Fusion de collections dans les métamoteurs

Jacques Savoy[1], Yves Rasolofo[1], Faïza Abbaci[2]

[1]Institut interfacultaire d'informatique – Pierre-à-Mazel 7 – 2000 Neuchâtel – Suisse

[2]Ecole nationale supérieure des Mines de Saint-Etienne – 158 cours Fauriel – 42023 St-Etienne cedex 2 – France

Abstract

We investigate the problem of combining ranked lists of documents provided by multiple search engines. Such a problem must be solved by meta-search engines. In this paper, we suggest a new merging strategy using only the rank of the retrieved items. Moreover, we evaluate various merging approaches based on both a corpus of 2 GB containing news, and a second test-collection of 10 GB of Web pages. Based on our evaluations, our merging approach presents interesting performance and it is well adapted for meta-search engines.

Résumé

Les métamoteurs disponibles sur le Web offrent la possibilité d'interroger de nombreux serveurs d'information soulevant le problème de la fusion des résultats provenant des différents moteurs interrogés. Dans cet article, nous proposons une nouvelle stratégie de fusion n'utilisant que le rang des documents dépistés par les divers moteurs de recherche consultés. De plus, nous évaluons plusieurs approches en utilisant un corpus de 2 GB comprenant des articles de quotidiens et une seconde collection de pages Web d'environ 10 GB. Basée sur nos expériences, notre stratégie, simple et efficace pour la fusion de collections, présente une performance intéressante et se révèle bien adaptée aux métamoteurs de recherche.

Mots-clés : recherche d'informations distribuée, fusion de collection, sélection de collection, Web.

1. Introduction

Pour dépister de l'information dans un environnement distribué comme le Web, la navigation a elle seule ne peut plus être considérée comme une stratégie adéquate et efficace, même avec l'introduction de différentes listes thématiques (par exemple, Yahoo!). Un mécanisme basé sur des requêtes, écrites en langue naturelle, doit être fourni aux internautes afin de rechercher efficacement les pages Web souhaitées. Dans ce but, les moteurs de recherche ont été créés et ils ont permis au Web de grandir dans les proportions que nous lui connaissons. Très souvent placés en tête de liste des sites les plus visités, ils sont utilisés par 85 % des utilisateurs (Schwartz, 1998) comme premier outil de recherche d'informations.

Mais ces outils de dépistage de l'information connaissent plusieurs lacunes. Ainsi, même des systèmes disposant d'une capacité de disque très importante n'indexent qu'une fraction de toute l'information disponible et la couverture de ces moteurs n'augmente pas aussi rapidement que la taille du Web. Par exemple, Northern Light, moteur disposant selon les dernières estimations de la plus grande couverture du Web, n'indexe que 16 % des pages tandis que Lycos couvre environ 2,5 % (Lawrence et Lee Giles, 1999). Afin de proposer une meilleure couverture des sources documentaires du Web, les métamoteurs comme, par exemple, www.MetaCrawler.com ou www.all4one.com interrogent différents moteurs de recherche. Ayant obtenu une liste ordonnée de la part de chaque moteur consulté, ces outils doivent alors fusionner ces réponses afin de ne présenter qu'une seule liste à l'internaute. Selon une étude

récente (Lawrence et Lee Giles, 1999), une couverture presque totale peut être obtenue en soumettant la requête à six moteurs ou plus. A côté de ces métamoteurs généraux, nous voyons apparaître de tels outils pour le Web francophone (par exemple www.ariane6.com ou www.kartoo.com) et d'autres se spécialisent pour apporter une réponse dans un domaine particulier comme, par exemple, les dépêches d'agences et les quotidiens (notre métamoteur disponible à l'adresse www.unine.ch/info/news).

Ce problème de fusion de collections se rencontre également dans les bibliothèques numériques. Dans ce cas, chaque fond documentaire peut être interrogé individuellement mais une réponse adéquate nécessite généralement la fusion des résultats provenant de plusieurs voire de toutes les sources. Pour être plus précis, notre démarche s'inscrit plus particulièrement dans le contexte des métamoteurs ne disposant que de listes ordonnées de résultats. En effet, tous les moteurs de recherche n'indiquent pas un score ou un degré de pertinence calculé pour chaque site retourné.

Cet article est organisé de la manière suivante. Dans le deuxième chapitre, nous décrirons les deux collections de documents utilisées et nous présenterons une première évaluation en optant pour une approche centralisée dans laquelle tous les documents disponibles sont regroupés pour former une seule banque de documents. Ensuite, nous distribuerons nos sources selon différents critères et nous analyserons la performance obtenue en interrogeant les différentes collections ainsi créées. Dans cette troisième partie, nous présenterons également une nouvelle stratégie pour résoudre efficacement et simplement le problème de fusion de collections.

