Un modèle comportemental d'interclassement de résultats dans un système de recherche d'information P2P

Rim Mghirbi [*,**] — **Khedija Arour** [*] — **Yahya Slimani** [*] — **Bruno Defude** [**]

[*] *Faculté des sciences de Tunis,Département des Sciences de l'informatique*
1060 Tunis, Tunisie

rim.mghirbi@laposte.net, yahya.slimani@fst.rnu.tn, Khedija.arour@issatm.rnu.tn
[**] *Institut de Télécom et Management Sud Paris, Département Informatique*
91011 Every Cedex, Paris, France

Bruno.Defude@it-sudparis.eu

RÉSUMÉ. *La Recherche d'Information Distribuée (RID) constitue aujourd'hui un domaine d'investigation en pleine effervescence. Afin de définir un processus de RID dans un système totalement distribué tel que les systèmes pair-à-pair, un intérêt particulier doit être porté à la phase de combinaison de résultats provenant de pairs autonomes, que nous appelons interclassement. Les approches classiques ne s'avèrent pas efficaces vu le manque de statistiques globales. Elles s'avèrent également trop généralistes pour s'adapter à des besoins particuliers. Toutes ces raisons nous ont incités à proposer un modèle d'interclassement de résultats qui tient compte des besoins de l'utilisateur (appelé modèle comportemental).*

ABSTRACT. *Distributed Information Retrieval (DIR) is already an important area of investigation due to the importance of ongoing access to relevant information. In a completely distributed system such as peer-to-peer systems, an important interest should be attributed to the rank aggregation phase. Classical approaches miss effectiveness due to lack of global statistics. This leads us to seek for a new aggregation model that takes into account the user needs (called behavioral model).*

MOTS-CLÉS : *Recherche d'information, Systèmes P2P, Interclassement de résultats, Profils utilisateurs*

KEYWORDS: *Information retrieval, P2P systems, Rank Aggregation, User profiles.*

1. Introduction au problème d'interclassement dans les systèmes pair-à-pair

Dans un processus de Recherche d'Information Distribuée (RID), l'interclassement de résultats (ou agrégation), consiste à combiner des listes de résultats provenant de différentes sources, dans une liste unique tout en veillant à satisfaire le plus possible les besoins de l'utilisateur. Ce problème, qui est relativement bien étudié dans le contexte de méta-recherche, ne s'avère pas aussi évident dans les systèmes de recherche pair-à-pair (P2P) totalement distribués. En effet, chaque pair est responsable d'interclasser les résultats des requêtes posées uniquement par ses utilisateurs. De ce fait, il ne détient pas d'index global des statistiques des autres pairs du réseau vu les contraintes d'anonymat et d'autonomie des pairs. En outre, le facteur d'échelle dans les systèmes P2P rend le classement des résultats critique, puisque nous devons classer les résultats provenant d'un grand nombre de pairs dans le réseau. Chaque pair est donc appelé à définir un classement fin des documents s'ajustant aux préférences des utilisateurs, en mettant les résultats les plus pertinents en premières positions. Le manque de statistiques globales (Yee *et al.,* 2005) d'une part, et l'utilisation de schémas d'indexation (Shokouhi *et al.,* 2007) et d'algorithmes de classement différents (Si *et al.,* 2003) d'autre part, nécessitent de recourir à un modèle d'agrégation efficace et indépendant des statistiques des collections. Ceci nous a incités à intégrer une information qui satisfait le mieux possible les préférences des utilisateurs, tout en étant facile à capturer sur chaque pair. Cette dimension, que nous appelons profil utilisateur, est définie par le comportement d'un utilisateur sur un pair lorsqu'il interagit avec le résultat d'une requête. Partant de cette notion de profil, nous proposons, dans ce papier, un modèle d'agrégation. Le reste de l'article est organisé comme suit : après une brève introduction au problème d'interclassement, la section 2 revise les modèles d'interclassement les plus populaires dans la littérature. Dans la section 3 nous définissons le modèle que nous proposons pour réaliser l'interclassement à partir des profils utilisateurs. Une étude empirique de ce modèle est présentée et discutée dans la section 4. Nous terminerons ce papier par une conclusion et quelques perspectives de travaux futurs.

2. Etat de l'art sur les principaux modèles d'interclassement

Une revue de la littérature nous a permis d'établir une catégorisation des principaux modèles d'interclassement en respectant les classifications données dans (Aslam *et al.,* 2001, Renda *et al.,* 2003). Nous parlons respectivement de modèles à base de scores et à base de rangs. Nous donnons dans les sections suivantes deux exemples de chaque famille de modèles à savoir le modèle vectoriel classique qui fut la base des modèles à base de scores et le modèle de langue. Dans la famille des modèles à base de rangs, nous parlerons du modèle de tourniquet plus connu par modèle Round Robin (RR) et le modèle Borda Count (BC) qui serviront par la suite comme références pour la comparaison de notre modèle.

