

---

# Apprentissage par Renforcement dans un système de filtrage adaptatif

**M. Boughanem<sup>\*</sup>, H.Tebri<sup>\*</sup>, M.Tmar<sup>\*\*</sup>**

<sup>\*</sup> *IRIT-SIG*

*Université Paul Sabatier de Toulouse*

*118, route de Narbonne*

*F-31062 Toulouse Cedex 4*

*{bougha,tebri}@irit.fr*

<sup>\*\*</sup> *ISIMS, Université de Sfax*

*M'Harza 3000 Sfax (Tunisie)*

*mohamedtmar@yahoo.fr*

---

**RÉSUMÉ.** Cet article présente une méthode incrémentale d'apprentissage des profils dans les systèmes de filtrage d'information. Cette méthode est basée sur le principe de renforcement. L'idée de base consiste à construire, à chaque arrivée d'un document pertinent, un profil " provisoire " permettant de sélectionner le document en question avec un score " fort ", puis intégrer ce profil, grâce à une descente de gradient, dans le profil global. Cette méthode est comparée à une version incrémentale de l'algorithme de Rocchio adapté au filtrage d'information. Des expérimentations effectuées sur la collection de test de référence TREC<sup>1</sup>, montrent que les résultats obtenus par la méthode de renforcement sont plus intéressants que ceux obtenus par l'algorithme de Rocchio.

**ABSTRACT.** This paper presents a profile learning method in the information filtering systems. This method is based on a reinforcement process. The basic idea consists to build, when a document is judged as relevant, the provisional profile which makes it possible to find this document with a strong score, then integrate this profile by using the gradient propagation formula in the global profile. The reinforcement method is compared to Rocchio learning algorithm which is adapted to information filtering. Experiments carried out on TREC-2002 collection showed the effectiveness of the reinforcement process.

**MOTS-CLÉS :** filtrage adaptatif, profil, apprentissage par renforcement, algorithme de Rocchio.

**KEYWORDS:** Adaptive filtering, profile, reinforcement learning, Rocchio algorithm.

---

---

1. Text REtrieval Conference

## 1. Introduction

Le filtrage d'information est un processus permettant, à partir d'un flot de documents provenant d'une source dynamique d'information (Internet, News, E-mail, etc.), de sélectionner ceux qui satisfont les besoins relativement stables des utilisateurs, appelés *profils* [BEL 92].

Compte tenu de la dualité entre la Recherche d'information (RI) et le Filtrage d'Information (FI), comme le montre Belkin dans [BEL 92], bon nombre de modèles de filtrage d'information sont basés sur des modèles de recherche d'information augmentés par une fonction de décision, le plus souvent de type seuil. D'une façon générale, les documents et les profils sont représentés par des listes de mots pondérés. Le filtrage d'information revient à comparer chaque document, qui arrive dans le système aux différents profils. Ceci consiste à mesurer un score de similarité entre le document et le profil. Si le score est supérieur au seuil le document est accepté sinon il est rejeté. Or, la problématique majeure en FI vient du fait qu'en l'absence de collection de référence, la détermination de ce seuil et des pondérations adéquates associées aux profils et aux documents est pratiquement impossible. Car, dans un système de filtrage d'information (SFI), au démarrage du processus de filtrage on ne dispose d'aucune connaissance sur les documents à filtrer.

La solution adoptée, dans la majorité des travaux actuels, consiste à démarrer le processus de filtrage en initialisant le profil avec une liste de mots clés extraits du texte du profil et le seuil à une valeur arbitraire, puis adapter et apprendre le seuil et le profil au fur et à mesure que les documents arrivent. Cette approche est appelée filtrage incrémental ou *Adaptive filtering* dans la terminologie TREC [VOO 01].

La majorité des techniques d'adaptation de profil proposées, sont inspirées du principe de reformulation de requêtes. Les techniques utilisées sont principalement basées sur une version incrémentale de l'algorithme de Rocchio [ROC 71], on y trouve notamment les travaux de [CAL 98] [SCH 98], ou des techniques basées sur les classifieurs Bayesiens [KIM 00], les réseaux de neurones [KWO 01] et des techniques génétiques [BOU 99].