2. Approche centralisée

La recherche d'informations s'appuie sur une longue tradition empirique encourageant les chercheurs à évaluer leur système de dépistage sur la base de collections-tests. Ces corpus disposent d'un nombre important de documents et un ensemble de requêtes. De plus, pour chaque requête, nous disposons de la liste des documents jugés pertinents, jugement posé par un être humain. Dans cet article, nous évaluerons différents modèles de recherche à l'aide de deux collections afin d'obtenir une plus grande validité pour nos résultats. En effet, les caractéristiques, souvent inconnues, d'un corpus peuvent favoriser une approche au détriment d'une autre. De plus, nos investigations ne seront pas limitées à un seul modèle de dépistage de l'information, mais nous évaluerons plusieurs stratégies de recherche permettant ainsi d'obtenir une meilleure compréhension de l'efficacité de divers modèles de dépistage de l'information.

2.1. Les collections-tests

Afin d'évaluer expérimentalement nos propositions, nous avons choisi un corpus de documents rédigés en langue anglaise correspondant à celui de la huitième conférence TREC, corpus nommé TREC8. Cet ensemble comprend 528 155 documents (pour un volume de 1 904 MB) se divisant logiquement en quatre collections soit des articles du *Financial Times*, des documents du *Federal Register*, des dépêches du *Foreign Broadcast Information Service* et des articles du journal *Los Angeles Times*. Avec ce corpus, nous disposons de 50 requêtes couvrant non pas un domaine restreint mais présentant un éventail assez large de thèmes (par exemple «Estonia, economy», «suicides», «airport security», «osteoporosis», «cosmic events»). Comme l'ordinateur ne possède que le titre des besoins d'information, le texte disponible se révèle très bref, comportant en moyenne, deux mots (écart type de 0,97). La table 1 présente quelques statistiques associées à ce corpus ainsi que le nombre de formes dis-

tinctes par collection, le nombre de requêtes pour lesquelles la collection possède au moins un document pertinent et le nombre de requêtes pour lesquelles le corpus retourne une réponse comportant au moins un document.

Afin de compléter cette première évaluation, nous avons également repris la collection de pages Web utilisée lors de la neuvième conférence TREC, corpus nommé TREC9. Cet ensemble comprend 1 692 096 pages Web écrites en anglais pour un volume de 11 033 MB. Cette seconde collection possède donc un volume approximativement six fois supérieur et présente des documents de nature très varié. Ainsi, il n'existe pas de vrai contrôle éditorial sur le contenu et le nombre de fautes d'orthographe se révèle plus important. Dans le cadre de ce corpus, une division logique selon les sources n'avait pas beaucoup de sens et nous avons donc généré huit collections possédant un nombre approximativement égal de documents. Selon les statistiques disponibles dans les tables 1 et 2, nous remarquerons que, pour nos deux collections-tests, chacune des collections ainsi générée ne possède pas des réponses pertinentes pour toutes les requêtes. Ainsi, si le corpus TREC9.2 possède au moins un article pertinent pour 44 des 50 requêtes, le corpus TREC9.4 ne dispose de réponses adéquates que pour 21 requêtes. La situation se présente de manière assez similaire pour le corpus TREC8.

Collection	Taille (en MB)	nb documents	nb de formes	nb req. pert.	nb req. réponse
FT	564	210 158	375 499	49	50
FR	395	55 630	196 220	18	50
FBIS	470	130 471	502 099	43	50
LA Times	475	131 896	337 492	45	50
TREC8	1 904	528 155	1 008 463	50	50

Table 1. Quelques statistiques sur les collections TREC8

Collection	Taille (en MB)	nb documents	nb de formes	nb req. pert.	nb req. réponse
TREC9.1	1 325	207 485	1 800 230	28	48
TREC9.2	1 474	207 429	2 274 318	44	48
TREC9.3	1 438	221 916	1 783 691	38	48
TREC9.4	1 316	202 049	1 684 452	21	48
TREC9.5	1 309	203 073	1 988 627	22	48
TREC9.6	1 311	215 451	2 136 650	24	48
TREC9.7	1 336	200 146	2 223 065	25	47
TREC9.8	1 524	234 547	2 134 040	43	48
TREC9	11 033	1 692 096		50	48

Table 2. Quelques statistiques sur les collections TREC9

Avec le corpus TREC9 correspondant plus à notre centre d'intérêt, nous disposons aussi de 50 requêtes différentes, couvrant plusieurs domaines et correspondant à des exemples soumis au moteur de recherche *Excite*. Les thèmes abordés sont très variables et le texte soumis se révèle très bref comportant en moyenne 2,4 mots (écart type de 0,6). De plus, les termes employés sont très souvent ambigus et très fréquents (par exemple, «deer», «baltimore»). L'orthographe n'est pas toujours exacte («tartin» pour «tartan», «angioplast7» ou «nativityscenes»), l'emploi des majuscules n'est pas systématique («CHEVROLET TRUCKS», «Toronto FIlm Awards» ou «steinbach nutcracker») et parfois la ponctuation est absente («what is the compostion of zirconium»). Ces exemples démontrent les difficultés du traitement de la langue naturelle dans notre contexte.