2.1. *Modèles à base de scores*

Ce type de modèles suppose que les listes de résultats à fusionner envoient des informations sur le contenu de documents et/ou leurs scores de similarité selon les stratégies de classement locales. Un score global pour chaque document est calculé sur la base de la fonction de correspondance du modèle d'agrégation proposé afin de le reclasser dans la liste finale. Une panoplie de modèles d'agrégation peut être considérée dans cette classe. Nous citons entre autres le modèle vectoriel classique, le modèle probabiliste et le modèle de langage.

2.1.1. *Le modèle vectoriel*

Le modèle vectoriel constitue l'un des modèles les plus utilisés dans la recherche d'information web ou la recherche textuelle classique. Dans ce contexte, le système doit recevoir en entrée les statistiques des documents selon chaque liste locale. Nous parlons essentiellement de :

– TF_{ij} : (Term Frequency) : la fréquence d'un terme t_i dans un document d_j

– $DF_{(ti)}$: (Document Frequency) : qui désigne le nombre de documents dans lesquels le terme ti apparait. Parfois c'est l'inverse de la fréquence documentaire qui est retourné par la liste locale afin de présenter le pouvoir discriminatif d'un terme donné : l'idf (Salton *et al.,* 1975) , où n désigne la taille de la collection qui a retourné le document.

Un document d_j sera présenté sous forme d'un vecteur de poids w_i relatifs aux termes d'indexation. Une requête soumise sera présentée sous le même format qu'un document. La troisième étape constitue l'étape fondamentale pour le classement final des documents. Elle permet de calculer selon une seule métrique le score des documents provenant des différentes listes. Ce dernier est dérivé de la relation géométrique entre les vecteurs documents et requête dans un espace à t dimensions (t étant le nombre de termes d'indexation).Une fonction cosinus est généralement utilisée.

$$S(d_j, q) = \cos(\overrightarrow{q}, \overrightarrow{d_j}) = \frac{d_j \times q}{|d_j| \times |q|} \quad [1]$$

La liste finale de documents sera produite par le classement des documents selon les scores calculés. Le principe du modèle vectoriel permet de fournir une approximation primaire effective des propriétés statistiques du système de recherche étudié tout en étant simple et rapide à calculer. Cependant, il représente une difficulté d'interprétation et délaisse toute dépendance sémantique entre termes indexés par les différents systèmes locaux, notamment les relations de synonymie (plusieurs termes pour désigner le même concept) et polysémie (même terme désignant des concepts différents) (Salton *et al.,* 1975). Cela augmente fortement l'espace des termes dans la matrice documents-requête finale et constitue une vraie limite de l'approche.

2.1.2. *Le modèle de langage*

Un modèle de langage statistique attribue une probabilité à une séquence de m mots $P(w_1, w_2, ..., w_m)$ par le biais d'une distribution de probabilité. La modélisation de langage est utilisée dans de nombreuses applications de traitement du langage naturel, notamment, dans la reconnaissance vocale, la traduction automatique,, l'analyse et la recherche d'information. Lorsqu'il est utilisé en recherche d'informations, un modèle de langage est associé à un document dans une collection. A la soumission d'une requête q, les documents recherchés sont classés sur la base de la probabilité que le modèle de langage du document puisse générer les termes de la requête, $P(q_{jMd})$. L'estimation de la probabilité des séquences peut devenir difficile de corpus, dans des expressions ou des phrases qui peuvent être arbitrairement longues et, par conséquent, certaines séquences ne sont pas observées au cours de l'apprentissage du modèle de langage (problème de la rareté des données ou overfitting). C'est pour cette raison que ces modèles sont souvent estimés en utilisant les modèles N-gram.

2.2. *Modèles à base de rang*

L'agrégation à base de rangs constitue une famille de modèle où les différentes listes de résultats sont combinées sans savoir aucune information sur les statistiques globales des documents présents dans ces listes. Seuls les rangs sont considérés pour le calcul de poids finaux de documents. Plusieurs approches à base de rangs sont connues dans la littérature. Nous passons en revue deux exemples de ces modèles à savoir le modèle Round Robin et le modèle de vote Borda count qui serviront par la suite pour faire une comparaison par rapport à notre modèle

2.2.1. *Round Robin (RR)*

Le modèle "Round Robin" (Greengrass, 2000) permet d'agréger des documents selon leurs rangs dans les listes de résultats retournées par les systèmes de recherche sélectionnés (les premiers puis les deuxièmes et ainsi de suite). Le premier document dans la liste finale sera le premier document dans la première liste retournée et ainsi de suite. Le choix des listes se fait de manière aléatoire. Callan affirme dans (Robertson *et al.,* 1995) que Round Robin est un choix à opérer si les scores individuels des documents sont complètement incomparables (lorsque les systèmes de recherche utilisent différents algorithmes ou lorsque les collections utilisent différentes statistiques). Cependant, les travaux de (Renda *et al.,* 2003) ont montré que les meilleurs résultats de RR sont observés lorsqu'il s'agit des mêmes stratégies de recherche. Le problème de la non comparabilité des listes de résultats peut être atténuée par la normalisation des rangs des différents documents. La simplicité de l'approche Round Robin ne fait pas d'elle certainement la meilleure puisqu'elle ne tient pas compte de l'importance relative des systèmes de recherche. En utilisant cette méthode, chaque système de recherche a les mêmes chances d'être classé premier, même s'il est moins pertinent que d'autres relativement à cette requête.

