Analyse d'hyperliens en vue d'une meilleure description des profils

Valéry Vandaele, Pascal Francq, Alain Delchambre

CAD/CAM – Université Libre de Bruxelles – Av. F.D. Roosevelt, 50 – CP 165/14 1050 Bruxelles – Belgique

{vavdaele, pfrancq, adelch}@ulb.ac.be

Abstract

The objective of this paper is to discuss the integration of various algorithms of link analysis into the GALILEI project. In the frame of this projet aiming at clustering users into virtual communities according to their centers of interests, one notices indeed that the system converges more rapidly when a document is consulted by more than one user. Knowing that the methods of link analysis could emphasize the interesting documents through the profiles, we thus hope to be able to increase the number of documents judged by more than one user. The benefits of various methods will be studied here.

Résumé

Le but de cet article est de discuter de l'intégration de différents algorithmes d'analyse de liens au sein du projet GALILEI. En effet, dans le cadre de ce projet, visant à regrouper des utilisateurs en communautés virtuelles en fonction de leurs centres d'intérêts, on remarque que le système converge plus rapidement lorsqu'un document est consulté par plus d'un utilisateur. Sachant que les méthodes d'analyse de liens pourraient faire ressortir des documents « références » à travers les profils, on espère ainsi augmenter le nombre de documents jugés par plus d'un utilisateur. Pour cela, nous allons utiliser plusieurs méthodes.

Mots-clés : hyperliens, liens, graphe, hub, authority.

1. Introduction

Dans cet article, nous présentons différents moyens pour améliorer le groupement de profils en utilisant l'analyse des structures hyperliées. Tous les travaux ont été effectués dans le cadre du projet GALILEI[1], décrit dans la section 2. La section suivante évoque la notion d'hyperliens et les extensions qu'on peut y apporter. Dans la section 4, les concepts des différents algorithmes d'analyse d'hyperliens sont présentés et brièvement expliqués. L'intégration de ces méthodes dans le projet est abordée dans la section 5 et la méthodologie de validation utilisée dans GALILEI est ensuite présentée (section 6). La section 7 décrit les expériences qui ont été réalisées sur les différentes méthodes de calcul de liens ; puis elle compare les résultats obtenus. L'article se termine par un point sur les développements futurs (section 8) et une conclusion (section 9).

2. Le projet GALILEI

La quantité d'information, disponible électroniquement, a considérablement augmenté, vu le nombre sans cesse croissant de systèmes orientés documents, tel qu'Internet (Lawrence et Giles, 1998 et 1999). Afin de pouvoir traiter cette masse d'informations, il est de plus en plus

[1] Ce projet est subventionné par la Région wallonne, sous le contrat 01/1/4675, et est disponible à l'adresse suivante : http ://galilei.ulb.ac.be

nécessaire de s'armer d'un système de recherche documentaire, qui facilite le processus d'extraction de l'information en la structurant et en la prétraitant. Un article récent a rappelé que trouver l'information pertinente est devenu un problème crucial (Fogarty et Bahls, 2002).

Le but principal du projet GALILEI (Francq, 2003) est de cerner, le plus précisément possible, les centres d'intérêts des utilisateurs afin de pouvoir ensuite les regrouper correctement. Les utilisateurs doivent tout d'abord se créer des profils, un par centre d'intérêt. Ils peuvent ensuite commencer à émettre des avis sur la pertinence[2] des documents qu'ils consultent. Ces derniers sont ajoutés au profil, de même que le jugement correspondant.

Le système calcule la description des profils en se basant sur le contenu des documents, ainsi que sur les jugements émis par les profils.

Le système regroupe ensuite les différents profils en fonction de leur description : les profils ayant une description similaire sont regroupés afin de constituer un certain nombre de **communautés virtuelles**. Ce processus est accompli de façon dynamique par un algorithme génétique spécialement développé à cet effet, le GVCA « *Genetic Virtual Community Algorithm* ». Celui-ci se base sur la similarité entre profils, mais aussi et surtout sur deux types de contraintes dures, à savoir le taux de comportement en accord et en désaccord :

Le taux de comportement en accord représente le rapport entre le nombre de documents jugés comme pertinents par deux profils et le nombre total de documents jugés par ces deux profils. Pour chaque paire de profils, p_k et p_l, le taux de comportement en accord, $0 <= B_{same} <= 1$, est défini par : $B_{same}(p_k, p_l) = \frac{|D^I_{k,l}|}{|D_{k,l}|}$ où $D_{k,l}$ représente le sous-ensemble de documents jugés par les deux profils p_k et p_l et où $D^I_{k,l}$ représente le sous-ensemble de documents jugés de la même façon par les deux profils.