2.2. Performance de l'approche centralisée

Afin de décrire un modèle de dépistage de l'information, nous devons répondre à trois questions. D'abord savoir comment l'ordinateur représente les documents, ensuite comment les requêtes seront traitées et représentées. Dans les différentes stratégies étudiées dans cet article, cette représentation se limite à un ensemble de termes simples pondérés. Dans tous les cas, un pré-traitement est prévu afin d'éliminer les formes très fréquentes et peu ou non porteuses de sens (comme les articles «the», «of», les prépositions «and», «or» ou les pronoms «I», «we») d'une part, et d'autre part, à éliminer la marque du pluriel («beavers» deviendra «beaver») voire d'autres dérivations suffixales (par exemple, «discussing» redonnera «discus») (Savoy, 1997).

Finalement, un modèle spécifie la manière dont l'appariement entre la requête et les documents se réalise. Pour ce dernier aspect, et dans toutes nos expériences, le degré de similarité de chaque document avec la requête (ou son degré de pertinence jugé par la machine) est obtenu par le calcul du produit interne (Salton, 1989, p. 318).

Pour indexer un document ou une requête, différentes stratégies de pondération existent. La manière la plus simple est d'effectuer une indexation binaire (approche notée «bnn»). Le système d'indexation retiendra les formes les plus représentatives du contenu sémantique du document ou de la requête. Si nous recourons à cette approche tant pour les documents et que pour les requêtes (approche notée «doc : bnn, requête : bnn»), le degré de similarité de chaque document indiquera le nombre de termes communs entre celui-ci et la requête.

Evidemment, d'autres stratégies d'indexation ont été proposées et nous pouvons tenir compte de la fréquence d'occurrence des termes d'indexation dans le document ou la requête (approche «nnn»). Ainsi, si un mot se répète dans une page, son importance sera accrue lors de l'indexation. Dans le modèle «doc : nnn, requête : nnn», le calcul du degré de similarité tiendra compte de la fréquence d'occurrence des termes communs entre le document et la requête. Pour le modèle vectoriel classique «doc : ntc, requête : ntc», l'indexation tient compte à la fois de la fréquence d'occurrence du terme dans le document et de l'inverse de sa fréquence documentaire (ou nombre de documents dans lesquels le terme apparaît). Ainsi, si un mot apparaît très souvent dans des documents, son poids diminuera lors de l'indexation. En effet, un tel terme ne permettra pas à l'ordinateur de distinguer les documents pertinents des autres. Finalement, dans cette stratégie «doc : ntc, requête : ntc», les poids sont normalisés selon la formulation du cosinus. D'autres stratégies tiendront compte également de la longueur du document ou de la requête (voir annexe 1).

Ces dernières années, et en particulier sous l'impulsion des conférences TREC, d'autres modèles de dépistage de l'information ont été proposés, en particulier diverses variantes du modèle vectoriel comme les stratégies «doc : Lnu, requête : ltc» (Buckley et al., 1996), «doc : dnu, requête : dtc» (Singhal et al., 1999) ou l'approche «doc : atn, requête : ntc». Depuis quelques années, l'approche probabiliste Okapi (Robertson et al., 1995) rencontre également un grand succès grâce à la bonne précision moyenne de cette stratégie. Ce modèle sera donc aussi repris dans nos évaluations.

Comme méthodologie d'évaluation, nous avons retenu la précision moyenne (Salton, 1983, Section 5.2) mesurant la qualité de la réponse fournie par l'ordinateur, mesure utilisée par la conférence TREC. L'évaluation des différentes stratégies de dépistage retenues au regard de nos deux collections-tests est indiquée dans la table 3. Finalement, pour décider si un système de dépistage est meilleur qu'un autre, nous admettons comme règle d'usage qu'une différence de 5 % dans la précision moyenne peut être considérée comme significative.

	Précision moyenne (% changement)	
collection	TREC8	TREC9
nombre doc. pertinents	4 728	2 617
modèle	50 requêtes	50 requêtes
doc: Okapi, requête: npn	25,66	19,86
doc : dnu, requête : dtn	24,03 (-6,4%)	15,42 (-22,4%)
doc : atn, requête : ntc	22,75 (-19,1%)	14,59 (-26,5%)
doc : Lnu, requête : ltc	21,63 (-15,7%)	16,81 (-15,4%)
doc : lnc, requête : ltc	14,44 (-43,7%)	3,79 (-80,9%)
doc : ltc, requête : ltc	13,60 (-47,0%)	5,20 (-73,8%)
doc : ntc, requête : ntc	12,21 (-52,4%)	8,00 (-59,7%)
doc : bnn, requête : bnn	11,51 (-50,1%)	5,13 (-74,2%)
doc : nnn, requête : nnn	4,47 (-82,6%)	4,06 (-79,6%)

Table 3. Précision moyenne des moteurs de recherche sur les deux corpus (approche centralisée)

Sur la base des évaluations indiquées dans la table 3, nous noterons que le modèle probabiliste Okapi présente la meilleure précision moyenne et que cette dernière est significativement supérieure aux autres approches (différence de précision moyenne supérieure à 5 %). En deuxième position, nous trouvons l'approche «doc : dnu, requête : dtn» pour le corpus TREC8 ou «doc : Lnu, requête : ltc» pour la collection TREC9 extraite du Web. Ces résultats indiquent également que le modèle vectoriel classique «doc : ntc, requête : ntc» occupe seulement le septième rang pour le corpus TREC8 et le cinquième dans le cadre de la collection TREC9.

