Choisir la meilleure configuration d'un système de recherche d'information. Adaptation à la difficulté du besoin

Anthony Bigot, Sébastien Déjean, Josiane Mothe

1. *Institut de Recherche en Informatique de Toulouse, Université de Toulouse, 118 Route de Narbonne, F-31062 Toulouse* CEDEX *9*
{anthony.bigot,josiane.mothe}@irit.fr

2. *Institut de Mathématiques de Toulouse, Université de Toulouse, 118 Route de Narbonne, F-31062 Toulouse* CEDEX *CEDEX 9*
sebastien.dejean@math.univ-toulouse.fr

RÉSUMÉ. *Les campagnes telles que Text REtrieval Conference (TREC) offrent un cadre qui permet d'évaluer des systèmes de recherche d'information (RI). L'évaluation utilise des mesures qui se basent sur une moyenne des résultats obtenus pour un ensemble de besoins en information : les succès et échecs sur chacun des besoins sont masqués. Dans un premier temps, cet article propose une analyse qui prend en compte la diversité des résultats et qui a pour but de sélectionner le système de RI le plus apte à traiter les besoins selon leur difficulté. L'évaluation montre un gain maximal en robustesse de 68 % et de 10 % sur les performances. Dans un second temps, nous montrons qu'il est possible d'apprendre la meilleure configuration de système en s'appuyant sur un sous-ensemble de documents dont la pertinence est connue.*

ABSTRACT. *Evaluation campaigns such as Text REtrieval Conference (TREC) offer a framework to evaluate information retrieval (IR) systems. Generally, these evaluation results are averaged over information needs so successes and failures on specific needs are hidden. At first, this article proposes an analysis which takes into account the diversity of results and select the IR system best suited to address the information needs according to their difficulty. An experiment is driven in order to select the best systems according to the information need difficulty. Evaluation results show a maximum gain of 68 % in reliability and of 10 % on performances. In a second step, we show that it is possible to learn the best system configuration based on a subset of documents which relevance is known.*

MOTS-CLÉS : *évaluation en RI, rang des systèmes, difficulté des besoins, classifications*

KEYWORDS: *IR evaluation, system ranks, information need difficulty, clustering*

DOI:10.3166/DN.147.2.125-147

1. Introduction

Les campagnes d'évaluation telles que Text REtrieval Conference (TREC) [1] ont permis de grandes avancées dans le monde de la recherche d'information (RI). Ces campagnes sont basées sur le modèle d'évaluation de Cranfield (Cleverdon *et al.*, 1966). Selon ce modèle, un système à évaluer recherche les documents à restituer à l'utilisateur pour un ensemble de besoins d'information et ce à partir d'une collection figée de documents ; la liste des documents pertinents étant connue pour chaque besoin. En comparant la liste de documents restituée par le système avec la liste de documents attendue pour un besoin donné, il est possible de calculer des mesures de performances. Ce cadre peut être utilisé pour comparer les systèmes entre eux sur le même ensemble de données. Plusieurs études ont montré la variabilité des résultats des systèmes (Harman, Buckley, 2009). Ainsi, un système S_1 peut être très performant sur un besoin A mais échouer sur un besoin B et un système S_2 peut avoir des performances inverses. Notre travail vise à privilégier le système S_1 pour traiter le besoin A et le système S_2 pour traiter le besoin B. La variabilité des résultats est issue de la variabilité qui existe au sein des systèmes de RI d'une part, et de celle qui existe parmi les besoins en information d'autre part.

Nous nous intéressons à la sélection du meilleur système parmi un ensemble afin de tirer profit de leur variabilité. Dans la littérature, différents travaux se sont intéressés à combiner les résultats provenant de différents systèmes. Lors de la campagne TREC3 ont eu lieu les premières expérimentations sur la fusion des systèmes de recherche d'information (Fox, Shaw, 1994). Dans cet article, l'auteur analyse une série de méthodes de fusion des systèmes de RI, dont la fonction CombMNZ, et montre une amélioration significative des résultats par rapport aux performances des systèmes initiaux. L'étude de ces méthodes de fusion montre que l'utilisation de rangs au lieu de la similarité produit de meilleurs résultats dans la combinaison des résultats lorsque les systèmes initiaux ont des rangs très différents (Lee, 1997). De nombreux auteurs se sont employés à la mise au point de nouveaux algorithmes de fusion des systèmes de RI existants. Par exemple, Lillis (Lillis *et al.*, 2006) étudie les résultats de différents algorithmes de RI et propose un modèle de fusion basé sur les probabilités (probFuse). L'auteur compare ces résultats à ceux obtenus par CombMNZ et montre une amélioration de la MAP allant de 15 % à 50 % selon la taille de la collection utilisée durant la phase d'apprentissage.

Des analyses ont également porté sur les différents paramètres intervenant dans la combinaison comme le nombre initial de systèmes, leurs performances et leur diversité (Liu *et al.*, 2012). Cette étude considère les rangs obtenus par les systèmes et montre que la performance de la combinaison n'augmente pas forcément avec le nombre de systèmes utilisés. L'auteur montre également que les performances sont améliorées lorsque les systèmes initiaux sont relativement performants et qu'ils sont suffisamment différents selon le critère RSC (*Rank-Score Characteristics*).

1. http~://trec.nist.gov/

Une seconde source de variabilité dans les performances des systèmes de RI provient de la variabilité des besoins en information et du contexte dans lequel ils sont formulés. La littérature montre de nombreuses analyses menées pour catégoriser les besoins en information. Mothe (Mothe, Tanguy, 2005) analyse 16 traits linguistiques afin de prédire la difficulté d'un besoin en information. Les auteurs montrent que 3 de ces traits ont un impact significatif sur les mesures de rappel et/ou de précision. Carmel (Carmel, Yom-Tov, 2010) s'intéresse à la prédiction de la difficulté d'un besoin en information nouvellement formulé et donc non connu au préalable.

La variabilité des besoins en information et du contexte dans lesquels ils sont formulés a conduit la RI à évoluer vers une RI adaptative (Joho *et al*., 2008). Les premiers travaux dans ce domaine s'intéressent aux modifications à apporter au processus de RI selon les informations tirées du contexte dans lequel il se déroule (Brown, Jones, 2001). D'autres travaux se focalisent sur la formulation du contexte sous forme d'un besoin en information afin d'améliorer la pertinence des résultats restitués par les systèmes de RI (Menegon *et al*., 2009). Cependant, la mise en place d'un cadre d'évaluation de tels algorithmes comportent de nombreuses difficultés (Mizzaro *et al*., 2008).

En s'appuyant sur l'hypothèse d'une variabilité dans la manière de traiter les besoins en information, Kompaore (Kompaoré *et al*., 2007) étudie le meilleur choix possible de systèmes de RI selon des critères linguistiques du besoin en information. Dans ce contexte, Bigot (Bigot *et al*., 2011) propose une méthode d'apprentissage du meilleur système de recherche d'information par classe de difficulté des besoins. Les auteurs montrent une amélioration significative de la mesure MAP de 10 % pour les besoins classés faciles et difficiles sur la collection de documents test. Le gain en performance augmente à 24 % sur les besoins classés moyennement difficiles à traiter.