Le taux de comportement en désaccord représente le rapport entre le nombre de documents jugés différemment par deux profils et le nombre total de documents jugés par ces deux profils. Pour chaque paire de profils, p_k et p_l, le taux de comportement en désaccord, $0 <= B_{diff} <= 1$, est défini par : $B_{diff}(s_k, s_l) = \frac{|D^D_{k,l}|}{|D_{k,l}|}$, où $D^D_{k,l}$ représente le sous-ensemble de documents jugés différemment par les deux profils.

Lorsque le taux de comportement en accord est supérieur à un seuil fixé, on force les deux profils à être groupés ; de même, lorsque le taux de comportement en désaccord est supérieur à un certain seuil, on force les deux profils à ne pas être groupés dans la même communauté virtuelle. On voit donc que ces deux contraintes dures jouent un rôle important dans le groupement, mais pour que celles-ci aient un sens, il faut que les profils aient jugé un nombre minimum de documents (par exemple dix).

Une fois ces communautés virtuelles définies, l'information pertinente peut être diffusée entre les différents membres de ce groupe. Par exemple, certains documents considérés comme pertinents par un grand nombre d'utilisateurs d'une même communauté virtuelle peuvent être mis en partage pour toute la communauté.

Le système a été validé sur plusieurs collections de documents grâce à une méthodologie bien définie ; les résultats obtenus sont très prometteurs (Francq, 2003). Une solution informatique, disponible sous licence GNU GPL, a été développée.

[2] La pertinence d'un document peut être jugée de trois façons différentes : « *Relevant* » : le document est pertinent, « *Fuzzy Relevant* », le document n'est pas pertinent mais appartient quand même au centre d'intérêt et « *Irrelevant* » : le document n'appartient pas au centre d'intérêt ; et ce juste par un click de souris sur le document.

3. Extension de la notion d'hyperliens

Les hyperliens dans les documents HTML sont bien connus : ils permettent à l'utilisateur de naviguer d'une ressource à une autre, juste en cliquant sur ceux-ci. Cette notion d'hyperliens peut être étendue à une multitude d'autres types de documents ne contenant pas explicitement d'hyperlien. En effet, ceux-ci doivent juste contenir une structure qui apparaît dans un grand nombre de ces documents et qui permet de relier au moins deux ressources. Par exemple, les cocitations — bibliographie — des articles PDF peuvent être vues comme des liens entre deux ressources distinctes. Ces types de liens ne permettent pas de retrouver directement l'emplacement du document, mais sont tout de même capables de créer des liens entre différentes ressources. Dans la suite de l'article, tous les raisonnements sont fondés sur les principes des hyperliens ; ils peuvent bien sur être généralisés à n'importe quel autre type de liens.

L'utilisation des liens a pour but est d'améliorer et d'élargir plus rapidement une collection de documents. En effet, les liens rencontrés dans les documents contiennent une certaine quantité d'information intéressante, appartenant généralement au même centre d'intérêt, puisque l'auteur prend la peine de citer les autres ressources.

4. Aperçu des algorithmes de liens

Soit un ensemble de documents Web se citant mutuellement, cette collection peut être vue comme un graphe dirigé. Les algorithmes d'analyse de liens construisent la matrice d'adjacence, reflet du graphe basé sur le modèle de citation utilisé. Par exemple, le lien qui lie deux documents i et j sera représenté par la valeur 1 de l'élément W_{ij}. Les pages les plus intéressantes peuvent ensuite être extraites en calculant les vecteurs propres du système. Au sens de Kleinberg (1998), ces pages peuvent être divisées en deux catégories :

les hubs : pages contenant peu d'informations pertinentes, mais beaucoup d'hyperliens.

les authorities : pages contenant peu de liens, mais beaucoup d'informations pertinentes.