Les requêtes étant différentes d'un corpus à l'autre, une comparaison directe entre les performances sur la collection TREC8 et TREC9 n'est donc pas possible. Nous remarquons que les modèles vectoriels «doc : lnc, requête : ltc» ou «doc : ltc, requête : ltc» ont plutôt été conçus pour des collections de documents possédant peu de fautes d'orthographe. Pour ces approches, la précision moyenne sur des pages Web se dégradent fortement. De plus, l'approche basée sur une indexation binaire («doc : bnn, requête : bnn») semble aussi connaître une baisse d'efficacité lorsque cette stratégie est utilisée sur le Web.

3. Fusion de collections

Dans un premier temps, l'ensemble des documents étaient réunis pour former une seule banque de documents. Cette approche constituait un axiome dans la recherche d'informations classique (Salton, 1989). Or, le corpus TREC8 peut se diviser logiquement en différentes sources correspondant aux différents quotidiens. Cette situation reflète bien le cas d'une bibliothèque numérique possédant plusieurs sources d'information pouvant être interrogées par le même moteur. Ainsi, en réponse à une requête d'un usager, le système d'interrogation envoie cette demande à l'ensemble des moteurs de recherche. Ces derniers calculent leur réponse et la retournent au système d'interrogation. Ce dernier peut soit présenter la liste obtenue par chaque source séparément, soit fusionner ces listes pour n'en visualiser qu'une seule. La manière de générer cette liste unique en ordonnant les résultats en fonction de leur pertinence par rapport à la requête soumise correspond au problème de fusion des collections. Ce phénomène se rencontre également dans les métamoteurs de recherche dont la qualité de réponse dépend fortement de la stratégie de fusion de collections utilisée.

3.1. Stratégies de fusions

Afin de résoudre ce problème de fusion, différentes propositions ont été avancées. Comme première approche, nous pouvons admettre que chaque collection contient un nombre approximativement égal de documents pertinents et que ceux-ci se retrouvent distribués de manière identique dans les réponses obtenues des serveurs (Voorhees et al., 1995). Selon ces hypothèses, nous pouvons construire la liste finale en prenant un élément dans chaque liste puis en recommençant. Cette stratégie, nommée «à chacun son tour», se rencontre souvent dans certains métamoteurs (Selberg, 1999) et, en particulier, cette approche a été préconisée par les premiers outils de ce type disponibles sur Internet.

Cependant, en plus du rang de chaque article dépisté, les moteurs de recherche fournissent parfois un score (degré de similarité) entre la requête et le document retourné. Or cette information n'est pas fournie de façon systématique par tous les moteurs de recherche. Dans le cas des métamoteurs, nous ne pouvons donc pas baser une stratégie de fusion sur ce renseignement, invalidant ainsi plusieurs stratégies de fusion proposées (Callan et al., 1995).

Dans un article récent, Yager et Rybalov (1998) suggèrent une généralisation de la fusion «à chacun son tour» à l'aide d'un paramètre noté α. Pour comprendre les grandes lignes de cette stratégie de fusion, un exemple nous aidera. Imaginons que nous ayons interrogé quatre collections dont la première retourne les neuf documents ordonnés suivants (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9), la deuxième cinq documents (b_1, b_2, b_3, b_4, b_5), la troisième trois documents (c_1, c_2, c_3) et la quatrième un seul document (d_1). La stratégie «à chacun son tour» effectuera la fusion en retournant la liste résultante suivante (a_1, b_1, c_1, d_1, a_2, b_2, c_2, a_3, b_3, c_3, a_4, b_4, a_5, b_5, a_6, a_7, a_8, a_9). Le résultat sera identique si l'on recourt à l'algorithme de Yager et Rybalov avec $\alpha = 0$. Dans ce cas, nous privilégions les documents retournés dans les premiers rangs par les divers moteurs. En revanche, si nous fixons le paramètre $\alpha = 1$, nous obtiendrons (a_1, a_2, a_3, a_4, a_5, b_1, a_6, b_2, a_7, b_3, c_1, a_8, b_4, c_2, a_9, b_5, c_3, d_1). Comme les listes résultantes possèdent des longueurs différentes, nous favorisons les plus longues listes au détriment des listes plus brèves. Entre ces deux possibilités extrêmes, nous pouvons faire varier le paramètre α pour donner au rang une plus grande importance (α faible) ou plutôt pour favoriser les longues listes (α proche de 1). En fixant le paramètre $\alpha = 0{,}5$, nous obtiendrons (a_1, a_2, a_3, b_1, a_4, b_2, c_1, a_5, b_3, c_2, d_1, a_6, b_4, c_3, a_7, b_5, a_8, a_9) ce qui correspond à accorder autant d'importance à la longueur des listes qu'au rang des documents dépistés.