Dans l'analyse que nous proposons ici, nous émettons l'hypothèse que des besoins différents doivent être traités par des systèmes de RI différents afin de tirer avantage de la variabilité de ces derniers comme dans (Bigot *et al*., 2011). Dans un premier temps, nous supposons que les meilleurs systèmes pour traiter les besoins les plus faciles sont différents des systèmes plus aptes à traiter les besoins les plus difficiles. Nous cherchons à sélectionner les meilleurs systèmes selon des classes de difficulté des besoins ; ces classes sont définies par les performances des systèmes de RI. Contrairement aux travaux proposés dans (Bigot *et al*., 2011), cet article utilise une combinaison de plusieurs mesures d'évaluation des systèmes et les sélectionne selon deux aspects différents : leur performance et leur robustesse. Dans un second temps, nous supposons que pour chaque requête on dispose d'un sous-ensemble de documents pour lesquels la pertinence est connue. Nous montrons qu'il est alors possible d'apprendre la meilleure configuration à associer à une requête et que la méthode améliore les résultats lorsque de nouveaux documents sont considérés. L'application dans un cadre réel concerne le cas des requêtes répétées qui sont nombreuses. Par ailleurs, ces travaux sont en lien avec ceux développés dans les approches sélectives. Par exemple, He et Ounis (He, Ounis, 2004) suggèrent une méthode pour choisir parmi plusieurs modèles de pondération des termes en fonction de la requête. Les requêtes sont représentées par diverses caractéristiques qui sont utilisées pour les regrouper. L'étape d'apprentissage associe

le meilleur schéma de pondération des termes à chaque groupe de requêtes. Après apprentissage, une nouvelle requête est d'abord associée à un des groupes existants et le système appris est utilisé pour la traiter. Chifu et Mothe (Chifu, Mothe, 2014) proposent une méthode d'expansion de requêtes sélective qui se base sur des prédicteurs de difficulté des requêtes, prédicteurs linguistiques et statistiques. Le modèle de décision est appris par un algorithme SVM (*Support Vector Machine*). Les performances sont améliorées par rapport à des méthodes non sélectives. Nos travaux proposent également des méthodes sélectives qui permettent de choisir le meilleur système parmi un ensemble.

L'article est structuré comme suit : la section 2 présente le contexte dans lequel les systèmes de RI sont évalués et explique la construction du jeu de données utilisé pour l'analyse. Dans la section 3, nous présentons la méthodologie de sélection du meilleur système par groupe de requêtes et définissons une implémentation pratique des méthodes proposées. La section 4 analyse l'expérimentation et propose une évaluation des méthodes de sélection. La section 5 présente une méthode pour apprendre le meilleur système pour chaque requête indépendamment. Enfin, la section 5 conclut et propose des pistes d'analyse sur le sujet. Cet article est une version étendue de la communication (Bigot, 2013) présentée à la conférence INFORSID.

2. Contexte et données

2.1. Text Retrieval Conference

Dans cette étude, nous cherchons à choisir les meilleurs systèmes de RI pour traiter les besoins en informations. Pour cela, nous avons besoin d'étudier les performances d'un nombre suffisamment grand de systèmes. Cranfield (Cleverdon *et al*., 1966) propose un cadre d'évaluation qui permet d'étudier les performances de systèmes de RI : les différents systèmes traitent des besoins en information prédéfinis sur une collection fixée de documents. Selon ce même cadre d'évaluation des systèmes, TREC propose chaque année des défis à relever pour les systèmes de RI. Nous avons choisi d'utiliser la collection de la tâche AdHoc issue de TREC. Pour cette tâche, TREC fournit une collection composée de documents, de besoins en information et de jugements de pertinence pour chacun des besoins.

Les besoins en information sont composés d'un titre, d'une description et d'une partie narrative. La requête est une représentation interne que chaque système de RI définit librement à partir du besoin en information traité. Les jugements de pertinence permettent d'évaluer les performances des systèmes. TREC fournit un outil d'évaluation (*trec_eval*) qui permet de calculer de nombreuses mesures de performances pour chaque couple (système ; besoin en information).

2.2. Structure des données

Dans ces travaux, nous nous appuyons sur différents types de données : des exécutions des participants à TREC et des exécutions que nous avons produites. Ces données sont présentées ci-dessous.

2.2.1. Résultats officiels issus de TREC

Nous utilisons les résultats obtenus par les participants à la tâche AdHoc TREC 8. La collection est composée de 528155 documents et de 50 besoins d'information. Chaque système participant doit produire une liste ordonnée de 1000 documents pour chaque besoin d'information. Ainsi chaque participant utilise un ou plusieurs systèmes de RI pour rechercher les documents de la collection susceptibles de répondre au besoin traité. Le résultat de cette recherche, appelée *run* ou exécution, est une liste ordonnée des documents restituée pour chaque besoin en information. A TREC 8, il y a au total 129 exécutions, qui correspondent soit à des systèmes produits par différentes équipes, soit à différentes configurations d'un même système. Il s'agit des résultats officiels produits dans le cadre de TREC. (Chrisment *et al*., 2005) a montré que les résultats obtenus par différentes configurations d'un même système sont statistiquement plus proches que les résultats obtenus par différents systèmes. Dans cet article, nous ne faisons pas de distinction entre ces différents types d'exécutions. L'inconvénient d'utiliser ces données est qu'elles ne sont disponibles que pour une seule collection. En effet, ne disposant pas des codes des systèmes, il est impossible de produire les résultats sur d'autres collections. Par ailleurs, parmi les 130 mesures de performances, calculées pour chacun des 129 exécutions et pour chacun des 50 besoins en information de la collection, nous sélectionnons les 118 mesures qui prennent leurs valeurs dans [0 ; 1].

2.2.2. Résultats construits

Nous avons également construit des résultats d'exécutions à partir de la plateforme Terrier. Par rapport aux précédentes données, l'avantage est que nous pouvons produire des résultats sur différentes collections. De plus, nous connaissons de façon détaillée les modules et paramètres qui ont permis de les obtenir. Nous avons ainsi produit 100 exécutions en faisant varier différents paramètres. Nous avons utilisé les collections de TREC AdHoc 7 et 8 qui ont l'avantage de se baser sur les mêmes documents ; nous avons ainsi 100 besoin d'information à notre disposition, 50 par an.

2.2.3. Représentation matricielle des données

Quelle que soit leur origine, officielles ou construites, nous avons besoin d'un format des données adapté à l'utilisation de méthodes mathématiques pour mener les analyses. Nous avons donc structuré les données sous forme de matrice. La matrice des données initiales est un cube de données à trois dimensions. La première dimension est constituée des besoins en information ; la seconde dimension représente les différents systèmes et la troisième dimension correspond aux mesures de performances. Chaque

cellule du cube contient la valeur de la mesure de performance pour l'exécution d'un système de RI sur un besoin en information.

2.3. Redondance des mesures de performance

Baccini (Baccini *et al*., 2011) montre que les 130 mesures de performances issues de trec_eval sont redondantes et les classe en 6 groupes. Les auteurs montrent également qu'un sous-ensemble composé d'une mesure représentative de chaque groupe est suffisant. Selon ces critères, nous retenons les 6 mesures suivantes pour construire le jeu de données dit "réduit" de notre analyse : la "mean average precision" (MAP) ; la précision à 30 (P30) ; la précision à 100 (P100) ; le rappel exact (exact_recall) ; la préférence binaire (bpref_retall) ; le rappel à 30 (R30). Baccini et al. (2011) ont montré que la corrélation entre les mesures moyennes des systèmes de RI en utilisant les données complètes et les mesures moyennes en utilisant les données réduites est supérieure à 0,99, ce qui confirme le choix de ces 6 mesures. Il faut noter que la mesure MAP reste la mesure la plus utilisée dans le domaine. (Voorhees, Harman, 1997) indique que cette mesure est particulièrement bien adaptée à la comparaison globale des systèmes entre eux.

3. Meilleurs systèmes de recherche d'information

3.1. Méthodes de classement des systèmes de recherche d'information

Dans cette section, nous proposons deux méthodes de sélection des meilleurs systèmes de RI. Ces méthodes sont générales et ne dépendent ni des besoins en information, ni des systèmes de RI considérés. Les résultats des méthodes sont ensuite comparés.