4.1. *L'algorithme HITS*

Kleinberg (1998) a proposé un algorithme, appelé HITS, capable d'identifier les meilleurs *hubs* et *authorities* au sein d'une collection hyperliée. Cet algorithme exploite la structure du graphe du web. Chaque document est vu comme un nœud du graphe dirigé et tout lien entre deux documents est interprété comme une arête entre deux nœuds. Partant d'une requête spécifique, appelée σ, l'algorithme crée tout d'abord un sous-graphe, S_σ ; il calcule ensuite le poids des *hubs* et *authorities* pour chaque nœud de S_σ.

Le principe utilisé par l'algorithme HITS est le suivant : un document a un poids *authority* élevé s'il est pointé par beaucoup de documents ayant un poids *hub* élevé et, vice versa, un document a un poids *hub* élevé s'il pointe vers beaucoup de documents ayant un poids *authority* élevé.

Plus précisément, en partant d'une série de documents hyperliés, l'algorithme HITS construit le graphe dirigé, associé à la collection. Idéalement, la collection S doit respecter les propriétés suivantes :

(i) S est relativement petit.
(ii) S contient beaucoup de pages pertinentes.
(iii) S contient la plupart des meilleurs *authorities*.

Le graphe est représenté par une matrice d'adjacence, W, de taille $n \times n$, où n représente le nombre de documents utilisés. L'élément W_{ij} prend la valeur 1 s'il existe une arête entre les

nœuds i et j dans le graphe dirigé, et 0 dans les autres cas. Généralement, la troisième condition n'est pas satisfaite et la collection S doit être étendue en explorant un certain nombre de liens du graphe (Kleinberg, 1998). L'algorithme peut ensuite calculer les relations de renforcement mutuel entre les *hubs* et les *authorities* en itérant les règles de mise à jour suivantes :

$$h_i^{(k+1)} = \sum_{j:(j\to i)}^{n} a_j^{(k)} \;;\; a_i^{(k+1)} = \sum_{j:(i\to j)}^{n} h_j^{(k)} \tag{1}$$

où « $i \to j$ » signifie que le document i pointe vers le document j. Les équations ci-dessus peuvent aussi s'écrire en utilisant la notation matricielle :

$$\left\{ \begin{array}{lcccl} h^{(k+1)} & = & Wa^{(k)} & = & (WW^T)h^{(k)} \\ a^{(k+1)} & = & W^T h^{(k)} & = & (W^T W)a^{(k)} \end{array} \right. \tag{2}$$

En initialisant les vecteurs poids avec un vecteur colonne unité $n \times 1 : h^{(0)} = a^{(0)} = [1, 1, 1, ...]^T$ et en normalisant les vecteurs résultant de l'application des règles de mise à jour, par une norme euclidienne, nous utilisons la « *power method* »[3], qui converge vers le vecteur propre principal de la matrice symétrique. Kleinberg a prouvé que h converge vers le vecteur propre principal normalisé de la matrice WW^T, alors que a converge vers le vecteur propre normalisé de la matrice $W^T W$.

4.2. *L'algorithme SALSA*

Dans cette section, nous introduisons SALSA, « *the Stochastic Approach for Link-Structure Analysis* », un algorithme basé sur la théorie des chaînes de Markov proposé par Lempel et Morgan (2000 et 2001). L'algorithme utilise les propriétés d'une marche aléatoire effectuée sur une collection de documents hyperliés. Tout comme l'algorithme de Kleinberg, SALSA commence par construire une collection de base « *base set* » issue graphe des liens. SALSA est basé sur l'intuition qu'une page « *authoritative* » doit être visible de beaucoup de pages du set de données. Une marche aléatoire dans ce graphe visitera donc, avec une probabilité relativement élevée, un certain nombre d'*authorities*. La théorie des marches aléatoires est combinée à la notion de *hubs* et *authorities*, ce qui nous mène à la nécessité d'analyser deux chaînes de Markov différentes : la chaîne des *hubs* visités et la chaîne des *authorities* visitées, ce qui nous donne, pour chaque page, deux poids distincts : celui des *hubs* et celui des *authorities*. Pour générer les états de transition de chacune de ces chaînes, deux arêtes du graphe doivent être traversées, la première vers l'avant (en suivant un lien sortant), la seconde vers l'arrière (en suivant un lien rentrant) ou vice versa. Les poids des *authorities* sont définis comme étant la distribution stationnaire de la chaîne explorant en premier un lien aléatoire vers l'arrière et ensuite un vers l'avant, alors que les poids des *hubs* sont définis comme étant la distribution stationnaire de la chaîne explorant en premier un lien aléatoire vers l'avant et ensuite vers l'arrière.