Pour être complet, signalons les travaux de (Le Calvé et Savoy, 2000) et de (Voorhees et al., 1995) qui proposent de recourir à un algorithme d'apprentissage (régression logistique respectivement classification) afin d'améliorer la fusion des listes de résultats. Or, dans la pratique, nous ne disposons pas souvent de requêtes passées avec leurs jugements de pertinence afin d'ajuster au mieux les paramètres sous-jacents à un modèle.

3.2. Stratégies de sélections

Afin d'améliorer la fusion des résultats provenant de différentes sources, nous avons imaginé une stratégie de sélection visant à exclure les moteurs dont la réponse n'apportait aucun document pertinent. En effet, en réponse à une requête très brève, les moteurs de recherche interrogés retournent très souvent une réponse dans laquelle les articles dépistés contiennent bien le (ou les) terme(s) de la requête mais utilisé(s) dans un contexte différent. Ainsi, face à la requête «hair transplant», un moteur dépistera des sites Web parlant de transplantation cardiaque ou de coiffure mais pas du sujet précis intéressant l'internaute.

Afin de mettre en œuvre cette stratégie de sélection, notre système d'interrogation inspecte le contenu des cinq premiers documents retournés à la recherche de l'occurrence de deux (ou plusieurs) termes de la requête dans un contexte restreint, comme le titre, une phrase ou un paragraphe. Or les expériences menées nous ont conduit dans une impasse. En effet, les pages Web ne disposent pas souvent d'un titre et souvent les deux termes de la requête n'apparaissent pas ensemble dans la partie logique encadrée par la balise <H1>. De plus, il était rare de voir conjointement plusieurs termes de la requête dans la même page. Enfin, l'évaluation de ces différentes approches n'a pas permis d'améliorer la précision moyenne ; l'effet de ces diverses stratégies de sélection était, le plus souvent, une baisse significative de la performance moyenne.

3.3. Notre stratégie de fusion

Ayant renoncé à opérer une sélection, nous avons conçu une stratégie de fusion fonctionnant en deux étapes. En premier lieu, nous évaluons l'importance de chaque collection consultée. Cette pondération est proportionnelle au nombre de documents extraits (ou longueur de la liste L_i retournée) par ce corpus par rapport au maximum des différentes listes retournées par tous les moteurs (maximum noté $\max(L_j)$). Cette pondération, attribuée à la i^e collection et représentée par la valeur α_i, s'évalue selon la formule suivante :

$$\alpha_i = (1 - k) + k \cdot [\log (1 + L_i) / \log(1 + \max(L_j))]$$

dans laquelle k est une constante (fixée à 0,4 dans nos évaluations). Dans notre modèle, le recours au logarithme se justifie par notre volonté d'attribuer une pondération de moins en moins forte par rapport au nombre croissant de documents retournés. Ainsi, si ce nombre passe de 10 à 11, nous considérons que cet accroissement unitaire doit être valorisé de manière plus importante que le passage de 210 à 211.

Sur la base de cette pondération, notre algorithme de fusion calcule une probabilité de pertinence, notée Prob [D | rg], dépendant du rang (noté rg) du document D extrait par la i^e collection selon l'équation suivante :

$$\text{Prob}[D \mid rg] = \exp(\alpha_i + \beta \cdot \log(rg)) / [1 + \exp(\alpha_i + \beta \cdot \log(rg))]$$

dans laquelle rg indique le rang obtenu par le document D dans la i^e collection considérée, β une constante (fixée à 0,05 dans nos évaluations). Cette dernière équation dérive directement de la méthode de la régression logistique (Bookstein et al., 1992). Durant la phase de fusion des listes, cette probabilité calculée pour chaque document dépisté est utilisée comme critère de tri, débutant par la probabilité la plus élevée vers la valeur la plus faible.

3.4. Evaluation

A priori, il se révèle difficile de sélectionner la meilleure stratégie de fusion de collections. Afin d'obtenir une meilleure vue d'ensemble de ces différentes approches, nous les avons évaluées à l'aide de nos deux collections-tests. Dans les tables 4 et 5, en plus de la performance obtenue par un moteur gérant de manière centralisée l'ensemble des documents disponibles, il nous paraît intéressant d'évaluer la précision moyenne de l'approche «à chacun son tour» optimale. Dans ce cas, le système de fusion connaît le rang de tous les documents pertinents dépistés. Dans une situation aussi favorable, il peut ignorer certaines collections et extraire autant d'éléments d'une liste que nécessaire. Selon nos expériences, la fusion optimale permet d'accroître la précision moyenne d'environ 11,5 % sur la collection TREC8 et 49,1 % sur le corpus TREC9.