3.1.1. Calcul des rangs.

Nous considérons le calcul à besoin en information fixé. Pour ce besoin, les systèmes de RI sont ordonnés par ordre décroissant de la mesure considérée. Le système ayant la plus forte valeur se retrouve en tête de liste et le rang 1 lui est attribué. Le système suivant de la liste est le second meilleur système et le rang 2 lui est attribué et ainsi de suite jusqu'au dernier système qui obtient le rang maximum. Si plusieurs systèmes obtiennent la même valeur de mesure, ils sont ex-aequo et obtiennent un rang équivalent à la moyenne des rangs attribués s'il n'y avait pas d'ex-aequo. L'algorithme d'attribution des rangs reprend son cours normal après le dernier ex-aequo.

3.1.2. Calcul de la mesure moyenne d'un système pour un besoin donné.

Pour chacune des méthodes de sélection, nous commençons par créer la matrice des mesures moyennes. Cette matrice est composée de *s* lignes (pour les systèmes) et de *m* colonnes (pour les mesures). Chaque cellule de la matrice est la performance moyenne $\bar{M}^{(S)}$ de chaque système *S* pour chaque mesure *M* donnée par :

$\bar{M}^{(S)} = \frac{1}{b}\sum_{B=1}^{b} m_B^{(S)}$ où $m_B^{(S)}$ est la valeur de la mesure attribuée pour la performance du système *S* sur le besoin *B*.

Dans la suite de cette section, $\bar{M}_j^i$ désignera la valeur de la moyenne de la mesure *j* pour le système *i*. De même, r^i désignera le rang du système *i*. Notez que ce calcul dépend des systèmes et des besoins ; la méthode est donc indépendante du nombre de documents présents dans la collection.

3.1.3. Mesure moyenne et rangs de la mesure moyenne.

La mesure moyenne "MeMo" d'un système consiste à calculer la moyenne générale de chaque système sur la matrice des mesures moyennes (FIGURE 1). Ensuite, les systèmes de RI sont ordonnés selon cette valeur pour obtenir les rangs de la mesure moyenne MeMo.

$$s\ ;systèmes \left\{ \overbrace{\begin{pmatrix} \bar{M}_1^1 & \cdots & \bar{M}_m^1 \\ \vdots & \ddots & \vdots \\ \bar{M}_1^s & \cdots & \bar{M}_m^s \end{pmatrix}}^{m\ ;mesures} \Longrightarrow \overbrace{\begin{pmatrix} \bar{\bar{M}}^1 \\ \vdots \\ \bar{\bar{M}}^s \end{pmatrix}}^{MeMo} \Longrightarrow \overbrace{\begin{pmatrix} r^1 \\ \vdots \\ r^s \end{pmatrix}}^{Rang\ ;MeMo} \right. \quad (1)$$

Figure 1. Calculs de la mesure moyenne et rangs de la mesure moyenne des systèmes

Les données réduites et la méthode de classement ci-dessus seront utilisées dans la suite de cet article afin de déterminer quels sont les meilleurs systèmes selon la difficulté des besoins à traiter. Nous opposerons les systèmes obtenus par les rangs MeMo avec les systèmes obtenus en travaillant directement avec la MeMo.

3.2. Définition des meilleurs systèmes de recherche d'information

Les systèmes de RI ne traitent pas les besoins en information de la même manière : là où certains systèmes réussissent, d'autres échouent, et inversement (Harman, Buckley, 2009). Nous faisons l'hypothèse que les systèmes de RI sont plus ou moins performants selon la difficulté des besoins traités. Ainsi, nous regroupons les besoins d'information en fonction de leur difficulté, puis nous définissons la qualité d'un système par deux caractéristiques : la performance et la robustesse.

3.2.1. Regroupement des besoins en information

Afin de déterminer différentes classes de besoins en information selon un critère de difficulté et afin de déterminer si elles nécessitent d'être traitées par des systèmes de RI différents, nous procédons à un regroupement des besoins en classes.

À partir des données présentées dans la section 2.2, il est possible de calculer une mesure moyenne représentative de la performance du système selon l'ensemble de ces performances sur un besoin donné. Ensuite, pour chaque besoin, le rang MeMo de

chaque système est calculé comme le montre la FIGURE 2.

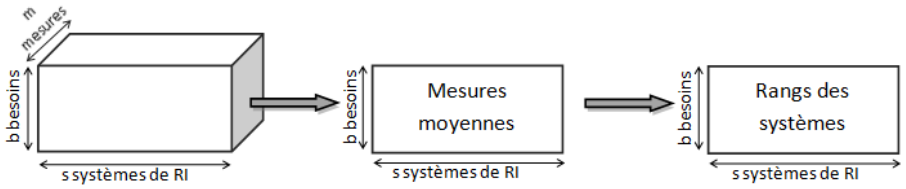

Figure 2. Transformation des mesures en rangs de la mesure moyenne

Pour grouper les besoins en information, une classification ascendante hiérarchique (CAH) est appliquée aux rangs MeMo. La CAH considère dans un premier temps chaque besoin en information comme un groupe à part entière puis assemble les deux groupes qui correspondent au critère choisi jusqu'à obtention d'une unique classe (Lebart *et al.*, 2006). Pour cette étude, le critère de Ward (Ward, 1963) est utilisé ; il minimise le gain en inertie entre les rangs d'un même groupe à chaque étape.

L'équation de décomposition de l'inertie nous indique que plus les besoins à l'intérieur des groupes sont similaires (i.e. plus l'inertie à l'intérieur de chaque groupe est petite), plus les groupes sont différents les uns des autres.

Le choix de 4 classes de besoins est raisonnable selon le critère usuel de rupture dans les inerties des groupes issus des différentes partitions possibles (Lebart *et al.*, 2006). Ici, le nombre de classes obtenues par la CAH est dépendant d'un choix subjectif que nous avons fait. Afin de diminuer le biais introduit par ce choix, une méthode de ré-allocation dynamique non-supervisée est appliquée sur les classes obtenues. Les centres de gravité des classes ont été utilisés comme point de départ de l'algorithme des K-moyennes (Jain *et al.*, 1999). Cette méthode consiste à agréger chaque élément un à un au groupe dont la moyenne est la plus proche. La figure 3 représente les boîtes à moustaches de chaque besoin d'information en considérant les différentes valeurs de MeMo obtenues par les différents systÃ¨mes. Une boîte à moustaches pour un besoin donné inclut, dans la partie rectangulaire, les valeurs du quartile Q1 au quartile Q3 ainsi que la médiane. Ainsi, la représentation est faite de sorte que :

- 25% des valeurs soient sous la boite (premier quartile : Q1) ;
- 50% des valeurs soient sous le trait gras (second quartile : médiane) ;
- 25% des valeurs soient au dessus de la boite (troisième quartile : Q3).

Les points isolés correspondent à des valeurs situées à une distance de la boite supérieure à une fois et demi l'écart inter-quartile (Q3-Q1).