Concernant le seuillage, les différentes méthodes proposées tentent de définir un seuil qui permet d'optimiser une fonction d'utilité. Une fonction d'utilité permet de mesurer la capacité d'un système de FI à ne sélectionner que des documents pertinents [VOO 01].

Nous nous intéressons dans cet article à l'apprentissage du profil dans un cadre purement incrémental. Contrairement à ce qui se fait dans les autres travaux, qui souvent utilisent des collections d'apprentissage ou effectuent l'apprentissage sur un lot de documents, nous proposons une méthode incrémentale pour l'apprentissage de profils, appelée apprentissage par renforcement. Pour évaluer l'intérêt de cette méthode, nous la comparons à l'un des algorithmes les plus utilisés dans le domaine de la RI, l'algorithme de Rocchio [ROC 71].

Cet article est organisé comme suit : la section 2 décrit l'algorithme de reformulation de requêtes développé par Rocchio. Dans la section 3, nous présentons notre modèle de base du système de filtrage d'information et la méthode d'apprentissage des profils basée sur le principe de renforcement. Enfin la section 4, est consacrée aux expérimentations et aux résultats obtenus par les deux méthodes. Les expérimentations sont effectuées sur une collection de test de référence TREC-2002 [VOO 02].

## 2. Algorithme de Rocchio

L'algorithme de reformulation de requêtes développé par Rocchio au milieu des années 60 [ROC 66] [ROC 71], est l'un des algorithmes d'apprentissage 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 :

$$\vec{Q}_{nle} = \alpha \vec{Q}_{init} + \beta \frac{1}{R} \sum_{d \in E_r} \vec{d} - \gamma \frac{1}{N - R} \sum_{d \notin E_r} \vec{d} \quad (1)$$

Où :

$\vec{Q}_{init}$  : le vecteur de la requête initiale, représenté par une liste de termes pondérés,

$\vec{Q}_{nle}$  : le vecteur de la nouvelle requête,

$E_r$  (resp.  $E_{nr}$ ) : l'ensemble des documents jugés pertinents (resp. non pertinents),

$R = |E_r|$  : le nombre de documents pertinents,

$N$  : le nombre de documents total (avec  $N = |E_r| + |E_{nr}|$ ),

$\alpha, \beta, \gamma$  : des paramètres constants, ils permettent de contrôler respectivement la contribution de la requête initiale, des documents pertinents et non pertinents dans le processus de construction de la nouvelle requête.

Une version incrémentale de l'algorithme de Rocchio a été utilisée pour l'apprentissage de profils. On y trouve notamment les travaux de [CAL 98] [SCH 98], Query Zoning (ou QZ) développé par Singhal et al. [SIN 97] et Dynamic Query Zoning (ou DQZ) [BUC 95].

Dans nos différentes expérimentations, nous comparons notre méthode d'apprentissage à une version incrémentale de l'algorithme de Rocchio tel qu'il est utilisé dans " Query Zoning ". La version incrémentale de QZ consiste à construire les deux ensembles,  $E_r$  et  $E_{nr}$ , à partir des documents sélectionnés et jugés pertinents par l'utilisateur à l'instant  $t$ . Le processus de d'apprentissage (construction du nouveau profil) est déclenché à chaque arrivée d'un document pertinent.

## 3. Apprentissage par renforcement

Avant de commencer à présenter la méthode d'apprentissage par renforcement, nous décrirons tout d'abord le modèle de filtrage de base utilisé dans cet article.

### 3.1. le modèle de filtrage de base

Le modèle de filtrage que nous utilisons est basé sur une approche vectorielle. Les documents et les profils sont représentés sous forme d'une liste de termes pondérés.

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

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 :  $w_i^{(0)} = t f p_i * (\max_j (t f p_j))^{-1}$ , où  $t f p_i$  est la fréquence du terme  $tp_i$  dans le profil. Ce poids sera ajusté par apprentissage à chaque fois un document est sélectionné et jugé pertinent.

