Évaluation d'outils de reformulation interactive de requêtes

Aurélien Saint Requier* — **Gérard Dupont*** ** — **Sébastien Adam*** — **Yves Lecourtier***

* *Université de Rouen, LITIS*
BP 12, 76801 Saint-Etienne-du-Rouvray, France

** *EADS Defence and Security, Information Processing Control and Cognition*
Val de Reuil, France

saintrequier.a@gmail.com

RÉSUMÉ. *Dans le cadre de travaux de recherche sur la modélisation de l'utilisateur et sur le développement d'un système de recherche d'information apprenant, nous présentons une nouvelle approche d'évaluation d'outils de reformulation interactive de requêtes prenant en compte le temps au cours d'une session de recherche. En suivant un protocole d'expérimentation utilisateur adapté, nous montrons que les performances globales d'un outil de reformulation de requêtes ne sont pas significatives de ses performances au cours d'une session de recherche et varient selon l'utilisateur. En conséquence, nous justifions l'intérêt de faire collaborer différents modules de suggestions de requêtes et nous proposons de modéliser le comportement de l'utilisateur à partir d'une analyse fine de ses interactions afin de sélectionner l'outil de reformulation de requêtes le plus performant à chaque instant.*

ABSTRACT. *As part of a user modeling and learning information retrieval research, we introduce a new approach of interactive query reformulation modules evaluation considering the search session time. Using an adapted user experimental protocol, we show that effectiveness is very changing for different user search sessions and during a same user search session and thus that suggestion mechanisms cannot be compared globally. In this way, we propose to use a fine analysis of user behavior to segment search session in state which may provide enough information to select the more effective suggestion mechanism.*

MOTS-CLÉS : *Recherche d'information, Modélisation utilisateur, Retour de pertinence, Reformulation de requêtes, Évaluation, Expérimentation utilisateur*

KEYWORDS: *Interactive information retrieval, User modeling, Relevance feedback, Query reformulation, Information retrieval, Evaluation, User experiment*

1. Introduction

Au cours des dix dernières années, les progrès en matière de production et de diffusion de documents numériques ont conduit à un accroissement exponentiel de la quantité de données stockées et disponibles. Les Systèmes de Recherche d'Information (SRI) ont donc été développés pour simplifier l'accès à cette masse d'information. Dans les SRI conventionnels, les utilisateurs expriment leur besoin en information sous la forme d'une requête, généralement composée d'une courte liste de mots clés ou d'une question (Jansen *et al.*, 2000). Or, il est évident qu'une requête construite à partir d'un ou deux mots clés permet de ne transcrire qu'une partie du besoin utilisateur. Si la formulation du besoin est une tâche complexe pour l'utilisateur de SRI, elle l'est d'autant plus lorsque ce besoin est imprécis et lorsque le corpus ne lui est pas familier. De ce fait, la nécessité d'approfondir la compréhension du besoin de l'utilisateur dans son contexte devient un point essentiel à l'amélioration des SRI. L'une des voies de recherche fréquemment adoptée est l'enrichissement de la formulation du besoin de l'utilisateur par un retour de pertinence implicite ou explicite.

Dans le but d'aider le SRI à répondre le plus pertinemment possible au besoin de l'utilisateur, des outils de reformulation du besoin ont été développés. Baeza-Yates et Ribeiro-Neto (Baeza-Yates *et al.*, 1999) ont défini trois types d'outils de reformulation de requêtes qui dépendent des données utilisées : des données provenant du jugement de la pertinence par l'utilisateur, des données récupérées à partir de l'ensemble de documents initialement retournés par le SRI et des données globales provenant d'une collection de documents et/ou de ressources externes.

Le retour de pertinence, introduit en 1971 pour les problèmes de reformulation de requêtes (Rocchio, 1971), est une méthode reconnue du premier type d'outils de reformulation de requêtes. Cependant, cette méthode s'est montrée peu populaire pour les utilisateurs. En effet, les utilisateurs sont souvent réticents à fournir un retour de pertinence explicite, qui est généralement ressenti comme une charge supplémentaire lors de leurs interactions avec le SRI (Spink *et al.*, 2000). Un axe de travail plus récent porte sur l'enregistrement de données provenant des interactions de l'utilisateur avec le système comme retour de pertinence implicite. L'utilisation de telles données est détaillée dans (Joachims *et al.*, 2005). Le retour de pertinence implicite est utilisé dans plusieurs systèmes de reformulation de requêtes et apporte de nombreux avantages aux SRI (White, 2004).

Le second type d'utilisation des données est basé sur une technique habituellement appelée retour de pertinence *local* ou *aveugle*. Celui-ci analyse les documents retournés par la requête initiale soumis au SRI. Étant donné que les premiers documents retournés par le SRI abordent le thème général de la requête, la sélection de termes provenant de ces documents permet de fournir des nouveaux termes pertinents (Buckley *et al.*, 1994). Cette technique s'est montrée efficace, plus particulièrement pour les requêtes courtes.

Pour finir, le dernier type d'information utilisé repose sur des techniques dites globales, correspondant à une analyse de corpus pour établir des relations entre les

termes (Xu *et al.,* 1996). Le but de cette technique est de compléter la requête initiale en utilisant de l'information globale provenant du corpus entier ou éventuellement de ressources extérieures. L'approche globale construit un thésaurus basé sur des relations termes à termes avec différentes approches d'agglomération de termes : *thésaurus* construits à partir d'associations de termes et de phrases (Jing *et al.,* 1994), *thésaurus* formés à partir de plusieurs sources d'expansion (Wordnet, dictionnaires généraux)(Mandala *et al.,* 1999) ou *thésaurus* basés sur la génération automatique de concepts à partir d'un ensemble de documents (Chang *et al.,* 2006).