Sur la figure 3, chaque groupe de besoins est représenté par un niveau de gris. Le groupe de besoins en information pour lesquels 75 % des systèmes ont leur MeMo supérieure à 0,5 est donc qualifié de groupe facile à traiter (en blanc). De la même manière, le groupe de besoins pour lesquels les systèmes obtiennent un Q1 supérieur à 0,3 et un Q3 inférieur à 0,6 est qualifié de groupe moyen (en gris clair) ; le troisième

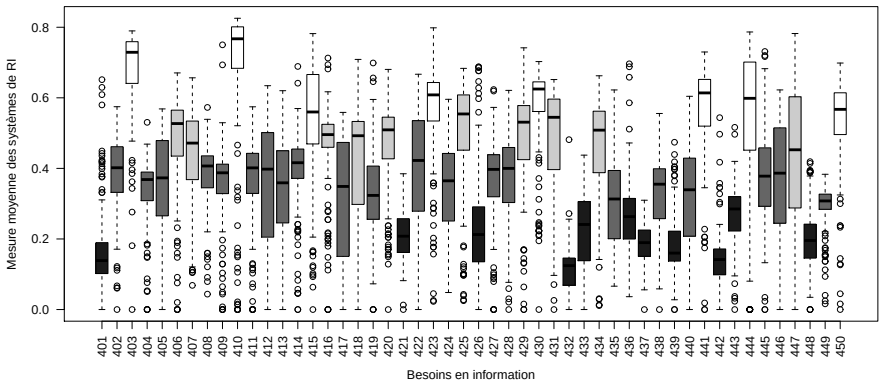

Figure 3. Caractérisation des classes de besoins en information

groupe est qualifié de groupe difficile à traiter (en gris foncé) avec une médiane des MeMo des systèmes comprise entre 0,3 et 0,4. Enfin, le groupe de besoins pour lesquels 75% des systèmes obtiennent une MeMo inférieure à 0,3 est qualifié de groupe très difficile (en noir).

La structure des besoins en information est commune à tous les besoins, celle-ci n'explique donc pas la différence de difficulté. Cependant, Harman et al. (Harman, Buckley, 2009) montre que les besoins utilisés lors des campagnes TREC les plus difficiles ont des caractéristiques propres. Ces derniers sont constitués de termes ayant plusieurs sens que les systèmes ne traitent pas en intégralité. Le nombre de documents pertinents dans la collection n'a en revanche pas d'effet notoire sur la difficulté.

3.2.2. Performance et robustesse des systèmes de recherche d'information

La première question qui se pose est quelle définition donner au meilleur système pour un groupe de besoins donné. Nous avons retenu deux approches en fonction de l'objectif visé. Le meilleur système peut présenter le meilleur classement possible pour la grande majorité des besoins en information, et avoir un très mauvais classement pour quelques-uns. Il s'agirait d'un système de RI qui donne les meilleurs résultats mais pas en toutes circonstances. Nous appelons ce système de RI le système le plus *performant*.

Alternativement, le meilleur système peut être un système avec un bon classement (pas nécessairement le meilleur) pour l'ensemble des besoins en information considérés. Il s'agit d'un système robuste en permanence mais ne donnant pas les meilleurs résultats possibles. Ce système de RI est qualifié de système le plus *robuste*. Idéalement, pour un groupe de besoins en information donné, le meilleur système de RI aura les deux propriétés susmentionnées : les meilleurs performances et cela pour l'ensemble des besoins en information. Dans les faits, un tel système n'existe pas for-

cément et les meilleurs systèmes évalués sur nos données connaissent des failles pour au moins un besoin. C'est pourquoi dans la suite de l'analyse, nous essayerons de trouver le système le plus performant d'une part, et le système le plus robuste d'autre part ; et ce pour chacun des groupes de besoins introduits dans la section 3.2.1.

4. Sélection des meilleurs systèmes en fonction de la difficulté des requêtes

4.1. Détection des meilleurs systèmes par groupe de besoins

Nous définissons deux règles qui correspondent aux données que nous traitons afin de déterminer les candidats aux deux types de *meilleurs* systèmes pour chacun des groupes de besoins considérés :

– candidats au système le plus *performant* : ces systèmes sont parmi les dix premiers pour au moins un besoin du groupe. Ces systèmes doivent également obtenir un rang inférieur à 20 pour au moins la moitié des besoins (i.e. la médiane de leurs rangs doit être inférieure à 20) ; ainsi nous sommes certains que les systèmes de RI candidats sont très performants pour au moins la moitié des besoins du groupe.

– candidats au système le plus *robuste* : ces systèmes ont un classement maximum inférieur à 80. De tels systèmes seront donc dans les deux premiers meilleurs tiers des systèmes de RI. La valeur 80 est fixée arbitrairement car suffisamment restrictive : en fixant ce seuil à un rang inférieur, aucun système de RI ne serait candidat au système le plus robuste pour certains groupes de besoins.
Il est clair que les règles présentées ici peuvent mener à plusieurs systèmes candidats concourants pour être le plus performant et/ou le plus robuste. Nous verrons par la suite que le choix d'un système unique parmi les candidats n'est pas toujours possible.

La figure 4 présente les 12 systèmes candidats pour le groupe des besoins que nous avons définis comme très difficiles en section 3.2.1. Un système candidat est un système qui correspond à au moins une des définitions présentées en début de cette section. Les différents seuils à respecter pour qu'un système soit candidat sont représentés par les lignes horizontales en pointillés. Les deux seuils bas correspondent aux valeurs de sélection pour le système le plus performant : une partie quelconque de la boite Ã moustache (valeurs extrêmes comprises) doit se trouver sous le premier seuil et la médiane doit se trouver sous le second seuil pour que le systme soit retenu. Le seuil haut correspond à la valeur limite pour qu'un systme soit retenu comme robuste : l'ensemble de la boite à moustache du système (valeurs extrêmes comprises) doit se trouver sous cette limite). Le signe "+" (respectivement "o") sous le nom des systèmes indique le(s) système(s) finalement retenus comme le(s) plus performant(s) (resp. robuste(s)).

D'après cette figure, sept systèmes de notre échantillon sont candidats au système de RI le plus performant. Cependant, le système READWARE2 (signe "+") est parmi les 10 meilleurs systèmes de RI pour tous les besoins classés très difficiles sauf deux. De plus, aucun autre candidat au système le plus performant pour ce groupe de besoins

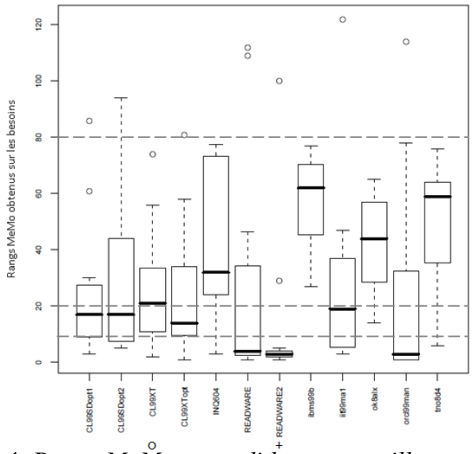

Figure 4. Rangs MeMo : candidats aux meilleurs systèmes - groupe très difficile

n'a de telles performances. READWARE2 est donc désigné comme le système le plus performant pour les besoins très difficiles.

Ici, cinq systèmes de RI peuvent correspondre au système le plus robuste car ils n'ont que des rangs inférieurs à 80 : CL99XT, INQ604, ibms99b, ok8alx et tno8d4. Cependant, CL99XT a 75% de ses rangs (Q3) inférieur à 40 ce qui signifie qu'il est classé 75 % du temps dans le premier tiers des systèmes pour les besoins très difficiles. En conséquence, le système CL99XT est choisi comme système le plus robuste pour ce groupe de besoins.
Nous pouvons procéder à des analyses similaires pour les différents groupes de besoins obtenus en 3.2.1. Le tableau 1 présente les résultats de ces analyses. Le tableau 1

Tableau 1. Résultats de la sélection des rangs MeMo par groupe de besoins

Groupes de besoins	**Système le plus**	
	perfomant	**robuste**
Faciles	MITSLStd	MITSLStd
Moyens	CL99XT	READWARE2
Difficiles	READWARE2	ibms99a
Très Difficiles	READWARE2	CL99XT

nous montre que la méthode de sélection basée sur les rangs MeMo permet de choisir le meilleur système à utiliser selon deux critères. Le premier critère est la difficulté du besoin en information à traiter. Le second concerne les attentes de celui qui exprime ce

besoin : attend-il les meilleurs résultats en acceptant quelques échecs ? ou préfère-t-il obtenir des résultats un peu moins bons mais qui le restent toujours ?

