorithme suivant :

*Pour un profil donné :*

- $E_r^{(0)} = \emptyset$ , l'ensemble des documents pertinents, à l'instant  $t = 0$
- $E_{nr}^{(0)} = \emptyset$ , l'ensemble des documents non pertinents, à l'instant  $t = 0$
- pour chaque document  $D_j^{(t)}$ 
  - calculer  $rsv(D_j^{(t)}, P^{(t)})$ ,
  - si  $rsv(D_j^{(t)}, P^{(t)}) > \theta^{(t)}$  alors
    - si  $D_j^{(t)}$  est pertinent alors
      - $E_r^{(t)} = E_r^{(t-1)} \cup \{D_j^{(t)}\}$ ,
      - appliquer Rocchio (équation (2.22) du chapitre 2),
      - appliquer les méthodes d'Okapi, TSV et NTSV (équations (3.26) et (3.28) du chapitre 3),

- *passer au document suivant,*
- *sinon*
- $E_{nr}^{(t)} = E_{nr}^{(t-1)} \cup \{D_j^{(t)}\},$
- *passer au document suivant,*
- *fin-si*
- *fin-si*
- *fin-pour*

Il faut souligner que nous retenons, pour chaque méthode d'apprentissage de profil, seulement les 60 premiers termes de poids élevés pour chaque apprentissage. Il est de même pour notre méthode d'apprentissage de profil par renforcement.

**Algorithme de Rocchio.** Plusieurs versions incrémentales de l'algorithme de Rocchio ont été utilisées pour l'apprentissage de profils en filtrage d'information. On y trouve notamment les travaux de [Callan, 1998] [Schapire et al., 1998], Query Zoning (ou QZ) développé par Singhal et al. [Singhal et al., 1997] et Dynamic Query Zoning (ou DQZ) [Buckley and Salton, 1985]. La version incrémentale de l'algorithme de Rocchio que nous avons utilisée dans notre expérimentation tient part de celle adoptée par singhal dans Query Zoning et décrit dans l'algorithme ci-dessus. Elle consiste d'une façon générale, à construire un nouveau profil à chaque sélection d'un document pertinent. Ce nouveau profil tient compte du profil initial et des documents pertinents et non pertinents effectivement sélectionnés à l'instant  $t$ .

**Adaptation par BM25 d'Okapi.** L'adaptation du profil, dans le système de filtrage d'Okapi consiste à apprendre le profil global à chaque sélection d'un document pertinent. L'ajustement des poids des termes dans le nouveau profil construit est donné par la formule de pondération standard de Robertson et Spark-Jones BM25 (présentée dans la section 3.25 du chapitre 3). Nous utilisons les deux méthodes présentées dans la section 3.10.2 du chapitre 3, pour les comparer avec notre méthode d'apprentissage par renforcement. Nous identifions la première méthode d'apprentissage TSV par Okapi/BSS<sup>1</sup>(1) et la deuxième méthode NTSV par Okapi/BSS(2).

Le tableau 5.3 illustre, pour des valeurs distinctes de  $\alpha$ ,  $\beta$  et  $\gamma$  (d'autres valeurs ont été testées, nous avons retenu seulement les meilleures), la valeur d'utilité moyenne sur les 50 profils de test obtenue par la version de l'algorithme de Rocchio. On remarque

---

<sup>1</sup>Okapi Basic Search System

que, les meilleures valeurs d'utilité sont obtenues lorsque  $\alpha = 0$ , c'est-à-dire, lorsqu'on ne tient pas en compte du profil initial dans le calcul du nouveau profil. La meilleure valeur d'utilité obtenue est donnée par  $\alpha = 0$ ,  $\beta = 2$  et  $\gamma = 1$ . Nous retenons ces valeurs pour comparer les résultats obtenus par l'algorithme de Rocchio par rapport aux autres méthodes d'apprentissage.

$\alpha$	$\beta$	$\gamma$	$T11SU$
0	1	1	0.3873
0	2	1	<b>0.4270</b>
0	3	1	0.4155
1	1	1	0.3744
1	2	1	0.3697
1	3	1	0.3768

Tableau 5.3 – Valeurs distinctes de  $\alpha$ ,  $\beta$  et  $\gamma$  : Algo. Rocchio

Le tableau 5.4 présente, pour chacune des méthodes, les valeurs moyennes de T11SU et T11U obtenues. On remarque que les valeurs d'utilité obtenues par notre méthode par renforcement sont nettement supérieures à celles obtenues par les différentes méthodes d'apprentissage utilisées.

	Rocchio	Okapi/ BSS(1)	Okapi/ BSS(2)	Renforcement
T11SU	0.4270	0.3549	0.4003	<b>0.4745</b>
T11U	52.52	27.94	33.90	<b>69.04</b>

Tableau 5.4 – Comparaison entre Rocchio, BM25 et renforcement

Au delà de cette comparaison quantitative, nous rappelons qu'un des intérêts de notre méthode d'apprentissage par rapport aux méthodes présentées dans ce chapitre est sa capacité d'apprendre de manière uniforme tout au long du processus de filtrage. A fin de montrer cette qualité d'apprentissage de notre méthode, la figure 5.4 illustre l'évolution de l'utilité moyenne cumulée pour les trois méthodes d'apprentissage. L'utilité cumulée consiste à calculer une valeur d'utilité à chaque passage d'un certain nombre de documents. Dans notre cas, nous calculons une valeur d'utilité cumulée à chaque passage

de  $10^5$  documents. Comme nous pouvons le constater, la méthode par renforcement croît considérablement et de manière uniforme dès la sélection des premiers documents pertinents. Nous remarquons également que Rocchio apprend mieux que la méthode d'apprentissage d'Okapi.

