Prédiction du SRI à utiliser en fonction des critères linguistiques de la requête

Désiré Kompaoré* — Josiane Mothe* — Alain Baccini ** — Sébastien Dejean **

** Institut de Recherche en Informatique de Toulouse, IRIT*
118 route de Narbonne, 31062, Toulouse cedex04, France
{kompaore, mothe}@irit.fr

*** Université Paul Sabatier*
Institut de Mathématiques de Toulouse, 31062 Toulouse cedex 04, France
{sebastien.dejean, baccini}@math.ups-tlse.fr

RÉSUMÉ. *En recherche d'information (RI), plusieurs techniques existent et sont utilisées par les systèmes pour répondre de manière efficace aux requêtes des utilisateurs. Nous nous intéressons dans ce papier à comment utiliser les caractéristiques linguistiques des requêtes pour prédire le(s) meilleur(s) système(s) à utiliser pour une requête donnée. Pour ce faire, nous avons utilisé 13 critères linguistiques définis dans (Mothe et al, 2005) pour catégoriser les requêtes de la campagne TREC 3, 5, 6 et 7. Chaque catégorie de requête est ensuite associée à un ou plusieurs systèmes. Les résultats montrent que notre méthode permet d'améliorer les performances de la recherche en fonction des campagnes TREC de 2,31% à 6,81% pour la P@15 et jusqu'à 16,27% pour la MAP.*

ABSTRACT. *In information retrieval (IR), many techniques are used by systems to improve their efficiency in the retrieval process. In this paper, we consider13 linguistic features for queries clustering (Mothe et al., 2005). Queries are TREC 3, 5, 6 and 7 topics. We then associate to each cluster a system that will be used for all queries in this cluster. Results show that P@15 can be improved for example from 2.31% to 6.81% and up to 16.27% for the MAP measure, depending of the year considered.*

MOTS-CLÉS: *recherche d'information, fusion de données, classification de requêtes, évaluation, TREC adhoc.*

KEYWORDS: *information retrieval, data fusion, queries clustering, evaluation, TREC adhoc.*

1. Introduction

De récentes études dans le domaine de la recherche d'information (RI) ont montré que les performances de différents systèmes de recherche (SRI) sont très variables (par exemple, le système A peut être très performant pour une requête donnée et être très médiocre pour une autre requête, alors que le système B a des résultats inversés). (Buckley et al., 2004) dans leur étude ont considéré que maîtriser la variabilité dans les résultats est complexe à cause d'un certain nombre de paramètres : la formulation de la requête, la relation entre la requête et les documents, ainsi que les caractéristiques des systèmes utilisés.

La requête et les documents: A ce niveau, l'utilisateur dispose d'une multitude de possibilités pour exprimer son besoin d'information, de même que les systèmes peuvent analyser une requête de plusieurs manières différentes. Les travaux en RI s'intéressent de plus en plus à la sémantique. L'hypothèse sous-jacente est qu'en étant capable de capturer la sémantique des requêtes et des documents, on améliore et enrichi le processus de recherche. Cependant, la représentation classique des requêtes sous forme de mots pris isolément ne permet pas d'associer une sémantique aux requêtes. Les techniques de traitement automatique de langues (TAL) s'adaptent bien quant à elles à ce contexte en essayant de reproduire une interprétation humaine du langage, en supposant que les différentes règles grammaticales ainsi que les relations entre les mots dans le langage peuvent être décomposées de manière scientifique. Un exemple de technique linguistique appliquée à la RI est l'expansion des termes de la requête avec des synonymes et des méronymes (Buscaldi et al., 2005). Plusieurs autres éléments linguistiques peuvent aussi être considérés lors de l'analyse de la requête.

Les systèmes utilisés : Les systèmes peuvent utiliser différentes techniques pour répondre à une requête donnée. En fonction de la technique utilisée, les résultats de la recherche peuvent donc varier. La fusion des données s'intéresse à ce genre de variabilité pour augmenter les performances des systèmes, en combinant les résultats issus de différentes stratégies de recherche (Fox et al., 1994). Il existe d'autres types de fusion mais nous ne nous intéressons ici qu'à la fusion des données.

Dans ce papier, nous nous intéressons à la variabilité des requêtes (comment elles sont exprimées) ainsi qu'à la variabilité des systèmes. Notre hypothèse se base sur le fait qu'un système A peut être plus performant pour un certain type de requêtes qu'un système B et vice-versa.

La campagne d'évaluation TREC permet d'évaluer sur une même base différents systèmes qui utilisent tous la même collection de documents pour répondre aux mêmes requêtes. Les systèmes sont évalués avec la mesure de *rappel* qui mesure la capacité du système à retrouver tous les documents pertinents, et la *précision* qui mesure la capacité des systèmes à ne retrouver que les documents pertinents.

Ce papier traite de la classification des requêtes issues des campagnes de TREC. Dans notre approche, les requêtes sont classifiées en fonction de critères

linguistiques (Mothe et al., 2005). Nous évaluons ensuite les performances des systèmes à travers un ensemble de mesures obtenues avec le program *trec_eval* ; il s'agit de la MAP, R-précision, P@5, P@10 et P@15. Cette évaluation est possible car TREC fournit les résultats des différents systèmes qui participent à la campagne d'évaluation. Nous montrons qu'en sélectionnant le système à utiliser pour une catégorie de requêtes données, nous pouvons améliorer la P@10 jusqu'à 16,33%, et la P@15 jusqu'à 14,64% par rapport aux résultats obtenus par le meilleur système pour chacune des mesures. Nous avons aussi évalué notre méthode en suivant un processus d'apprentissage et de test. En fonction de la campagne TREC, les résultats sont améliorés de 3,72% à 5,97% pour la P@5, et de 1,48% à 6,73% pour la P@10 par exemple.

