Analyse des paramètres de recherche d'information: Etude de l'influence des paramètres sur les résultats

Josiane Mothe [*,**], Marion Moulinou[*]

[*] Institut de Recherche en Informatique de Toulouse, IRIT
UMR 5505 CNRS, Université de Toulouse,
118 Route de Narbonne 31062 Toulouse CEDEX 9
[**]ESPE, Ecole interne Université Toulouse II Jean Jaurès,
56 av. de l'URSS 31079, Toulouse
prenom.nom@irit.fr, http://www.irit.fr/~ Prenom.Nom

Résumé. Cet article présente une analyse détaillée d'un ensemble de 2 millions de résultats de recherche d'information obtenus par différents paramétrages de systèmes de recherche d'information. Plus spécifiquement, nous avons utilisé la plateforme Terrier et l'interface RunGeneration pour créer différentes exécutions (run en anglais) en modifiant les modèles d'indexation et de recherche. Nous avons ensuite évalué chacun des résultats obtenus selon différentes mesures de performance de recherche d'information. Une analyse systématique a été menée sur ces données afin de déterminer d'une part quels étaient les paramètres qui ont le plus d'influence, d'autre part quels étaient les valeurs de ces paramètres les plus susceptibles de conduire à de bonnes performances du système.

1 Introduction

Un système de recherche d'information (SRI) est un module logiciel qui sélectionne, à partir d'une collection de documents, une liste de documents potentiellement pertinents en réponse à une requête utilisateur. Le processus suivi par un SRI est composé de 3 étapes.

Indexation. Cette étape permet de passer d'un document textuel à un document qui peut être utilisé dans la RI. Elle se base sur l'extraction des mots les plus importants des textes. Lors de cette étape, les mots vides tels que *le*, *la*, *les* sont généralement éliminés ; les termes sont ensuite racinisés, c'est-à-dire que des règles de transformation sur les termes sont appliquées afin d'obtenir un radical, limitant les variantes des termes à une forme unique ; enfin une pondération reflète l'importance des différents radicaux obtenus. Dans un cadre non dynamique, l'indexation est réalisée sur l'ensemble des documents, avant toute recherche.

Calcul des scores de pertinence des documents. Lorsqu'une requête est soumise au système, des scores de pertinence sont attribués aux termes qui la composent, en tenant compte de leur présence dans les documents. Ces scores sont ensuite combinés pour calculer le score global de chacun des documents de la collection. Il existe de nombreux modèles de pondération. La plupart sont basés sur les facteurs TF et IDF. L'expression TF (Term Frequency) correspond à la fréquence du terme dans le document, tandis que l'IDF (Inverse Document

Frequency) désigne la fréquence inverse du terme dans le document, inversement proportionnel au nombre de documents qui contiennent le terme.

Reformulation de la requête. Cette étape permet de créer une requête plus adéquate à la RI que celle initialement formulée par l'utilisateur. Le principe de la reformulation automatique est de modifier la requête de l'utilisateur en ajoutant des termes significatifs ou en ré-estimant leurs poids. Dans sa version automatique, il s'agit de considérer les premiers documents restitués comme pertinents et d'ajouter des termes issus de ces documents ; de nouveaux poids sont également estimés et les scores des documents recalculés pour fournir la réponse finale du système. Ce paramètre n'est pas étudié dans le travail présenté dans cet article.

Chacune de ces étapes fait intervenir différents paramètres : par exemple lors de l'indexation, il est possible de choisir entre différents outils de racinisation, lors du calcul des scores de pertinence des documents, différents modèles de pondération peuvent être choisis. Les différents paramètres étudiés dans cet article sont présentés en figure 1.

L'efficacité d'un SRI est évaluée en calculant des mesures de performance comme le rappel, la précision et d'autres mesures associées. Depuis ses débuts, le domaine de la RI est très actif pour fournir de nouvelles propositions correspondant à une évolution de ces trois étapes. Lorsqu'un nouveau modèle de RI est proposé, ses paramètres sont étudiés, mais sans considérer les effets croisés. Par exemple dans Ponte et Croft (1998), ce sont les paramètres du modèle lui-même qui sont étudiés sans regarder l'influence du choix de l'algorithme de racinisation. Les modèles d'apprentissage d'ordonnancement (*Learning to rank* en anglais) considèrent de nombreux paramètres tels que les fréquences TF et IDF, la taille des documents et des caractéristiques comme les scores BM25, LMIR, PageRank des documents (Qin et al., 2010). Ces approches ont pour objectif d'optimiser l'ordonnancement des documents mais ne cherchent pas à connaître l'impact des paramètres. Quelques travaux visent à sélectionner les variables importantes, donc à étudier leur influence (Laporte et al., 2014; Naini et Altingovde, 2014).