De nos jours, les utilisateurs sont de plus en plus pris en compte dans le processus de recherche dans le but d'adapter le comportement du SRI au comportement de l'utilisateur. Ainsi, les approches de personnalisation utilisant le comportement de l'utilisateur, comme dans (Chirita *et al.,* 2007) et (Jie *et al.,* 2006), combinent ces différentes techniques. Nous pouvons aussi citer les travaux de Jansen et McNeese dans (Jansen *et al.,* 2005) qui présentent un SRI qui intègre 5 modèles de suggestion utilisant un mécanisme de combinaison statique.

La technique couramment utilisée et éprouvée pour l'évaluation des différents outils de reformulation de requêtes (Belkin *et al.,* 2001) suit le modèle TREC (Text REtrieval Conference) qui trouve ses origines dans le modèle des études de Cleverdon à Cranfield (Cleverdon, 1962). Il repose sur une évaluation globale des performances des différentes méthodes de reformulation. Dans cet article, nous montrons qu'une évaluation globale d'outils de reformulation ne traduit pas les performances au cours d'une session de recherche. Nous proposons des pistes pour répondre à ce problème, notamment une approche d'évaluation qui prend en compte le temps lors d'une session de recherche. Dans une première section, nous présentons brièvement l'approche classique d'évaluation des outils de reformulation de requêtes. Dans une seconde section, nous présentons le système expérimental et les outils de suggestion évalués, puis nous détaillons le protocole d'expérimentation suivi. Enfin, nous présentons les résultats de l'expérimentation et nous concluons sur l'introduction d'une nouvelle approche d'évaluation des outils de reformulation de requêtes.

2. Évaluation classique des outils de reformulation de requêtes

L'approche classique d'évaluation des outils de suggestion est principalement celle adoptée lors de la conférence annuelle de recherche documentaire (TREC). Pour chaque TREC, le National Institute of Standards and Technology (NIST) fournit des corpus de documents avec les tâches (requêtes, questions, tâches de recherche, ...) correspondantes. Les participants procèdent à l'expérimentation de leur SRI sur les données fournies et retournent à NIST l'ensemble des listes de documents retrouvés. En utilisant la technique du "pooling" (Kuriyama *et al.,* 2002), des experts jugent la pertinence des listes de documents retournées par chaque système par rapport à une liste définie (par des juges) de documents pertinents et évaluent les performances globales des SRI. Le protocle d'évaluation TREC permet donc une comparaison des performances des SRI, grâce à un ensemble de tâches, de corpus de documents et de

métriques cohérent. Cependant, des critiques justifiées montrent que le modèle d'évaluation proposé par TREC ne convient pas pour l'évaluation de SRI interactif (Borlund *et al.*, 1997). En effet le modèle d'évaluation TREC n'implique pas l'utilisateur dans le jugement de la pertinence de la liste renvoyée par le système, liste construite à partir des besoins du moment de l'utilisateur. Or, l'utilisateur a des connaissances et un besoin d'information différents de ceux des experts.

Pour répondre à la problématique de l'évaluation des SRI interactifs, TREC a développé des tâches interactives (Interactive Tracks) qui sont spécifiques aux évaluations de ce type de SRI (voir (Over, 2001) ou (Dumais *et al.*, 2005) pour une description complète). Les tâches interactives se différencient des autres tâches par l'implication de l'utilisateur dans le protocole d'évaluation. En effet, les tâches interactives consistent à simuler un besoin d'information sur des thèmes précis, afin que l'utilisateur formule des requêtes pour combler ce besoin et juge de la pertinence des documents retournés par le système pour former des listes utilisateurs. Ce protocole permet donc d'étudier les problématiques liées au comportement de l'utilisateur face à un SRI, les interactions de l'utlisateur avec le SRI et les mécanismes propres au SRI. On notera que dans la suite, les références aux expérimentations TREC ne prennent ici pas en compte cette tâche TREC interactive qui n'a que très rarement été utilisée pour l'évaluation d'outils de reformulation du besoin. Les limites de ces évaluations proviennent du fait que les listes utilisateurs sont toujours évaluées en fonction d'une liste construite par des experts, ce qui renvoie aux problèmes cités dans le paragraphe précédent, et aussi à un problème de généralisation des résultats observés lors des évaluations TREC (Jones, 2006).

Dans ce contexte, notre étude aborde une nouvelle approche d'évaluation au cours du temps lors d'une session de recherche. La section suivante présente le système expérimental utilisé et les outils de reformulation de requêtes intégrés au système.

3. Le système expérimental

Dans cette section, nous présentons dans un premier temps le système expérimental utilisé pour l'évaluation des différents services de reformulation de requêtes que nous détaillons dans un second temps.