Nos travaux sont présentés dans ce papier de la manière suivante : nous présentons dans la section 2 les différents travaux qui ont été effectués dans la littérature. Nous présenterons ensuite le jeu de données que nous avons utilisé pour nos tests ainsi que les différents critères linguistiques que nous avons utilisé pour catégoriser nos requêtes (section 3). La section 4 présente la méthode de classification des requêtes et de sélection du système à utiliser. Nous donnons ensuite dans la section 5 les résultats que nous avons obtenus. Nous évaluons dans la section 6 notre méthode sur les 4 années de TREC (3, 5, 6, et 7). Nous terminons ce papier en section 7 par la conclusion de nos travaux.

2. Travaux du domaine

Plusieurs variabilités existent dans le processus de recherche d'information. Ces variabilités concernent aussi bien l'expression de la requête que les différentes techniques de recherche utilisées. Lors du séminaire sur l'accès fiable à l'information (RIA workshop), (Buckley et al ., 2004) ont analysés les raisons de la variation des performances des systèmes en fonction des requêtes. Les auteurs se sont intéressés aux requêtes pour lesquelles les systèmes participants à TREC ont échoués et ont ainsi déterminé 10 catégories d'échec (Buckley, 2004). Une des conclusions de ce séminaire était que « comparer les résultats des systèmes lorsque ces derniers utilisent toutes les parties des requêtes TREC et ceux des systèmes lorsque les requêtes TREC sont utilisées partiellement, permet d'avoir une idée dans les résultats de l'importance des parties de la requête qui sont utilisées pour la recherche ». En effet, certains des échecs sont dus au fait que les systèmes favorisent seulement certains aspects de la requête.

La variabilité des requêtes a été étudiée dans (Buckley et al., 2000) lors de leurs expériences dans la tâche « requête » de TREC8. Les auteurs ont utilisé 4 versions pour chacune des 50 requêtes de TREC. Ils ont étudié la variation entre le fait d'utiliser des requêtes longues ou courtes ; leurs résultats montrent que les requêtes courtes permettent d'avoir de meilleurs résultats. La variabilité des requêtes a aussi été étudiée dans (Beitzel et al., 2004). Dans leur papier, les auteurs ont montré que la

normalisation de la taille de la requête qu'ils ont utilisée dans leurs travaux donne de meilleurs résultats que les méthodes de fusion des données. (Chang et al., 2004) se sont intéressés quant à eux aux méthodes d'expansion automatique des requêtes. Ils ont extrait des caractéristiques de l'espace des documents et ont utilisés ces caractéristiques pour catégoriser les requêtes.

La difficulté de la requête est quant à elle une autre tâche. L'hypothèse est que malgré les variabilités des systèmes, certaines requêtes sont difficiles pour tous les systèmes. Le concept de « score de clarté » (clarity score) a été proposé pour essayer de quantifier la difficulté de la requête (Cronen-Townsend et al., 2002). Ce score semble être corrélé positivement avec la mesure de précision moyenne, et peut être considéré comme une mesure de prédiction de la difficulté de la requête. Dans leurs travaux, les auteurs ont utilisé les documents de la collection pour prédire la difficulté de la requête en utilisant un « score de clarté » dépendant aussi bien de la requête que de la collection cible de documents dans laquelle s'effectue la recherche. (Mothe et al., 2005) ont montré dans leur étude que deux critères linguistiques sont corrélés négativement aux mesures de *rappel* et de *précision* (« syntactic links span » pour le rappel, et « polysemy value » pour la précision).

Des études ont également été menées dans la littérature sur la variabilité des systèmes (techniques de recherche utilisées). Un certain nombre de travaux s'intéressent à l'utilisation de cette variabilité pour augmenter les performances de la recherche. (Fox et al., 1994) ont étudié l'effet de la fusion de données sur l'efficacité de la recherche. Les auteurs ont montré que la fusion des résultats de plusieurs SRI augmente la performance par rapport à celle obtenue par les systèmes pris isolément. La meilleure méthode de combinaison (CombSUM) consiste en la sommation de toutes les valeurs de similarité de l'ensemble des documents. (Lee, 1997) utilise CombMNZ (CombMNZ considère en plus de CombSUM le rang de chaque document) et a étudié le degré de chevauchement entre les documents pertinents et non pertinents. Cette étude a montré que la fusion de systèmes ayant un grand degré de chevauchement de documents pertinents donne des résultats plus satisfaisants. (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 INQUERY (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.

Dans ce papier, nous considérons que la combinaison de systèmes permet d'augmenter l'efficacité de la recherche. Cependant, notre combinaison de systèmes sélectionne parmi un ensemble de système le(s) meilleur(s) système(s) à utiliser pour répondre à une requête spécifique. La sélection de système est basée sur des catégories de requêtes déterminées à l'aide de critères linguistiques.