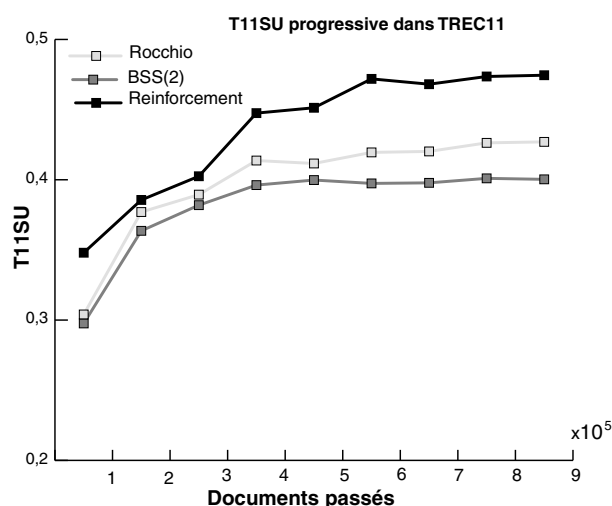


Figure 5.4 – Comparaison des performances des différentes méthodes d'apprentissage

Nous avons également effectué une autre expérimentation pour vérifier si les deux techniques d'apprentissage, Rocchio et renforcement, permettent de construire les mêmes profils. Nous nous limitons dans cette expérimentation à l'algorithme de Rocchio, car il obtient de meilleurs résultats par rapport aux autres méthodes d'Okapi. Pour réaliser cette expérimentation, nous avons mesuré le nombre de termes communs entre le profil appris par la méthode de Rocchio et celui appris par la méthode de renforcement. Pour ce faire, nous considérons le cas du profil 101, où la valeur d'utilité obtenue par la méthode de Rocchio est de 0,7311 et celle obtenue par la méthode de renforcement est de 0,9098.

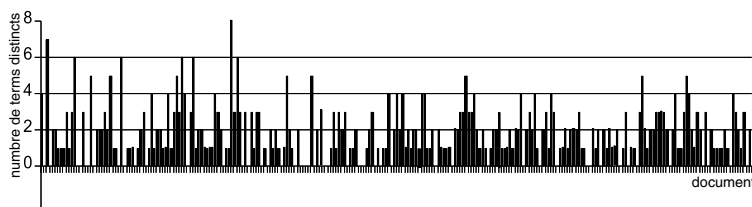


Figure 5.5 – Nombre de termes distincts entre deux profils :Renforcement-Rocchio

	LDS	KUN
T11SU	0.4745	0.3664
T11F	0.4624	0.3408
T11U	69.04	27.86

Tableau 5.5 – Résultats LDS - KUN

La figure 5.5 présente le nombre de termes différents entre les deux profils. Ce nombre est calculé chaque fois que les deux profils sont modifiés simultanément, en d’autres termes, chaque fois qu’un même document pertinent est sélectionné par les deux profils. Il est égal au nombre de termes qui apparaissent dans le profil appris par renforcement et qui n’apparaissent pas dans le profil appris par Rocchio plus le nombre de termes qui apparaissent dans le profil appris par Rocchio et qui n’apparaissent pas dans le profil appris par renforcement.

Nous constatons que le nombre de termes distincts entre les deux profils varie en moyenne entre 0 et 4 (sur 60 termes contenus dans chaque profil). Cette différence ne devrait pas influencer sur les performances entre les deux méthodes. Or, on remarque que la valeur d’utilité obtenue par notre méthode est supérieure de 20 % par rapport à la méthode de Rocchio. La différence d’utilité s’explique par les poids affectés aux différents termes discriminants par ces deux méthodes.

#### 5.4.3.2 Adaptation de la fonction de seuillage dans KUN

L’une des premières méthodes ayant utilisé le principe de distribution des scores dans le processus d’adaptation de la fonction de seuillage est présentée par Arampatzis dans KUN [Arampatzis et al., 2000]. La méthode d’adaptation de la fonction de seuillage proposée par Arampatzis, rappelons-le, suppose que la distribution des scores des documents pertinents (resp. non pertinents) suit une loi normale (resp. une loi exponentielle), contrairement à notre approche, où les deux distributions sont construites par la linéarisation des scores. Le but de cette expérimentation est de comparer ces deux méthodes. Nous utilisons la notation LDS pour désigner notre méthode d’adaptation de la fonction de seuillage, par référence à la Linéarisation de la Distribution des Scores, et KUN pour celle d’Arampatzis.

Le tableau 5.5 montre les résultats, en terme de  $T11SU$ ,  $T11F$  et  $T11U$ , obtenus par les deux méthodes. Le premier résultat clair que l’on peut tirer est que notre approche améliore

mois	Utilité cumul.			Utilité périodique		
	KUN	LDS	Perf.	KUN	LDS	Perf.
1996 – 10	0.2882	0.3606	25%	0.2882	0.3605	25%
1996 – 11	0.3140	0.3815	22%	0.3663	0.4305	18%
1996 – 12	0.3275	0.4000	22%	0.3862	0.4197	9%
1997 – 01	0.3420	0.4335	27%	0.4059	0.4711	16%
1997 – 02	0.3414	0.4461	31%	0.3792	0.4571	21%
1997 – 03	0.3558	0.4516	27%	0.4048	0.4471	10%
1997 – 04	0.3532	0.4573	29%	0.3594	0.4342	21%
1997 – 05	0.3576	0.4710	32%	0.3120	0.3485	12%
1997 – 06	0.3585	0.4685	31%	0.3349	0.3837	15%
1997 – 07	0.3649	0.4754	30%	0.4103	0.4801	17%
1997 – 08	0.3664	0.4745	30%	0.3235	0.3550	10%

Tableau 5.6 – Evolution de l'utilité

considérablement les différentes mesures utilisées par rapport à KUN. Nous rappelons que dans l'expérimentation réalisée pour cette comparaison, la méthode d'apprentissage entre KUN et LDS est identique, la différence réside dans le manière de construire la distribution de probabilités et le calcul du seuil optimal.

Afin d'analyser finement ces résultats, notamment sur la manière dont évolue l'utilité dans le temps, nous avons mesuré l'utilité par période de temps. Comme les documents de la collection de test sont publiés entre 1996-1997, nous les avons alors regroupé en plusieurs périodes de temps (selon la date de leur publication).

Le tableau 5.6 montre les utilités périodiques, calculées pour la période considérée, et cumulées, calculées par rapport à tous les documents filtrés du début du filtrage jusqu'à la période considérée. Une période est identifiée par l'année et le mois de publication des documents (ex. 1996-10) obtenus par chacune des deux. On constate tout d'abord si l'on regarde les colonnes d'utilité cumulées, des deux méthodes, les résultats de notre approche sont meilleurs que ceux de KUN dès le début du filtrage. Cette différence se creuse au fur et à mesure que l'on avance dans le filtrage. En effet, la différence de performances entre LDS et KUN passe de 22% à 30%, ce qui montre qu'on gagne de l'utilité tout au long du filtrage avec LDS comparativement à KUN.

#### 5.4.4 Evaluation comparative avec les résultats TREC-2002

L'objectif de cette section est de confronter nos résultats à ceux présentés par les participants de TREC-2002.