Cependant, pour un groupe de besoins fixé, le choix des meilleurs systèmes de RI n'est pas toujours trivial et nécessitent une étude plus approfondie sur les rangs des systèmes candidats. Nous supposons que cela est dû aux ex-aequos possibles lors de la transformation des MeMo en rangs. Nous procédons donc à une étude similaire des meilleurs systèmes de RI par groupe de besoins en information, cette fois basée directement sur les MeMo.

4.2. Détection des meilleurs systèmes par groupe de besoins par la mesure moyenne

Les critères de sélection du système le plus performant et du système le plus robuste sont analogues à celles présentées précédemment. L'étude est adaptée à la mesure moyenne. Dans l'étude précédente, une valeur de rang élevée était mauvaise puisque cela signifie que le système était classé parmi les moins bons. Désormais nous travaillons sur des mesures de performance : une valeur élevée signifie que le système fait partie des meilleurs.

La FIGURE 5 présente les systèmes de RI candidats selon les MeMo pour le groupe de besoins très difficiles. La lecture de cette figure est analogue à celle de la FIGURE 4. Cependant, plutot que de resprésenter la dispersion des rangs MeMo, dans la FIGURE 5 les boites représentent la dispersion de la MeMo.

Les médianes respectives du système "READWARE2" et du système "orcl99man" sont nettement supérieures aux médianes des autres systèmes. La position de la médiane de "READWARE2" proche du troisième quartile montre une forte densité des MeMo obtenues dans l'intervalle [0,45 ; 0,50]. Cela nous pousse à le choisir comme unique système le plus performant pour ce groupe. Étant donné que les moins bonnes valeurs de MeMo de READWARE2 sont supérieures aux MeMo de tous les autres candidats, le système "READWARE2" est également choisi ici comme système le plus robuste. Le tableau 2 présente l'ensemble des systèmes retenus par groupe de besoins en information selon les MeMo.

Tableau 2. Résultats de la sélection des MeMo par groupe de besoins

Groupes de besoins	**Système(s) retenu(s) comme le(s) plus performant(s)**	**Système(s) retenu(s) comme le(s) plus robuste(s)**
Faciles	—	—
Moyens	ii99mal	Flab8x, ok8amxc, CL99SDopt1
Difficiles	READWARE2	ibms99a
Très Difficiles	READWARE2	READWARE2

Pour les groupes de besoins faciles et moyens, le choix d'un unique meilleur système est une tâche difficile voire impossible. En effet, ces groupes de besoins sont définis tels que tous les systèmes réussissent à bien les traiter : les performances

Figure 5. MeMo : candidats aux meilleurs systèmes - groupe très difficile

obtenues par les systèmes sur le groupe facile sont très bonnes et donc très peu discriminantes pour choisir les "meilleurs" systèmes. Ceci n'est pas un problème, car comme tous les systèmes sont performants sur ces besoins, améliorer les résultats est une tâche compliquée. De plus, améliorer un processus déjà très robuste et performant n'est pas nécessaire.

4.3. Comparaison des systèmes retenus par les deux approches

Le tableau 3 montre que les deux méthodes de sélection produisent des résultats similaires pour les besoins en information les plus difficiles (groupes difficiles et très difficiles) et de plus larges variations dans les choix pour les besoins moins difficiles (classés moyens et faciles). Nous remarquons également que la méthode de sélection basée sur les rangs a tendance à gommer les faibles écarts entre différents systèmes. Cela présente l'avantage de proposer un choix unique de "meilleur" système lorsque plusieurs obtiennent des performances proches les unes des autres.

Les besoins les plus difficiles sont aussi les plus discriminants pour les systèmes, c'est à dire qu'ils permettent de repérer plus facilement les meilleurs systèmes de RI et donc de les sélectionner. Nous avons vu que la méthode de sélection des systèmes basée sur le rang MeMo permet toujours de faire le choix d'un unique système au contraire de la sélection directement basée sur les MeMo. Finalement, la méthode de sélection basée sur les rangs MeMo donnent des résultats similaires à la sélection

Tableau 3. Comparaison des méthodes de sélection par groupe de besoins

Groupes de besoins	**Système le plus performant**		**Système le plus robuste**	
	Rangs MeMo	MeMo	Rangs MeMo	MeMo
Faciles	MITSLStd	—	MITSLStd	—
Moyens	CL99XT	ii99mal	READWARE2	Flab8x ok8amxc CL99SDopt1
Difficiles	READWARE2	READWARE2	ibms99a	ibms99a
Très Difficiles	READWARE2	READWARE2	CL99XT	READWARE2

basée sur les MeMo pour les besoins les plus difficiles. Elle permet aussi d'opérer une sélection unique pour les besoins les moins difficiles donc la sélection sur les rangs MeMo devrait être préférée. Afin de valider ces observations, nous procédons à l'évaluation des deux méthodes de sélection.

4.4. Évaluation des méthodes de sélection

Ici, nous cherchons à déterminer si la robustesse (resp. la performance) des méthodes de sélection est meilleure que la robustesse (resp. la performance) des systèmes considérés initialement. En d'autres termes, nous cherchons à vérifier si les méthodes de sélection peuvent apporter une plus-value à un ensemble de systèmes de RI existants.

4.4.1. Sélection des systèmes de référence.

Un système de référence doit être sélectionné pour tester les méthodes selon l'approche désirée : la performance ou la robustesse. Pour tester la performance, le système qui a obtenu la meilleure mesure moyenne sur l'ensemble des besoins est sélectionné. Ce système est "orcl99man" avec une mesure moyenne générale de 0,5220. Cette moyenne générale est utilisée comme référence pour évaluer la performance des méthodes de sélection sur l'ensemble des besoins. Pour affiner l'évaluation, nous procédons également à des tests sur chacun des groupes de besoins. La référence alors utilisée est la moyenne des mesures moyennes obtenues par "orcl99man" sur les besoins du groupe étudié.
Pour tester la robustesse, nous pourrions nous contenter d'utiliser le système ayant obtenu la variance la plus faible. Le risque est d'obtenir une référence certes très stable mais avec des performances faibles. Pour palier à cela, nous sélectionnons les deux meilleurs tiers des systèmes sur l'ensemble des besoins considérés ; parmi les systèmes restants, celui à plus faible variance est choisi comme référence.

4.4.2. Création des méta-exécutions à partir des sélections.

La procédure d'évaluation est identique pour les deux méthodes. Pour chaque groupe de besoins, le système sélectionné pour ce groupe est utilisé. La moyenne des mesures de performance du système est calculée pour chaque besoin du groupe.

Si pour un groupe de besoin, plusieurs systèmes sont retenus, la moyenne de leurs mesures moyennes est utilisée. En concaténant les résultats obtenus pour chacun des groupes, nous obtenons les mesures moyennes d'une méta-exécution de tous les besoins. Une méta-exécution est utilisée pour les tests de performance et une seconde est utilisée pour les tests de robustesse.

4.4.3. Tests statistiques.

La procédure de test est identique pour chaque groupe de besoins et pour l'ensemble des besoins. Pour tester si la différence des performances entre le système de référence et la méta-exécution est significative, nous procédons à un test de Student (*t-test*) au seuil de 5 %. Pour cela, nous comparons les mesures moyennes obtenues par le système référence avec les mesures moyennes de la méta-exécution sur les besoins considérés. Le t-test unilatéral est employé afin de déterminer si la performance de la méta-exécution est significativement supérieure (resp. significativement inférieure) au point de comparaison lorsque celle-ci est supérieure (resp. inférieure) à la référence. Pour tester si la robustesse de la méta-exécution est significativement différente de la robustesse de la référence, nous procédons au test de Fisher de comparaison de deux variances. Ce test nous permet d'établir si la variance des mesures moyennes obtenues par la méta-exécution est significativement plus petite ou plus grande que la variance des mesures moyennes du système de référence. Un t-test est également réalisé sur les moyennes pour comparer les performances de ces exécutions.