A chaque arrivée d'un document  $d_i^{(t)}$ , celui-ci est indexé. Le résultat de cette opération est une liste de termes pondérés. Le poids de chaque terme  $d_{i_k}^{(t)}$  dans le document  $d_i^{(t)}$  est calculé par une fonction de pondération utilisée dans le système de recherche d'information **Mercure** [BOU 00] :

$$d_{i_k}^{(t)} = \frac{t f_k^{(t)}}{h_3 + h_4 \frac{\Delta l^{(t)}}{\Delta l^{(t)}} + t f_k^{(t)}} \log\left(\frac{N^{(t)}}{n_k^{(t)}} + 1\right) \quad (2)$$

Où :

$t f_k^{(t)}$  : fréquence du terme  $t_k$  dans le document  $d_i^{(t)}$ , à l'instant  $t$ ,

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

$\Delta l^{(t)}$  : nombre de termes dans le document  $d_i^{(t)}$ ,

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

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

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

Le filtrage consiste à mesurer un score, noté  $rsv(d_i^{(t)}, p^{(t)})$ , entre le document et le profil. Ce score est défini par le produit scalaire entre le document  $d_i^{(t)}$  et le profil  $p^{(t)}$ . Ensuite, ce score est comparé à un seuil ( $\theta$ ) de filtrage, pour décider si le document est accepté ou non : si  $rsv(d_i^{(t)}, p^{(t)}) \geq \theta^{(t)}$  alors le document  $d_i^{(t)}$  est sélectionné, sinon il est rejeté. Le profil et les statistiques liées à la pondération des termes des documents sont appris à chaque arrivée d'un document pertinent.

Nous présentons dans la section suivante la méthode d'apprentissage par renforcement. Nous ne détaillons pas la technique d'adaptation du seuil, les lecteurs intéressés peuvent se référer à [Tmar 02].

### 3.2. Apprentissage des profils par renforcement

L'apprentissage des profils que nous utilisons est basé sur un principe de renforcement [SUT 98]. A cet effet, nous considérons que quand un document  $d_i^{(t)}$  est sélectionné et jugé pertinent, il faut trouver une représentation du profil  $p_x^{(t)}$  qui permet de retrouver ce document avec un score " fort ", soit  $\lambda$ , puis intégrer cette solution pour apprendre le profil global de l'utilisateur. Ceci revient donc à trouver le profil tel que  $rsv(d_i^{(t)}, p_x^{(t)}) = \lambda$ . Autrement dit, il faut chercher les  $pw_j^{(t)}$ , poids du terme  $tp_j$  dans le profil  $p_x^{(t)}$ , qui satisfont l'équation suivante :

$$\sum_{t_k \in d_i^{(t)}, tp_j \in p_x^{(t)}, t_k = tp_j} d_{i_k}^{(t)} \cdot pw_j^{(t)} = \lambda \quad (3)$$

Cette équation admet évidemment une infinité de solutions. Pour pallier ce problème, nous proposons d'ajouter une contrainte pour 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 à intégrer est la suivante : compte tenu de l'aspect incrémental de l'apprentissage, nous considérons que les termes du profil solution de l'équation [3] 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  $tp_i$  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 à résoudre devient alors :

$$\begin{cases} \sum_{t_k \in d_i^{(t)}, tp_j \in p_x^{(t)}, t_k = tp_j} d_{i_k}^{(t)} \cdot pw_j^{(t)} = \lambda \\ \forall (t_i, t_j) \in d_i^{(t)2}, \frac{pw_i^{(t)}}{f_i^{(t)}} = \frac{pw_j^{(t)}}{f_j^{(t)}} \end{cases} \quad (4)$$

La solution du système 4 est l'ensemble des poids du profil  $p_x^{(t)}$  qui permet de retrouver le document  $d_i^{(t)}$ .