2.2.2. *Le modèle Borda count(BC)*

Faisant partie de modèles de vote positionnels, le modèle BC sert à combiner les préférences de plusieurs experts. L'entrée d'un système de vote est dite profil. Afin d'adapter les algorithmes de vote au problème de la recherche d'information distribuée, l'ensemble des documents à classer sont considérés comme candidats. Les votants sont simulés par différents systèmes de recherche se présentant en entrée de l'algorithme d'interclassement. Borda count se base sur le principe suivant : Chaque votant classe un ensemble de "C" candidats selon un ordre de préférence (Renda *et al.,* 2003). Pour chaque votant, le premier candidat acquière "c" points, le deuxième acquière "$c-1$" et ainsi de suite. S'il reste des candidats non classés par les votants, les points restants seront divisés équitablement entre les candidats non classés. L'interclassement se fait en triant la liste des documents par ordre décroissant du nombre de points cumulés. L'un des inconvénients des modèles basés sur Borda Count est qu'ils se contentent de diviser le nombre de points restant de façon équitable entre les documents manquants dans la liste. Ceci réduit considérablement l'importance des documents manquants, qui se voient toujours classés en dernières positions. Le deuxième inconvénient du modèle Borda Count est qu'il attribue une importance égale à tous les systèmes de recherche ce qui n'est pas réaliste. Une mesure de préférence peut améliorer expressivement la performance du modèle.

3. Modèle d'interclassement à base de profils

Une revue de la littérature a révélé que la majorité des méthodes d'interclassement, notamment les méthodes à base de scores et (ou) à base de rangs (Renda *et al.,* 2003), ont une approche généraliste. Ce type d'approche ne tient pas compte généralement des caractéristiques des utilisateurs d'un site ou d'un pair. Ceci a pour conséquence de limiter la qualité des résultats obtenus. Dans le but d'améliorer l'efficacité de l'opération d'agrégation, nous avons défini un modèle d'agrégation basé sur les profils des utilisateurs et tenant compte des corrélations observées entre requêtes, pairs et documents.

3.1. *Notion de profil utilisateur*

"Un profil utilisateur (ou encore modèle utilisateur) est un ensemble de données qui concernent l'utilisateur d'un service informatique. C'est une source de connaissances qui contient des acquisitions sur tous les aspects de l'utilisateur qui peuvent être utiles pour le comportement du système" (Wahlster *et al.,* 1986). Dans notre contexte, un profil utilisateur représente des informations extraites de l'historique des requêtes de cet utilisateur et qui sont stockées dans un fichier journal, appelé fichier log. Nous considérons deux types de profils :

– Les corrélations sémantiques entre les requêtes passées et les pairs positifs.

– Les corrélations sémantiques entre les requêtes passées et les documents positifs.

Notons ici qu'un pair positif est un pair à partir duquel il y a eu un téléchargement de documents pour une requête donnée. Ces documents représentent un intérêt particulier pour une requête similaire qui pourra être formulée dans le futur. Un document positif est un document téléchargé (donc considéré comme pertinent par un utilisateur) pour une requête donnée. La corrélation sémantique pour un profil est modélisée par un contexte formel qui représente une association entre objets et propriétés qui seront déduits du fichier log. L'ensemble de profils de même type constitue une base de connaissances qui sera utilisée pour agréger les résultats. retournés par les différents pairs. Nous présentons dans la section suivante les détails de la génération de ces profils.

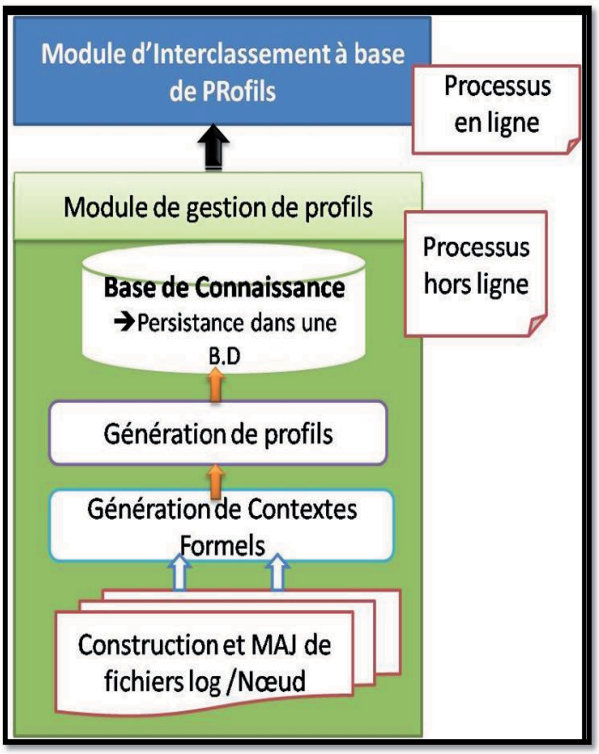

Figure 1. *Modèle d'interclassement à base de Profils : Architecture Générale*

3.2. *Architecture générale du modèle d'agrégation à base de profils*

L'architecture du modèle d'interclassement que nous proposons est schématisée par la Figure 1. Il s'agit d'une architecture à deux couches : une couche de gestion

de profils et une couche d'interclassement des résultats qui se base sur la première couche.