Plus précisément, partant d'une collection de documents hyperliés, nous pouvons construire le graphe dirigé G. Soit $Back(i) = \{k : k \to i\}$, l'ensemble des nœuds qui pointent vers i, par

[3] La « *power method* » est une méthode numérique itérative pour le calcul du vecteur propre dominant d'une matrice symétrique pour le problème aux valeurs propres suivant :

$$\left\{ \begin{array}{lcl} h & \propto & Wa \\ a & \propto & W^T h \end{array} \right.$$

exemple les nœuds qui peuvent être atteints à partir de i en suivant un lien en arrière, et soit $Forw(i) = \{k : i \rightarrow k\}$, l'ensemble de tous les nœuds qui peuvent être atteints en partant de i en suivant un lien vers l'avant. $|Back(i)|$ représente le nombre de nœuds qui pointent vers i, tout comme $|Forw(i)|$ représente le nombre de nœuds vers lesquels i pointe.

Nous pouvons maintenant définir les deux matrices stochastiques, qui contiennent les probabilités de transition pour les chaînes de Markov respectivement pour les *hubs* et les *authorities*.

1. La matrice pour les *hubs*, H :

$$h_{i,j} = \sum_{k:k \in Forw(i) \cap Forw(j)} \frac{1}{|Forw(i)|} \frac{1}{|Back(k)|} \tag{3}$$

2. La matrice pour les *authorities*, A :

$$a_{i,j} = \sum_{k:k \in Back(i) \cap Back(j)} \frac{1}{|Back(i)|} \frac{1}{|Forw(k)|} \tag{4}$$

Un élément $a_{i,j} > 0$ signifie qu'au moins un nœud k pointe vers les deux nœuds i et j. Le nœud j est donc atteignable en partant du nœud i en deux étapes : la première en remontant le lien $i \rightarrow k$, la deuxième en suivant le lien $k \rightarrow j$.

Les équations (3) et (4) peuvent être écrites sous forme matricielle. On obtient dès lors un système similaire à celui obtenu en (2), pouvant aussi être résolu grâce à la « *power method* ».

5. Intégration des méthodes de liens dans GALILEI

Les utilisateurs jugent chacun un ensemble de documents. Tous les profils du système sont donc composés d'ensembles de documents. Le système va ensuite appliquer les algorithmes d'analyse de liens pour chaque ensemble de documents pertinents en étendant les graphes, comme dans la section 4.1. Le but de ces algorithmes est d'affecter à chaque document un poids *hub* et un poids *authority*. Les meilleures pages de chacune des catégories sont ajoutées aux profils. On espère ainsi que les documents les plus intéressants se retrouveront dans les profils correspondant au centre d'intérêt. De plus, les documents de type *hub*, pointent vers des documents contenus dans les profils ; on espère ainsi que deux profils, n'ayant pas jugé de documents communs mais appartenant à une même communauté virtuelle, se verront attribuer un même document *hub*. Il sera dès lors encore plus facile de les regrouper.

6. Méthodologie de validation de GALILEI

La méthodologie de validation définit des cycles de tests simulant le comportement des utilisateurs (Francq, 2003).

Seuls deux types de tests, permettant de quantifier l'apport des méthodes de liens, seront exposés ici. Il s'agit, d'une part, d'un test de simulation du système réel et, d'autre part, de tests permettant de mesurer les différents taux de comportement, exposés précédemment.

6.1. *Set de données et catégorisation*

Les différents tests seront effectués sur une collection de documents précatégorisés par centre d'intérêt. Cette collection a été construite par l'aspiration de plusieurs sites traitant de différents sujets. L'ensemble du set de données contient 5935 documents et est composé de six catégories, comme le montre le tableau 1.