3. Données utilisées

3.1. Collections et mesures

Les collections TREC adhoc que nous utilisons dans notre étude sont composées des résultats issus de différents systèmes ayant participés aux campagnes d'évaluation de TREC. Les collections TREC adhoc comportent un ensemble de documents (plus de 3Gb dans TREC3 par exemple), un ensemble de 50 requêtes et les *« runs »* (documents retrouvés pour chacune des requêtes) de tous les systèmes participants à l'évaluation TREC. Les collections TREC adhoc existent pour différentes années de TREC. Un programme nommé *trec_eval* est utilisé pour évaluer les systèmes en fonction d'un certain nombre de mesures (rappel, précision, MAP, R-précision, et P@). L'intérêt d'utiliser ces mesures dans notre étude est qu'il est possible d'évaluer nos résultats en fonction de plusieurs cas de figure. En effet, les mesures P@ nous permettent de nous intéresser aux systèmes qui retournent les bonnes réponses en tête de liste (exemple : les recherches sur Internet). La MAP quand à elle nous indique les performances globales des systèmes sur l'ensemble de leurs réponses. Nous nous limitons dans ces travaux à ces mesures pour évaluer nos résultats. Les données que nous utilisons pour nos expérimentations proviennent des campagnes TREC3, 5, 6, et 7. En plus d'être une campagne d'évaluation internationale, TREC permet une évaluation homogène des systèmes car ces derniers sont évalués sur la même base, et les requêtes sont assez longues pour permettre un traitement linguistique. Nous présentons dans la section suivante les critères linguistiques que l'on utilise dans notre étude.

3.2. Critères linguistiques

Dans les campagnes TREC, chaque requête est caractérisée par un titre, une description et une partie narrative. Cette structure de requête permet de leur appliquer des techniques de TAL. Nous utilisons les critères linguistiques qui ont été proposés par (Mothe et al., 2005) pour caractériser nos requêtes et qui sont présentés ci-dessous :

- NBWORDS: longueur moyenne des termes de la requête, mesurée en nombre de caractères.

- AVGMORPH: nombre moyen de *morphèmes* par mot. Cette mesure est obtenue en utilisant la base de données morphologique CELEX, qui décrit pour approximativement 40000 lemmes, leur construction morphologique.

- SUFFIX: *nombre d'éléments suffixés*. Une technique de *bootstrap* a été utilisée pour extraire de la base CELEX les suffixes les plus fréquents. Ces suffixes sont ensuite comparés à chaque lemme de la requête.

- PN: *nombre de noms propres.* Cette valeur est obtenue en utilisant l'analyse POS de tagger ainsi qu'une méthode plus robuste basée sur les formes majuscules des mots.

- ACRO, NUM les acronymes et les numéros sont détectés grâce à une technique de comparaison avec des modèles.

- UNKNOWN: les m*ots inconnus* sont ceux qui ont été marqués comme tel avec POS.

- CONJ, PREP, PP: *Conjonctions, prépositions et pronoms* détectés en utilisant exclusivement l'étiquetage POS.

- SYNTDEPTH, SYNDIST: la profondeur syntaxique et l'envergure des liens syntaxiques sont calculés à partir des résultats de l'analyseur syntaxique. La profondeur syntaxique est une mesure de complexité syntaxique en termes de hiérarchie. Il correspond au nombre maximal de composants syntaxiques imbriqués de la requête. Pour déterminer la valeur de l'envergure des liens syntaxiques, on calcule le nombre de mots concernés par chaque lien syntaxique et on fait la moyenne sur le nombre de liens syntaxiques existants.

- SYNSETS: nombre de polysèmes (champ synsets dans WordNet)

Pour extraire tous ces éléments, la requête est d'abord analysée en utilisant des techniques génériques. En fonction des données étiquetées, des programmes simples traitent l'information correspondante. Les outils suivants ont été utilisés :

- Tree Tagger[1] pour la lemmatisation et l'étiquetage de parties de discours: cet outil attribue une simple catégorie morphosyntaxique à tous les mots contenus dans le texte passé en paramètre. Il utilise un lexique général et un modèle de langage;

- Syntex (Fabre and Bourigault, 2001) pour la détection de liens syntaxiques: cet analyseur identifie les relations syntaxiques entre les mots d'une phrase en utilisant des règles grammaticales.

Les ressources suivantes ont aussi été utilisées dans le cadre de cette étude :

1*TreeTagger*, by H. Schmidt; available at
www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

- WordNet 1.6 réseau sémantique pour traiter les ambiguïtés sémantiques. Cette base de données fournit entre autre, le sens possible des mots;

- CELEX[2] base de données pour la morphologie dérivationnelle. Cette ressource donne la décomposition morphologique des mots.

Dans la section suivante, nous montrons comment nous utilisons les critères linguistiques pour catégoriser les requêtes.

4. Méthode utilisée

Notre étude se base sur les critères linguistiques présentés dans la section précédente. Ces critères linguistiques nous servent à classifier les requêtes dans la première phase de notre démarche.

4.1. Représentation des requêtes et méthode de classification

Classification ascendante hiérarchique

Les données sont stockées dans une matrice 200x13 qui comporte les individus (représentés par les requêtes) et les variables (représentées par les critères linguistiques). Les données ont été préalablement centrées et réduites selon les colonnes pour homogénéiser les variables. En effet, l'échelle de données n'est pas la même pour certains critères linguistiques (ex : NBWORDS : 400, et AVGMORPH : 0,23). Plus d'informations sur la manipulation des données statistiques peuvent être obtenues en consultant (Lebart et al., 2006).