3.2.1. *Module de gestion de profils*

Pour gérer les modules, nous proposons un module composé des étapes suivantes :

1. Construction et mise à jour d'un fichier log : afin de garder une trace des interactions de l'utilisateur avec le système de recherche d'information pair-à-pair (RI-P2P), nous avons opté pour la construction d'un fichier log au niveau de chaque pair. Ce fichier est formalisé comme suit : Soit l'ensemble des termes $T_i = \{t_{i1}, t_{i2}, \ldots, t_{ik}\}$ d'une requête soumise par un utilisateur à partir d'un pair donné d'identité pairID. Soit DR l'ensemble $\{< d_1; id_1 >, < d_2; id_2 > , \ldots, < d_p; id_p >\}$ de couples $< document; pairId >$ décrivant les documents téléchargés à partir d'un ensemble de pairs. A la réception d'une réponse relative à une requête donnée, le module met à jour le fichier log en y ajoutant les informations relatives à cette requête, à savoir l'identifiant de la requête, son thème (déduit à partir de l'ensemble de ses mots clés), les documents téléchargés par l'utilisateur et les pairs associés.

2. La génération des contextes formels : c'est une étape intermédiaire qui consiste à manipuler des fichiers log en vu de générer ultérieurement les connaissances. Ces dernières seront stockées dans notre système afin de lui fournir les éléments nécessaires pour définir le profil d'un utilisateur. La génération des contextes formels est régie par le principe suivant : Soient O un ensemble d'objets (ou exemples), P un ensemble de propriétés (ou attributs) et R une relation binaire définie entre O et P. Un contexte formel est défini par le triplet (O, P, R). Les éléments de O sont appelés les objets et les éléments de P sont appelés les propriétés du contexte. Pour exprimer qu'un objet o de O est en relation avec une propriété p de P, nous écrirons oRp. Ceci signifie que l'objet o a la propriété p, tel que o est le domaine du profil, noté $dom(PR)$, et p est le co-domaine du profil, noté $codom(PR)$.

3. La génération de profils : afin de générer les profils utilisateurs, nous avons utilisé une approche formelle de construction de ces profils, basée sur l'Analyse Formelle de Concepts (AFC). Dans notre cas, nous nous sommes basés sur deux contextes qui sont des projections du fichier log. La première représente le lien entre les thèmes des requêtes passées et les pairs positifs, appelée contexte CPT. La seconde projection représente le lien entre les thèmes des requêtes passées et les documents associés, que nous appellerons contexte CDT .

La génération de l'ensemble des profils se fait hors ligne au niveau de chaque pair et de manière périodique. Un algorithme de génération de concepts formels est par la suite appliqué pour générer deux couvertures, notées $CE1$ et $CE2$. Les profils de $CE1$, respectivement de $CE2$, sont définis sous la forme suivante : $(\{p_1, \ldots, p_i\}; \{t_1, \ldots, t_j\})$, respectivement, $(\{d_1, \ldots, d_t\}; \{t_1, \ldots, t_j\})$, tel que $\{p_1, \ldots, p_i\}$ est un ensemble de

pairs, $\{t_1, \ldots, t_j\}$ est l'ensemble des termes des requêtes passées et $\{d_1, \ldots, d_t\}$ est un ensemble de documents. Ces ensembles constituent une base, que nous noterons par $B(CE1, CE2)$ et qui servira par la suite comme base de connaissances pour l'algorithme d'interclassement des résultats. Un profil utilisateur, PR, est ainsi défini comme étant un couple regroupant deux ensembles nommés, respectivement, domaine $dom(PR)$ et co-domaine $codom(PR)$ de profils. Une formalisation des principales opérations sur les profils sera détaillée dans la section 3.3.2.
Un exemple ullistratif de génération des profils est donné dans la figure 2 :