Le tableau 5.7 présente le rang et les valeurs moyenne de l'utilité  $T11SU$ ,  $T11F$  et de  $T11U$  obtenus par les différents participants à la tâche de filtrage adaptatif de TREC-2002 (14 participants). Nous avons inclus dans ce tableau les résultats de notre approche et mis en évidence le rang qu'on aurait obtenu dans ce cas. On constate tout d'abord que comparativement à notre modèle initial, correspondant à la ligne (IRIT Toulouse), l'utilité passe de 0.386% à 0.474%, soit une accroissement de 23%. Ensuite le résultat important que l'on peut tirer de ce tableau est que notre approche obtient le meilleur résultat (rang 1) si l'on considère les mesures  $T11U$  et  $T11F$ , et nous suivons au second rang pour  $T11SU$ . La mesure  $T11U$  indique la proportion de documents pertinents retrouvés parmi les documents sélectionnés. La valeur  $T11U$  obtenue par notre nouveau modèle (69.04) montre effectivement que nous sélectionnons plus de documents pertinents que de documents non pertinents.

Participant	Moy. T11F	Moy. T11SU	Moy. T11U
<b>Nouveau modèle</b>	<b>0.462</b>	<b>0.474</b>	<b>69.04</b>
Chinese Academy of Sc.	0.427	0.475	60.76
Microsoft R. Cambridge	0.421	0.435	49.4
Tsinghua Univ.	0.417	0.395	43.78
Carnegie Mellon Univ.	0.41	0.447	51.3
KerMIT Consortium	0.376	0.459	59.04
CLIPS IMAG Lab	0.369	0.424	44.12
Fundan Univ.	0.346	0.397	23.8
<b>IRIT Toulouse</b>	<b>0.327</b>	<b>0.386</b>	<b>34.76</b>
Independent C.Lewis	0.318	0.293	14.08
Queens College CUNY	0.196	0.154	-156.82
Rutgers Univ.-Kantor	0.187	0.337	12.62
Univ. of Iowa	0.174	0.333	6.64
Johns Hopkins Univ.	0.104	0.342	5.5
Univ. of Buffalo-Cedar	0.014	0.013	-39404.06

Quelques indications sur certains modèles décrits dans ce tableau :  
le modèle de Chinesse Academy est proche du modèle de Wu et al. présenté dans le chapitre 3, notamment dans le processus d'apprentissage du profil ;  
le modèle de Microsoft R. Cambridge c'est le système Keenbow/Okapi du chapitre 3 ; Carnegie Mellon Univ. utilise le modèle de KUN  
utilise le modèle de KUN.

Tableau 5.7 – Liste des participants à TREC-2002 : Filtrage adaptatif

Nous avons également effectué une comparaison des valeurs d'utilité obtenues, pour chaque profil, par notre nouveau modèle par rapport à l'utilité moyenne obtenue par tous les participants de TREC. La figure 5.6 illustre cette comparaison, où pour chaque profil nous calculons la différence entre la valeur d'utilité que nous obtenons et l'utilité moyenne des différents participants. On constate que les valeurs d'utilité obtenues pour presque

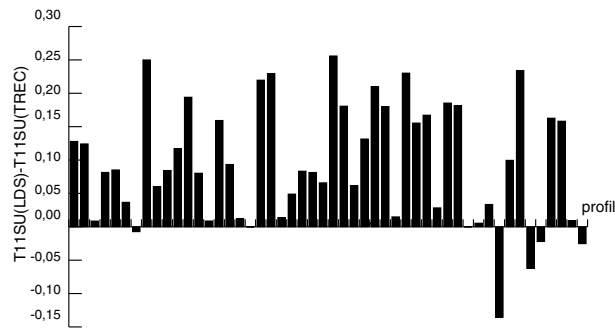


Figure 5.6 – Différence par rapport à la moyenne dans TREC-2002

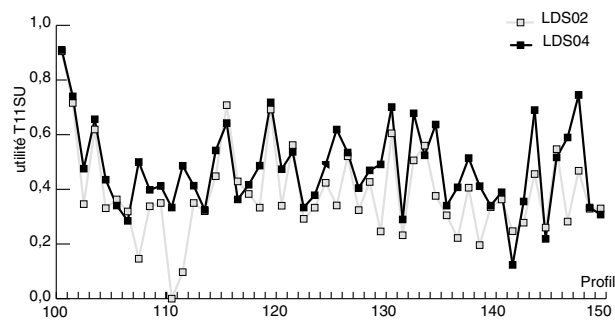


Figure 5.7 – Valeur d'utilité (T11SU) par profil

tous les profils sont au dessus de la moyenne (43 profils). Ceci nous permet d'affirmer que les performances obtenues ne sont pas le "coup du sort" de quelques profils qui surclassent les autres, mais cette performance est répartie sur un ensemble important de profils. Ceci permet en fait en partie de détacher les résultats vis-à-vis de la collection utilisée. On pourrait on fait penser que ces performances peuvent être reproduites sur d'autres collections.

Un dernier point que nous souhaitons mettre en évidence, concerne les performances obtenues par notre approche vis-à-vis du modèle initial sur chaque profil utilisé. La figure 5.7 illustre la différence, sur les 50 profils, entre les valeurs d'utilité T11SU obtenues par notre nouveau modèle et le modèle initial pour chaque profil. On remarque que nous avons pu améliorer l'utilité sur un plus grand nombre de profils, soit 40 profils sont améliorés.

## 5.5 Conclusion

Dans ce chapitre nous avons présenté les expérimentations et les résultats obtenus par notre modèle de filtrage adaptatif. Nos expérimentations ont été basées sur le programme d'évaluation des systèmes de filtrage adaptatif de la tâche de TREC-2002. Les résultats obtenus par notre modèle sont nettement supérieurs à ceux obtenus lors de notre participation à la campagne d'évaluation de TREC-2002. Nous avons pu améliorer notre modèle de filtrage adaptatif de 28% par rapport au modèle de filtrage initial. Avec cette amélioration, nous nous sommes repositionné par rapport aux autres participants de TREC-2002. En effet, nous sommes passé de la 8ème position à la deuxième position lorsque la mesure d'évaluation utilisée est la fonction d'utilité T11SU. Cependant, nous avons amélioré considérablement les performances de notre système (1ère position) lorsque la mesure d'évaluation utilisée est T11F. Pour situer nos méthodes d'apprentissage du profil et d'adaptation de la fonction de seuillage, nous les avons comparé à quelques méthodes proposées dans la littérature. Les résultats obtenus par nos méthodes sont très satisfaisants comparés à ceux obtenus par ces méthodes.