La classification ascendante hiérarchique (CAH) est une méthode d'analyse de données que nous utilisons pour classer les requêtes. Cette méthode effectue des regroupements sans connaissance a priori du nombre de groupes à constituer ni de leur structure. La CAH peut être vue comme un algorithme itératif qui considère à son démarrage que tout individu forme une classe à lui seul. A la fin de la CAH, tous les individus sont regroupés au sein d'une seule classe. A chaque itération, les 2 classes les plus proches sont agrégées. Pour fonctionner, la CAH a besoin des éléments suivants :

- Une mesure pour calculer la distance entre chaque paire d'individus (mesure d'éloignement) ;

- Un critère qui permet d'en déduire une distance entre 2 classes.

En général, le choix pour le premier élément est la distance Euclidienne. Cependant, ce choix est naturel lorsque les données à analyser sont réduites. La distance Euclidienne entre 2 requêtes X et Y est définie comme :

$$d(X,Y) = \sqrt{\sum_{i=1}^{13} (Xi - Yi)^2}$$

2*CELEX English database* (1993). Available at www.mpi.nl/world/celex

Pour le deuxième élément les statisticiens recommandent en général le critère de Ward. Il consiste à agréger les 2 classes minimisant la décroissance de la variance interclasses (Seber 1984).

Les résultats de la CAH peuvent être représentés sous forme d'arbre ou dendrogramme, où les nœuds correspondent à l'union de 2 individus ou 2 classes.

4.2. Classification des requêtes

Classification ascendante hiérarchique

Comme mentionné précédemment, la distance Euclidienne et le critère de Ward ont été utilisés dans notre étude.

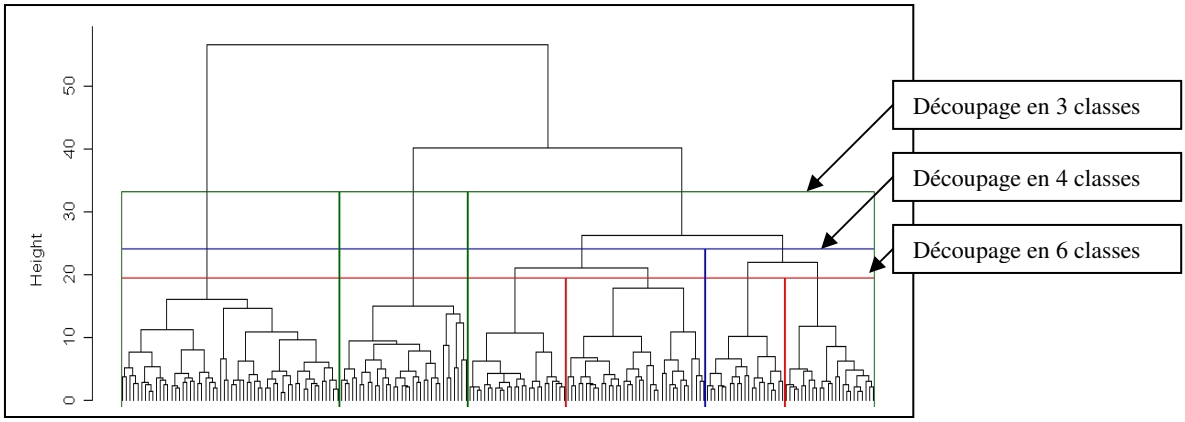

Figure 1. *Dendrogramme représentant les classes de requêtes*

L'interprétation d'un dendrogramme se fait en fonction d'un niveau de coupe de l'arbre fixé a posteriori. Ce niveau est habituellement fixé à une hauteur qui sépare "nettement" 2 noeuds successifs. Ici plusieurs choix pertinents sont possibles : 6, 4 ou 3 classes (voir figure 1). Ces choix diffèrent exclusivement par un découpage plus ou moins poussé du 3ème groupe (à droite du dendrogramme sur la figure 1) en 1, 2 ou 4 classes. Dans une première approche, nous choisissons de favoriser la classification la plus grossière consistant à ne retenir que 3 groupes. La suite de ces travaux visera notamment à tirer profit d'une classification plus fine des requêtes.

Pour utiliser les 3 classes obtenues, nous avons utilisé un algorithme de type K-means en partant des barycentres des classes obtenues par la CAH (Lebart et al., 2006).

Affectation des requêtes aux classes

Nous avons déterminés nos classes à partir de 200 requêtes provenant de différentes campagnes d'évaluation TREC. Nous avons donc effectué un filtrage des requêtes en fonction de la campagne TREC à laquelle elles appartiennent. Le tableau

ci-dessous donne la répartition des requêtes pour chaque année de TREC et pour chaque classe de requêtes. Dans cette table *Nq* représente le nombre de requêtes par classe.

	Nq	TREC3	TREC5	TREC6	TREC7
Classe1	*58 (29%)*	22,41%	20,69%	29,31%	25,86%
Classe2	*34 (17%)*	44,12%	29,41%	14,71%	11,76%
Classe3	*108 (54%)*	19,44%	25,00%	26,85%	28,7%

Tableau 1: *Répartition des requêtes TREC dans les classes*

Le tableau 1 montre que la classe 3 contient 54% des requêtes et que 28,7% de ces requêtes proviennent de TREC7. En fonction de ces 3 classes de requêtes, nous avons évalué les performances des systèmes. Nous avons utilisé le programme *trec_eval* pour calculer les performances des systèmes participants. Ce programme fournit en sortie pour chaque système un ensemble de mesures. Nous avons choisi parmi ces mesures la précision moyenne (MAP), la R-précision, la P@5, P@10, et P@15. Pour chacune de ces mesures, nous avons classé les systèmes en fonction de leur performance, et cela pour chaque session de TREC.