Ainsi, généralement, les paramètres sont étudiés de façon indépendante, sans considérer les effets croisés des paramètres. Compte tenu du nombre de paramètres, l'étude des effets croisés est difficile et implique au préalable de collecter des données suffisantes pour le faire. Cet article s'attaque à ce problème. Ainsi, dans cette étude, nous nous appuyons sur un ensemble de données massif (2 millions de configurations) dans lequel les différents paramètres varient.

La littérature du domaine ne s'est que peu intéressée à une analyse de cette nature. Presque tous les articles et les thèses du domaine de la RI rapportent des études montrant la variation de mesures de performance en fonction d'un ou plusieurs paramètres, mais il ne s'agit pas d'une analyse en parallèle de paramètres variés. Quelques travaux se sont cependant intéressés à utiliser les méthodes d'analyse pour étudier les résultats de moteurs de recherche sur un ensemble de requêtes. Banks et al. (1999) ont réalisé l'analyse de résultats de TREC. Ils ont considéré une matrice dans laquelle les lignes et les colonnes correspondent aux systèmes et aux besoins d'information (topics) et les cellules aux valeurs de la mesure de performance AP (Average Precision). Cette matrice est alors utilisée pour analyser les regroupements qui peuvent en être extraits en s'appuyant sur différentes méthodes d'analyse de données. Les auteurs concluent que les analyses ne permettent pas d'extraire des conclusions. Chrisment et al. (2005) utilisent le même type de matrice pour visualiser les corrélations entre systèmes et besoins d'information ; les auteurs s'appuient sur des classifications hiérarchiques et des analyses factorielles. Ils montrent que les variations entre différentes variantes d'un même système (modification de valeurs de paramètres) ont moins d'influence que des variations de moteurs eux-mêmes (diffé-

rents participants). Mizzaro et Robertson (2007) se sont intéressés à analyser le même type de données mais pour définir un sous-ensemble minimum de besoins d'information permettant de distinguer les systèmes performants des systèmes non performants. Les auteurs concluent que ce sont les requêtes faciles qui permettent le mieux de faire cette distinction. Dinçer (2007) utilise une analyse en composante principale dans l'objectif de comparer les performances de différentes stratégies de recherche. Par ailleurs, Bigot et al. (2011) ont montré qu'il était possible, en utilisant ce type de données et ces méthodes d'analyse, de regrouper les besoins d'information par difficulté et d'appliquer certains moteurs en fonction du type de besoin rencontré. Compaoré et al. (2011) présentent une étude qui a les mêmes objectifs que ceux de ce papier. Cependant, le nombre d'éléments analysés et donc les combinaisons de paramètres est bien moindre. Dans leurs études, les auteurs montrent que les paramètres qui ont le plus d'influence sont différents en fonction que l'on considère les besoins d'information faciles ou difficiles. Bigot et al. (2014) utilisent les résultats d'une analyse pour sélectionner la configuration de système la plus adaptée en fonction de la difficulté du besoin d'information.

L'objectif de l'analyse que nous présentons dans le présent papier est d'étudier à grande échelle les caractéristiques des SRI dans le but de déterminer les meilleures combinaisons de paramètres selon certaines mesures de performance. Ce travail a été mené dans le cadre du projet ANR CAAS (Contextual Analysis and Adaptive Search) ANR-10-CORD-001-01.

La suite de cet article est structurée comme suit. La section 2 présente la méthode utilisée pour obtenir les données ainsi que les données elles-mêmes. La section 3 présente l'analyse de la dépendance entre les paramètres et leur influence mutuelle. La section 4 s'attache à déterminer quelles sont les valeurs de paramètres les plus susceptibles de conduire à de bonnes performances du moteur de recherche correspondant. La section 5 conclut cet article.

2 Variantes de moteurs de recherche et paramétrage

Les données ont été générées via l'interface RunGeneration présentée dans Louédec et Mothe (2013) et qui est une sur-couche à la plateforme Terrier.