	Précision moyenne				
stratégie de fusion modèle	collection centralisée TREC8	«à chacun son tour» optimale	«à chacun son tour»	Yager *et al.* $\alpha = 0,5$	notre approche
doc:Okapi, req:npn	25,66	27,63	18,84	19,30	22,14
doc : dnu, req : dtn	24,03	25,60	17,40	18,12	20,77
doc : atn, req : ntc	22,75	24,17	16,18	16,58	18,93
doc : Lnu, req : ltc	21,63	22,06	15,31	17,07	18,57
doc : lnc, req : ltc	14,44	15,37	10,23	11,43	12,63
doc : ltc, req : ltc	13,60	15,42	10,24	10,44	12,05
doc : ntc, req : ntc	12,21	12,24	8,17	9,99	9,97
doc : bnn, req : bnn	11,51	12,45	7,58	8,28	9,39
doc : nnn, req : nnn	4,47	6,82	4,37	5,36	5,20
gain / perte moyen en %		+11,47 %	-26,18 %	-18,67 %	-11,40 %

Table 4. Précision moyenne de différentes stratégies de fusion (TREC8)

La stratégie «à chacun son tour» permet d'obtenir une précision moyenne d'environ 26,2 % à 28,9 % inférieure à l'interrogation d'une collection unique regroupant tous les documents. Nous pouvons donc affirmer que chaque collection contient un nombre variable de documents pertinents et que ceux-ci ne se retrouvent pas distribués de manière identique dans les réponses obtenues des serveurs. Pour mesurer l'efficacité de l'approche proposée par (Yager et Rybalov, 1998), nous avons fixé le paramètre α à 0,5, valeur attribuant autant de poids au rang qu'à la longueur des listes retournées. Cette dernière présente une meilleure efficacité pour le corpus TREC8 (table 4) que la fusion «à chacun son tour». Cette constatation ne se révèle pas confirmée par le corpus extrait du Web (table 5).

Notre modèle de fusion propose la meilleure stratégie pour les deux corpus utilisés. Tenir compte de l'efficacité des différents moteurs de recherche en prenant en compte le nombre d'articles retournés par chacun d'eux s'avère une stratégie intéressante. Cependant, l'accroissement n'est pas aussi net et significatif pour la collection TREC9. En se limitant au meilleur moteur de dépistage de l'information, c'est-à-dire le modèle Okapi, notre stratégie de fusion propose, pour le corpus TREC8, une précision moyenne de 22,14 contre 18,84 pour l'approche «à chacun son tour» soit une augmentation de 17,5 %. Pour la collection TREC9, la performance obtenue s'élève à 12,6, soit 1,8 % de mieux que la fusion «à chacun son tour».

	Précision moyenne				
stratégie de fusion modèle	collection centralisée TREC9	«à chacun son tour» optimale	«à chacun son tour»	Yager *et al.* $\alpha = 0,5$	notre approche
doc:Okapi, req:npn	19,86	22,58	12,38	11,76	12,60
doc : Lnu,req : ltc	16,81	19,29	9,81	9,05	9,88
doc : dnu, req : dtn	15,42	18,82	10,37	9,53	10,63
doc : atn, req : ntc	14,59	19,01	9,93	10,63	10,18
doc : ntc, req : ntc	8,00	11,99	5,37	5,14	5,88
doc : ltc, req : ltc	5,20	10,01	4,40	4,46	4,43
doc : bnn, req : bnn	5,13	8,27	3,91	3,47	3,96
doc : nnn, req : nnn	4,06	6,22	2,89	3,16	3,26
doc : lnc, req : ltc	3,79	7,74	3,21	3,20	3,15
gain / perte moyen en %		+49,09 %	-28,91 %	-30,26 %	-26,64 %

Table 5. Précision moyenne de différentes stratégies de fusion (TREC9)

La performance calculée dans les tables 4 et 5 s'est effectuée en utilisant le même moteur pour interroger les quatre respectivement huit collections des corpus TREC8 et TREC9. Or, les métamoteurs interrogent des moteurs œuvrant avec des stratégies de dépistage différentes pour chaque collection. Afin d'analyser cette situation, nous allons interroger les deux corpus en recourant à différents moteurs de recherche. Dans notre cas, nous avons retenu le modèle probabiliste Okapi et les trois approches vectorielles dont la distribution pour la collection TREC8 est la suivante FT interrogée par «doc : Okapi, requête : npn», FR par «doc: dnu, requête : dtn», FBIS via «doc : Lnu, requête : ltc» et LA Times par «doc : atn, requête : ntc». Pour le corpus TREC9, la distribution des stratégies de recherche est la suivante : TREC9.1 et TREC9.5 par «doc : Okapi, requête : npn», TREC9.2 et TREC9.6 via «doc : dnu, requête : dtn», TREC9.3 et TREC9.7 par «doc : Lnu, requête : ltc» et TREC9.4 et TREC9.8 par «doc : atn, requête : ntc». Dans cette situation, nous ne pouvons donc plus donner la performance obtenue par une approche centralisée pour fin de comparaison.