5. Résultats préliminaires

Dans cette section, nous présentons les résultats obtenus avec les données de TREC5.

5.1. Évaluation locale (par classe de requêtes)

Dans le tableau 2, nous analysons les résultats obtenus pour les 3 classes de requêtes en utilisant les données de TREC5. Pour chaque mesure, nous comparons le résultat du meilleur système sur l'ensemble des requêtes de la classe choisie avec les résultats obtenus par le même meilleur système, mais cette fois sur l'ensemble des requêtes de la collection (50 requêtes par session de TREC). Par exemple, en considérant la classe 1 de requêtes, le système *ETHme1* obtient la meilleure MAP (0,3407) sur les requêtes de la classe (on l'appelle alors *Best_Cl1*). Ce même système obtient 0,3070 comme valeur de MAP sur l'ensemble des 50 requêtes (*ETHme1_all*). Il peut cependant arriver que la meilleure performance pour une classe donnée soit obtenue par plusieurs systèmes. Dans ce cas, nous mentionnons les résultats de l'ensemble des systèmes (ex : P@5 pour la classe 2 de requêtes ; pour cette mesure, 4 systèmes obtiennent la meilleure performance). Nous pouvons voir dans le tableau 2 que le système *ETHme1* gagnerait à être utilisé pour répondre aux requêtes de la classe 1, indépendamment de la mesure choisie. Par exemple, pour cette classe, le système *ETHme1* obtient la meilleure MAP (0,3407) alors qu'il obtient une MAP de 0,3070 sur l'ensemble des requêtes. Notons que ce système (*ETHme1*) correspond aussi au système qui obtient les meilleures performances

globales sur toutes les requêtes. Le même type de commentaires peut être fait pour la classe 3. Contrairement aux classes 1 et 3, lorsque nous nous intéressons à la classe 2, les meilleurs systèmes pour la haute précision dans la classe 2 sont moins performants que s'ils avaient été utilisés pour l'ensemble des 50 requêtes. On peut dire que pour ces mesures de haute précision, et pour la classe 2, nous ne détectons pas le meilleur système à utiliser.

TREC5					
Classe1		Classe2		Classe3	
	MAP		**MAP**		**MAP**
Best_Cl1	***0,3407***	*Best_Cl2*	***0,3396***	*Best_Cl3*	***0,3255***
Ethme1_all	0,3070	*Uwgcx0_all*	0,2944	*LNaDesc2_all*	0,2585
	R-prec		**R-prec**		**R-prec**
Best_Cl1	***0,3633***	*Best_Cl2*	***0,3829***	*Best_Cl3*	***0,3528***
Ethme1_all	*0,3305*	*Uwgcx0_all*	0,3444	*LNaDesc2_all*	0,2929
	P@5		**P@5**		**P@5**
Best_Cl1	***0,7481***	*Best_Cl2*	*0,5200*	*Best_Cl3*	***0,6769***
Ethme1_all	0,6160	*Uwgcx0_all*	0,5560	*LNaDesc2_all*	0,4960
		Uwgcx1_all	0,5520		
		Genrl3_all	0,5800		
		Genrl4_all	0,5240		
	P@10		**P@10**		**P@10**
Best_Cl1	***0,6778***	*Best_Cl2*	*0,4400*	*Best_Cl3*	***0,5923***
Ethme1_all	0,5460	*Genrl3_all*	0,4700	*LNaDesc2_all*	0,4780
	P@15		**P@15**		**P@15**
Best_Cl1	***0,6420***	*Best_Cl2*	*0,3867*	*Best_Cl3*	***0,5026***
Ethme1_all	0,5147	*Ethme1_all*	0,5147	*LNaDesc2_all*	0,4280

Tableau 2: *Performances des systèmes de TREC5 sur les classes de requêtes*

Le tableau 3 indique en colonne 2, la moyenne par rapport à la classe de requêtes (moyenne de *Best_Cl1, Best_Cl2, et Best_Cl3 pour la MAP, R-précision, etc.*). Dans la colonne 3, nous indiquons les résultats du meilleur système pour la mesure concernée sur l'ensemble des requêtes. En fait, le meilleur système en termes de MAP, R-précision, P@ pour TREC5 est aussi le meilleur système pour toutes les autres mesures, sauf pour la R-précision. Le tableau 3 montre qu'en moyenne les résultats peuvent être améliorés. Par exemple, le choix du meilleur système pour chacune des classes permet d'obtenir une MAP de 0,3353 alors qu'elle était de 0,3070 pour le meilleur des systèmes participants à TREC5, soit une amélioration de 9,22%.

	Moyenne sur les 3 classes de requêtes (meilleur système pour chaque classe)	Meilleur système en termes de MAP, R-précision, P@5, P@10, et P@15
MAP	**0,3353**	0,3070
R-Prec	**0,3663**	0,3444
P@5	**0,6448**	0,6160
P@10	**0,5700**	0,5460
P@15	0,5104	0,5147

Tableau 3: *Performances moyennes des systèmes de TREC5 par rapport au meilleur système*

5.2. Évaluation globale

Les résultats présentés dans la section précédente comparent les performances par classe de requête avec les performances des systèmes pour toutes les requêtes de TREC. Etant donné que pour une année de TREC les 50 requêtes sont réparties en 3 classes, nous étudions dans cette section les performances que nous obtenons en sélectionnant le meilleur système à utiliser pour l'ensemble des classes (donc sur 50 requêtes) par rapport au meilleur système global (performances initiales sur 50 requêtes).