6.2. *Simulation du système réel*

L'objectif est de simuler l'évolution du système en fonctionnement réel pendant une période de 200 unités de temps. Les profils doivent être calculés et regroupés en communautés virtuelles afin de pouvoir mesurer la convergence et la stabilité du système. On part du principe que chacune des catégories correspond à une **communauté virtuelle idéale**. Le but recherché est que le système simulé puisse converger vers le groupement idéal, c'est à dire que chaque communauté virtuelle corresponde à un centre d'intérêt prédéfini. Le test commence par l'initialisation et répète ensuite aléatoirement, jusqu'à la fin de la période, une phase de création ou de rétroaction.

Initialisation : deux profils issus de deux catégories différentes sont créés. On simule ensuite une série de jugements sur des documents (en respectant un certain pourcentage d'erreurs par rapport à la situation idéale). Le système calcule ensuite les deux profils et les communautés virtuelles. Le résultat est ensuite comparé au groupement idéal.

Rétroaction : pendant une ou plusieurs unités de temps, certains documents pertinents sont partagés entre les utilisateurs d'une même communauté. Le système recalcule ensuite les profils et les communautés virtuelles. Le résultat est ensuite comparé au groupement idéal.

Création : on crée un nouveau profil associé à une catégorie aléatoire parmi celles prédéfinies. Ceci permet de simuler l'arrivée d'un nouvel utilisateur dans le système et éventuellement l'arrivée d'un nouveau centre d'intérêt si celui-ci appartient à une nouvelle catégorie. On génère pour ce profil un ensemble de jugements sur des documents. Les profils et communautés virtuelles sont recalculés. On compare ensuite le groupement calculé au groupement idéal.

6.3. *Mesures*

***adjusted RAND Index* :** A chaque unité de temps, l'*adjusted RAND Index* (Hubert et Arabie, 1985) du système simulé est calculé. Cette valeur, comprise entre -1 et 1, représente la qualité de la réponse par rapport au groupement idéal. Lorsque cette valeur est nulle, le groupement calculé est aléatoire. Ce test permet d'observer l'évolution du système, ainsi que d'en déduire sa convergence et sa stabilité.

Nous avons vu précédemment (section 2), que d'autres critères permettent de quantifier la solution de l'algorithme de groupement. Afin de pouvoir relever ces différentes mesures, nous prenons une photographie du système à un instant t donné. Ce processus est réalisé une dizaine de fois de manière à avoir un aperçu du comportement du système à différents moments.

Nombre moyen de profils ayant jugé un même document : Ce test simule le système et

N°	Centre d'intérêt	Nbr
1	C++	1283
2	Html	535
3	Java	1741
4	Linux	972
5	Perl	368
6	Windows	1046

Tableau 1. Description du set de données

prend une configuration particulière au temps t ; il calcule, pour chaque document connu du système, le nombre de profils ayant émis un jugement sur le document en question. La moyenne des valeurs obtenues est ensuite calculée. Ce test n'a de sens que s'il est exécuté par différentes méthodes, afin que les courbes puissent être comparées entre elles. Il donne un aperçu du recouvrement entre profils et donc une bonne estimation de la facilité à les grouper. La comparaison des résultats sera discutée dans la section 7.2.

Taux moyen de comportement en accord : Ce test se base sur le taux de comportement en accord. Comme vu précédemment dans la section 1.3, plus la valeur de ce paramètre est élevée, plus le groupement sera aisé, puisque plus de profils auront jugé les même pages. Ce test simule le système, prend une configuration à un temps t et calcule la moyenne sur les différents profils. La comparaison des résultats sera discutée dans la section 7.3.

Taux moyen de comportement en désaccord : Ce test se base sur le taux de comportement en désaccord. Tout comme le taux de comportement en accord, plus la valeur de ce paramètre est élevée, plus le groupement est facilité, puisque ces deux profils ne devront pas être groupés ensemble. Le test se déroule de la même façon que le test exposé ci-dessus. Les résultats sont commentés dans la section 7.4.