Le système 4 peut être réécrit en :  $\forall i \in \{1 \dots n\}$

$$\left\{ \begin{array}{l} \sum_{t_k \in d_i^{(t)}, tp_j \in p_x^{(t)}, t_k = tp_j} d_{i_k}^{(t)} . pw_j^{(t)} = \lambda \\ \frac{pw_1^{(t)}}{f_1^{(t)}} = \frac{pw_i^{(t)}}{f_i^{(t)}} \\ \vdots \\ \frac{pw_n^{(t)}}{f_n^{(t)}} = \frac{pw_i^{(t)}}{f_i^{(t)}} \end{array} \right. \Leftrightarrow \begin{array}{l} d_{j_1}^{(t)} . pw_1^{(t)} = f_1^{(t)} d_{j_1}^{(t)} \frac{pw_i^{(t)}}{f_i^{(t)}} \\ \vdots \\ d_{j_n}^{(t)} . pw_n^{(t)} = f_n^{(t)} d_{j_n}^{(t)} \frac{pw_i^{(t)}}{f_i^{(t)}} \end{array} \quad (5)$$

où  $j_k$  correspond à l'index dans le document du terme indexé par  $k$  dans le profil ( $t_k = tp_{j_k}$ ).

En additionnant le premier opérande de chaque équation, on obtient :

$$\forall i \in \{1 \dots n\}, \sum_{k=1}^n d_{j_k}^{(t)} . pw_k^{(t)} = \sum_{k=1}^n f_k^{(t)} d_{j_k}^{(t)} \frac{pw_i^{(t)}}{f_i^{(t)}} \quad (6)$$

or  $\sum_{k=1}^n d_{j_k}^{(t)} . pw_k^{(t)} = \lambda$ , d'où :

$$\sum_{k=1}^n f_k^{(t)} d_{j_k}^{(t)} \frac{pw_i^{(t)}}{f_i^{(t)}} = \lambda \Leftrightarrow pw_i^{(t)} = \frac{\lambda f_i^{(t)}}{\sum_{k=1}^n f_k^{(t)} d_{j_k}^{(t)}} \quad (7)$$

Pour chaque terme du document, le poids solution du système 4 est le suivant :

$$\forall i, pw_i^{(t)} = \frac{\lambda f_i^{(t)}}{\sum_{k=1}^n f_k^{(t)} d_{j_k}^{(t)}} \quad (8)$$

Concernant le poids idéal, il est clair que nous n'avons aucun moyen théorique permettant de donner une mesure de ce poids. Notre but est de donner une fonction qui permet de l'estimer. Ainsi, sur la base de ce qui se fait dans le domaine de la recherche d'information, 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 pertinents sélectionnés, etc. Nous avons expérimenté plusieurs fonctions et avons opté pour une fonction dérivée de la formule de Robertson-Sparck Jones **BM25** [ROB 76] :

$$f(d_{i_k}^{(t)}, r_k^{(t)}, s_k^{(t)}) = d_{i_k}^{(t)} * \log\left(1 + \frac{r_k^{(t)}(S^{(t)} - s_k^{(t)})}{(s_k^{(t)} + 1)(R^{(t)} - r_k^{(t)} + 1)}\right) \quad (9)$$

Où  $R^{(t)}$  (resp.  $S^{(t)}$ ) représente le nombre de documents pertinents (resp. non pertinents) sélectionnés par le système à l'instant  $t$ .  $r_k^{(t)}$  (resp.  $s_k^{(t)}$ ) est le nombre de documents pertinents (resp. non pertinents) contenant le terme  $t_k$  à l'instant  $t$ .

Il faut noter que  $p_x^{(t)}$  est la solution pour sélectionner le document  $d_i^{(t)}$ . Il faut maintenant intégrer cette solution dans le profil global. Nous utilisons, pour cela, la formule de distribution de gradient suivante :

$$w_i^{(t+1)} = w_i^{(t)} + 0.1 * \log(1 + pw_i^{(t)})$$

## 4. Expérimentation et résultats

Dans cette section nous décrivons nos expérimentations et les résultats correspondants.