TREC5									
MAP		**R-prec**		**P@5**		**P@10**		**P@15**	
Best_Cl		***Best_Cl***		***Best_Cl***		***Best_Cl***		***Best_Cl***	
(0,3353)	**%A**	***(0,3663)***	**%A**	***(0,6483)***	**%A**	***(0,57)***	**%A**	***(0,5104)***	**%A**
ETHme1 (0,3070)	*9,22%*	*ETHme1 (0,3305)*	*10,83%*	*ETHme1 (0,6160)*	*5,24%*	*ETHme1 (0,5460)*	*4,4%*	***ETHme1*** *(0,5147)*	*-0,8%*
Uwgcx0 (0,2944)	*13,89%*	*Uwgcx0 (0,3444)*	*6,36%*	*Uwgcx0 (0,5560)*	*16,6%*	*genrl3 (0,47)*	*21,28%*	*LNaDesc2 (0,4280)*	*19,25%*
LNaDesc2 (0,2585)	*29,71%*	*LNaDesc2 (0,2929)*	*25,06%*	*uwgcx1 (0,5520)*	*17,45%*	*LNaDesc2 (0,4780)*	*19,25%*		
				genrl3 (0,58)	*11,78%*				
				genrl4 (0,5240)	*23,72%*				
				LNaDesc2 (0,4960)	*30,71%*				

Tableau 4: *Comparaison globale des performances des systèmes de TREC5 avec Best_Cl*

Dans le tableau 4, nous comparons le système que nous recommandons pour une mesure donnée au système ayant obtenu la meilleure performance pour cette mesure. Dans ce tableau, les nombres entre parenthèses représentent les performances des systèmes. La première ligne nous donne les performances moyennes (sur les 3 classes de requête) du système que nous recommandons (*Best_Cl*). Ce dernier fait une moyenne des performances des systèmes que nous recommandons pour un type de mesure et une classe de requête donnée (*Best_Cl1, Best_Cl2, et Best_Cl3*) (cf. Tableau 2). *%A* représente dans le tableau 4 le pourcentage d'amélioration lorsqu'on compare les performances du système *Best_Cl* avec celles des meilleurs systèmes. Par exemple, en considérant la mesure de R-précision, le système Best_Cl obtient une valeur de 0,3663 ; cette valeur est obtenue en faisant la moyenne des valeurs obtenues par les meilleurs systèmes de chacune des 3 classes (*ETHme1*, *uwgcx0*, et

LNaDesc2). Dans cet exemple, %A représente l'amélioration que l'on obtient en comparant les résultats de *Best_Cl* avec ceux de *ETHme1*, *uwgcx0* et *LNaDesc2*. Nous remarquons que dans ce cas, le meilleur système pour la R-précision (*uwgcx0*) obtient 0,3444 comme valeur de performance, ce qui correspond à une amélioration de 10,83% lorsqu'on utilise *Best_Cl* à la place de *uwgcx0*. Les mêmes analyses peuvent être faites pour les autres mesures et les autres années de TREC. Sur les 61 systèmes ayant participé à la campagne TREC5, nous sommes alors capables de détecter à 98,36% (sur un ensemble de systèmes) le meilleur système à utiliser pour une mesure donnée.

6. Évaluation

L'analyse présentée dans la section 5 correspond à une première étape de nos travaux. Dans cette section, nous validons notre hypothèse en utilisant une technique d'apprentissage. Nous avons donc déterminé un ensemble de requêtes d'entraînement qui vont nous permettre de trouver les classes de requêtes, et finalement des ensembles de test sont utilisés pour vérifier nos hypothèses de départ.

6.1. Principe

Dans notre évaluation, notre ensemble d'entraînement est composé de 180 requêtes ; les requêtes de test quant à elles sont composées de 20 requêtes tirées au hasard. Ces 2 ensembles sont construits aléatoirement en sélectionnant des requêtes sur les 200 requêtes dont nous disposons. Cette procédure d'entraînement/test a été renouvelée 10 fois. De plus, une nouvelle classification des requêtes d'entraînement est effectuée à chaque nouvelle itération.

Nous présentons dans les paragraphes suivants les résultats que nous avons obtenus après nos expériences.

6.2. Résultats

Évaluation locale

La figure 2 donne la répartition des requêtes dans les classes à chacune des itérations.

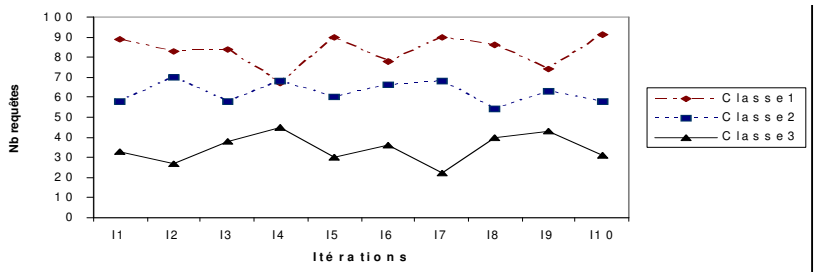

Figure 2: *Répartition des requêtes d'entraînement dans les classes par itération*

Les variations entre le nombre de requêtes par classe et par itération sont très homogènes. En moyenne, la classe 1 contient le plus grand nombre de requêtes et la classe 3 le plus faible nombre de requêtes. Cela témoigne de la stabilité de notre classification.

