

Réinjection de pertinence pour la fusion de systèmes

Gilles Hubert(*), Désiré Kompaore(*), Josiane Mothe(*,**)

hubert@irit.fr, kompaore@irit.fr, mothe@irit.fr

(*) Institut de Recherche en Informatique de Toulouse,
118 Route de Narbonne, 31062 Toulouse Cedex 04, France

(**) Institut Universitaire de Formation des Maîtres
56, av. de l'URSS, 31078 Toulouse Cedex, France

Mots clés :

recherche d'information, fusion de systèmes, réinjection de pertinence

Keywords :

information retrieval, post-retrieval fusion, relevance feedback

Palabras claves :

Recuperación de Información, fusión de sistemas, Realimentación de relevancia

Résumé

Utilisant des techniques ou des modèles différents, les systèmes de recherche d'information (SRI) présentent une grande variabilité dans les résultats qu'ils retrouvent. Ainsi, deux systèmes qui ont des performances globales (sur un ensemble de requêtes) comparables peuvent avoir des performances locales différentes (selon les requêtes). De plus, deux systèmes obtenant des performances comparables pour une requête donnée peuvent en fait retourner des documents différents. Dans cet article, nous nous intéressons à la fusion post recherche de systèmes en prenant en compte les performances locales de différents systèmes. Plus spécifiquement, nous utilisons la pertinence des premiers documents retournés pour choisir parmi un ensemble de moteurs celui qui sera utilisé pour conduire la recherche pour une requête donnée. Cette sélection automatique du meilleur moteur pour chaque requête permet une amélioration des performances proche de 10%. Nous avons évalué notre méthode sur 4 années de la tâche adhoc du programme d'évaluation de SRI TREC (« Text REtrieval Conference ») en se basant sur la fusion de 2, 5 ou plus systèmes.

1 Introduction

L'évaluation des systèmes de recherche d'information (SRI) est une préoccupation de la communauté scientifique depuis les années 60 (Cleverdon, 1967). De façon classique, un SRI est évalué sur la base d'une collection de test comprenant un ensemble de requêtes, un ensemble de documents et pour chaque requête la réponse attendue. Les taux de rappel, de précision et la mesure-F permettent de comparer les systèmes entre eux. Généralement, les systèmes sont comparés de façon globale, c'est-à-dire en calculant des valeurs moyennes obtenues pour l'ensemble des requêtes.

Cependant, l'étude de phénomènes locaux a montré, d'une part, que les systèmes ne sélectionnent pas les mêmes documents en réponse à une requête. Les mécanismes de fusion de résultats ou post recherche (Fox & Shaw, 1994) exploitent cette complémentarité dans les listes documents restitués par différents systèmes. L'hypothèse sous-jacente est qu'en combinant les réponses de différents systèmes, il est possible d'améliorer la sélection des documents et donc les performances globales. Les mécanismes retenus reposent sur une fusion des listes des réponses de différents systèmes pour une requête donnée.

D'autre part, l'étude des phénomènes locaux a également montré qu'il existait une spécificité des systèmes et des requêtes. En effet, même si certains besoins sont difficiles à satisfaire pour une majorité de systèmes (Dkaki et al., 2004 ; Buckley & Harman, 2004), certains systèmes ont des comportements spécifiques qui leur permettent de mieux répondre à certaines requêtes.

Par exemple, les tableaux 1 et 2 indiquent les performances locales et globales de deux exécutions (« runs ») de deux systèmes différents extraites des résultats de la tâche adhoc de la 7^{ème} édition du programme d'évaluation de SRI TREC (TREC7).

Le tableau 1 indique les performances des deux systèmes pour deux requêtes données. Le nombre de documents retrouvés par chaque système (Num_ret), le nombre de documents pertinents retrouvés (Num_rel_ret), le nombre de documents pertinents pour la requête (Num_rel) ainsi que différentes valeurs de critères sont indiqués. Les critères reportés correspondent aux mesures d'évaluation utilisées traditionnellement dans les programmes d'évaluation comme TREC. Ces mesures sont :

- la précision moyenne (proportion des documents pertinents par rapport aux documents retrouvés),
- la R-précision (précision lorsque exactement R documents sont retrouvés. R est le nombre de documents pertinents pour la requête),
- les précisions lorsque n documents sont retrouvés (P@n) pour des valeurs de n égales à 5, 10, 15, 20.