### 4.1. Collection de test

La collection de test que nous avons utilisée dans nos différentes expérimentations est issue de la tâche de filtrage de TREC-2002, plus précisément du corpus de test de Reuters entre 20 août 1996 et 19 août 1997. Ce corpus est constitué d'environ 810.000 documents. Nous avons utilisé les 50 premiers topics (101-150) créés par les assesseurs du NIST<sup>1</sup> dans nos différents tests. Ces topics sont utilisés pour construire les profils de tests initiaux. Trois documents d'entraînement supposés pertinents sont donnés pour initialiser le processus de filtrage (apprendre le profil et le seuil initial). L'adaptation du seuil n'est pas décrite dans cet article.

Le but de nos expérimentations est de comparer et d'évaluer la méthode d'apprentissage par renforcement à celle de Rocchio. La technique de seuillage basée sur la distribution de probabilités des scores des est utilisée dans chacune des deux méthodes pour apprendre la valeur du seuil [Tmar 02]. Le seuil est appris d'une manière incrémentale, à chaque sélection d'un document pertinent le processus d'adaptation du seuil est déclenché.

### 4.2. Méthodologie

#### 4.2.1. Application de la méthode d'apprentissage Rocchio

La version incrémentale de l'algorithme de Rocchio que nous avons utilisée dans notre expérimentation est décrite dans l'algorithme ci-dessous. Elle consiste d'une façon générale, à construire un nouveau profil, en appliquant la fonction 1, 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  $t$ . Seuls les 60 premiers termes de poids élevés sont retenus dans le nouveau profil.

---

1. National Institute of Standards and Technology –(<http://trec.nist.gov/>)

*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  $d^{(t)}$ 
  - calculer  $rsv(d^{(t)}, p^{(t)})$ ,
  - si  $rsv(d^{(t)}, p^{(t)}) > seuil^{(t)}$  alors
    - si  $d^{(t)}$  est pertinent alors
      - $E_r^{(t)} = E_r^{(t-1)} \cup \{d^{(t)}\}$ ,
      - appliquer Rocchio (équation [1], où  $R = |E_r^{(t)}|$  et  $N = |E_r^{(t)}| + |E_{nr}^{(t)}|$ ,
      - passer au document suivant,
    - sinon
      - $E_{nr}^{(t)} = E_{nr}^{(t-1)} \cup \{d^{(t)}\}$ ,
      - passer au document suivant,

#### 4.2.2. Application de la méthode par renforcement

L'application de la méthode d'apprentissage par renforcement, consiste tout d'abord à initialiser le score idéal  $\lambda$ . Différentes valeurs de  $\lambda$  ont été testées. Nous avons retenu  $\lambda = 200$ . Le processus d'apprentissage est déclenché à chaque sélection d'un document pertinent. Le processus d'apprentissage se résume comme suit :

*Pour un profil donné :*

- pour chaque document  $d^{(t)}$ 
  - calculer  $rsv(d^{(t)}, p^{(t)})$
  - si  $rsv(d^{(t)}, p^{(t)}) > seuil^{(t)}$  alors
    - si  $d^{(t)}$  est pertinent alors
      - pour chaque terme  $t_i$  dans  $d^{(t)}$  :  $r_i^{(t)} \leftarrow r_i^{(t-1)} + 1$
      - $R^{(t)} \leftarrow R^{(t-1)} + 1$  //nombre de documents pertinents sélectionnés
      - apprendre le profil
      - passer au document suivant
    - sinon
      - pour chaque terme  $t_i$  dans  $d^{(t)}$  :  $s_i^{(t)} \leftarrow s_i^{(t-1)} + 1$
      - $S^{(t)} \leftarrow S^{(t-1)} + 1$  //nombre de documents non pertinents sélectionnés
      - passer au document suivant

On tient à souligner que le nombre de termes qu'on retient dans chaque profil, appris par chacune des deux méthodes, est de 60 termes.



### 4.3. Evaluation des résultats obtenus

Pour évaluer les résultats obtenus par ces deux méthodes, nous avons utilisé la mesure d'utilité  $T11SU$  (Robertson 2002) suivante :

$$\begin{aligned}
 T11SU &= \frac{\max(T11NU, \min NU) - \min NU}{1 - \min NU} \\
 T11NU &= \frac{T11U}{\max NU} \\
 T11U &= 2R_+ - S_+ \\
 \min NU &= -0.5 \\
 \max NU &= 2 * R
 \end{aligned} \tag{10}$$

Où  $R$  est le nombre de documents total,  $R_+$  (resp.  $S_+$ ) le nombre documents pertinents (resp. non pertinents) sélectionnés.