3.1. *Présentation générale du système*

Le système mis en place pour l'expérimentation est basé sur la plateforme WebLab[1], développée en open source[2] par l'entreprise EADS dans le but de faciliter le développement d'applications dédiées au traitement de documents multimédia. Celle-ci repose sur une décomposition en couches (voir figure 1) et sur les architectures

1. http ://weblab-project.org
2. http ://forge.ow2.org/projects/weblab/

orientées services qui permettent la construction d'applications à partir de briques élémentaires respectant des interfaces standardisées. Le système est donc une "application" basée sur le socle d'intégration WebLab. Il met en œuvre des composants en Web Services pour le traitement et des portlets intégrées dans le portail permettent la composition de l'interface utilisateur.

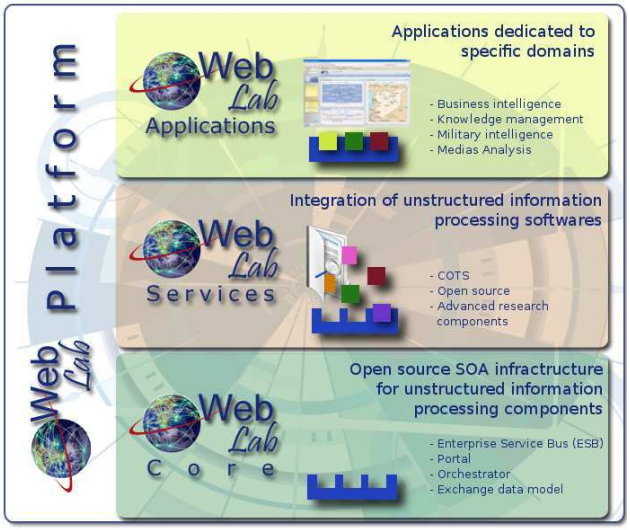


Figure 1. *Les différentes couche de la plateforme WebLab*

Le système développé est un moteur de recherche qui a pour objectif d'adapter la réponse présentée à l'utilisateur en exploitant des données d'interaction. Le système dispose de plusieurs moteurs de suggestion de requêtes ainsi que d'un mécanisme de récupération en temps réel des actions utilisateurs (temps de lecture, clics...). Il extrait donc des caractéristiques globales issues de ces données comportementales afin de faire varier son mode de réponse en fonction de celles-ci. L'apprentissage des liens entre comportements et réponse adaptée se fait par la mise en œuvre d'un apprentissage par renforcement utilisant la théorie des processus de décision markoviens (MDP). La figure 2 donne une vue globale de l'architecture du système. Cette architecture permet la récupération en temps réel des données d'interaction (par des capteurs spécifiques au niveau de l'interface utilisateur), un apprentissage temps réel multi-utilisateurs et multi-sessions et aussi l'intégration de différents services dédiés à la recherche d'information dont plusieurs services de reformulation de requêtes. Cet article ne porte pas sur le moteur d'apprentissage de comportement directement mais a pour objectif de confirmer les hypothèses sur lesquelles repose le système.

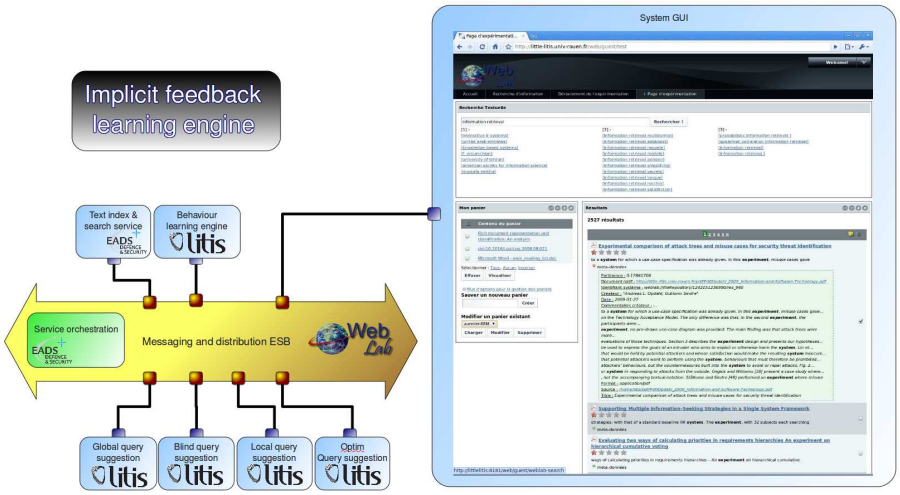


Figure 2. *Vue globale de l'architecture du système.*

3.2. *Les services de reformulation de requêtes*

Pour notre étude, nous avons intégré au système quatre services de reformulation de requêtes proposant chacun un mode de fonctionnement différent que nous avons nommé : **global**, **blind**, **local** et **optim**.

Le mode **global** est un mode de reformulation incrémental basé sur un modèle de capitalisation des sessions utilisateurs. Un modèle simple de pondération est utilisé pour chaque requête q_i déjà reçue par le système en fonction d'une requête entrante q_0 et éventuellement du dernier document consulté d_0 :

$$w(q_i, d_0) = freq(q_i)\,(sim(q_0, q_i) + \alpha sim(d_0, q_i)) \qquad [1]$$

avec $freq()$ correspondant à la fréquence de la requête dans la base des historiques des sessions passées, $sim()$ étant un opérateur de calcul de similarité basé sur la représentation vectorielle des données textes et α un facteur de pondération fixé empiriquement.