4.4.4. Résultats de l'évaluation.

Les tableaux suivants présentent les résultats obtenus par les méthodes de sélection pour le système le plus performant (tableau 4) et pour le système le plus robuste (tableau 5). Dans ces tableaux, "(++)" (resp. "(- -)") indique que la sélection est significativement supérieure (resp. inférieure) à la référence selon le test employé ; "(+)" et "(-)" indiquent un écart non significatif ; "(=)" indique qu'en valeur absolue l'écart est inférieur à 1 %.

Tableau 4. Évaluation de la sélection du système le plus performant - test sur la moyenne

Groupes de besoins	**MeMo**		**Rangs MeMo**	
	Référence	Sélection	Référence	Sélection
Faciles	—	—	0,6792	(+) 0,6974
Moyens	0,6110	(+) 0,6253	0,6110	(=) 0,6082
Difficiles	0,4723	(+) 0,5195	0,4723	(+) 0,5195
Très Difficiles	0,4215	(=) 0,4207	0,4215	(=) 0,4207
Tous	0,4920	(+) 0,5188	0,5220	(+) 0,5440

Dans le tableau 4, nous observons que hormis pour les besoins les plus difficiles, la sélection basée sur la MeMo obtient des performances supérieures à celles de la référence. La sélection sur les rangs MeMo est également meilleure pour les besoins faciles et difficiles et obtient des résultats très proches de la référence pour les deux

autres groupes de besoins. Le meilleur gain est obtenu pour le groupe de besoins difficiles avec une amélioration de 10,0 % par rapport à la référence.

Tableau 5. Évaluation de la sélection du système le plus robuste - tests sur la variance et la moyenne

Groupes de besoins	**MeMo**				**Rangs MeMo**			
	Référence		Sélection		Référence		Sélection	
	Moyenne	Variance	Moyenne	Variance	Moyenne	Variance	Moyenne	Variance
Faciles	—	—	—	—	0,6066	0,00703	(++) 0,6974	(-) 0,00439
Moyens	0,4805	0,00431	(++) 0,5831	(- -) 0,00138	0,4805	0,00431	(++) 0,6298	(+) 0,00498
Difficiles	0,3992	0,00553	(++) 0,4513	(-) 0,00368	0,3992	0,00553	(++) 0,4513	(-) 0,00368
Très Difficiles	0,2306	0,00470	(++) 0,4207	(++) 0,02749	0,2306	0,00470	(++) 0,3321	(++) 0,02134
Tous	0,3744	0,01341	(++) 0,4752	(-) 0,01273	0,4116	0,01962	(++) 0,4789	(+) 0,02987

Dans le tableau 5, pour les besoins de difficulté moyenne, nous observons que la variance de la sélection MeMo est significativement inférieure à celle de la référence ; on en déduit que la sélection est significativement plus robuste que la référence pour ces besoins avec un gain de 68 % en robustesse avec des performances 21 % plus élevées que celles de la référence. Au contraire, sur les besoins très difficiles, la référence est significativement plus robuste. Cependant, la sélection MeMo obtient des performances significativement plus élevées quel que soit le groupe de besoin considéré.

La sélection sur les rangs MeMo permet d'obtenir une meilleure robustesse lorsque les besoins faciles et difficiles sont traités. Là encore, la robustesse est significativement moins bonne que celle de la référence pour les besoins très difficile malgré des performances supérieures.

Lorsque tous les besoins sont considérés, les deux méthodes de sélection des systèmes selon la robustesse permettent d'obtenir des meilleurs performances que la référence. La sélection MeMo améliore la robustesse de 5 % en conservant des performances de 27 % supérieures à la référence. La sélection sur les rangs MeMo quant à elle dégrade la robustesse de moitié en conservant des performances 16 % meilleures.

5. Sélection basée sur l'apprentissage sur un sous-ensemble de documents

Dans cette section, nous nous intéressons à la définition d'une méthode de recherche d'information sélective. Il s'agit d'apprendre à choisir quel système doit traiter chaque requête de sorte d'obtenir les résultats optimaux. Cette méthode tire donc avantage de la variabilité des systèmes. Concrètement, comme nous apprenons pour chaque requête quelle est la méthode optimale, cette approche est applicable au cas des requêtes répétées.

Durant la phase d'apprentissage, il s'agit de choisir la configuration de systèmes ou la classe de systèmes la plus adaptée en fonction des performances obtenues pour une requête donnée sur un sous-ensemble de documents pour lesquels la pertinence vis à vis de cette requête est connue. Après apprentissage, chaque fois que la même requête est soumise au système, la configuration qui a été considérée comme la meilleure sera choisie pour être appliquée sur l'ensemble de la collection de documents.

5.1. Description de la méthode

La méthode que nous proposons ici est basée sur la même hypothèse que les méthodes présentées précédemment. Plus précisément, elle fait l'hypothèse que les paramètres du système ont un effet différent sur différentes requêtes. Nous considérons que pour une requête donnée, certains paramètres du système donnent de bons résultats similaires tandis que d'autres aboutiront à des résultats plus faibles. Pour mettre en œuvre cette idée, la méthode proposée apprend la configuration du système qui doit traiter chaque requête (répétée).

5.1.1. Phase d'apprentissage

La phase d'apprentissage vise à choisir la configuration qui doit traiter chaque requête d'apprentissage. Au cours de cette étape, nous considérons les diverses configurations de système et les résultats qu'ils ont obtenus pour chacune des requêtes d'apprentissage en utilisant un échantillon de documents annotés. Pour ces documents, nous savons s'ils sont pertinents pour la requête ou non. A partir des listes de documents retrouvées par chaque configuration, nous calculons la performance pour chaque configuration de système et chaque requête. La meilleure configuration est alors choisie pour chaque requête.

5.1.2. Phase de test

Après la phase d'apprentissage, le méta-système connait quelle configuration de système doit traiter chaque requête apprise. Chaque fois qu'une requête identique est soumise, le méta-système traite la requête avec la configuration apprise.

5.2. Evaluation

5.2.1. Mesures d'évaluation

Nous utilisons le logiciel d'évaluation TREC trec_eval [2], version 9.0 qui calcule de nombreuses mesures de performance. Dans ce travail, nous ne considérons que la précision moyenne (AP) pour chaque requête et la moyenne des précisions moyennes sur l'ensemble des requêtes (MAP). AP correspond à la moyenne de la précision (nombre de documents pertinents retrouvés sur les documents restitués) chaque fois qu'un document pertinent est restitué par le système. Elle est calculée par la formule :

$$AP = \frac{\sum_{i=1}^{n} P(i) * \text{rel}(i)}{\text{nombre de documents pertinents}}$$

où :

– n est le nombre de documents retrouvés ;

2. http ://trec.nist.gov/trec_eval/

– $P(i)$ est la précision lorsque les i premiers documents retrouvés sont considérés ;

– rel(i) est égal à 1 is le document est pertinent, 0 sinon.

MAP permet de caractériser les performances d'un système en calculant la moyenne des AP obtenues sur l'ensemble des besoins d'infromation.

$$MAP = \frac{\sum_{t=1}^{T} AP(t)}{T}$$

où T est le nombre des besoins d'information.
En utilisant l'AP pour évaluer la configuration de système, nous construisons une table dans laquelle les lignes correspondent aux besoins d'information, les colonnes à des configurations de système ; cette table contient la valeur de la mesure AP obtenue. Par exemple, la table 6 présente un extrait de la table qui est obtenue.