## 5.6 Conclusion générale et perspectives

Les travaux présentés dans ce mémoire se situent dans le contexte général des systèmes d'accès à l'information et plus particulièrement dans le cadre des systèmes de filtrage d'information. Un Système de Filtrage d'Information (SFI) permet d'extraire, à partir d'un flot d'informations, celles qui sont susceptibles d'intéresser un utilisateur ou un groupe d'utilisateurs ayant des besoins en information relativement stables, besoins modélisés au travers du concept de *profils utilisateurs*. Nous nous sommes intéressé dans ce mémoire plus précisément aux systèmes de filtrage basés sur le contenu, appelés également filtrage cognitif. Un système de filtrage cognitif se base sur des modèles de recherche d'information augmentés d'une fonction de décision. D'une façon générale, il traite des documents provenant de flot d'informations et décide à la volée, si le document correspond ou pas aux besoins des utilisateurs. Ceci revient à calculer un score de similarité entre le profil et le document. Si ce score dépasse un certain seuil, ce document est sélectionné, donc acheminé vers l'utilisateur du profil sinon il est rejeté. Dans le but d'adapter le processus de sélection d'information pertinente au flot d'informations, un SFI intègre un processus d'apprentissage qui se base sur des informations cumulées, issues des documents déjà filtrés. L'apprentissage concerne plus particulièrement les profils et la fonction de décision.

Notre contribution se situe plus précisément au niveau de l'apprentissage des profils et de l'adaptation de la fonction de décision. Les processus d'apprentissage et d'adaptation, présentés dans ce mémoire, sont incrémentaux, c'est-à-dire ils s'adaptent au flot d'informations.

La méthode d'apprentissage du profil que nous avons proposé, appelée *apprentissage par renforcement*, est purement incrémentale. Elle ne nécessite aucune connaissance autre que le profil initial, au démarrage du processus de filtrage. Notre processus d'apprentissage est déclenché pour chaque document jugé pertinent par l'utilisateur. Elle consiste tout d'abord à construire un profil temporaire à partir de ce document. Ce profil devrait permettre de sélectionner le document en question, avec un score le plus élevé possible, appelé score de renforcement. Ce profil temporaire est ensuite intégré dans le profil global de l'utilisateur. Compte tenu de la fonction de décision utilisée, il s'est avéré que l'apprentissage du profil entraîne systématiquement un accroissement des scores des documents pertinents. Ceci le score de renforcement incontrôlable au fur et à mesure que les documents pertinents sont sélectionnés. Comme nous l'avons tout au long de ce mémoire, ce travail a été tout d'abord présenté dans le modèle initial relevant des travaux effectués dans le cadre de la

thèse M. Tmar. Nous avons également participé à la mise en oeuvre de certaines solutions, notamment dans notre participation à TREC<sup>2</sup>.

Nous avons alors proposé une méthode qui permet de ramener ces scores dans un intervalle contrôlable, en l'occurrence [0..1]. Nous avons pour cela réécrit le système d'équations permettant de construire le profil temporaire on y intégrant un facteur permettant la normalisation des scores. Nous avons ensuite proposé une méthode de résolution de ce système pour déterminer le profil temporaire. Nous avons montré à travers les expérimentations effectuées, que cette nouvelle technique d'apprentissage permet d'apprendre les profils de manière uniforme tout au long du processus de filtrage, comparativement aux différentes méthodes proposées dans le domaine.

Concernant, le seuillage, nous proposons une méthode d'adaptation du seuil qui s'inscrit dans la catégorie des méthodes basées sur la distribution des scores des documents. Notre méthode suppose que les distributions des scores des documents sont inconnues, mais propose d'estimer les probabilités discrètes des scores des documents, puis de "dessiner" la distribution des scores en utilisant une régression linéaire. Une fois ces distributions construites, nous proposons de réécrire la fonction d'utilité en fonction de ces distributions, puis de déduire le score (seuil) qui permet d'optimiser cette fonction. La construction d'une distribution de probabilité des scores des documents consiste tout d'abord à décomposer les scores en plusieurs intervalles, puis à calculer la probabilité des scores dans ces intervalles. Une régression linéaire est ensuite utilisée pour convertir la distribution de probabilités discrètes en une distribution de probabilités continue. Ce travail rentrait dans le cadre du modèle initial. Nous avons tout d'abord apporté une amélioration concernant l'estimation des intervalles. Nous avons constaté que la répartition des scores des documents sur des intervalles, calculés à partir de l'étendu des scores, entraîne souvent une succession d'intervalles vides. L'utilisation d'intervalles vides lors de la linéarisation génère souvent des probabilités indésirables. Pour résoudre ce problème, nous avons proposé deux solutions possibles, la première consiste à convertir les probabilités négatives en probabilités positives, et la seconde prend en compte la variance des scores pour déterminer ces intervalles.

Nous avons ensuite proposé une nouvelle formalisation de la fonction d'optimisation du seuil et une méthode de résolution de cette fonction. Contrairement, au modèle initial, la fonction d'optimisation du seuil a été tout d'abord réécrite en fonction des densités de probabilités théoriques des scores documents pertinents et non pertinents. Nous avons ensuite ramener ces densités à un calcul de surfaces. De plus, l'estimation du seuil dans le

---

<sup>2</sup>Text REtrieval Conference

modèle initial consistait à détecter une valeur optimale du seuil de manière empirique, en considérant une liste de scores possibles dans un intervalle donné. Ce procédé présente une limite car il ne permet pas de trouver la valeur exacte du seuil. La méthode que nous avons proposé est déterministe, elle permet effectivement de résoudre directement la fonction d'optimisation du seuil et de ce fait de déterminer la valeur optimale du seuil.

Pour valider ces propositions nous avons effectué une série d'expérimentations sur des collections issues du programme TREC. La démarche d'évaluation que nous avons suivie respecte le canevas défini dans TREC. Ce choix est effectué pour pouvoir comparer et situer nos travaux par rapport à ceux présentés notamment dans le cadre de TREC 2002.

Les premières expérimentations ont été effectuées pour déterminer les meilleurs paramètres pour l'apprentissage du profil et l'adaptation de la fonction de décision. Ces paramètres concernent la valeur du score de renforcement, la normalisation ou non des scores, l'identification des intervalles et enfin l'intérêt de la linéarisation.