La formule précédente permet donc d'associer un poids à ces requêtes et donc de sélectionner les dix requêtes systèmes les plus pertinentes vis à vis de la dernière requête q_0 et du dernier document d_0. On notera qu'il n'est pas indispensable d'avoir la référence d_0 (si aucun document n'a été récemment consulté) et l'on revient alors à une simple proposition des requêtes les plus similaires et les plus populaires (Bar-Yossef *et al.*, 2008), d'où le qualificatif de globale pour cette approche.

Le mode **blind** est basé sur un retour de pertinence dit "aveugle" (blind feedback) introduit dans (NIST, 1995) : les cinq premiers résultats d'une requête sont supposés pertinents et ainsi on extrait les termes les plus pertinents issus de ces documents en exploitant simplement leur représentation vectorielle (voir (Thompson *et al.,* 1995) pour une description complète). Pour ne pas simplement proposer des termes uniques, on combine ces termes à la requête initiale, simulant ainsi une suggestion de requêtes par un système de suggestion de termes.

Le mode **local** est basé sur la nouvelle approche de construction de modèle de besoin en temps réel en fonction des interactions utilisateurs. L'implémentation initiale proposée exploite les observations x_t du comportement de l'utilisateur qui a la possibilité d'executer des actions λ sur les documents u_i qui lui sont présentés. Ces actions recouvrent toutes les interactions de l'utilisateur avec le système : requête, clic sur un menu ou un document, sélection de morceaux de texte, impression... Ce concept reprends et étant donc les données que l'on peut retrouver dans les données de logs systèmes. Parmi ces actions, un sous-ensemble $\hat{\Lambda}$ contient celles qui sont reconnues comme ayant une interprétation possible vis à vis de la pertinence d'un document.

$$\rho_{t,u_i} = \begin{cases} w_{\lambda_t} & \text{if } \exists x | \left(u_t = u_i \wedge \lambda_t \in \hat{\Lambda} \right) \\ 0 & \text{otherwise} \end{cases} \quad [2]$$

Ainsi, à un instant t, la pertinence d'une unité d'information u_i sur laquelle l'utilisateur a effectué une action λ correspond à un poids fixé à priori w_λ si cette action fait partie de l'ensemble $\hat{\Lambda}$ des actions révélatrices de pertinence (ou de non pertinence avec dans ce cas un poids négatif). Cette même pertinenence est arbitrairement fixée à 0 dans le cas contraire. Un effet perte de mémoire avec latence est ensuite appliqué sur cette pondération afin d'obtenir à tout instant une estimation de la pertinence (positive ou négative) des documents qui ont été présentés à l'utilisateur. Une fois les documents ordonnés par pertinence, on procède de maniére similaire au mode **blind** pour déduire les requêtes suggérées.

Le mode **optim** repose sur le principe d'optimisation de requêtes. La technique utilisée reprend une approche d'apprentissage automatique de requêtes utilisant le paradigme "Inductive Query By Example" (Chen *et al.,* 1998) et l'algorithme évolutionnaire multiobjectif NSGA-II (Deb *et al.,* 2002). Une population initiale de termes candidats, extraits en utilisant une représentation vectorielle d'un ensemble de documents jugés pertinents, est crée pour former des requêtes candidates. Ensuite, deux objectifs sont optimisés : un objectif de similarité entre une liste initiale (les 10 premiers documents retournés par la requête initiale) et les listes candidates (les 10 documents retournés par les requêtes candidates) et un objectif de complexité correspondant au nombre de termes des requêtes candidates. Les requêtes candidates obtenant les meilleurs compromis entre ces deux objectifs appartenant donc au Front de Pareto du problème multiobjectif défini sont suggérées à l'utilisateur.

La section suivante présente le protocole d'expérimentation suivi dans le but d'évaluer la pertinence des requêtes suggérées de ces différentes approches.

4. Protocole expérimental

L'expérimentation mise en place correspond à une évaluation de laboratoire avec des utilisateurs réalistes. Les sous-sections suivantes présentent les différents aspects de l'étude.

4.1. *Sujets*

L'expérimentation s'est déroulée au sein du laboratoire LITIS avec la participation volontaire de 10 utilisateurs chacun devant accomplir 3 tâches de recherche. Les participants, qualifiés au travers de questionnaires démographiques, étaient principalement des étudiants en 5ème année (60%) ainsi que des enseignants chercheurs (40%). Tous appartennaient au laboratoire d'informatique mais ne faisaient pas partie de l'équipe responsable de l'étude. A l'issue du questionnaire, ils ont tous affirmé avoir une bonne expérience des outils de recherche d'information sur le web. La sélection des sujets n'a pas suivi un modèle d'échantillonnage strict, cependant les caractéristiques et les compétences des sujets de l'étude ont été mises en correspondance avec les utilisateurs réels d'une application de recherche ou de veille d'information (c.à.d. un niveau d'éducation élevé et une expérience de la recherche d'information).

4.2. *Documents et corpus*

Le corpus utilisé contenait 6000 documents de type article scientifique au format pdf. Il a été initié avec 3000 documents issus des bibliographies personnelles des volontaires. Les domaines abordés ont alors été listés afin de compléter la collection avec 3000 nouveaux documents de revues récentes (2008-2009), issus d'une bibliothèque sur internet en gardant les mêmes thématiques. Les documents eux-mêmes étaient variables en taille, du simple résumé de communication (2-4 pages) jusqu'au manuscrit de thèse (plus de 200 pages).
L'objectif était ici de constituer un corpus sur des thématiques maîtrisées par les sujets tout en limitant le biais apporté par les documents connus. Cela permet d'assurer l'intérêt des sujets pour les tâches de recherche et de contraindre les tâches sur des thématiques connues. Celles-ci ont été restreintes à 7 domaines principaux tels que la *reconnaissance de caractères manuscrits*, l'*apprentissage statistique* ou encore le *traitement de signal*.