Tableau 6. Extrait des valeurs de AP

	Sys1	Sys2	...	Sys99	Sys100
T351	0,5819	0,5722	...	0,5432	0,5135
T352	0,0001	0,0011	...	0,0016	0,0005
...	...	...	...	...	...
T449	0,0528	0,0275	...	0,1324	0,0117
T450	0,3353	0,1699	...	0,3067	0,2959

5.2.2. Collection utilisée

Nous avons utilisé la collection présentée dans la section 2.2.2 comprenant 100 besoins d'information. Grâce à la plateforme Terrier, nous avons généré différentes configurations de systèmes. Terrier permet de paramétrer une chaine de recherche d'information selon les trois étapes du processus : indexation (racinisation, taille des blocs, mots vides), la mise en correspondance entre requêtes et documents (modèle d'appariement, les champs du besoin d'information à utiliser, ...) et l'expansion de la requête (QE) (modèle de QE, le nombre de termes à ajouter à la requête, le nombre de documents à considérer, le nombre de termes rajoutés, ...). Dans nos expériences, nous avons fixé le modèle d'indexation ; en effet, celui-ci est coûteux en temps de calcul ; par ailleurs, dans un modèle rééel, il est plus facile d'envisager plusieurs façons d'utiliser un index que l'utilisation simultanée de plusieurs index. Ainsi, les seuls paramètres que nous avons fait varier sont ceux de la mise en correspondance entre les documents et la requête et ceux concernant l'expansion de la requête. La phase d'indexation choisie utilise la liste par défaut de mots vides de Terrier, l'algorithme de Porter pour la racinisation, une taille de bloc de 1 et ne tient pas compte des documents vides. Ces choix sont cohérents avec les études présentées dans (Compaoré *et al.*, 2011). Cela nous a conduits à considérer plus de 100 configurations, mais nous n'en avons retenu seulement 100 qui obtiennent une MAP supérieure à $0,2$ au moins sur 50 % des besoins d'information. Ce seuil de $0,2$ correspond à la moitié de la valeur de MAP du meilleur système utilisé dans (Bigot *et al.*, 2011) et a été choisie afin de

veiller à ce que les performances des configurations de systèmes sélectionnés soient assez bonnes.

La répétition de requêtes est simulée par le traitement du même besoin d'information sur le reste de la collection de documents (documents de test). L'évaluation s'appuie sur les jugements de pertinence fournis par le programme TREC (après avoir retiré les documents d'apprentissage).
Dans nos expériences, nous divisons la collection de documents selon une partie d'apprentissage et une partie de test.

5.2.3. Principe de la validation croisée

Pour évaluer l'apprentissage du choix de la meilleure configuration à associer à chaque requête, nous avons appliqué le principe de validation croisée. En effet, une seule partition des données selon une partie d'apprentissage et une partie de test ne suffit pas pour tirer des conclusions solides. Pour cette raison, nous avons réalisé 10 partitions différentes de la collection composées chacune d'une partie d'apprentissage de $\frac{2}{3}$ et d'une partie de test de $\frac{1}{3}$ choisi au hasard. Ensuite, nous calculons la moyenne des résultats au cours des 10 collections.

5.2.4. Fonction de référence

Pour chacune des 10 collections, le système qui obtient la meilleure MAP moyenne sur tous les besoins d'information est sélectionné comme point de référence. La moyenne des dix références est la référence générale.

5.3. Résultats

Utiliser 100 configurations de systèmes peut être coûteux pour des applications réelles. Pour cette raison, nous avons étudié l'impact du nombre de configurations de système à utiliser. Nous considérons différents seuils de sélection (10, 20, 30 ... 100 configurations). Plutôt que de choisir au hasard les systèmes nous avons plutôt voulu sélectionner des configurations de système qui se comportent différemment sur les besoins d'information. Pour cette raison, nous avons regroupé les configurations de systèmes qui conduisent à des performances proches en termes de AP pour l'ensemble des besoins d'information. Pour chaque groupe de configurations de systèmes, nous choisissons alors une configuration unique qui sera utilisée dans la phase d'apprentissage.

Cette manière de procéder permet de s'assurer que les configurations choisies sont assez différentes. Nous considérons alors plusieurs seuils : nous analysons les résultats lorsque 100 % des configurations sont utilisées (sans regroupement de configurations), lorsque que 90 % des configurations intiales sont utilisées (ce qui signifie que nous regroupons les configurations en 90 groupes), 80 % ... jusqu'à 10 % (ce qui signifie que seulement 10 configurations sont utilisées, correspondant à 10 groupes différents ou profils de configuration différents). La figure FIGURE 6 montre les résultats de cette exprimentation. Les lignes en gris correspondent à la phase d'entraînement et

les lignes noires à la phase de test. Les lignes en pointillé correspondent au système de référence alors que les lignes pleines sont celles de la méthode de sélection proposée. Les différents points correspondent aux résultats obtenus sur la moyenne des 10 sous-collections utilisées comme présentés dans la section 5.2.3.

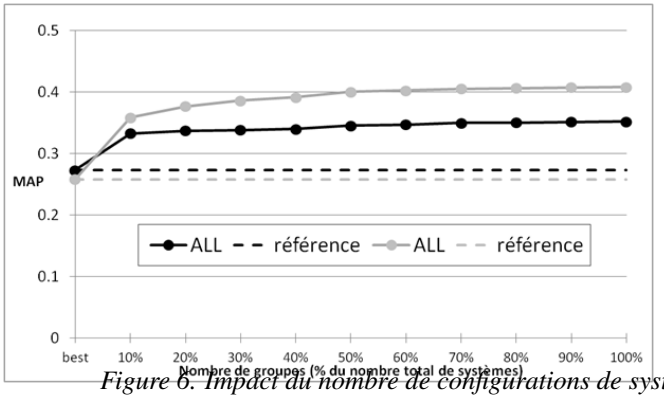

Figure 6. Impact du nombre de configurations de système utilisées.

Les résultats montrent que l'utilisation de seulement 10 groupes différents, et donc de 10 différentes configurations de système permet d'améliorer les résultats de 22 %. Au delàs, l'augmentation du nombre de configurations n'améliore pas beaucoup les résultats.

6. Conclusion et travaux à venir

Nous avons vu deux méthodes de sélection de systèmes. Une méthode consiste à calculer une unique mesure pour quantifier la performance moyenne de chaque système. La seconde méthode introduit les rangs des systèmes selon leur performance moyenne. Nous avons montré qu'avec ces méthodes il est possible de choisir le meilleur système selon un critère de robustesse ou de performance pour différents groupes de difficulté des besoins en information. La sélection des meilleurs systèmes est compliquée pour les besoins classés faciles ou moyens : ceux-ci sont très peu discriminants pour les systèmes de RI car par définition, les systèmes réussissent tous à traiter ces besoins avec succès. Pour les groupes de besoins difficiles et très difficiles, le choix d'un unique meilleur système basé sur la performance ou sur la robustesse est plus aisé. L'évaluation montre que, à partir d'un ensemble de systèmes de RI donné, les méthodes de sélection analysées dans cet article sont capables d'améliorer les résultats obtenus par les systèmes de manière individuelle. L'évaluation montre que l'on peut atteindre une amélioration de 10 % des performances si celle-ci est privilégiée. Il est également montré que la robustesse peut être améliorée significativement jusqu'à 68 % de gain tout en conservant de bonnes performances.

Pour les deux méthodes, un choix final des systèmes est opéré à la main et introduit un biais subjectif dans la sélection. Une future analyse devrait proposer une façon d'automatiser la détection des seuils fixés arbitrairement dans cet article et analyser les différences de performances selon les valeurs de seuil retenues. L'emploi de méthodes statistiques telle que les skylines, les arbres aléatoires ou les analyses factorielles pourraient être étudiées pour détecter les valeurs des seuils à employer.