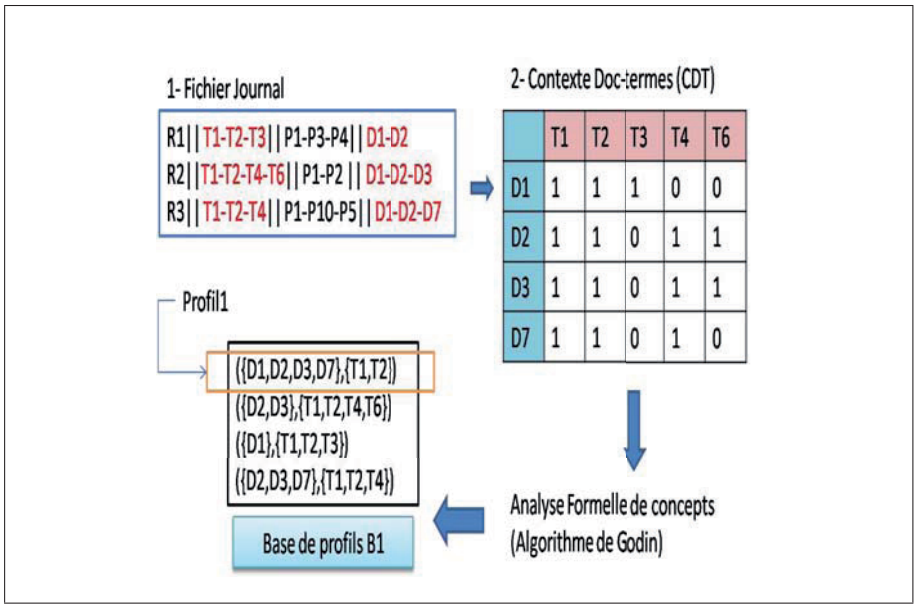

Figure 2. *Exemple de génération de profils*

3.3. *Module d'interclassement*

Nous considérons que le jugement d'un utilisateur représente un élément fondamental pour déterminer la pertinence d'une réponse par rapport à une requête. En effet de par ses compétences et son expérience, un utilisateur est capable de porter un jugement sur les réponses qu'il a obtenu suite à une requête. Partant de ce principe notre module d'interclassement (voir Algorithme 1) utilise deux filtres :

– Filtre 1 : La couverture CE1 est utilisée pour attribuer des scores aux pairs retournant des résultats.

– Filtre 2 : La couverture CE2 est utilisée pour attribuer des scores partiels aux documents retournés par les pairs.

Un score global pour chaque document retourné est calculé afin d'interclasser les résultats dans une seule liste. Ce score met en évidence quatre paramètres indispensables à notre approche d'agrégation :

a. Rang du document (SRd) : il représente le rang local du document dans le résultat retourné par un pair donné et reflète l'importance de ce document dans le pair. Il est fourni explicitement dans le résultat de recherche (à partir de son rang).

b. Score du pair (Sp) : ce paramètre représente l'importance du pair par rapport à la requête en puisant dans la base de connaissance des profils collectés. Il représente le degré de confiance que nous pouvons attribuer à ce pair ; il est fourni par la procédure $SimilartyPeerQuery$ de l'algorithme 1.

c. Ancienne Popularité d'un document (SPd) : il définit le taux de présence d'un document résultat dans l'historique des requêtes similaires modélisées. Plus un document est présent, plus il a été téléchargé et donc il est intéressant. Ce taux est fourni à l'aide de la procédure $SimilartyDocQuery$ de l'algorithme 1

d. Popularité d'un document (Pd) : représente le taux de présence d'un document dans les résultats des différents pairs. le choix de ce paramètre est justifié par le fait que plus le document est élu par les pairs lors de la réponse à une requête, plus cela reflète son importance pour la requête. Il est fourni à l'aide de la procédure $CardinalityDocPeers$ de l'algorithme 1 qui se base sur la couverture $CE2$.

L'ensemble de ces paramètres nous ont conduit à définir un "premier" score global de document résultat, qui tient compte de l'importance du document dans la réponse locale au niveau de chaque nœud, i.e., son rang SR_d, puis sa popularité dans la réponse à des requêtes similaires passées, SP_d, ainsi que le taux de satisfaction des utilisateurs dégagé à partir de l'historique des requêtes. Ce score, calculé à l'aide de la fonction $setGlobalScore$ de l'algorithme 1, est formulé comme suit :

$$SG = \frac{\Sigma SP(SP_d + SR_d)\frac{P_d}{NP}}{NP} \quad [2]$$

NP : est le nombre total des pairs répondant à la requête. L'utilisation de ce paramètre a pour objectif de normaliser le score. Le calcul de l'ancienne popularité du nœud "Score du pair : S_p" et celui du document repose sur la notion de distance sémantique expliquée dans la section suivante.

3.3.1. *Distance sémantique*

La génération des profils similaires au profil d'une requête est basée sur les propriétés communes entre les co-domaines des profils. La fonction de similarité utilisée dans notre cas est celle proposée par Salton dans (Salton, 1989). La similarité de l'ensemble de deux objets a et b de O se caractérise par des propriétés communes. D'une manière formelle, cette similarité s'exprime comme suit : Soit P_a et P_b les ensembles

Algorithme : INTERCLASSEMENT $(B, Q, Lrp,)$
Entrées :
B : Base de connaissances $(CE1, CE2)$.
Q : Requête soumise.
Lrp : Liste de résultats par différents pairs.
p_i, d_j : un pair p_i ; un document d_j retourné par p_i.
Sortie :
LF : Liste Finale classée
Traitement
  $similarityPeerQuery(CE1, Q)$;
  $similarityDocQuery(CE2, Q)$;
  tant que $Lrp \neq \emptyset$ **faire**
    $EC1 := selecteConcepts(CE1, Lrp)$;
    tant que $Lrp.p_i \neq \emptyset$ **faire**
      $EC2 := selecteConcepts(CE2, Lrp.pi)$;
      $card := cardinalityDocPeers(dj, Lrp)$;
    $setGlobalScore(Lrp.pi.dj)$;
  $LF := mergeSort(Lrp)$;
fin