Nous avons également effectué une série d'expérimentations pour situer nos propositions vis-à-vis de celles connues et reconnues pour leurs performances dans le domaine de la RI. Concernant l'apprentissage du profil, nous avons comparé notre méthode avec deux méthodes distinctes : une version incrémentale de l'algorithme de Rocchio et l'expansion de requêtes dans le modèle d'Okapi. Outre leurs performances reconnues dans le domaine, ces méthodes ont été sélectionnées car leurs performances dépendent de la taille de l'échantillon d'apprentissage. Nous avons montré que notre méthode d'apprentissage permet en effet d'apprendre de manière uniforme et obtient de meilleurs résultats, en terme d'utilité, tout au long du processus de filtrage. Nous avons également comparé notre méthode d'adaptation de la fonction de décision à une méthode qui suppose que les distributions de probabilités sont connues *à priori*, en l'occurrence la méthode KUN. Les résultats obtenus avec notre méthode sont clairement meilleurs dès le début du filtrage ; cette différence se creuse au fur et à mesure que l'on avance dans le processus de filtrage

Enfin, nous avons dressé à la fin de ces expérimentations une étude comparative entre notre modèle de filtrage et les modèles présentés dans TREC-2002. L'objectif de la comparaison est de confronter nos résultats à ceux présentés par les participants de TREC-2002. Le premier résultat important que l'on peut tirer de cette comparaison est que les performances de notre modèle de filtrage sont meilleures que tous les modèles présentés à l'édition TREC-2002. Un second résultat qui découle du premier concerne les performances de notre modèle vis-à-vis du modèle initial, présent à TREC-2002. Ces performances ont été améliorées d'environ 30%.

Les perspectives envisageables à nos travaux portent essentiellement sur plusieurs points :

- élargir le contenu du profil de l'utilisateur pour prendre en compte d'autres éléments informationnels tels que ses préférences, ses habitudes de recherche, son environnement etc. Ceci permet d'intégrer davantage l'utilisateur dans le processus de filtrage personnalisé ;
- identifier l'information nouvelle vis-à-vis de l'information redondante, c'est-à-dire déjà vues, dans le flot d'informations. Une information nouvelle devrait apporter de nouveaux éléments par rapport aux informations déjà filtrées. Cette information doit être prise en compte de manière spécifique dans le processus d'apprentissage.
- optimiser les processus d'apprentissage, c'est-à-dire arriver à détecter l'instant de convergence du système. Nous avons constaté dans nos expérimentations que les valeurs d'utilité tendent à se stabiliser après une période donnée du processus filtrage.

# Bibliographie

- [Allan, 1996] Allan, J. (1996). Incremental relevance feedback for information filtering. In *Proceedings of the 19th annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 270–278.
- [Anderson and Pérez-Carballo, 2001] Anderson, J. and Pérez-Carballo, J. (2001). The nature of indexing : how humans and machines analyze messages and texts for retrieval : part ii : machine indexing, and the allocation of human versus machine effort. In *Information Processing and Management 37*, pages 255–277. Tarrytown, NY, USA, Pergamon Press, Inc.
- [Arampatzis et al., 2000] Arampatzis, A., Beney, J., Koster, C., and Van-Der-Weide, T. (2000). Incrementality, half-life, and threshold optimization for adaptive document filtering. In *Proceedings of the 9th Text REtrivel Conference (TREC-9)*. NIST, Gaithersburg, Maryland.
- [Arampatzis and Hameren, 2001] Arampatzis, A. and Hameren, A. V. (2001). The score-distributional threshold optimization for adaptive binary classification tasks. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 589. New Orleans Louisiana.
- [Ault and Yang, 2000] Ault, T. and Yang, Y. (2000). knn at trec9 : A failure analysis. In *Proceedings of the 9th Text REtrivel Conference (TREC-9)*. NIST, Gaithersburg, Maryland.
- [Baclace, 1992] Baclace, P. (1992). Competitive agents for information filtering. *Communications of the ACM*, 35(12).
- [Balabanovic, 1998] Balabanovic, M. (1998). An interface for learning multi-topic user profiles from implicit feedback. In *Recommender Systems, Paper from the 1998 Workshop*, pages 6–10. Madison, WI. Menlo Park.
- [Belkin and Croft, 1992] Belkin, N. and Croft, W. (1992). Information retrieval and information filtering : two sides of the same coin? *Communications of the ACM*, 35(12).

- [Bennett, 2003] Bennett, P. (2003). Using asymmetric distributions to improve text classifier probability estimates. In *Proceedings of the 26th annual international ACM SIGIR Conference on Research and development in Information Retrieval*. Toronto, Canada.
- [Berrut and Denos, 2003] Berrut, C. and Denos, N. (2003). *Filtrage collaboratif*. Assistance intelligente à la recherche d'informations, Hermes - Lavoisier, chapitre 8.
- [Boughanem, 1992] Boughanem, M. (1992). *Systèmes de recherche d'informations : d'un modèle classique à un modèle connexionniste*. Phd, Université Paul Sabatier de Toulouse.
- [Boughanem, 2000] Boughanem, M. (2000). *Formalisation et spécification des systèmes de recherche et de filtrage d'information*. Hdr, Univ. Paul Sabatier de Toulouse.
- [Boughanem et al., 1999a] Boughanem, M., Chrisment, C., and Soule-Dupuy, C. (1999a). Query modification based on relevance back-propagation in ad-hoc environment. *Information processing and Management*, 35(12) :121–139.
- [Boughanem et al., 1999b] Boughanem, M., Chrisment, C., and Tamine, L. (1999b). Query space exploration based on genetic algorithms. *Information Retrieval Journal*.
- [Boughanem et al., 2001] Boughanem, M., Chrisment, C., and Tmar, M. (2001). Mercure and mercurefiltre applied for web and filtering tasks at trec-10. In *Proceedings of the 10th Text REtrieval Conference (TREC-10)*. NIST, Gaithersburg, Maryland.
- [Boughanem and Tamine, 2004] Boughanem, M. and Tamine, L. (2004). *Connexionnisme et génétique pour la recherche d'information*. Dans : Les systèmes de recherche d'informations, (Eds.), Hermes-Lavoisier, Lavoisier.
- [Boughanem et al., 2002] Boughanem, M., Tebri, H., and Tmar, M. (2002). Irit at trec 2002 : Filtering track. In *Proceedings of the 11th Text REtrieval Conference (TREC-11)*. NIST, Gaithersburg, Maryland.
- [Boughanem et al., 2004a] Boughanem, M., Tebri, H., and Tmar, M. (2004a). Apprentissage incrémental des profils dans un système de filtrage d'information. In *4èmes journées d'Extraction et de Gestion des Connaissances*. Clermont Ferrand.
- [Boughanem et al., 2004b] Boughanem, M., Tebri, H., and Tmar, M. (2004b). Apprentissage par renforcement dans un système de filtrage adaptatif. In *CONFérence en Recherche d'Information et Application (CORIA'2004)*, pages 10–12. Toulouse.
- [Boughanem et al., 2004c] Boughanem, M., Tmar, M., and Tebri, H. (2004c). *Méthodes avancées pour les systèmes de recherche d'informations*. Traité des sciences et techniques de l'information, Hermes sciences Lavoisier.
- [Bourne and Anderson, 1979] Bourne, C. and Anderson, B. (1979). Dialog : Labworkbook. In *Lockheed Information Systems*, pages 640–644. PaloAlto, Californie (USA).