4.3. *Tâches de recherche et protocole d'expérimentation*

Le protocole expérimental suivi reprend les bases du protocole expérimental "hybride" défini par Borlund (Borlund, 2000) qui est un compromis entre une approche centrée sur le système et une approche centrée sur l'utilisateur. Ce protocole repose sur la réalisation de simulation de besoin d'information et du jugement de la pertinence par l'utilisateur.

Nous avons donc défini sept thèmes de recherche, 3 types de tâches de recherche, chacune d'entre elle étant déclinée selon 3 niveaux de difficulté (facile, moyen et difficile, respectivement niveau 1, 2 et 3) selon le descriptif du besoin fourni au sujet et les résultats attendus. La thématique de la tâche de recherche était laissée libre, cependant le sujet était tenu de préciser son choix parmis les domaines principaux référencés et le participant ne devait pas choisir un même thème pour différentes tâches. On notera que dans les résultats, les données issues du premier sujet ont été retirées des analyses pour avoir changé de thématique en cours de session. Ceci reprend donc le concept de simulation de besoin d'information, adapté à notre corpus de documents. Les 3 types de tâches représentent 3 étapes du processus de recherche d'information définies dans (Kuhlthau, 1993) : exploration, reformulation et collecte d'information précise, que nous avons codées respectivement type E, R et C. Pour minimiser l'effet d'apprentissage durant l'expérimentation, l'attribution des tâches a suivi un carré gréco-latin (Tague-Sutcliffe, 1992) en fonction du type et du niveau des tâches de recherche.

Chaque participant suivait tout d'abord un entrainement guidé d'une dizaine de minutes pour se familiariser librement avec le système. Un temps d'exécution effectif (hors questionnaire) de 20 min était conseillé pour chaque tâche de recherche. Cependant pour préserver un comportement réaliste, il n'a pas été appliqué de contrainte forte à la durée de la session de recherche et certaines sessions ont pu durer jusqu'à 45min selon l'utilisateur. Au final, chaque session d'expérimentation durait environ une heure et demie au cours de laquelle les participants devaient réaliser trois tâches. Pour chaque tâche, les participants sauvegardaient les documents qu'ils jugeaient pertinents et répondaient à un questionnaire de fin de tâche.

Nous allons, dans la section suivante, présenter les résultats de l'expérimentation décrite ci-dessus.

5. Résulats et discussion

Dans cette section, nous nous intéressons à l'évaluation comparative des quatre outils de reformulation interactive de requêtes présentés dans la section précédente au cours d'une session de recherche utilisateur. Nous présentons dans un premier temps les résultats de l'évaluation, puis dans un second temps, une discussion sur les résultats obtenus.

5.1. *Résultats*

La réalisation de l'expérimentation utilisateur nous a permis de récupérer des sessions de recherche pour lesquelles les utilisateurs ont construit un ensemble de documents pertinents par rapport à un besoin défini (la tâche de recherche). À chaque session de recherche correspond donc une liste L_u de documents jugés pertinents par l'utilisateur. Pour l'évaluation des outils de reformulation interactive de requêtes, nous avons rejoué les sessions utilisateur et évalué la liste obtenue par chaque requête suggérée des différents modes de reformulation par rapport à la liste finale de documents L_u au cours de la session de recherche. Nous avons calculé la moyenne de la précision et du rappel obtenus sur l'ensemble des requêtes suggérées par chaque outil, pour chaque requête formulée par l'utilisateur au cours des sessions de recherche :

$$precision = \frac{|P \cap R|}{|R|} \qquad rappel = \frac{|P \cap R|}{|P|} \qquad [3]$$

avec P l'ensemble des documents pertinents et R l'ensemble des documents retrouvés. À partir de ces deux mesures, nous en avons déduit la $F_1 - mesure$ (Van Rijsbergen, 1979) :

$$F_1 - mesure = 2 * (precision * rappel)/(precision + rappel) \qquad [4]$$

Nous avons choisi une valeur de coupure de dix documents, représentant le nombre de résultats visibles sur la première page du système expérimental. L'intérêt de ce mode d'évaluation est de proposer une mesure des performances en précision-rappel, du point de vue de l'utilisateur.

Dans un premier temps, nous avons calculé les performances globales en prenant la moyenne de la $precision$, du $rappel$ et de la $F_1 - mesure$ sur toutes les sessions de recherche. Cette évaluation globale a pour objectif de se rapprocher des évaluations en "pooling" de type TREC. Les performances obtenues sont présentées sur la figure 3. Nous constatons tout d'abord qu'il n'y a que pour le rappel qu'il existe un consensus clair qui révèle la supériorité du mode de suggestion "global" avec un écart de performance supérieur à 30%. Du point de vue de la précision, on constate tout d'abord des performances quantitatives relativement faibles (un maximum de 0.05). Ceci s'explique par le choix de laisser la composition de la liste des documents pertinents P, à la discrétion des utilisateurs. Il en résulte que le nombre de documents pertinents est relativement faible et cela pénalise naturellement les valeurs de précision. Néanmoins on constate une légère supériorité du mode "optim" et surtout une nette infériorité du mode "local". Cette analyse se retrouve globalement sur la F1-mesure qui synthétise les deux métriques précédentes. En conclusion, on notera qu'à l'exception du mode "local", les autres modes semblent avoir des performances globales équivalentes avec des écarts dans le rapport précision/rappel. Le mode de reformulation local semble donc être rejeté et il nous est difficile de départager les autres modes de reformulation.