Algorithme 1 : ALGORITHME D'INTERCLASSEMENT DE RÉSULTATS

des propriétés respectives des objets a et b. Les propriétés communes de ces objets est l'ensemble $P_a \bigcap P_b$. La similarité se définit selon la formule suivante (Salton, 1989) :

$$Sim(P_a, P_b) = \frac{|P_a \bigcap P_b|}{|P_a \bigcup P_b|} \quad [3]$$

3.3.2. *Opérations sur les profils*

Un profil PR est défini comme étant une paire de deux ensembles dits $dom(PR)$ ou domaine de PR et $codom(PR)$ ou co-domaine de PR. Dans cette section nous tenons à définir l'ensemble d'opérations servant à manipuler l'ensemble de nos profils. Soient $PR_1(D_1, C_1)$, $PR_2(D_2, C_2)$ deux profils de domaines D_1 et D_2 et co-domaines C_1 et C_2. Les opérations que nous utilisons pour manipuler nos profils sont les suivantes :

– Inclusion de profils : $= \{PR_1 \subseteq PR_1 | D_1 = D_2\}$

– Egalité de profils : $PR_1 = PR_2 \Rightarrow PR_1 \subseteq PR_2$ et $PR_2 \subseteq PR_1\}$.

– Produit Cartesien : $PR_1 \times PR_2 = PR | dom(PR) = D_1 \cup D_2$ et $Codom(PR) = C_1 \cup C_2$.

– Différence de profils $PR_1 - PR_2 = PR | dom(PR) = D_1 - (D_1 \cap D_2)$ et $Codom(PR) = C_1 - (C_1 \cap C_2)$

– Projection : $PR_1 \perp PR_2 : PR|dom(PR) = D_1 \cup D_2$ et $Codom(PR) = C_1 \cup C_2$ et $C_1 \cup C_2 \neq \emptyset$

4. Etude empirique

4.1. *Environnement d'expérimentation*

Pour tester l'approche proposée dans cet article, nous avons choisi le simulateur PeerSim (Jelasity *et al.,* 2007), qui est un outil Opensource écrit en Java. Il présente l'avantage d'être déjà spécialisé pour l'étude des systèmes P2P et il dispose d'une architecture ouverte et modulaire, ce qui lui permet d'être adapté à des besoins spécifiques. Pour la recherche d'information dans les systèmes pair-à-pair, nous avons opté pour le simulateur générique de Recherche d'Information (RI) réalisé dans le cadre du projet RARE (Defude, 2008) . Ce simulateur est construit au dessus de PeerSim et peut être vu comme une spécialisation de PeerSim pour la recherche d'information. Pour tester notre approche au sein de ce système, nous avons ajouté deux couches : la première est relative à la création des fichiers log alors que la deuxième sera consacrée à l'interclassement des résultats provenant des différents pairs et utilisant les profils. La génération des profils a été faite en utilisant l'algorithme de Godin (Godin *et al.,* 1995), implémenté dans la plate-forme Galicia V 3 (Valtchev *et al.,* 2003).

4.2. *Benchmark*

Afin de tester notre algorithme d'interclassement, nous avons utilisé le jeu de données "BigDataSet", produit dans le cadre du projet RARE (Defude, 2008). Ce jeu de données a été obtenu à partir d'une analyse statistique sur des données collectées à partir du système Gnutella (H., 2007) et des données de la collection TREC (TREC, 2008), ce qui nous permet de réaliser des simulations en conditions réelles. BigDataSet est composé de 25 000 documents et de 4999 requêtes répartis sur 500 pairs. Il fournit des fichiers XML décrivant les nœuds du système et les documents qu'ils possèdent, ainsi que les requêtes qui seront lancées sur le réseau. La distribution des documents sur les pairs est faite d'une manière complètement aléatoire. De même pour la distribution des requêtes, mais sous la contrainte que la répartition des requêtes soit proportionnelle à celle des documents c'est à dire les pairs contenant un grand nombre de documents sont ceux qui envoient le plus de requêtes et vice versa.

4.3. *Phase d'apprentissage*

L'algorithme IPR a besoin d'une base de connaissances (les profils) pour chaque pair qui va simuler le modèle comportemental. Pour cela, nous avons lancé les 10 000 premières requêtes (sur 500 nœuds en simulation), afin de construire un fichier log initial pour chaque pair. Par la suite, nous avons lancé le module de gestion de profils

pour construire une base de connaissances pour chaque pair à partir de son fichier log, que nous avons noté B.

4.4. *Phase de test*

Pour la phase de test, nous avons lancé le reste des requêtes sur 100 nœuds. Parallèlement à l'implantation de notre approche, des approches classiques d'interclassement ont été testées afin de les comparer à la notre. Nous avons principalement faits des tests avec les méthodes Round Robin (Greengrass, 2000) et Borda Count (Renda *et al.,* 2003). L'évaluation s'est basée sur certaines métriques classiques et de nouvelles qui ont été définies dans le cadre de nos travaux.