- [Bouzeghoub et al., 2004] Bouzeghoub, M., Berrut, C., Boughanem, M., Doucet, A., and Rumpler, B. (2004). Action spécifique sur la personnalisation de l’information. In *Rapport CNRS-AS98/RTP9*.
- [Bowen et al., 1992] Bowen, T., Gopal, G., Harman, G., Hickey, T., Lee, K., Mansfield, W., Raitz, J., and Weiribnrib, A. (1992). The datacycle architecture. *Communications of the ACM*, 35(12) :71–80.
- [Buckley and Salton, 1985] Buckley, C. and Salton, G. (1985). Optimization of relevance feedback weights. In *Proceedings of the 8th annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 351–357. New York, Association for Computing Machinery.
- [Callan, 1995] Callan, J. (1995). Passage-level evidence in document retrieval. In *Proceedings of the 18th annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 302–310.
- [Callan, 1998] Callan, J. (1998). Learning while filtering documents. In *Proceedings of the 21st annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 224–231.
- [Callan et al., 1992] Callan, J., Croft, W., and Harding, S. (1992). The inquiry retrieval system. In Springer-Verlag, editor, *Proceedings of the 3rd International Conference on Database and Expert Systems Applications*, pages 78–83. Valencia, Spain.
- [Crestani, 1994] Crestani, F. (1994). Comparing neural and probabilistic relevance feedback in an interactive information retrieval system. In *Proceedings of the 1994 IEEE International Conference on Neural Networks*, pages 3426–3430. Orlando, Florida, USA.
- [Croft and Xu, 1995] Croft, W. and Xu, J. (1995). Corpus-specific stemming using word from co-occurrence. In *Proceedings of the fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR95)*, pages 147–159.
- [Deerwester et al., 1990] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latex semantic analysis. *Journal of the American Society for Information Science*, 41(6) :391–407.
- [Denning, 1982] Denning, P. (1982). Electronic junk. *Communications of the ACM*, 32(3) :163–165.
- [Dumais et al., 1996] Dumais, S., Landauer, T., and Littman, M. (1996). Automatic cross-linguistic information retrieval using latent semantic indexing. In *SIGIR’96 - Workshop on Cross-Linguistic Information Retrieval*, pages 16–23.

- [Dunning, 1993] Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, pages 61–74.
- [Foltz and Dumais, 1992] Foltz, P. and Dumais, S. (1992). Personalized information delivery : An analysis of information filtering methods. *Communications of the ACM*, 35(12) :51–60.
- [Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *ACM SIGIR Forum*, 12(35) :61–70.
- [Goldberg et al., 2000] Goldberg, K., Roeder, T., Huptan, D., and Perkins, C. (2000). Eigentaste : A constant time collaborative filtering algorithm. In *Technical Report M00/41, IEOR and EECS Departments*. UC Berkeley.
- [Harman, 1992a] Harman, D. (1992a). The darpa tipster project. *ACM SIGIR Forum*, 2(26) :26–28.
- [Harman, 1992b] Harman, D. (1992b). Overview of the first text retrieval conference (trec-1),. In *Proceedings of the 1st Text REtrieval Conference (TREC-1)*. NIST, Gaithersburg, Maryland.
- [Harman, 1992c] Harman, D. (1992c). Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 1–10. Copenhagen, Denmark.
- [Harman, 2000] Harman, D. (2000). What we have learned, and not learned, from trec. In *Proceedings of BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*.
- [Hiemstra, 1998] Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *Second European Conference, ECDL'98*, pages 35–41.
- [Hiemstra, 2002] Hiemstra, D. (2002). Term specific smoothing for the language modeling approach to information retrieval : The importance of a query term. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–41. Finland.
- [Hirschman, 1991] Hirschman, L. (1991). Comparing muck-ii and muc-3 : Assessing the difficulty of different tasks. In *Proceedings, Third Message Understanding Conference (MUC-3)*, pages 25–30. DARPA, Morgan Kaufmann.
- [Hoashi et al., 1999] Hoashi, K., Matsumoto, K., Inoue, N., and Hashimoto, K. (1999). Experiments on the trec-8 filtering track. In *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, pages 457–463. NIST, Gaithersburg, Maryland.