Requête	352		354	
	CLARIT98COMB	T7miti1	CLARIT98COMB	T7miti1
Run				
Num_ret	1000	210	1000	486
Num_rel	246	246	361	361
Num_rel_ret	216	117	124	190
Précision moyenne	0,5068	0,3081	0,1675	0,2767
R-précision	0,5325	0,4756	0,2271	0,4100
P@5	1,0000	0,6000	1,0000	0,4000
P@10	1,0000	0,7000	1,0000	0,3000
P@15	0,9333	0,6000	1,0000	0,3333
P@20	0,8000	0,6500	0,8500	0,4500

TAB 1 : performances locales des systèmes CLARIT98COMB et T7miti1 dans la tâche adhoc TREC7 (2 requêtes et 2 systèmes issus de TREC 7 adhoc)

Le tableau 2 indique les performances globales des deux systèmes correspondant aux valeurs moyennes des mesures sur l'ensemble des requêtes notamment la précision globale moyenne (« Mean Average Precision » MAP). Pour chaque requête, la précision obtenue chaque fois qu'un nouveau

document est retrouvé est calculée ; on calcule ensuite la moyenne de ces valeurs. La MAP correspond à la moyenne de cette valeur sur un ensemble de requêtes.

Run	CLARIT98COMB	T7miti1
MAP	0,3702	0,3675
R-précision	0,4140	0,4392
P@5	0,6920	0,6640
P@10	0,6940	0,6400
P@15	0,6613	0,6213
P@20	0,6180	0,5780

TAB 2 : performances globales des systèmes CLARIT98COMB et T7miti1 dans la tâche adhoc TREC7 (2 systèmes issus de TREC 7 adhoc)

Au regard des performances des deux systèmes illustrées dans les tableaux 1 et 2, s'il était possible de décider a priori d'utiliser le système CLARI98COMB pour la requête 352 et le système T7miti1 pour la requête 354, il serait possible d'optimiser les performances globales telles que la MAP et la R-précision.

Notre étude s'inscrit dans ce cadre. Nous nous appuyons sur le fait que les systèmes n'obtiennent pas localement les mêmes performances, c'est-à-dire que, même si deux systèmes sont globalement équivalents sur un ensemble de requêtes, les systèmes n'obtiennent pas des performances semblables lorsque chaque requête est considérée de façon indépendante. Ainsi, notre hypothèse est qu'un système basé sur un ensemble de moteurs et qui choisit le moteur qui traitera la requête en fonction de celle-ci obtiendra des performances globales meilleures que chacun des moteurs s'ils sont pris indépendamment.

L'idée que nous développons dans cet article est que le choix du « meilleur » moteur peut être basé sur l'analyse des premiers documents restitués par le moteur. Ainsi, une première recherche est réalisée par chacun des moteurs et en fonction du jugement de pertinence sur les premiers documents restitués, l'un ou l'autre des systèmes sera utilisé pour la requête et restituera les documents finaux.

L'article est structuré comme suit. Dans la section 2, nous proposons une revue des travaux dans le domaine de la fusion de moteurs. La section 3 présente la méthode de fusion que nous proposons ainsi que les collections utilisées pour les tests. La section 4 présente les résultats des expérimentations réalisées ainsi que leur analyse. La dernière section conclut cette étude.

2 Travaux du domaine

Le principe des mécanismes de fusion post recherche utilisés dans la littérature du domaine consiste à obtenir différentes listes de documents sélectionnés pour une requête donnée et à combiner ces listes pour obtenir les documents effectivement restitués à l'utilisateur.

Fox and Shaw (1994) ont montré que combiner les résultats de plusieurs recherches améliore les performances par rapport au résultat d'une seule recherche. Les expérimentations qu'ils ont menées sont basées sur neuf sous-collections différentes de TREC et combinent cinq recherches effectuées de différentes façons. Le même système est utilisé ; il permet d'ordonner les réponses du système en fonction du degré de similarité avec la requête. En revanche, différentes représentations de requêtes sont utilisées. Fox et Shaw ont montré que la combinaison la plus efficace (CombSUM) consiste à additionner les valeurs de similarité pour chaque document. Sur la collection TREC-2, ils ont montré des améliorations de la R-Précision de l'ordre de 13 %. Belkin et al. (1994) se sont également intéressés à ce type de fusion. Pour chaque besoin d'informations (dix au total), dix requêtes ont été formulées manuellement et combinées pour être soumises au système INQUIRY (Callan et al. 1992). Ces expérimentations ont montré que les résultats sont fortement liés à la formulation des requêtes et que la combinaison des requêtes améliore la performance générale.