	Précision moyenne (% changement)			
stratégie de fusion modèle	«à chacun son tour»	«à chacun son tour» optimale	Yager *et al.* $\alpha = 0,5$	notre approche
TREC8	16,85	24,73	17,64	19,95
TREC8 + permutation	16,83	24,02	18,02	19,92
TREC8 + permutation	17,12	24,47	17,97	20,27
TREC8 + permutation	17,16	24,99	17,33	20,06
moyenne TREC8	16,99	24,55 (+44,5%)	17,74 (+4,4%)	20,05 (+18%)
TREC9	10,19	19,83	10,42	10,35
TREC9 + permutation	10,88	19,19	10,86	10,96
TREC9 + permutation	11,42	20,87	10,49	11,37
TREC9 + permutation	9,86	19,34	9,37	10,15
moyenne TREC9	10,59	19,81 (+19,8%)	10,29 (-2,9%)	10,71 (+1,1%)

Table 6. Précision moyenne de différentes stratégies de fusion

La table 6 indique la précision moyenne de quatre stratégies de fusion. Selon ces dernières expériences, nous constatons que notre stratégie de fusion présente encore la meilleure performance. Afin de renouveler notre expérience, nous avons permuté trois fois la distribution des quatre stratégies de recherche par rapport aux différentes collections. En observant l'efficacité dans chaque cas ou en calculant la moyenne, notre approche propose une meilleure efficacité qui se révèle même significative dans le cas de la collection TREC8 et presque significative dans le cadre du corpus TREC9 (10,71 contre 10,29, augmentation de 4,1 % par rapport à l'approche de Yager et Rybalov (1998)).

4. Conclusion

Sur la base de nos expériences basées sur deux collections-tests, neuf stratégies de dépistage et des requêtes extrêmement courtes (mais similaires à celles que nous rencontrons sur le Web), les conclusions suivantes peuvent être tirées. D'abord, le modèle probabiliste Okapi présente une performance très attractive. Il propose significativement la meilleure réponse dans le cadre de nos deux collections-tests œuvrant sur des volumes importants (2 GB et 10 GB). Ensuite, le modèle vectoriel classique «doc : ntc, requête : ntc» ne propose pas une stratégie d'indexation et de dépistage efficace. Or cette approche est souvent recommandée. Finalement, notre stratégie de fusion propose une alternative simple et efficace dont la précision moyenne se révèle meilleure que celle de la fusion «à chacun son tour» ou que

l'approche proposée par (Yager et Rybalov, 1998). N'utilisant que le rang et la longueur des listes retournées, notre stratégie de fusion est donc très adaptée aux métamoteurs ou aux bibliothèques numériques œuvrant sur un ensemble de sources documentaires interrogées avec le même moteur de dépistage de l'information. Nos travaux en cours tentent à trouver de nouvelles stratégies afin d'améliorer encore la performance des métamoteurs de recherche (voir notre métamoteur, encore en phase expérimentale, à l'adresse www.unine.ch/info/news). Enfin, le problème de fusion de collections se rencontre également dans la recherche multilingue (Peters, 2001) dans laquelle la requête soumise dans une langue donnée doit dépister des documents écrits dans différentes langues. Dans de tels systèmes, la requête soumise est souvent traduite automatiquement et est soumise à diverses collections, chacune renfermant des documents rédigés dans une langue donnée. La fusion doit donc s'opérer dans un contexte quelque peu différent ; chaque collection étant interrogée par une requête différente car écrite avec des mots appartenant à diverses langues naturelles (Savoy, 2002).

Remerciements

Cette recherche a été subventionnée en partie par le FNS avec le subside 21-58 813.99 (J. Savoy et Y. Rasolofo) et par la Région Rhône-Alpes (bourse Eurodoc de F. Abbaci).

Références

Bookstein A., O'Neil E., Dillon M., Stephen D. (1992). Applications of Loglinear Models for Informetric Phenomena. *Information Processing & Management*, vol.(28):75-88.

Buckley C., Singhal A., Mitra M., Salton G. (1996). New Retrieval Approaches using SMART. Proceedings of *TREC'4*, pages 25-48.

Callan J. P., Lu Z., Croft W. B. (1995). Searching Distributed Collections with Inference Networks. Proceedings of the *ACM-SIGIR'95*, pages 21-28.

Gordon M., Pathak P. (1999). Finding Information on the World Wide Web : The Retrieval Effectiveness of Search Engines. *Information Processing & Management*, vol.(35):141-180.