Les tableaux (1) et (2) illustrent, pour des valeurs distinctes de  $\alpha$ ,  $\beta$  et  $\gamma$ , la moyenne de la fonction d'utilité  $T11SU$  sur les 50 profils de test. Dans la table (1) nous avons repris les valeurs testées par Singhal et al. (Singhal et al. 1997), dans leurs différentes expérimentations. On remarque que les meilleures utilités 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 utilité obtenue est donnée par  $\alpha = 0$ ,  $\beta = 2$  et  $\gamma = 1$ .

$\alpha = 8$			
	$\gamma = 16$	$\gamma = 32$	$\gamma = 64$
$\beta = 16$	0.2135	0.2137	0.2137
$\beta = 32$	0.2126	0.2129	0.2125
$\beta = 64$	0.2124	0.2127	0.2134

Tableau. 1 : Résultats de Rocchio ( $\alpha$  fixe)

$\alpha$	$\beta$	$\gamma$	$T11SU$
0	1	1	<b>0.3262</b>
0	2	1	<b>0.3460</b>
1	2	1	0.2922
1	2	2	0.2933
1	4	2	0.2914

Tableau. 2 : Résultats de Rocchio ( $\alpha$  varie)

Le tableau (3) présente, pour chacune des deux méthodes, la valeur d'utilité par chaque ensemble de profils, c'est-à-dire, nous calculons pour chaque 10 profils une valeur d'utilité moyenne. On remarque que les valeurs d'utilités obtenues par la méthode par renforcement sont nettement supérieures à celles obtenues par la méthode de Rocchio.

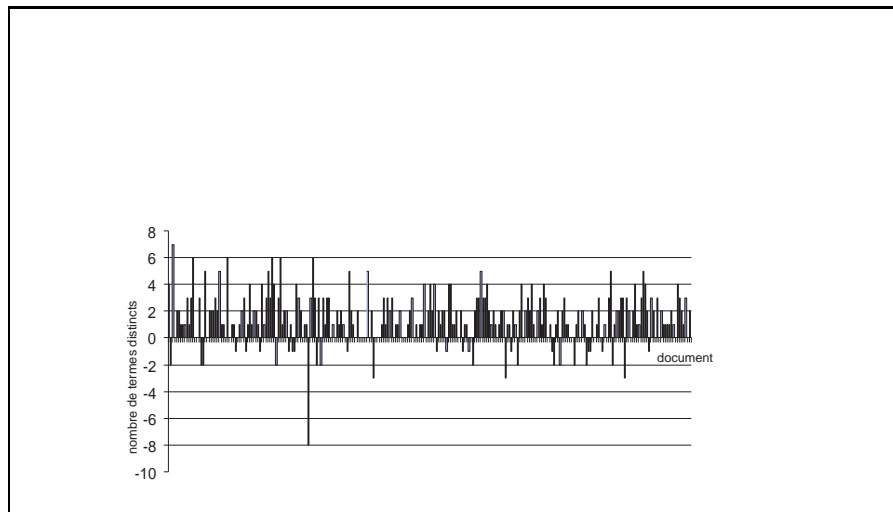
Nous avons également effectué une autre expérimentation pour vérifier si les deux techniques d'apprentissage permettent de construire les mêmes profils. Nous avons pour cela mesuré le nombre de termes communs entre le profil appris par la méthode de Rocchio et celui appris par la méthode de renforcement. 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 0.8784.

ensemble de profils	Rocchio	Renforcement
101 – 110	0.4126	0.4824
111 – 120	0.3240	0.3958
121 – 130	0.3433	0.4333
131 – 140	0.3247	0.4514
141 – 150	0.3258	0.4414
Totalités des profils (101-150)	0.3460	0.4409

Tableau 3. Les valeurs d'utilités [101-150]

La figure 1 présente le nombre de termes différents entre les deux profils. Ce nombre est calculé chaque fois les deux profils sont modifiés simultanément, en d'autres termes, chaque fois un même document pertinent est sélectionné par les deux profils.