De son côté, Lee (1997) a montré que CombSUM peut être combinée de façon efficace en considérant la différence de niveau de chevauchement entre documents pertinents et documents non

pertinents. Ainsi, il a montré qu'il vaut mieux fusionner des systèmes qui ont un plus fort chevauchement de documents pertinents que de documents non pertinents. Beitzel et al. (2003) ont un peu contredit cette hypothèse en montrant que l'amélioration n'est pas tant liée au taux de chevauchement qu'au nombre de documents pertinents qui n'apparaissent que dans un résultat de recherche. Dans (He, 2003), différentes fonctions de pondération de termes sont utilisées en fonction des besoins d'information et de caractéristiques de requêtes. Cette méthode, appliquée à la tâche 'Robuste' de TREC est efficace pour les requêtes aux performances faibles.

Contrairement aux travaux de la littérature, notre méthode ne consiste pas à combiner différentes listes de documents retrouvés par différents systèmes pour une requête donnée. Nous préconisons plutôt une méthode qui sélectionne parmi un ensemble de systèmes, celui qui devra effectuer la recherche pour une requête donnée.

3 Méthode de fusion

Notre méthode de fusion est basée sur l'utilisation d'un ensemble de moteurs. L'évaluation d'un tel mécanisme nécessite donc un ensemble de moteurs utilisés sur les mêmes collections de documents et de requêtes. TREC fournit ce type de données.

3.1 Collections et critères d'évaluation

Les évaluations que nous avons réalisées s'appuient sur les données issues de TREC dans la tâche adhoc. Une collection de TREC contient :

- un ensemble de documents correspondant à la collection à interroger,
- un ensemble de requêtes (généralement 50 pour une tâche et une année donnée),
- l'ensemble des documents que le système idéal aurait dû retrouver (jugés manuellement),
- pour chaque système participant, la liste des documents retrouvés pour chaque requête.

Dans cette étude, nous nous sommes appuyés sur les données TREC 3, 5, 6 et 7 de la tâche adhoc. Les caractéristiques de ces collections sont indiquées dans le tableau 2.

Caractéristique	TREC3	TREC5	TREC6	TREC7
Documents	≈ 1 Go	≈ 2 Go	≈ 2 Go	≈ 2 Go
Requêtes	Topics 151-200	Topics 251-300	Topics 301-350	Topics 351-400
Nombre de systèmes participants	48	77	79	103

TAB 3 : Caractéristiques des collections TREC adhoc utilisées dans l'expérimentation.

Trec-eval est un programme utilisé dans les évaluations du programme TREC pour permettre la comparaison des systèmes. Les critères classiques de rappel, précision et F-mesure moyenne pour une requête sont calculées par rapport à l'ensemble fourni des documents que le système idéal aurait dû retrouver pour chaque requête. Ce programme permet de calculer la précision à n-documents (précision lorsque n documents sont retrouvés) ainsi que la précision globale moyenne (« Mean Average Precision » MAP) qui donne une mesure globale de la qualité d'un résultat en prenant en compte à la fois le rappel et la précision. Il s'agit là d'une des mesures globales les plus utilisées pour comparer différents systèmes.

3.2 Expérimentation préliminaire

L'hypothèse sur laquelle nous nous appuyons est qu'il est possible de combiner intelligemment différents moteurs de recherche et de sélectionner celui qui sera le plus performant pour une requête donnée. Pour valider une partie de cette hypothèse, nous avons réalisé une expérimentation préliminaire qui a consisté à utiliser différents systèmes pour chacune des requêtes et à sélectionner « manuellement » le système le plus performant pour chaque requête. Nous maximisons donc ici les résultats qu'il est possible d'obtenir par le principe de fusion de systèmes que nous proposons. Nous avons pu montrer que dans le cas de l'utilisation des deux meilleurs systèmes ayant participé chaque

année à la tâche adhoc de TREC, si un système était capable de choisir toujours le « meilleur » d'entre eux, les performances en termes de MAP seraient améliorées de 10 à près de 20%, en fonction des années par rapport au meilleur système. Lorsque les 5 meilleurs systèmes sont choisis, cette amélioration varie de 12 à 30% (voir tableaux 4 et 5, colonne « Optimal »). Nous montrons par cette expérimentation préliminaire l'intérêt de l'étude puisqu'il est potentiellement possible d'augmenter les performances d'un système de recherche d'information en se basant sur différents moteurs et en sélectionnant celui qui est le plus adapté en fonction de la requête traitée. Il est évident qu'une telle approche n'est concevable que s'il est possible de détecter à priori le système à utiliser.