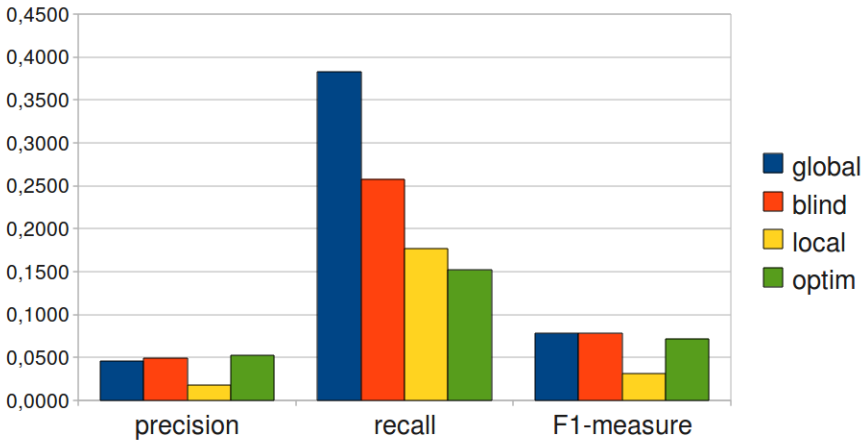
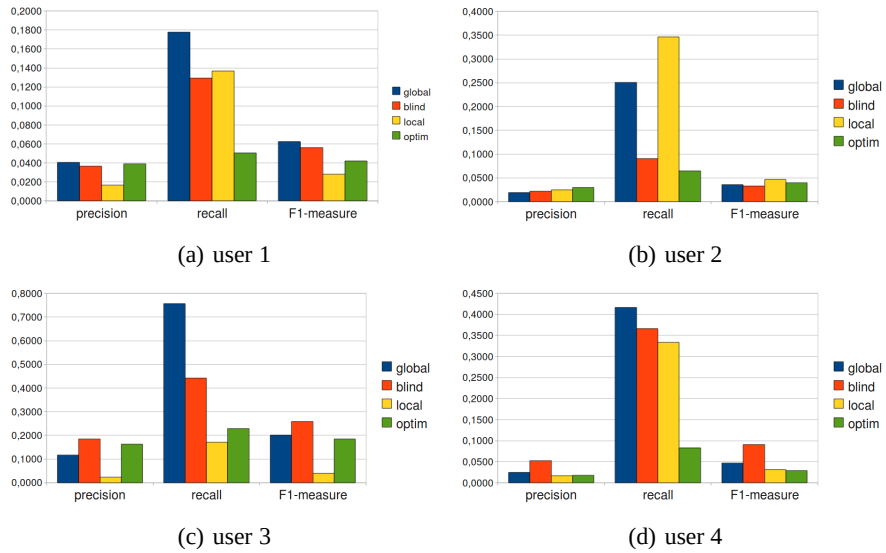


Figure 3. *Comparaison globale des 4 modes de reformulation sur toutes les sessions de recherche*

La figure 4 montre une analyse différente pour laquelle nous avons isolé les sessions de quatre utilisateurs. Nous observons la variabilité des performances des différents modes de reformulation sur chacun des utilisateurs. Pour l'utilisateur 2, le mode de reformulation local obtient une performance équivalente (en précision) ou meilleure (en rappel ou en F1-mesure) que les trois autres, et pour les autres modes, les performances sont globalement intéressantes, soit d'un point de vue précision, soit d'un point de vue rappel.

Pour appuyer l'idée que les performances des modes de reformulation sont liées au comportement de l'utilisateur, la figure 5 montre les performances des quatres modes de reformulation au cours de quatre sessions de recherche sélectionnées, en se limitant à la F1-mesure pour faciliter la lecture. La figure 5 donne donc un aperçu des performances de chaque mode de suggestion sur quatre sessions différentes en conservant l'aspect temporel : les performances ne sont pas moyennées pour toute la session, mais évaluées à chaque moment important de la session (à noter : les échelles de temps ne sont pas identiques). Nous observons que les quatre modes de reformulation donnent des performances variables en fonction de moments particuliers. Si nous comparons avec les performances globales, nous constatons qu'elles ne peuvent traduire les performances des modes de reformulation au cours des sessions de recherche. Pour exemple, nous prenons le graphique 4(d). Le mode de reformulation blind est nettement plus performant que le mode de reformulation local. Or, si nous regardons le graphe 5(d), nous remarquons que le mode de reformulation blind n'est pas performant en fin de session de recherche contrairement au mode de reformulation local.