- [Hoashi et al., 2000] Hoashi, K., Matsumoto, K., Inoue, N., and Hashimoto, K. (2000). Document filtering method using non-relevant information profile. In *Proceedings of the 23rd annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 176–183.
- [Housman, 1969] Housman, E. (1969). Survey of current systems for selective dissemination of information. In *Technical Report SIG/SDI-1, American Society for Information Science Special Interest Group on SDI*, pages 176–183. Washington.
- [Ide, 1971] Ide, E. (1971). New experiments in relevance feedback. In System, T. S. R., editor, *The SMART Retrieval System*, pages 337–354. G. Salton.
- [Kim et al., 2000] Kim, Y., Hahn, S., and Zhang, B. (2000). Text filtering by boosting naive bayesian classifiers. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 294–302.
- [Kohonen, 1989] Kohonen, T. (1989). Self-organization and associative memory. In *Springer Verlag*. ISBN 0387513876.
- [Kohonen et al., 1996] Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., and Torkkola, K. (1996). *LVQ PAK : The learning vector quantization program package*. Technical report, Helsinki University of Technology.
- [Kohonen et al., 2000] Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., and Saarela, A. (2000). Self organization of a massive text document collection. In *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, pages 574–585. ISBN 0387513876.
- [Kraaij, 2004] Kraaij, W. (2004). *Variations on Language Modeling for Information Retrieval*. Phd thesis, University of Twente.
- [Kwok, 1989] Kwok, K. (1989). A neural network for probabilistic information retrieval. In *Proceedings of ACM SIGIR*, pages 21–30.
- [Kwok et al., 2000] Kwok, K., Grunfeld, L., and Dinstl, N. (2000). Trec-9 cross language, web and question-answering track experiments using pircs. In *Proceedings of the 10th Text REtrieval Conference (TREC-9)*. Gaithersburg, Maryland.
- [Lelu and François, 1992] Lelu, A. and François, C. (1992). Information retrieval based on neural unsupervised extraction of thematic fuzzy clusters. In *Fifth International Conference, Neural Networks and their Applications : NEURO NIMES*, pages 93–104.
- [Lewis, 1995] Lewis, D. (1995). Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th ACM SIGIR on Research and Development in Information Retrieval*, pages 246–254.

- [Loeb, 1992] Loeb, S. (1992). Architecting personalized delivery of multimedia information. *Communications of the ACM*, 35(12) :39–48.
- [Luhn, 1957] Luhn, H. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM*, 1(4) :309–317.
- [Luhn, 1958] Luhn, H. (1958). A business intelligent system. *IBM Journal of Research and Development*, 2(4) :390–402.
- [Mackay et al., 1989] Mackay, W. E., Malone, T. W., Growston, K., Rao, R., Rosenlitt, D., and Card, S. K. (1989). How do experienced information lens use rules. In *Proceedings of SIGCHI*, pages 211–216.
- [Malone et al., 1987] Malone, T., Grant, K., Turbak, F., Brobst, S., and Cohen, M. (1987). Intelligent information sharing systems. *Communications of the ACM*, 30(5) :390–402.
- [Manning and Schütze, 1999] Manning, C. and Schütze, H. (1999). Chapter 6 : Statistical estimation : n-gram models over sparse data. In *Foundations of Statistical Natural Language Processing*. MIT Press.
- [Maron and Kuhns, 1960] Maron, M. and Kuhns, J. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the Association for Computing Machinery*, 7 :216–244.
- [Milic-Frayling et al., 1997] Milic-Frayling, N., Zhai, C., Tong, X., Jansen, P., and Evans, D. (1997). Experiments in query optimization, the clarit system trec-6 report. In *Proceedings of the 6th Text REtrieval Conference (TREC-6)*, pages 415–454. NIST, Gaithersburg, Maryland.
- [Miller et al., 1997] Miller, B., Riedl, J., and Konstan, J. (1997). Experiences with groupLens : making usenet useful again. In *Proceedings of the 1997 USENIX Winter Technical Conference*. Anahem, CA.
- [Mizzaro, 1997] Mizzaro, S. (1997). Relevance : The whole history. *Journal of the American Society for Information Science*, 49(9) :810–832.
- [Morita and Shinoda, 1994] Morita, M. and Shinoda, Y. (1994). Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 272–281. Dublin, Ireland.
- [Mothe, 1994] Mothe, J. (1994). *Modèle connexionniste pour la recherche d'information, expansion dirigée de requête et apprentissage*. Phd, Université Paul Sabatier de Toulouse.
- [Mothe, 2000] Mothe, J. (2000). *Recherche et exploration d'information Découverte de connaissance pour l'accès à l'information*. Hdr, Université Paul Sabatier de Toulouse.

- [Nichols, 1997] Nichols, D. (1997). Implicit ratings and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36. Budapest.
- [Oard and Marchionini, 1996] Oard, D. and Marchionini, G. (1996). *A conceptual framework for text filtering*. Report ee-tr-96-25, Université de Maryland.
- [Ponte and Croft, 1998] Ponte, J. and Croft, W. (1998). A language modelling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 40–48.
- [Porter, 1980] Porter, M. (1980). An algorithm for suffix stripping. *Program 14*, 1(3) :130–137.
- [Qui and Frei, 1993] Qui, Y. and Frei, H. P. (1993). Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169.
- [Ram, 1992] Ram, A. (1992). Natural language understanding for information filtering systems. *CACM*, 35(12) :80–81.
- [Rijsbergen, 1979] Rijsbergen, C. V. (1979). Information retrieval. In *Information retrieval experiments*. Butterworths, London, 2nd edition.
- [Rijsbergen, 1981] Rijsbergen, C. V. (1981). Retrieval effectiveness. In *Information retrieval experiments*. Butterworths, London, Karen Sparck-Jones.
- [Rijsbergen and Sparck-Jones, 1973] Rijsbergen, C. V. and Sparck-Jones, K. (1973). A test for the separation of relevant and non-relevant documents in experimental retrieval collections. In *Journal of Documentation 29*, pages 251–257.
- [Robertson, 1977] Robertson, S. (1977). The probability ranking principle in ir. In *Journal of Documentation*, pages 294–304.
- [Robertson, 1990] Robertson, S. (1990). On term selection for query expansion. *Journal of Documentation 46*, pages 359–364.
- [Robertson and Sparck-Jones, 1976] Robertson, S. and Sparck-Jones, K. (1976). Relevance weighting of search terms. *JASIS*, 27(3) :129–146.
- [Robertson and Sparck-Jones, 1997] Robertson, S. and Sparck-Jones, K. (1997). *Simple, proven approaches to text retrieval*. Tech. rep. tr356, Computer Laboratory University of Cambridge.
- [Robertson and S.Walker, 2000] Robertson, S. and S.Walker (2000). Threshold setting in adaptive filtering. *Journal of documentation*, 56 :312–331.