Nous constatons que le nombre de termes distincts entre les deux profils varie en moyen 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 l'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 par ces deux méthodes.



**Figure 1.** *Le nombre de termes distincts entre deux profils*

## 5. Conclusion

Nous nous sommes intéressés dans cet article, plus particulièrement au problème d'adaptation incrémentale du profil. L'adaptation du profil est déclenchée à chaque

arrivée d'un document pertinent. A cet effet, nous avons proposé une méthode d'apprentissage basée sur le principe de renforcement. Ceci revient à résoudre une équation consistant à trouver le profil permettant de sélectionner ce document pertinent avec un score fort, puis intégrer ces solutions dans le profil global de l'utilisateur. Pour évaluer l'intérêt de cette méthode, nous l'avons comparé à une version incrémentale de l'algorithme de Rocchio. Des expérimentations réalisées sur une collection de test Reuters issue de TREC-2002, montrent que la méthode d'apprentissage par renforcement est plus efficace que l'algorithme de Rocchio.

## 6. Références

- [BEL 92] BELKIN N.J., CROFT W.B., « Information retrieval and information filtering : Two sides of the same coin ? », *CACM*, 1992, p. 29-38.
- [BOU 99] BOUGHANEM M., CHRISMENT C., TAMINE L., « Query space exploration based on genetic algorithms », *Information Retrieval Journal*, 1999.
- [BOU 00] BOUGHANEM M., « Formalisation et spécification des systèmes de recherche et de filtrage d'information », HDR de l'université Paul Sabatier de Toulouse, 2000, IRIT.
- [BUC 95] BUCKLEY C., SALTON G., « Optimization of relevance feedback », *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, 1995, In Edward Fox, Peter Ingwersen, and Raya Fidel, editors, Association for Computing Machinery, p. 351-357.
- [CAL 98] CALLAN J., « Learning while filtering documents », *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, p. 224-231.
- [KIM 00] KIM Y.H., HAHN S.Y., , ZHANG B.T., « Text filtering by boosting Naïve Bayes classifiers », *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000, ACM Press, p. 168-175.
- [KWO 01] KWOK K.L., GRUNFELD L., DINSTL N., CHAN M., « TREC-9 Cross Language, Web and Question-Answering Track Experiments using PIRCS », *Proceedings of TREC-9*, NIST, 2001, p. 417-426.
- [ROB 76] ROBERTSON S., SPARCK JONES K., « Relevance weighting of search terms », *JASIS*, vol. 27, n° 3, 1976, p. 129-146.
- [ROC 66] ROCCHIO J.J., « Document retrieval Systems-Optimization and evaluation », Phd thesis, Harvard Computational Laboratory, 1961, Cambridge.
- [ROC 71] ROCCHIO J.J., « Relevance feedback in information retrieval », *In The SMART retrieval system experiments in automatic document processing*, 1971, Prentice Hall Inc, p. 313-323.
- [SCH 98] SCHAPIRE R.E., SINGER Y., SINGHAL A. « Boosting and Rocchio applied to text filtering », *Proceedings of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, p. 215-223.
- [SIN 97] SINGHAL A., MITRA M., BUCKLEY C., « Learning routing query zone », *Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 1997, Association for Computing Machinery, p. 25-32.

- [SUT 98] SUTTON R.S., BARTO A.G., « Reinforcement learning : An introduction », *MIT Press*, Cambridge, MA, 1998.
- [Tmar 02] BRAUN T., TMAR M., ROCA V., « Modèle auto-adaptatif de Filtrage d'Information : Apprentissage incrémental du profil et de la fonction de décision », Thèse de l'Université Paul Sabatier de Toulouse, 2002, IRIT.
- [VOO 01] VOORHEES E.M., « Overview of TREC'10 », *The 10th Text REtrieval Conference*, 2001.
- [VOO 02] VOORHEES E.M., « Overview of TREC'11 », *The 11th Text REtrieval Conference*, 2002.