6.4. *Résultats antérieurs*

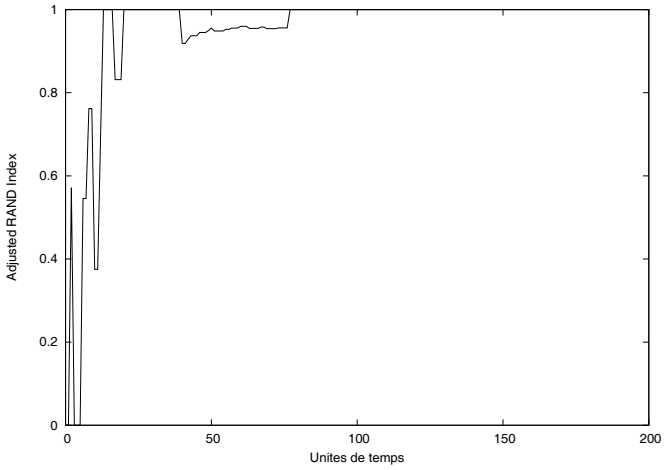


Figure 1. Résultat de la simulation du système réel

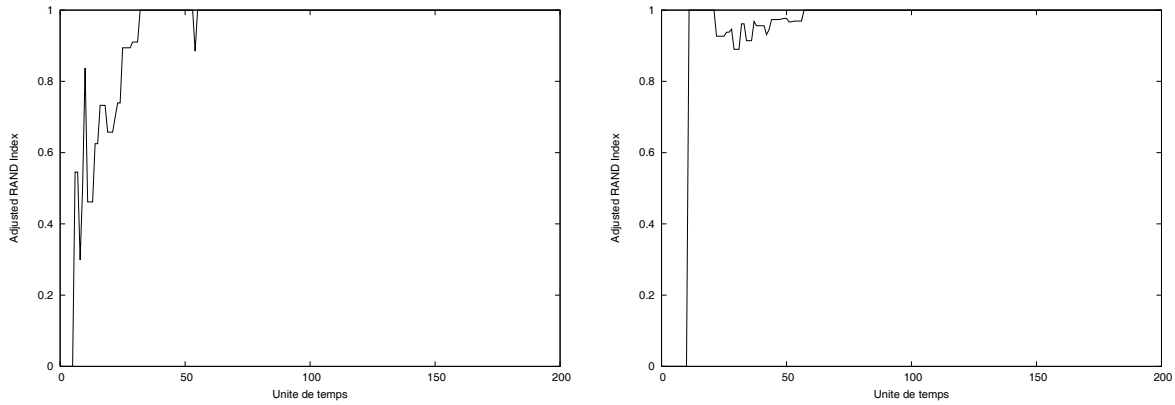
La figure 1 représente la simulation du système réel sans l'analyse des liens. On remarque clairement que le système est fortement instable lors des premières unités de temps. En effet, le peu de documents jugés par les utilisateurs ne donne pas suffisamment d'informations pour permettre aux algorithmes de grouper correctement les différents profils lorsque de nouveaux profils sont créés. Le système va ensuite converger vers le groupement idéal, mais il lui faudra encore quelques unités de temps avant de devenir complètement stable.

7. Méthode de validation pour les algorithmes de liens

Les différentes expériences ont pour but de quantifier l'apport des méthodes de liens au système. Les tests seront toujours effectués avec et sans les méthodes d'analyse de liens, afin de pouvoir comparer les résultats.

7.1. *Simulation réelle du système*

La figure 2(a) montre les résultats de la simulation du même système qu'à la section 6.2, mais utilisant en plus l'algorithme de liens HITS. En comparant la figure 2(a) à la figure 1, on peut tirer un certain nombre de conclusions. Le système présente de moins bons résultats lors des premières itérations, ce qui est dû au non respect des conditions initiales de la méthode (voir section 4.1.). En effet, les profils ne contiennent pas assez de documents pour construire un graphe dirigé correct et les documents ainsi rajoutés perturbent le groupement. Par la suite, le système converge de la même manière, mais il devient stable plus rapidement.



(a) avec la méthode HITS (b) avec la méthode SALSA

Figure 2. Simulation du système réel

En comparant la figure 2(b), qui présente les résultats de la simulation en utilisant SALSA, à la figure 1, on constate que lors des dix premières itérations la méthode fausse complètement le groupement, pour la même raison que celle évoquée précédemment. Passé ce cap, le groupement converge beaucoup plus rapidement et de façon beaucoup plus stable.