La méthode que nous proposons vise donc à permettre une sélection automatique, dans notre cas, sur la base de la précision à 5 documents.

3.3 Réinjection de pertinence pour le choix des systèmes

Nous proposons de baser la sélection du système sur des informations minimales sur la pertinence des premiers documents retrouvés. Dans notre cas, cette information sur la pertinence des premiers documents est utilisée pour choisir le système à considérer. Nous utilisons la précision à 5 documents pour choisir parmi les systèmes celui qui traitera la requête pour l'utilisateur. La figure 1 présente l'algorithme proposé :

```

Pour chaque requête  $Q_j$ 
  Pour chaque système  $S_i$ 
    Rechercher les documents pertinents
    Restituer les 5 premiers documents à l'utilisateur pour jugement de pertinence
    Calculer la pertinence des 5 premiers documents ( $P@5$ )
  Fin Pour
  Ordonner les systèmes par  $P@5$  décroissante
  Choisir le premier système pour traiter la requête
  Restituer les documents retrouvés par ce système
Fin Pour

```

Figure 1 : algorithme de sélection du système pour chaque requête

Ce principe de sélection a été évalué sur les quatre collections de TREC adhoc. Dans un premier temps, nous avons considéré pour chaque collection les deux meilleurs systèmes puis dans un second temps les cinq meilleurs systèmes pour la collection considérée. Les meilleurs systèmes sont ceux qui ont obtenu les meilleurs résultats globaux mesuré en termes de MAP.

4 Expérimentation

4.1 Utilisation des 2 meilleurs systèmes

Dans cette expérimentation, nous avons sélectionné les 2 systèmes qui ont obtenu la meilleure valeur de MAP globale pour une année donnée de TREC. En fonction de la valeur de la précision à 5 documents (pour chaque requête) l'un ou l'autre des systèmes est utilisé. Le tableau 4 présente les résultats. Dans ce tableau, la colonne « Optimal » indique la valeur de MAP si, pour chaque requête, le meilleur système avait été sélectionné. La colonne « Fusion » correspond aux résultats obtenus par fusion automatique. Les résultats sont indiqués pour les 5 premières requêtes de chaque collection ainsi que sur l'ensemble des requêtes (Global). Sur ces lignes, les chiffres entre parenthèses indiquent la différence en termes de % par rapport au meilleur système sur la collection considérée.

TREC3	Inq102 (1 ^{er})	Citya1 (2 nd)	Optimal	Fusion
Local (5 premières requêtes)	0,6259	0,5783	0,6259	0,6259
	0,2699	0,5667	0,5667	0,5667
	0,1806	0,2681	0,2681	0,2681
	0,7372	0,7354	0,7372	0,7372
	0,2504	0,0035	0,2504	0,2504
Global	0,4226	0,4012	0,4647 (+9,96%)	0,4576 (+8,28%)
TREC5	ETHme1 (1 ^{er})	Uwgcx1 (2 nd)	Optimal	Fusion
Local (5 premières requêtes)	0,0673	0,2215	0,2215	0,2215
	0,0453	0,0932	0,0932	0,0453
	0,6813	0,8600	0,8600	0,6813
	0,3262	0,2909	0,3262	0,3262
	0,1660	0,0543	0,1660	0,1660
Global	0,3165	0,3098	0,3900 (+23,22%)	0,3684 (+16,40%)
TREC6	uwmt6a0 (1 ^{er})	CLAUG (2 nd)	Optimal	Fusion
Local (5 premières requêtes)	0,3185	0,4753	0,4753	0,4753
	0,7671	0,5819	0,7671	0,7671
	0,6556	0,6779	0,6779	0,6556
	0,5000	0,2599	0,5000	0,2599
	0,0302	0,0600	0,0600	0,0600
Global	0,4631	0,3742	0,5079 (+9,67%)	0,4773 (+3,04%)
TREC7	CLARIT98COMB (1 ^{er})	T7miti (2 nd)	Optimal	Fusion
Local (5 premières requêtes)	0,7112	0,8366	0,8366	0,7112
	0,5068	0,3081	0,5068	0,5068
	0,4281	0,3388	0,4281	0,3388
	0,1675	0,2767	0,2767	0,1675
	0,4555	0,5429	0,5429	0,5429
Global	0,3702	0,3675	0,4341 (+17,26%)	0,4069 (+9,91%)