- [Robertson et al., 1998] Robertson, S., Walker, S., , and Beaulieu, M. (1998). Okapi at trec-7 : automatic ad hoc, filtering, vlc and interactive track. In *Proceedings of the 7th Text REtrivel Conference (TREC-7)*. NIST, Gaithersburg, Maryland.
- [Robertson and Walker, 1994] Robertson, S. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighting. In *Proceedings of ACM SIGIR*, pages 232–241.
- [Robertson and Walker, 1999] Robertson, S. and Walker, S. (1999). Okapi/keenbow at trec-8. In *Proceedings of the 8th Text REtrival Conference (TREC-8)*. NIST, Gaithersburg, Maryland.
- [Robertson and Walker, 2000] Robertson, S. and Walker, S. (2000). Microsoft cambridge at trec-9 : Filtering track. In *Proceedings of the 9th Text REtrival Conference (TREC-9)*. NIST, Gaithersburg, Maryland.
- [Robertson et al., 1994] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at trec-3. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, pages 109–126. NIST, Gaithersburg, Maryland.
- [Robertson et al., 2002] Robertson, S., Walker, S., Zaragoza, H., and Herbrich, R. (2002). Microsoft cambridge at trec 2002 : Filtering. In *Proceedings of the 11th Text REtrivel Conference (TREC-11)*. NIST, Gaithersburg, Maryland.
- [Rocchio, 1966] Rocchio, J. (1966). *Document retrieval Systems-Optimization and evaluation*. Phd thesis, Harvard Computational Laboratory.
- [Rocchio, 1971] Rocchio, J. (1971). Relevance feedback in information retrieval. In *The SMART retrieval system - experiments in automatic document processing*, pages 313–323. Prentice Hall Inc.
- [Salton, 1971] Salton, G. (1971). *The SMART Retrieval System - Experiment in Automatic Document Processing*. Englewood Cliffs, NJ : Prentice-Hall.
- [Salton, 1983] Salton, G. (1983). *Introduction to modern information retrieval*. New York, McGraw-Hill.
- [Salton, 1989] Salton, G. (1989). *Automatic text processing : The transformation, analysis and retrieval of information by computer*. Addison-Wesley publishing, MA.
- [Salton et al., 1983] Salton, G., Fox, E., and Wu, H. (1983). Extended boolean information retrieval. *Communications of the ACM*, 26(11) :1022–1036.
- [Salton and Yang, 1973] Salton, G. and Yang, C. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*, 29 :351–372.

- [Saporta, 1990] Saporta, G. (1990). *Probabilités analyse des données et statistiques*. Editions Technip, Editions Technip Paris.
- [Schapire et al., 1998] Schapire, R., Singer, Y., and Singhal, A. (1998). Boosting and rocchio applied to text filtering. In *Proceedings of the 21st annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 215–223.
- [Shardanand, 1994] Shardanand, U. (1994). *Social Information Filtering for Music Recommendation*. Master’s thesis, Department of Electronical Engineering and Computer Science.
- [Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering : algorithms for automating ’word of mouth’. In *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems*, pages 210–217. New York.
- [Sheridan and Smeaton, 1992] Sheridan, P. and Smeaton, A. (1992). The application of morpho-syntactic language processing to effective phrase matching. *Information Processing and Management*, 28(3) :349–370.
- [Singhal et al., 1997] Singhal, A., Mitra, M., and Buckley, C. (1997). Learning routing queries in a query zone. In *Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32. Philadelphia.
- [Song and Croft, 1999] Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–280.
- [Stadnyk and Kass, 1992] Stadnyk, I. and Kass, R. (1992). Modeling user’s interests in information filters. *Communications of the ACM*, 35(12) :49–50.
- [Stevens, 1992] Stevens, C. (1992). *Knowledge-Based Assistance for Accessing Large, Poorly Structured Information Spaces*. Phd thesis, University of Colorado, Department of Computer Science, Boulder.
- [Tamine, 2000] Tamine, L. (2000). *Optimisation de requêtes dans un système de recherche d’information*. Phd, Université Paul Sabatier de Toulouse.
- [Tebri et al., 2005] Tebri, H., Boughanem, M., Chrisment, C., and Tmar, M. (2005). Incremental profile learning based on a reinforcement method. In *The 20th ACM Symposium on Applied Computing, SAC2005(à paraître)*. Santa Fe, New Mexico, USA, ACM.
- [Tmar, 2002] Tmar, M. (2002). *Modèle auto-adaptatif de filtrage d’information : apprentissage incrémental du profil et de la fonction de décision*. Phd, Université Paul Sabatier de Toulouse.

- [Tmar et al., 2002] Tmar, M., H.Tebri, and Boughanem, M. (2002). Détection de convergence en vue de l'optimisation d'un système de filtrage adaptatif. In *9ème journées d'études sur les systèmes d'information élaborée : Bibliométrie - Informatique stratégique - Veille technologique*. Ile-Rousse (Corse).
- [Turtle, 1991] Turtle, H. (1991). *Inference Networks for Document Retrieval*. Phd thesis, University of Massachusetts.
- [Voorhees, 2002] Voorhees, E. (2002). Overview of the trec 2002. In *Proceedings of the 11th Text REtrivel Conference (TREC-11)*. NIST, Gaithersburg, Maryland.
- [Wang et al., 2001] Wang, B., Xu, H., Yang, Z., Liu, Y., Cheng, X., Bu, D., and Bai, S. (2001). Trec-10 experiments at cas-ict : Filtering, web and qa. In *Proceedings of the 10th Text REtrivel Conference (TREC-10)*. NIST, Gaithersburg, Maryland.
- [Wong et al., 1985] Wong, S., Ziarko, W., and Wong, P. (1985). Generalized vector space model information retrieval. In *Proceedings of the 8th annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 18–25. ACM.
- [Wu et al., 2001] Wu, L., Huang, X., Niu, J., Xia, Y., and Feng, Z. (2001). Fdu at trec-10 : Filtering, qa, web and video tasks. In *Proceedings of the 10th Text REtrivel Conference (TREC-10)*. NIST, Gaithersburg, Maryland.
- [Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and control* 8, pages 338–353.
- [Zhai et al., 1998] Zhai, C., Jansen, P., Stoica, E., Grot, N., and Evans, D. (1998). Threshold calibration in clarit adaptive filtering. In *Proceedings of the 7th Text REtrivel Conference (TREC-7)*, pages 149–156. NIST, Gaithersburg, Maryland.
- [Zhang, 2004] Zhang, Y. (2004). Using bayesian priors to combine classifiers for adaptive filtering. In *Proceedings of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 354–352. Sheffield UK.
- [Zhang and Callan, 2001] Zhang, Y. and Callan, J. (2001). Maximum likelihood estimation for filtering thresholds. In *Proceedings of the 24th annual international ACM SIGIR Conference on Research and development in Information Retrieval*, pages 294–302.
- [Zipf, 1949] Zipf, G. (1949). *Human Behaviour and the Principle of Least Effort*. 1st edition, Addison Wesley.