(a) user 1 (b) user 2

(c) user 3 (d) user 4

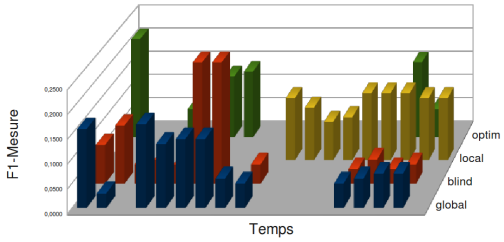
Figure 4. *Comparaison globale des modes de reformulation pour 4 utilisateurs*

De plus, aucun mode de reformulation ne domine globalement les autres lors d'une session de recherche. En effet, Nous observons des dominances des modes pour des intervalles de temps lors des sessions de recherche. Par exemple, nous remarquons que le mode de reformulation "local" domine à 100% sur un intervalle pour trois sessions de recherche. Les modes de reformulation "global", "blind" et "optim" se montrent plus performants en début de session de recherche alors qu'ils semblent inefficaces en fin de session de recherche, contrairement au mode de reformulation "local". Ces observations peuvent s'expliquer par le principe de fonctionnement des modes de reformulation, en particulier du mode "local". En effet, le mode "local" doit capitaliser un certain nombre d'interactions utilisateurs afin d'être pertinent.

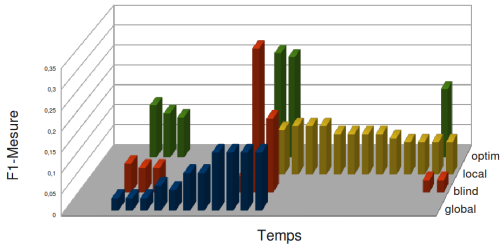
Les performances globales d'un mode de suggestion ne traduisent donc pas correctement les performances au cours d'une session de recherche. Nous discuterons donc dans la sous-section suivante d'une approche d'évaluation des modes de reformulation au cours d'une session de recherche pour une adaptation des modes au besoin utilisateur.

5.2. *Discussion*

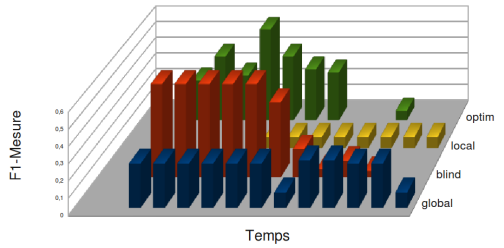
Dans la section précédente, nous avons montré que les performances globales d'un mode de reformulation ne sont pas significatives des performances au cours d'une session de recherche. Nous avons montré qu'un mode de reformulation globalement



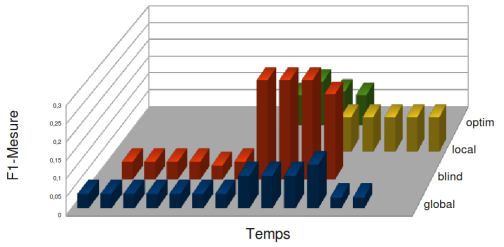
(a) user1-tâche E01



(b) user2-tâche R03



(c) user3-tâche E01



(d) user4-tâche R03

Figure 5. *Comportement des modes de reformulation lors de différentes sessions utilisateurs avec en abscisse le temps et en ordonée la F1-mesure*

performant pour une session pouvait avoir des faibles performances lors d'une session d'un utilisateur particulier ou même au cours de certains intervalles de temps de la même session de recherche. Enfin, nous avons montré qu'un mode de suggestion globalement moins performant que d'autres pouvait avoir localement des pics de performances importants vis à vis des autres modes.

Cette expérimentation montre donc que réaliser l'évaluation globale d'outils de reformulation par un ensemble de requêtes de test ne peut être significatif et que les performances globales masquent une partie des apports de certaines approches particulières. Nous pensons qu'adopter une vision "temps réel" des performances permet de mettre en évidence le bénéfice qu'il y aurait à mettre en œuvre simultanément différentes approches du support à la recherche, chacune ayant leur apport à différents moments au cours d'une session de recherche.

Dans l'optique d'adapter le mode de suggestion au besoin utilisateur, nous avons introduit une approche d'évaluation de modes de suggestion en fonction du temps dans une session de recherche. Cette nouvelle approche s'oppose à une évaluation globale des performances, et insiste sur la nécessité d'évaluer les outils de reformulation au cours d'une session de recherche pour adapter le mode de support à la recherche au comportement de l'utilisateur. Pour cela, nous proposons d'utiliser une analyse spécifique des interactions entre l'utilisateur et le système afin de pouvoir segmenter les sessions de recherche en différents états. L'objectif par la suite est alors de savoir associer à chaque état le mode de support à la recherche qui est le plus performant. Une description compléte de cette approche et de la mise en œuvre d'apprentissage par renforcement fera l'objet d'une prochaine publication.

6. Conclusion

Dans le cadre de travaux de recherche sur la modélisation de l'utilisateur et sur le développement d'un SRI apprenant, nous avons réalisé de premières expérimentations utilisateurs en suivant un protocole d'évaluation adapté. Les résultats de l'évaluation nous ont permis de montrer que les performances globales d'un outil de reformulation ne sont pas significatives de ses performances au cours d'une session de recherche et varient selon les utilisateurs. De nouvelles expérimentations utilisateurs sur un corpus Web sont en cours de réalisation pour confirmer les résultats des premières expérimentations.

Notre travail s'inscrit dans le cadre du développement d'un SRI apprenant capable de modéliser le comportement de l'utilisateur. La suite de ce travail consiste à définir une méthode permettant de segmenter les différents états du comportement de l'utilisateur au cours d'une session de recherche pour faire correspondre l'outil de reformulation le plus adapté à chaque état.