2.1 Terrier et RunGeneration

La plateforme de RI TERRIER (Ounis et al., 2006) possède de nombreuses possibilités de paramétrage, tant au niveau de l'indexation (indexation par blocs de différentes tailles, choix de la racinisation, etc.) que de la recherche (différents modèles de pondération pour la mise en correspondance entre la requête et les documents, différentes normalisations des poids) et de la reformulation de requêtes. Une fois paramétré, Terrier permet, pour un besoin d'information ou un ensemble de besoins, de retrouver les documents susceptibles de répondre à ce besoin.

L'interface RunGeneration a comme objectif de faciliter le paramétrage d'une chaine de traitement sous Terrier. Une fois les paramètres sélectionnés au travers de l'interface, celle-ci crée le fichier "terrier.properties" indispensable à Terrier et contenant l'ensemble des paramètres. Le second objectif de l'interface RunGeneration est de permettre de lancer plusieurs combinaisons de paramètres simultanément, ce qu'il n'est pas possible de faire en utilisant la plateforme Terrier. Ainsi plusieurs indexations et recherches sont effectuées sur les mêmes données via une seule intervention de l'utilisateur. Celui-ci peut par exemple demander en une

Variable Terrier	Description	Modalités
BlocsSize	Taille des blocs	11 &1, 2, 4, 6, 8, 10, 14, 22, 44, 100, 1000
IgnoreEmpty Documents	Prise en compte des documents vides	Vrai / Faux
Stemmer	Outil de racinisation utilisé	Crop, ESS, FSS, PS, TRPS, TRWPS, WPS
Retrieving Model	Modèle de pondération	BB2, DLH, InexpB2, PL2, BM25, DLH13, InexpC2, TFIDF, DFI0, DPH, InL2, XSqrAM, DFRBM25, HiemstraLM, JsKLs, DFRee, IFB2, LemurTFIDF, DirichletLM, InB2, LGD
TrecQueryTags process	Champs pris en compte dans la requête	T, TD, TDN, D, N, TN, DN
IgnoreLowIdf Terms	Prise en compte des termes qui ont un IDF bas	Vrai / Faux
Topic	N° du besoin d'information	401, 402, ... 449, 450

FIG. 1 – *Paramètres de la génération des données.*

action plusieurs indexations de documents avec des paramètres différents. L'interface a été développée pour fonctionner sur les versions 3.0 et 3.5 de Terrier [1] (Louédec et Mothe, 2013).

Ainsi, lors de la génération des données utiles à notre analyse, le principe est le suivant : pour chaque combinaison de paramètres, une liste de documents retrouvés en réponse au besoin d'information est constituée. Cette réponse du système est alors évaluée sur la base de mesures de RI. Ainsi, pour une combinaison de paramètres, nous connaissons la valeur de chacune des caractéristiques correspondant aux paramètres du moteur et aux mesures de performance.

2.2 Paramètres utilisés lors de la génération de données

La figure 1 indique les variables utilisées lors de la génération des données ainsi que les modalités de ces variables paramètres. Le nombre de combinaisons obtenues est de 2 263 800. Les variables correspondant à des paramètres du système sont qualitatives.

2.3 Collection d'évaluation utilisée

Compte tenu du nombre de combinaisons et des temps nécessaires pour générer les données, dans cette étude, nous n'avons utilisé qu'une seule collection de documents : la collection TREC-8 de la tâche *adhoc* de TREC [2]. Elle comprend environ 530 000 documents soit 2 Go ; chaque document est composé en moyenne de 532 mots. La collection comprend également 50 besoins d'information et les jugements de pertinence des documents associés à ces besoins.

1. http ://terrier.org/docs/v3.5/
2. trec.nist.gov

Sur ce jeu de données, nous n'avons pas pris en compte les paramètres de reformulation de requêtes afin de ne pas rendre le nombre de combinaisons possibles trop grand pour être généré. Plutôt nous avons fait l'hypothèse qu'une première analyse permettrait de faire ressortir les paramètres principaux qui eux pourront être combinés avec les paramètres de reformulation. En effet, les principes de reformulation (implantés dans Terrier) s'appuient tous sur l'utilisation des premiers documents retrouvés suite à une première recherche. Aussi, optimiser la précision dans ces premiers documents, optimise à priori la reformulation de requêtes.