Par ailleurs, nous avons montré qu'il est possible d'améliorer les performance en apprenant la meilleure configuration de recherche en fonction des requêtes. En utilisant la même indexation des documents et en s'appuyant sur un ensemble de documents d'apprentissage pour lesquels la pertinence vis à vis d'une requête est connue, il est possible ainsi d'augmenter d'environ 20 % la MAP.

Dans nos travaux futurs nous aimerions pouvoir apprendre la meilleure configuration de système non pas sur la base de chaque requête mais plutôt sur la base de caractéristiques de requêtes. L'idée serait alors de pouvoir traiter toute nouvelle requête, même si elle n'est pas dans l'ensemble d'apprentissage simplement par son partage de caractéristiques avec certaines requêtes apprises.

Aussi, les méthodes de sélection sont indépendantes du nombre de documents présents dans la collection. Cependant la méthode d'évaluation ne l'est pas puisque les documents sont séparés en deux groupes (entrainement et test). Il faudrait donc étudier si les choix réalisés dans cette étude sont directement applicables à d'autres collections plus volumineuse ou bien analyser si les optimisations sont dépendantes de la collection de documents.

7. Remerciements

Nous tenons à remercier l'ANR pour son soutien financier à ces recherches au travers du projet ANR-2010-CORD-001-01 CAAS (contextual analysis and adaptive search, http://www.irit.fr/CAAS/) ainsi que la fédération CNRS FREMIT FR3424 (http://www.irit.fr/FREMIT/). Enfin, nous remercions J. Poirier et B. Sansas pour leur travail réalisé lors de leur projet long (Poirier, Sansas, 2010).

Bibliographie

Baccini A., Déjean S., Mothe J., Lafage L. (2011). How many performance measures to evaluate information retrieval systems ? *Knowledge and Information Systems*, vol. 30, n° 3, p. 693-713. http://hal.archives-ouvertes.fr/hal-00658732/en/

Bigot A. (2013, mai). Adapter les moteurs de recherche aux besoins en information - Prise en compte de la difficulté du besoin . In *INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID)*, p. 59–74. http://www.univ-paris1.fr/. %20ftp://ftp.irit.fr/IRIT/SIG/Bigot_Anthony_INFORSID_2013.pdf

Bigot A., Chrisment C., Dkaki T., Hubert G., Mothe J. (2011). Fusing different information retrieval systems according to query-topics: a study based on correlation in information

retrieval systems and TREC topics. *Information Retrieval Journal*, vol. 14, n° 6, p. 617–648. http://dx.doi.org/10.1007/s10791-011-9169-5

Brown P. J., Jones G. J. F. (2001). Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Personal and Ubiquitous Computing*, vol. 5, n° 4, p. 253–263. http://dx.doi.org/10.1007/s007790170004

Carmel D., Yom-Tov E. (2010). *Estimating the query difficulty for information retrieval*. Morgan & Claypool (ed.). http://dx.doi.org/10.2200/S00235ED1V01Y201004ICR015;http://dx.doi.org/10.2200/S00235ED1V01Y201004ICR015

Chifu A., Mothe J. (2014). Expansion sélective de requêtes par apprentissage. In *Soumis à coria 2014*, p. Soumis.

Chrisment C., Dkaki T., Mothe J., Poulain S., Tanguy L. (2005). Recherche d information - analyse des résultats de différents systèmes réalisant la même tâche. *Revue des Sciences et Technologies de l'Information*, vol. 10, n° 1, p. 31–55. ftp://ftp.irit.fr/IRIT/SIG/2005_RSTI_CDMPT.pdf

Cleverdon C., Mills J., Keen M. (1966). *Factors determining the performance of indexing systems* (vol. 1). ASLIB Cranfield Research Project.

Compaoré J., Déjean S., Gueye A. M., Mothe J., Randriamparany J. (2011). Mining information retrieval results: Significant IR parameters (regular paper). In *Advances in Information Mining and Management*. IARIA. ftp://ftp.irit.fr/IRIT/SIG/2011_IMMM_CGDMR.pdf

Fox E. A., Shaw J. A. (1994). Combination of multiple searches. In *Trec-2, proceedings of the second text retrieval conference*, p. 243–249. D. Harman (ed.).

Harman D., Buckley C. (2009). Overview of the reliable information access workshop. *Information Retrieval*, vol. 12, n° 6, p. 615–641. http://dx.doi.org/10.1007/s10791-009-9101-4

He B., Ounis I. (2004). Inferring query performance using pre-retrieval predictors. In *11th international conference, spire 2004, proceedings*, p. 43 – 54. Padova, Springer Berlin Heidelberg. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.102

Jain A. K., Murty M. N., Flynn P. J. (1999). Data clustering: A review. *CSURV: Computing Surveys*, vol. 31, n° 3, p. 264-323.

Joho H., Urban J., Villa R., Jose J. M., Rijsbergen C. J. van. (2008). AIR 2006: First international workshop on adaptive information retrieval. *SIGIR Forum : Special Interest Group on Information Retrieval Forum*, vol. 42, n° 1, p. 63–66. http://doi.acm.org/10.1145/1394251.1394265

Kompaoré D., Mothe J., Baccini A., Déjean S. (2007). Prédiction du SRI à utiliser en fonction des critères linguistiques de la requête. In *Conference en recherche d'information et applications*, p. 239–254. Université de Saint-Étienne. http://asso-aria.org/coria/2007/239.pdf

Lebart L., Piron M., Morineau A. (2006). *Statistique exploratoire multidimensionnelle : visualisations et inférences en fouille de données*. Dunod.

Lee J.-H. (1997). Analyses of multiple evidence combination. In *Sigir : Special interest group on information retrieval*, p. 267–276. ACM. http://doi.acm.org/10.1145/258525.258587

Lillis D., Toolan F., Mur A., Peng L., Collier R. W., Dunnion J. (2006). Probability-based fusion of information retrieval result sets. *Artificial Intelligence Review*, vol. 25, n° 1-2, p. 179–191. http://dx.doi.org/10.1007/s10462-007-9021-x

Liu H., Wu Z., Hsu D. F. (2012). Combination of multiple retrieval systems using rank-score function and cognitive diversity. In *Aina : Advanced information networking and applications IEEE 26th international conference on advanced information networking and applications, AINA, 2012, fukuoka, japan, march 26-29, 2012*, p. 167–174. IEEE. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6184368

Menegon D., Mizzaro S., Nazzi E., Vassena L. (2009). Evaluating mobile proactive context-aware retrieval: An incremental benchmark. In *Advances in information retrieval theory, second international conference on the theory of information retrieval, ICTIR 2009, cambridge, UK, september 10-12, 2009, proceedings*, vol. 5766, p. 362–365. Springer.

Mizzaro S., Nazzi E., Vassena L. (2008). Retrieval of context-aware applications on mobile devices: how to evaluate? In *Proceedings of the 2nd international conference on information interaction in context, IIiX 2008, london, UK, october 14-17, 2008*, vol. 348, p. 65–71. ACM.

Mothe J., Tanguy L. (2005). Linguistic features to predict query difficulty - a case study on previous TREC campaigns. In *ACM Conference on research and Development in Information Retrieval, SIGIR : Special Interest Group on Information Retrieval, Predicting query difficulty - methods and applications workshop*, p. 7–10.

Poirier J., Sansas B. (2010). *Evaluation des systèmes de recherche d'information*. IRIT.

Voorhees E. M., Harman D. (1997). Overview of the sixth text retrieval conference (TREC-6). In *Trec*, p. 1-24.

Ward J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, vol. 58, p. 236–244.