7. Bibliographie

Baeza-Yates R. A., Ribeiro-Neto B., *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

Bar-Yossef Z., Gurevich M., « Mining Search Engine Query Logs via Suggestion Sampling », *Proceedings of the VLDB Endowment*, p. 54-65, 2008.

Belkin N. J., Cool C., Kelly D., Lin S. J., Park S. Y., Perez-Carballo J., Sikora C., « Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval », *Information Processing & Management*, vol. 37, n 3, p. 403 - 434, 2001.

Borlund P., « Experimental Components for the Evaluation of Interactive Information Retrieval Systems », *Journal of Documentation*, vol. 56 No. 1, p. 71-90, 2000.

Borlund P., Ingwersen P., « The development of a method for the evaluation of interactive information retrieval systems », *Journal of Documentation*, vol. 53, p. 225-250, 1997.

Buckley C., Salton G., Allan J., Singhal A., « Automatic Query Expansion Using SMART : TREC 3 », *TREC*, p. 69-80, 1994.

Chang Y., Ounis I., Kim M., « Query reformulation using automatically generated query concepts from a document space », *Information Processing & Management*, vol. 42, n 2, p. 453 - 468, 2006.

Chen H., Shankaranarayanan G., She L., « A machine learning approach to inductive query by examples : An experiment using relevance feedback ID3, genetic algorithms, and simulated annealing », *Journal of the American Society for Information Science*, vol. 49, p. 693-705, 1998.

Chirita P.-A., Firan C. S., Nejdl W., « Personalized Query Expansion for the Web », *SIGIR 2007 Proceedings*, p. 7 - 14, 2007.

Cleverdon C. W., Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems, Technical report, Cranfield Coll. of Aeronautics, Cranfield, England, 1962.

Deb K., Pratap A., Agarwal S., Meyarivan T., « A fast and elitist multiobjective genetic algorithm : NSGA-II », *Evolutionary Computation, IEEE Transactions on*, vol. 6, p. 182-197, 2002.

Dumais S. T., Belkin N., *TREC : Experiment and Evaluation in Information Retrieval*, MIT Press, chapter The Interactive TREC Track : Putting the user into search., p. 123-153, 2005.

Jansen B. J., McNeese M. D., « Evaluating the effectiveness of and patterns of interactions with automated searching assistance : Research Articles », *J. Am. Soc. Inf. Sci. Technol.*, vol. 56, p. 1480-1503, 2005.

Jansen B. J., Spink A., Saracevic T., « Real life, real users, and real needs : a study and analysis of user queries on the web », *Information Processing & Management*, vol. 36, n 2, p. 207-227, March, 2000.

Jie H., Zhang Y., « Personalized Faceted Query Expansion », *Proc of the First SIGIR'2006 Workshop on Faceted Search*, 2006.

Jing Y., Croft, An Association Thesaurus for Information Retrieval, Technical report, Amherst, MA, USA, 1994.

Joachims T., Granka L., Pan B., Hembrooke H., Gay G., « Accurately interpreting clickthrough data as implicit feedback », *SIGIR '05 : Proceedings of the 28th annual international ACM*

SIGIR conference on Research and development in information retrieval, ACM Press, New York, NY, USA, p. 154-161, 2005.

Jones K. S., « What's the value of TREC : is there a gap to jump or a chasm to bridge ? », *SIGIR Forum*, vol. 40, p. 10-20, June, 2006.

Kuhlthau C., *Seeking Meaning : A Process to Library and Information Services*, Norwood, 1993.

Kuriyama K., Kando N., Nozue T., Eguchi K., « Pooling for a Large-Scale Test Collection : An Analysis of the Search Results from the First NTCIR Workshop », *Inf. Retr.*, vol. 5, n 1, p. 41-59, 2002.

Mandala R., Tokunaga T., Tanaka H., « Combining multiple evidence from different types of thesaurus for query expansion », *SIGIR '99 : Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, p. 191-197, 1999.

NIST, « Third Text REtrieval Conference (TREC-3) », *Proc. of the Text REtrieval Conference*, 1995.

Over P., « The TREC interactive track : an annotated bibliography », *Inf. Process. Manage.*, vol. 37, n 3, p. 369-381, 2001.

Rocchio J. I., « Relevance feedback in information retrieval. In The SMART Retrieval System », *Prentice-Hall*p. 313-323, 1971.

Spink A., Jansen B. J., Ozmultu C. H., « Use of query reformulation and relevance feedback by Excite users », *Internet Research : Electronic Networking Applications and Policy*, vol. 10, n 4, p. 317-328, 2000.

Tague-Sutcliffe J., « The Pragmatics of Information Retrieval Experimentation Revisited », *Inf. Process. Manage.*, vol. 28, n 4, p. 467-490, 1992.

Thompson P., Turtle H., Yang B., Flood J., « TREC-3 ad-hoc retrieval and routing experiments using the WIN system », *in* , Gaithersburg (ed.), *Proc. of the Text REtrieval Conference*, USA, p. 211-218, 1995.

Van Rijsbergen C. J., *Information retrieval*, 2 edn, Butterworths, London, 1979.

White R. W., Implicit Feedback for Interactive Information Retrieval, PhD thesis, Department of Computing Science Faculty of Computing Science Mathematics and Statistics University of Glasgow, 2004.

Xu J., Croft B. W., « Query Expansion Using Local and Global Document Analysis », *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 4-11, 1996.