En comparant les deux figures 2(a) et 2(b), on constate que, sur cette collection de documents, la méthode SALSA donne de meilleurs résultats, puisque le système converge plus vite et qu'il reste plus stable.

7.2. *Nombre moyen de profils ayant jugé un même document*

Plus les valeurs de ce test sont importantes, plus les groupes seront déterminés facilement. Les deux courbes ainsi obtenues (voir figure 3) montrent l'impact des méthodes de liens sur le système. En effet, grâce à l'analyse des liens, d'autres documents considérés comme pertinents sont rajoutés aux profils. On constate donc un accroissement du nombre de pages jugées en commun par plusieurs profils, et donc un groupement plus aisé.

7.3. *Taux moyen de comportement en accord*

On a vu précédemment (section 2) que si le taux de comportement en accord entre deux profils est supérieur à un seuil imposé, alors ces deux profils issus d'un même centre d'intérêt sont mis dans la même communauté virtuelle. Comme le montre la figure 3, le taux moyen de comportement en accord augmente quelle que soit la méthode de liens utilisée. Ces résultats indiquent bien que trouver le meilleur groupement sera plus aisé lorsque l'on adjoint au système les méthodes d'analyse de liens.

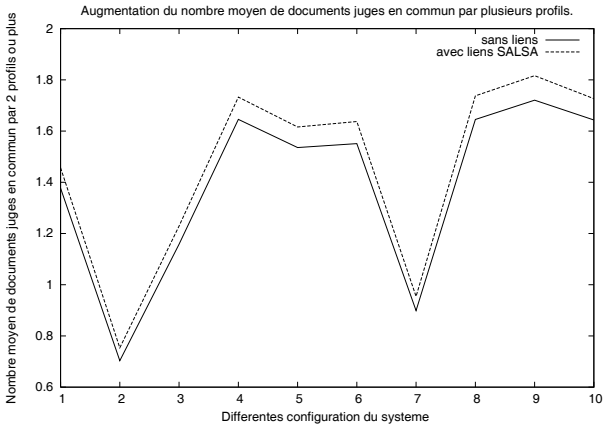


Figure 3. Average Ratio

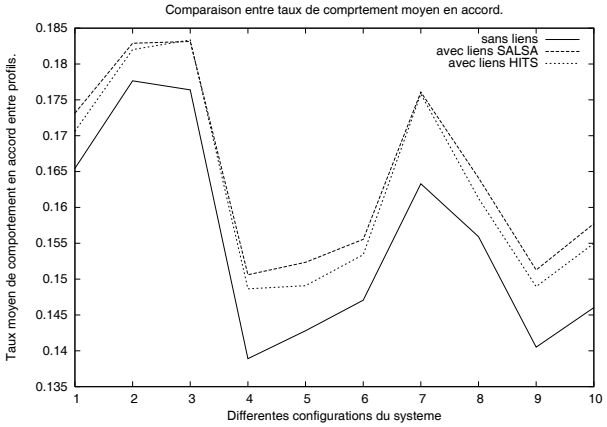


Figure 4. Taux de comportement en accord

7.4. *Taux moyen de comportement en désaccord*

Comme vu précédemment (section 2), si le taux de comportement de désaccord entre deux profils est supérieur à un seuil imposé, alors les deux profils, issus de centres d'intérêt différents, ne seront pas placés dans la même communauté virtuelle.

On voit clairement, sur la figure 4, l'amélioration obteneu lorsque l'on utilise les méthodes d'analyse de liens. Ceci s'explique facilement par l'ajout automatique de documents aux profils. En effet, deux profils venant de communautés virtuelles différentes recevront chacun des documents propres à leur centre d'intérêt. Cet ajout d'informations séparera d'autant plus les deux profils et le groupement en sera d'autant plus aisé.

8. Développements futurs : l'algorithme hybride

Nous remarque que les deux méthodes d'analyse de liens donnent plus ou moins le même genre de résultats quel que soit le type de test, pour la collection de documents utilisée pour les tests. D'autres collections de documents ne donnent pas des résultats aussi homogènes pour les différentes méthodes. C'est pourquoi une méthode hybride est en cours de développement.