2.4 Caractéristiques d'évaluation associées

Afin d'évaluer les résultats obtenus nous avons utilisé trec_eval [3] qui calcule plus de 100 mesures de performance telles que $bpref$, AP et $P@5$. Cependant, nous avons restreint les variables utilisées à celles qui sont les moins corrélées. Ainsi, nous avons conservé les 6 mesures de performance préconisées dans Baccini et al. (2012). Elles sont résumées dans la table 1. Elles sont toutes quantitatives à valeur continue et leurs valeurs sont comprises entre 0 et 1.

Variable 1	Description
P30	Précision après que 30 documents aient été retrouvés
P100	Précision après que 100 documents aient été retrouvés
Rprec	Précision après que le nombre total de documents pertinents a été retrouvé
bpref	Préférence binaire, nbre de fois que les documents jugés non pertinents sont retournés avant un document pertinent
iprec@recall 0.30	Précision interpolée avec 30 documents retrouvés
AP	Moyenne des précisions obtenues à chaque fois qu'un document pertinent est retrouvé

TAB. 1 – *Mesures de performance retenues - les moins corrélées selon Baccini et al. (2012).*

3 Dépendance entre les variables : influence des paramètres

Cette première analyse a pour objectif d'étudier d'une part la distribution des valeurs des variables correspondant à des mesures de performance, d'autre part d'étudier les corrélations entre variables de même type (corrélation entre paramètres du système d'un côté et entre mesures de performance de l'autre côté) et entre variables de types différents.

3.1 Distribution des valeurs des variables de mesure de performance

La figure 2 montre la distribution des valeurs des variables correspondant aux mesures d'évaluation de la performance. On note que le minimum 0 est atteint pour chacune des variables ; en revanche le maximum 1 n'est atteint que pour la $P@30$ et la $iprec@recall0.30$. La moyenne est d'environ $0,2$, ce qui est faible. De plus, le troisième quantile (colonne 3èmeQ du

3. trec.nist.gov/trec_eval

	Min	Max	1er Q	Médiane	3ème Q	Moy	Var	Ecart-type
P_30	0.000	1.000	0.067	0.200	0.367	0.250	0.048	0.220
P_100	0.000	0.840	0.050	0.110	0.200	0.154	0.023	0.151
Rprec	0.000	0.875	0.064	0.169	0.312	0.216	0.035	0.188
Bpref	0.000	0.929	0.053	0.142	0.260	0.194	0.035	0.187
iprec_at_recall_0.30	0.000	1.000	0.000	0.100	0.343	0.230	0.084	0.289
Map	0.000	0.948	0.019	0.096	0.239	0.171	0.039	0.199

FIG. 2 – *Distribution des valeurs des mesures d'évaluation.*

tableau) a une valeur moyenne de $0,3$; cela signifie que seulement $1/4$ des observations ont des mesures de performance moyennes ou élevées. Les $3/4$ restants ont donc des mesures de performance faibles ou très faibles.

Nous avons ensuite étudié la dispersion de chacune de ces variables à l'aide de deux graphiques : histogrammes et boîtes à moustaches. Nous constatons une très forte asymétrie de la distribution pour chaque variable quantitative sur les histogrammes représentés en figure 3 . Les variables ne semblent donc pas suivre une loi normale. La variable P@30 ne prend qu'une trentaine de valeurs différentes et l'histogramme est divisé en 20 classes. Certaines classes utilisent une seule valeur de P@30 tandis que d'autres vont en utiliser deux, d'où l'observation de barres successivement hautes et basses.

La figure 4 correspondant à des boîtes à moustaches confirme que les mesures de performance ne suivent pas une loi normale centrée réduite. De plus, on peut noter que chacune des variables possède un certain nombre de valeurs atypiques.

L'ensemble de ces figures montre que les valeurs sont faibles ; la valeur 0 est la plus fréquente, montrant ainsi que beaucoup de configurations échouent dans la RI. Par ailleurs, comme les données ne suivent pas une loi normale, il faudra faire attention aux analyses réalisées par la suite pour ne choisir que celles qui s'appliquent à des variables qui ne suivent pas une loi normale. Cependant, les tests et méthodes que nous utilisons dans la suite restent valident car nous travaillons sur un grand jeu de données.