Lawrence S., Lee Giles C. (1999). Accessibility of Information on the Web. *Nature,* vol.(400):107-110.

Le Calvé A., Savoy J. (2000). Database Merging Strategy Based on Logistic Regression. *Information Processing & Management*, vol.(36):341-359.

Peters, C. (Ed.) (2001). *Cross-Language Information Retrieval and Evaluation, Workshop of the Cross-Language Evaluation Forum, CLEF-2000*. Lecture Notes in Computer Science, vol.(2069), Berlin : Springer-Verlag.

Robertson S. E., Walker S., Hancock-Beaulieu M. M. (1995). Large Test Collection Experiments on an Operational, Interactive System : OKAPI at TREC. *Information Processing & Management*, vol. (31):345-360.

Salton G. (1989). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Reading (MA) : Addison-Wesley.

Salton G., McGill M. J. (1983). *Introduction to Modern Information Retrieval*. New York (NY) : McGraw-Hill.

Savoy J., Rasolofo Y. (2000). Recherche d'informations dans un environnement distribué. Actes de *TALN 2000*, pp. 317-326.

Savoy J. (2002). Report on CLEF-2001 Experiments: Effective Combined Query-Translation Approach. In Peters, C., editor, *Cross-Language Information Retrieval and Evaluation, Workshop*

of the Cross-Language Evaluation Forum, CLEF-2001. Lecture Notes in Computer Science. Berlin : Springer-Verlag.

Savoy J. (1997). Statistical Inference in Retrieval Effectiveness Evaluation. *Information Processing & Management*, vol.(33):495-512.

Schwartz C. (1998). Web Search Engines, *Journal of the American Society for Information Science*, vol. (49):973-982.

Selberg E. W. (1999). *Towards Comprehensive Web Search.* Ph.D. Thesis, University of Washington.

Singhal A., Choi J., Hindle D., Lewis D. D., Pereira F. (1999). AT&T at TREC-8. Proceedings of *TREC-8*, pp. 239-251.

Voorhees E. M., Gupta N. K., Johnson-Laird B. (1995). Learning Collection Fusion Strategies. Proceedings of the *ACM-SIGIR'95*, pp. 172-179.

Yager R. R., Rybalov A. (1998). On the Fusion of Documents from Multiples Collection Information Retrieval Systems. *Journal of the American Society for Information Science*, vol.(49):1177-1184.

Annexe 1. Formules d'indexation

Afin d'attribuer un poids w_{ij} reflétant l'importance de chaque terme d'indexation T_j, j = 1, 2, ..., t, dans un document D_i, nous pouvons recourir à l'une des formules décrites dans la table ci-dessous. Dans cette dernière, tf_{ij} indique la fréquence d'occurrence du terme T_j dans le document D_i (ou dans la requête), n représente le nombre de documents D_i dans la collection, df_j le nombre de documents dans lesquels le terme T_j apparaît (fréquence documentaire), et idf_j l'inverse de la fréquence documentaire ($idf_j = \log[n/df_j]$). Les constantes ont été fixées aux valeurs suivantes : slope=0,2, pivot=150, b=0,75, k=2, k_1=1,2, advl=900. De plus, la longueur du document D_i (ou le nombre de termes d'indexation associé à ce document) est notée par nt_i, la somme de valeurs tf_{ij} par l_i et $K = k \cdot [(1 - b) + b \cdot (l_i/advl)]$.

bnn	$w_{ij} = 1$	nnn	$w_{ij} = tf_{ij}$
dtn	$w_{ij} = [\log(\log(tf_{ij}) + 1) + 1] \cdot idf_j$	atn	$w_{ij} = idf_j \cdot [0{,}5 + 0{,}5 \cdot tf_{ij} / \max tf_{i.}]$
Okapi	$w_{ij} = \left((k_1 + 1) \cdot tf_{ij}\right) \Big/ \left(K + tf_{ij}\right)$	npn	$w_{ij} = tf_{ij} \cdot \log\left[\left(n - df_j\right) \Big/ df_j\right]$
lnc	$w_{ij} = \dfrac{\log(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} \left(\log(tf_{ik}) + 1\right)^2}}$	ntc	$w_{ij} = \dfrac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik} \cdot idf_k\right)^2}}$
ltc	$w_{ij} = \left(\log(tf_{ij}) + 1\right) \cdot idf_j \Big/ \sqrt{\sum_{k=1}^{t} \left(\left(\log(tf_{ik}) + 1\right) \cdot idf_k\right)^2}$		
dnu	$w_{ij} = \dfrac{\left(1 + \log\left(1 + \log(tf_j)\right)\right) \Big/ (1 + pivot)}{(1 - slope) \cdot pivot + slope \cdot nt_i}$		
Lnu	$w_{ij} = \dfrac{\left(1 + \log(tf_j)\right) \Big/ (1 + pivot)}{(1 - slope) \cdot pivot + slope \cdot nt_i}$		

Table 7. Formules de pondération lors de l'indexation