Comme nous l'avons mentionné dans les premières parties de ce papier, nous avons filtré les requêtes et les systèmes en fonction de la campagne TREC à laquelle ils se réfèrent. Le tableau suivant nous donne la répartition moyenne (sur l'ensemble des 10 itérations) des requêtes de test dans les différentes classes en fonction des années de TREC.

	TREC3	TREC5	TREC6	TREC7	Moyenne
Classe1	8,00%	14,00%	12,50%	11,50%	*11,50%*
Classe2	8,50%	5,50%	10,50%	9,00%	*8,38%*
Classe3	8,00%	5,50%	2,00%	4,00%	*4,88%*

Tableau 6: *Répartition des requêtes de test dans les classes*

En moyenne, la classe 1 de requêtes comprend le plus grand nombre de requêtes. En calculant la proportion des requêtes de test par rapport aux requêtes d'entraînement, nous pouvons dire que la répartition dans le tableau 6 est correcte.

Évaluation globale

Nous mesurons ici les performances moyennes de notre classification sur les 3 classes de requêtes. Le premier résultat que nous présentons ici montre le nombre de systèmes qui ont des résultats supérieurs (inférieurs, ou égaux) à ceux obtenus par le système que nous avons sélectionné. Dans le tableau suivant, *Best_Cl* représente comme dans la section précédente la moyenne des performances sur l'ensemble des meilleurs systèmes que nous proposons d'utiliser pour chaque classe de requêtes.

	TREC3	TREC5	TREC6	TREC7	Moyenne
Nb > Best_Cl	19,21%	20,58%	10,26%	14,85%	**16,22%**
Nb < Best_Cl	67,55%	61,75%	80,63%	74,42%	**71,09%**
Nb = Best_Cl	13,24%	17,67%	9,11%	10,73%	**12,69%**

Tableau 7: *Comparaison des performances des systèmes participants à TREC par rapport aux performances du système Best_Cl*

Nous constatons que pour chacune des années de TREC, une très grande majorité de systèmes participants obtiennent des performances inférieures à celle que nous obtenons si le système que nous proposons est utilisé.

Dans la figure 3 ci dessous, nous faisons une comparaison globale des meilleurs systèmes avec notre système. Nous appelons ici meilleurs systèmes, les systèmes

ayant obtenus les meilleures performances pour une mesure donnée et une classe de requêtes. Il peut toutefois arriver que ces meilleurs systèmes soient également ceux qui ont obtenus les meilleures performances pour l'ensemble des requêtes et pour la même mesure. Dans la figure, nous comparons les améliorations (en pourcentage) que l'on obtient si on utilise le système *Best_Cl* à la place des meilleurs systèmes. En analysant cette figure, *Best_Cl* permet d'améliorer les performances par rapport aux meilleurs systèmes de TREC sauf pour la mesure MAP (Best_Cl : 0,4181 et inq102: 0,4226, amélioration : -1,08%) et R-précision (Best_Cl : 0,4491 et inq102: 0,4524, amélioration : -0,72%) de TREC3, ainsi que pour la MAP (Best_Cl : 0,4611 et uwmt6a0: 0,4631, amélioration : -0,43%) et la R-précision (Best_Cl : 0,4872 et uwmt6a0: 0,4893, amélioration : -0,48%) de TREC6.

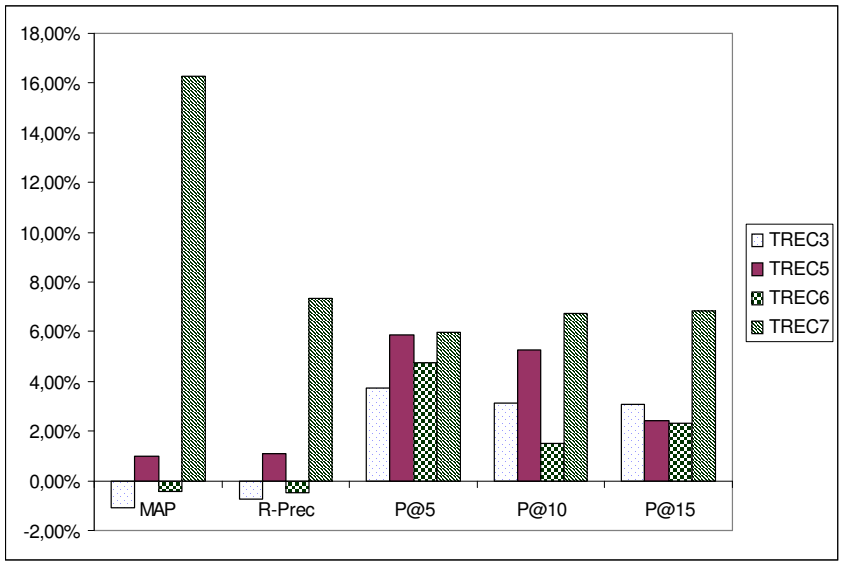

Figure 3: *Comparaison globale des différents pourcentages d'amélioration obtenus en utilisant le système Best_Cl à la place des meilleurs systèmes*