Tab 4 : MAP locale et globale après fusion des deux meilleurs systèmes

Globalement, quelle que soit la collection, l'amélioration des performances globales en termes de MAP par la technique de fusion proposée est proche des 10%. Si l'on considère les 5 premières requêtes de chaque collection, dans la majorité des cas, la sélection est pertinente. Les performances sont plus nuancées pour la collection TREC6. Ces résultats peuvent s'expliquer par les très bonnes performances initiales cette année là du meilleur système (0,46 de MAP).

Il est également intéressant de noter que la différence entre les MAP globales des systèmes initiaux ne permet pas de prédire la pertinence de la fusion. Par exemple, pour la collection TREC5, la différence entre les résultats obtenus par les deux meilleurs systèmes ETHme1 et Uwgcx1 est de 0,0067, pourtant, leur fusion est très efficace (potentiellement elle est de plus de 23% et notre méthode permet d'augmenter les performances de plus de 16%. A l'opposé, pour la collection TREC6, la différence entre les résultats obtenus par les deux meilleurs systèmes uwmt6a0 et CLAUG est de 0,0889 mais leur fusion ne permet qu'une amélioration de moins de 10% potentiellement et de 3% par notre méthode.

4.2 Utilisation des 5 meilleurs systèmes

Dans cette expérimentation, nous avons répété l'expérimentation précédente, sur la base des 5 systèmes qui ont obtenu la meilleure valeur de MAP globale pour la collection considérée. En fonction de la valeur de la précision locale à 5 documents (pour chaque requête), l'un ou l'autre des systèmes est utilisé. Le tableau 5 présente les résultats en utilisant les mêmes conventions que dans le tableau 4.

TREC3	lnq102 (1 ^{er})	...	Assctv2 (5 ^{ème})	Optimal	Fusion
Local	0,6259	...	0,4810	0,6259	0,6259
(5	0,2699	...	0,3987	0,6115	0,5667
premières	0,1806	...	0,3002	0,3803	0,2681
requêtes)	0,7372	...	0,6164	0,7372	0,7372
	0,2504	...	0,3762	0,3762	0,2504
Global	0,4226	...	0,3539	0,4837 (+14,46%)	0,4593 (+8,68%)
TREC5	ETHme1 (1 ^{er})	...	Cor5M2rf (5 ^{ème})	Optimal	Fusion
Local	0,0673	...	0,0773	0,2215	0,2215
(5	0,0453	...	0,0396	0,0932	0,0453
premières	0,6813	...	0,7971	0,8760	0,6813
requêtes)	0,3262	...	0,2632	0,3262	0,3262
	0,1660	...	0,0534	0,1660	0,1660
Global	0,3165	...	0,2931	0,4128 (+30,43%)	0,3786 (+19,62%)
TREC6	uwmt6a0 (1 ^{er})	...	LNmShort (5 ^{ème})	Optimal	Fusion
Local	0,3185	...	0,2127	0,4934	0,4753
(5	0,7671	...	0,4686	0,7671	0,7671
premières	0,6556	...	0,4750	0,6909	0,6556
requêtes)	0,5000	...	0,0879	0,5000	0,2599
	0,0302	...	0,0227	0,0600	0,0600
Global	0,4631	...	0,2902	0,5217 (+12,65%)	0,4703 (+1,55%)
TREC7	CLARI98COMB (1 ^{er})	...	CLART98RANK (5 ^{ème})	Optimal	Fusion
Local	0,7112	...	0,7054	0,8468	0,7112
(5	0,5068	...	0,4833	0,5068	0,5068
premières	0,4281	...	0,3804	0,4546	0,446
requêtes)	0,1675	...	0,0866	0,4204	0,1675
	0,4555	...	0,4334	0,5429	0,5429
Global	0,3702	...	0,3351	0,4820 (+30,20%)	0,4067 (+9,86%)