4.4.1. *Métriques d'évaluation utilisées*

Dans un système de RI, le succès du système ou son rejet est fonction des métriques d'évaluation utilisées. Dans la communauté de la RI, les mesures de Précision et de Rappel sont couramment utilisées comme métriques de base pour tester l'efficacité de méthodes de RI (Renda *et al.,* 2003). Pour raffiner d'avantage la qualité des systèmes de RI, nous proposons d'autres mesures. Ces mesures sont définies comme suit : Soient $NP-k$ et NP, respectivement le nombre de documents pertinents retournés aux k premières positions et le nombre de tous les documents pertinents pour une requête donnée. Les métriques que nous avons utilisées ici pour évaluer et comparer l'ensemble de méthodes d'interclassement sont les suivantes :

$$Precision@k = \frac{NP-k}{k} \quad [4]$$

$$Rappel@k = \frac{NP-k}{NP} \quad [5]$$

Nous remarquons que ces métriques ne reflètent aucune qualité pour le classement : est ce qu'on a préservé le meilleur classement ou plutôt de combien d'écart ce classement a été perturbé ? Pour ce faire nous avons défini deux nouvelles métriques que nous allons présenter dans la section suivante.

4.4.2. *Définition de nouvelles métriques*

Nous proposons des métriques d'évaluation des performances tenant compte de l'aspect multivalué de la pertinence. Ces métriques permettent de procéder à une évaluation comparative du gain d'information obtenu sur des processus de recherche d'information déployés sur différentes collections. Ainsi, nous introduisons ces mesures qui nous permettent de qualifier de quel ordre une amélioration basée sur le comportement utilisateur améliorera le score global du système. Dans notre cas, les scores sont calculés sur les k premiers documents retournés. Ainsi, pour avoir une évaluation qualitative plutôt que quantitative de notre approche, nous avons défini, dans un premier

temps, le pourcentage de positions similaires Np identique à la liste centralisée. Cette mesure est définie comme suit :

$$SimPos@k = \frac{\sum Np_{identiques}}{k} \quad [6]$$

La deuxième métrique, que nous proposons, mesure le taux de perturbation du classement, Np, par rapport au classement centralisé. Nous la définissons formellement par l'équation suivante :

$$\Delta DecalPos@k = \frac{\sum Np\Delta - k}{k} \quad [7]$$

4.4.3. *Résultats expérimentaux*

Nous avons comparé notre approche (notée IPR sur la figure 3) avec l'approche Round Robin (RR) et la méthode de vote Borda Count (BC) pour les k premières positions. La Figure 2 présente les résultats obtenus pour les métriques $Precision - k$, $SimPos - k$ et $\Delta DecalPos - k$. Les premiers tests présentés dans cette figure sont, à notre avis, très encourageants. La comparaison de notre approche avec celles existantes montre que notre approche est compétitive. En effet, pour les métriques $Precision - k$ (a) et $SimPos - k$ (b), la Figure 3 montre que notre approche donne les meilleurs résultats bien que la différence ne soit pas très significative. Cette différence est par contre plus importante pour la métrique $\Delta DecalPos - k$ (c), pour la comparaison de notre méthode avec la méthode BC pour les trois premiers documents. Ceci veut dire que même si on ne garde pas les mêmes positions par rapport à la liste centralisée, nous préservons un petit décalage par rapport aux autres méthodes. Il nous semble que ces tests pourront être étendus par un nombre plus importants de requêtes, pour un k plus grand et en utilisant d'autres benchmarks.

5. Conclusion

Nous avons proposé, dans ce papier, une approche d'interclassement de résultats à base de profils utilisateurs dans le contexte d'une recherche d'information pour des systèmes P2P. La motivation sous-jacente à l'utilisation de ces profils était de tenir compte des préférences des utilisateurs d'un site ou d'un pair afin d'améliorer l'efficacité du processus d'agrégation. Notre approche est complètement décentralisée, c'est-à-dire que d'une part l'autonomie des pairs est respectée (il n'y a pas de statistiques globales) et d'autre part la construction des profils utilisateurs est également purement locale. Pour valider notre modèle, nous avons réalisé un certain nombre d'expérimentations et nous l'avons comparé à des approches existantes. L'étude empirique que nous avons menée nous a permis de proposer de nouvelles métriques d'évaluation. Les résultats obtenus à travers ces expérimentations ont permis de montrer que notre approche est compétitive par rapport aux approches existantes. Les premiers résultats obtenus sont assez encourageants et nous laissent envisager de nouvelles améliorations