Dans la figure 3, nous remarquons des améliorations de performance pour chaque année de TREC pour toutes les mesures de haute précision. Par exemple pour la P@10, les améliorations varient de 1,48% (TREC 6) à 6,73% (TREC 7). La plus grande amélioration est obtenue pour la MAP et est de 16,27% (TREC 7). Inversement, la plus petite amélioration est de 0,97% pour la MAP et pour TREC 5. Le pourcentage d'amélioration moyen sur l'ensemble des campagnes TREC varie de 1,81% (R-précision) à 5,08% (P@5)

7. Conclusion

Dans cet article, nous avons fait l'hypothèse que les requêtes pouvaient être classifiées en fonction de critères linguistiques. Nous avons aussi fait l'hypothèse qu'il est possible d'associer un système à utiliser pour chaque classe de requêtes ainsi déterminée. Pour évaluer nos hypothèses, nous avons procédé en deux étapes : nous avons dans un premier temps considéré les résultats que nous aurions obtenus si on était capable de choisir le système à utiliser pour une catégorie de requêtes. La sélection des systèmes est basée sur les mesures MAP, Précision, P@5, P@10, et P@15. Les catégories de requêtes sont obtenues en utilisant les caractéristiques linguistiques de ces dernières. En considérant que nous sommes capable de détecter le meilleur système à utiliser pour une classe de requête, nous montrons que nous améliorons les résultats de la recherche. Dans une deuxième étape, nous avons évalués la première hypothèse en utilisant de l'apprentissage (phase d'entraînement pour déterminer les classes de requêtes, et phase de test pour voir si l'on classifie correctement les requêtes de test). Nous avons trouvés que dans ce cas aussi, nous améliorons les résultats que les meilleurs systèmes ont obtenus. De plus, nous avons choisi de réaliser nos expérimentations à partir des données de TREC. Il s'agit d'un cadre standard de validation de nos résultats.

Des travaux futurs nous permettrons de tester des caractéristiques autres que les critères linguistiques. Une piste serait de combiner les caractéristiques des requêtes avec celles des documents pertinents retrouvés. Des études dans le domaine de la fusion de données ont montré qu'une telle combinaison permet d'améliorer les résultats. Nous essayerons de voir de quelle manière ces critères peuvent être inclus dans un processus d'entraînement et quel est leur impact sur la prédiction du meilleur système à utiliser. Nous verrons ensuite comment les techniques de fusion peuvent être utilisées pour améliorer les performances lors de la prédiction du système à utiliser. Une étude plus approfondie sur les critères linguistiques est en cours et cela nous permettra d'expliquer le comportement des systèmes par rapports à ces critères.

12. Bibliographie

Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O., Goharian N. "Fusion of Effective retrieval strategies in the same information retrieval system." *J. Am Soc. Inf. Sci. Technol*., 55(10): 859-868, 2004

Belkin J., Koenemann J., Quatrain R., Cool C., Belkin N., J., "The Interactive Searching Behavior of Expert Online Searches using INQUERY". *New Tools and Old Habits*, 1994

Buckley, C., "Why current IR engines fail"; In *Proceedings of the 27th annual International ACM SIGIR conference on Research and development in information retrieval*. ACM Press, 584-585, 2004

Buckley, C., Harman, D., "Reliable information access". Final report, *27th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Sheffield: ACM Press, 528 – 529, 2004

Buckley C., Waltz J., "SMART in TREC 8". In The *Eighth Text Retrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000

Buscaldi, D., Rosso, P., Sanchis Arnal, E., "A WordNet-based Query Expansion method for Geographical Information Retrieval", 2005, *CLEF, http://www.clef-campaign.org/2005/working_notes/workingnotes2005/buscaldi05.pdf.*

Chang, Y., Kim, M., Ounis, I., "Construction of query concepts in a document space based on data mining techniques". *Proceedings of the 6th International Conference On Flexible Query Answering Systems (FQAS,2004).* Lecture notes in Artificial Intelligence, Lyon, France, June 24-26, 137-149, 2004

Cronen-Townsend, S., Zhou, Y., Croft, W.B., "Predicting query performance". *Proceedings of the 25th annual international ACM-SIGIR conference on research and development in information retrieval, Tampere*, 299-306, 2002

Fabre, C., Bourigeault, D., "Linguistic clues for corpus-based acquisition of lexical dependencies", in *Proceeding of Corpus Linguistics*, Lancaster, 2001

Fox, E.A., Shaw, J.A., "Combination of multiple searches". *Proceedings of the 2nd Text Retrieval Conference (TREC-2)*, NIST special publication, 243-252, 1994

Lebart, L., Morineau, A., Piron, M., « Statistique exploratoire multidimensionnelle : Visualisations et inférences en fouille de données». 4e *Edition, Dunod*, Paris, 6 juillet 2006

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

Mandl, T, Womser-Hacker, C., "Linguistic and statistical analysis for the CLEF topics". *Peters C, Braschler M, Gonzalo J and Kluck M, Eds. Advances in Cross-Language Information Retrieval: Results of the CLEF 2002 Evaluation Campaign, LNCS 2785*. Spinger Verlag, 505-511, 2003

Mardia, K.V., Kent, J.T., Bibby, J.M., "Multivariate Analysis". *Academic Press.7th Printing*. 1989

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

Seber, G., "Multivariate Observations". New York: Willey, 1984