Tab 5 : MAP locale et globale après fusion des cinq meilleurs systèmes

Lorsque cinq systèmes sont utilisés plutôt que deux, le potentiel d'augmentation augmente quelle que soit la collection de façon significative (de 10 à 14% pour TREC3, de 17 à 30% pour TREC7). L'amélioration des résultats par la méthode de fusion en revanche n'est pas aussi efficace que son potentiel, même si globalement, 5 systèmes permettent tout de même d'améliorer un peu plus les résultats (+9,40% de la MAP par rapport au meilleur système en moyenne sur les 4 collections avec 2 systèmes et +9,93% avec 5 systèmes).

5 Conclusions

Dans cet article nous avons proposé une nouvelle méthode de fusion qui sélectionne le moteur qui doit se charger de la requête, en fonction de la requête traitée. Le choix est effectué sur la base des performances obtenues lorsque les cinq premiers documents sont restitués. Il s'agit donc d'une nouvelle application de la réinjection de pertinence. En effet, dans la littérature, la réinjection de pertinence est utilisée pour reformuler automatiquement la requête initiale à soumettre à un moteur. Cette technique consiste à considérer les premiers documents retrouvés après une première recherche ainsi que leur pertinence. La requête initiale est alors reformulée en donnant plus d'importance aux termes issus des documents pertinents et moins à ceux issus des documents non pertinents. Dans cette étude, la réinjection de pertinence est utilisée pour décider du moteur qui va traiter la requête.

Nous avons montré que des améliorations significatives de la précision moyenne pouvaient ainsi être obtenues. La réinjection de pertinence sous entend qu'un certain nombre de documents ont été vus par l'utilisateur ; ces documents devraient donc être supprimés des collections lors de l'évaluation.

Concernant le prolongement de ces travaux, le principe d'un système choisissant le moteur le plus adapté en fonction de la requête va être approfondi. Plus spécifiquement, nous souhaitons étudier la sélection des moteurs sur la base de critères extraits des requêtes elles-mêmes. En effet, certains travaux ont montré que les performances des moteurs en termes de rappel et de précision sont

corrélées à des critères intrinsèques des requêtes (Mothe & Tanguy, 2005). Une autre piste consiste à proposer des mécanismes de fusion des listes restituées par plusieurs systèmes dépendant des requêtes.

Bibliographie

S. M. Beitzel, O. Frieder, E. C. Jensen, D. Grossman, A. Chowdhury, & N. Goharian. “*Disproving the fusion hypothesis: an analysis of data fusion via effective information retrieval strategies*”, SAC '03”, ACM symposium on Applied computing, pp. 823-827, 2003.

N. J. Belkin, C. Cool, W. B. Croft, & J. P. Callan. “*The effect of multiple query representations on information retrieval performance*”, 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 339-346, 1993.

C. Buckley & D. Harman. “*The NRRC reliable information access (RIA) workshop*”, 27th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 528 – 529, 2004.

J. P. Callan, W. B. Croft, & S. M. Harding. “*The INQUERY Retrieval System*”. Tjoa A. M. and Ramos I. editors, *Database and Expert Systems Applications*, International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 78–83, 1992.

C. W. Cleverdon. The Cranfield tests on English language devices. Aslib Proceedings, vol. 19, Issue 6, pp. 173-194, 1967.

T. Dkaki, G. Hubert, J. Mothe, & E. Orain, “*Recherche de la nouveauté dans les textes: une tâche difficile*”, VSST, Veille Stratégique Scientifique & Technologique, vol. 2, pp. 359-368, 2004.

E. A. Fox and J. A. Shaw. “*Combination of Multiple Searches*”. 2nd Text Retrieval Conference (TREC-2), NIST Special Publication 500-215, pp. 243-252 - 1994.

B. He & I Ounis, “*University of Glasgow at the Robust Track – A query-based Model Selection Approach for Poorly-performing Queries*”, Text Retrieval Conference, 2003.

J. Lee. “*Analysis of multiple evidence combination*”, 22th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 267-276, 1997.

J. Mothe & L. Tanguy, *Linguistic features to predict query difficulty- A case study on previous TREC campaign*, SIGIR workshop on Predicting Query Difficulty - Methods and Applications, 2005.