Les deux algorithmes, présentés à la section 4, ont certes une série de propriétés très intéressantes, mais ont aussi leurs propres limites. Un des problèmes les plus indésirables rencontrés avec l'algorithme HITS est ce qu'on appelle le « *topic drift* » : l'algorithme converge vers la communauté la plus hyperliée au sein du set de base, mais cette communauté n'a parfois

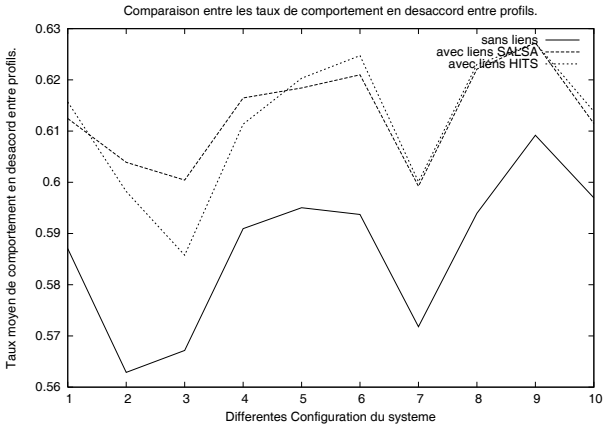


Figure 5. Taux de comportement en désaccord

rien à voir avec le centre d'intérêt du groupe. SALSA peut être vu comme un premier moyen de contourner ce type d'instabilité ; certaines modifications, détaillées ci-dessous, peuvent encore améliorer les résultats.

L'idée retenue consiste à appliquer un seuil durant la phase de calcul des *hubs* et *authorities*, afin de ne tenir compte que des pages dont le poids est élevé ; cela revient à dire que lors de la phase de calcul on ne garde que les documents considérés comme bon *hub* ou bonne *authority*. Nous détaillerons ici brièvement la méthode que l'on a appelé « *Authority-Mean-Treshold*», laquelle consiste à appliquer un seuil lors du calcul des poids des *hubs*. Lors de ce calcul, pour une page i, l'algorithme ne prend pas en compte toutes les pages pointées par i mais seulement celles qui font partie des meilleures (le nombre de ces meilleurs pages est représenté par le seuil).

Cette philosophie peut aussi bien être appliquée lors de la phase de calcul des *authorities*, ainsi que lors du calcul des *hubs* et des *authorities* simultanément. Cet algorithme sera paramétrable de manière à pouvoir choisir la valeur des seuils, ainsi que le moment où de leur application.

9. Conclusions

Nous avons intégré plusieurs algorithmes d'analyse de liens au sein du projet GALILEI, le but premier étant d'utiliser ces liens pour déterminer les meilleurs documents *hub* et *authority* d'un profil. Ces documents sont ensuite proposés au profil en question dans l'espoir qu'ils soient du même centre d'intérêt. Les résultats nous confirment la validité des algorithmes. Le nombre de documents associés aux profils augmente d'avantage et les documents ainsi ajoutés sont pertinents par rapport aux centres d'intérêt du profil. Les communautés virtuelles sont donc plus vite définies et de façon plus précise. Le système convergera et se stabilisera plus rapidement.

Références

Fogarty M. et Bahls (2002). Information Overload : Feel the pressure ? *The Scientist*, vol. (16/16).

Francq P. (2003). *Structured and Collaborative Search : An integrated approach to share documents among users*. PhD Thesis. Université Libre de Bruxelles.

Hubert L. et Arabie P. (1985). Comparing partitions. *Journal of Classification* : 193-218.

Kleinberg J.M. (1998). Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithm* : 668-677.

Lawrence S. et Giles C.L. (1998). Searching the World Wide Web. *Science*, vol. (280) : 98-100.

Lawrence S. et Giles C.L. (1999). Accessibility of information on the Web. *Nature*, vol. (400) : 107-109.

Lempel R. et Morgan S. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *9th International World Wide Web Conference*.

Lempel R. et Morgan S. (2001). The Stochastic approach for link-structure analysis. *Transaction on Information Systems*, vol. (19/2) : 131-160.