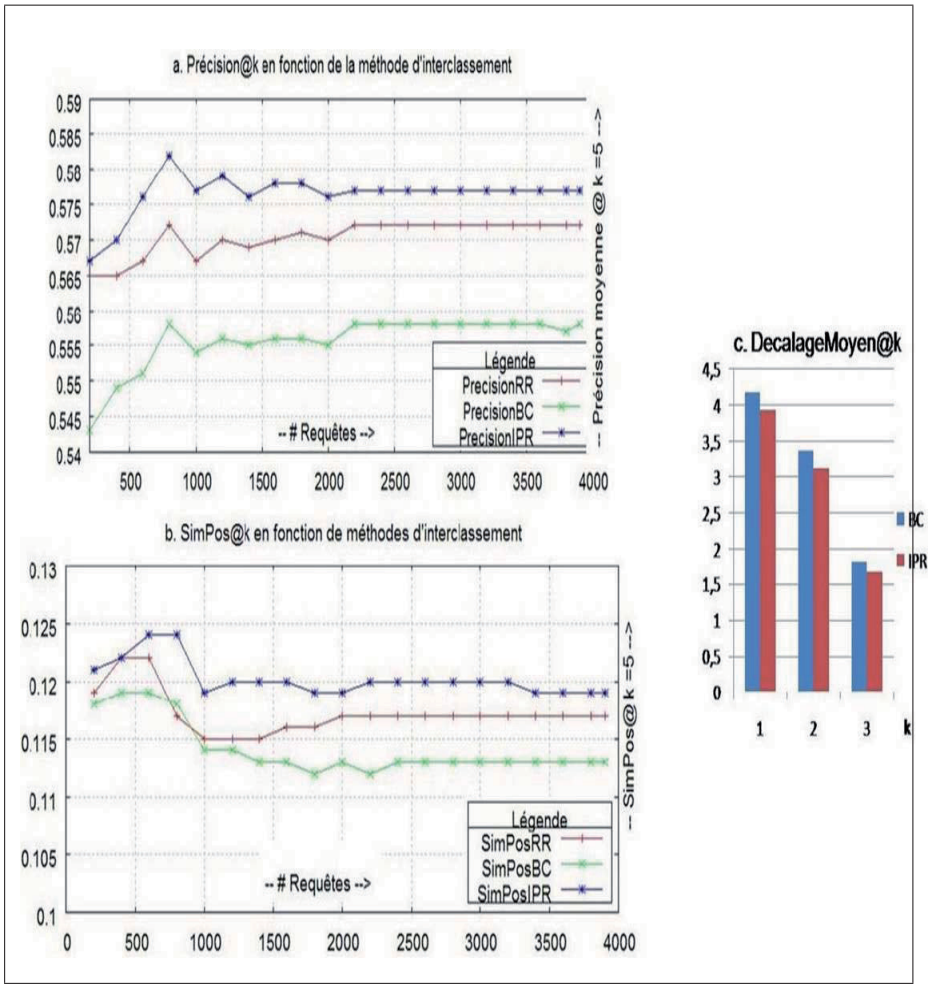

Figure 3. *Evaluation du Modèle d'interclassement*

ainsi que des expérimentations beaucoup plus larges que celles que nous avons effectuées. En effet, il est possible de jouer sur le nombre de pairs testés, sur certains paramètres de distribution et de score. Du point de vue expérimentations, il nous semble nécessaire de faire des tests avec un nombre plus important de requêtes, d'augmenter le paramètre k (nombre de documents pertinents retournés aux k premières positions) et d'utiliser d'autres benchmarks.

6. Bibliographie

Aslam J. A., Montague M., « Models for metasearch », *SIGIR '01 : Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, p. 276-284, 2001.

Defude B., « Le projet RARE : Routage optimisé par Apprentissage de Requêtes », http ://www-inf.int-evry.fr/defude/RARE/, 2008.

Godin R., Missaoui R., Alaoui H., « Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices », *Computational Intelligence*, vol. 11, p. 246-267, 1995.

Greengrass E., « Information Retrieval : A Survey », 2000.

H. G. R. M. R. A., « Incremental concept formation algorithms based on galois », http ://www.gnutella.com/, January, 2007.

Jelasity M., Montresor A., Jesi G. P., Voulgaris S., « Peersim Simulator, a peer-to-peer simulator », 2007.

Renda M. E., Straccia U., « Web metasearch : rank vs. score based rank aggregation methods », *SAC '03 : Proceedings of the 2003 ACM symposium on Applied computing*, ACM, New York, NY, USA, p. 841-846, 2003.

Robertson S. E., Walker S., Beaulieu M. M., Gatford M., « Okapi at TREC-4 », *In Proceedings of the 4th Text REtrieval Conference (TREC-4*, p. 73-96, 1995.

Salton G., *Automatic text processing : the transformation, analysis, and retrieval of information by computer*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

Salton G., Wong A., Yang C. S., « A vector space model for automatic indexing », *Commun. ACM*, vol. 18, n° 11, p. 613-620, 1975.

Shokouhi M., Zobel J., Bernstein Y., « Distributed text retrieval from overlapping collections », *ADC '07 : Proceedings of the eighteenth conference on Australasian database*, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, p. 141-150, 2007.

Si L., Callan J., « A semisupervised learning method to merge search engine results », *ACM Trans. Inf. Syst.*, vol. 21, n° 4, p. 457-491, 2003.

TREC, « Text REtrival Conference », 2008.

Valtchev P., Grosser D., Roume C., Hacene M. R., « Galicia : An Open Platform for Lattices », *In Using Conceptual Structures : Contributions to the 11th Intl. Conference on Conceptual Structures (ICCS'03*, Shaker Verlag, p. 241-254, 2003.

Wahlster W., Kobsa A., « Dialogue-based user models », *IEEE*, p. 948-960, 1986.

Yee W. G., Frieder O., « On search in peer-to-peer file sharing systems », *SAC '05 : Proceedings of the 2005 ACM symposium on Applied computing*, ACM, New York, NY, USA, p. 1023-1030, 2005.