3.2 Corrélations entre variables de même type

Nous avons étudié la corrélation d'une part entre les variables correspondant aux paramètres du SRI et d'autre part entre variables d'évaluation des moteurs.

Nous avons analysé le lien entre les paramètres du moteur de RI, pris deux à deux afin de savoir si certaines de ces variables paramètres ont des rôles similaires ou sont liées entre elles. Pour cela, nous avons réalisé le test du chi^2. Soit l'hypothèse $H0$ *«les deux variables sont indépendantes»* contre $H1$ *«les deux variables ont un lien»*. Nous rejetons l'hypothèse $H0$ si la p-value est inférieure à $0,05$. Après avoir réalisé le test pour chacune des variables, nous concluons que toutes les variables qualitatives sont indépendantes deux à deux. Elles ont donc chacune leur rôle spécifique.

En revanche, en ce qui concerne les variables quantitatives correspondant aux mesures de performance, nous avons constaté qu'il existe une corrélation. Cette corrélation est plus ou moins importante en fonction des mesures que l'on compare. La figure 5 montre cette corrélation. On considère généralement qu'entre $-0,5$ et $0,5$, la corrélation existe mais qu'elle

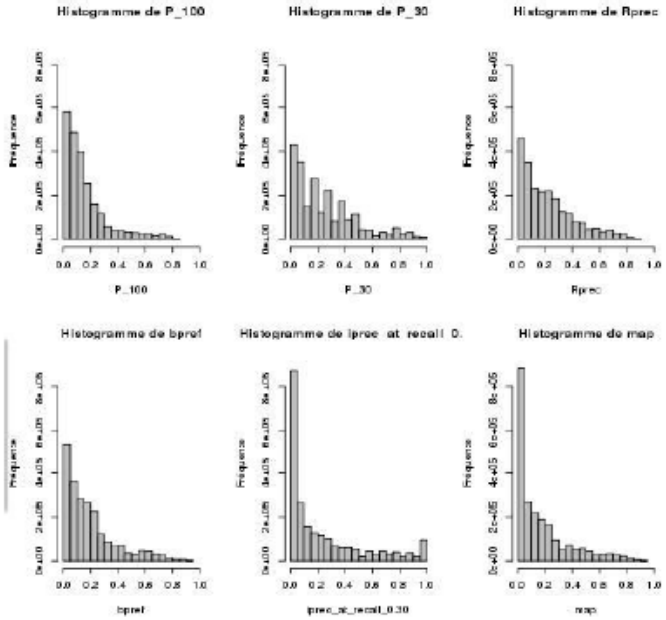

FIG. 3 – *Distribution des variables d'évaluation de la performance − Histogramme.*

est faible. Toutes les variables sont donc plus ou moins liées entre elles. Dans notre cas, toutes les corrélations sont supérieures à $0,5$. Nous remarquons que les plus fortes corrélations sont entre $bpref$ et $Rprec$ et entre AP et $bpref$ avec des coefficients de corrélation supérieurs à $0,9$. Ces mesures sont toutes orientées précision. La plus basse corrélation est entre $P@100$ et AP avec $0,517$. De plus, ici, toutes les corrélations sont positives. Les mesures de performance varient donc toutes dans le même sens.

Ainsi les mesures de performance, déjà réduites à 6 pour plus de 100 au départ sont assez redondantes dans leur capacité à mesurer les performances des systèmes puisque corrélées, même si elles ne mesurent pas le même phénomène. Les paramètres du système en revanche n'étant pas corrélés, cela a un sens de chercher à optimiser chacun de ces paramètres.

3.3 Corrélations entre variables paramètres et variables d'évaluation

Afin d'étudier l'effet des paramètres sur les mesures de performance, nous avons effectué une analyse de la variance (ANOVA).

Soit l'hypothèse $H0$ *«Le paramètre n'a pas d'effet sur la mesure de performance»* contre $H1$ *«le paramètre a un effet sur la mesure de performance»*. Nous rejetons $H0$ si la p-value est $< 0,05$.

Nous avons étudié cette corrélation sur chacune des mesures de performance. La table 2 présente les résultats. La taille des blocs utilisée lors de l'indexation et le fait d'éliminer les documents vides ou les termes ayant un faible IDF n'a donc pas d'effet significatif. Les autres

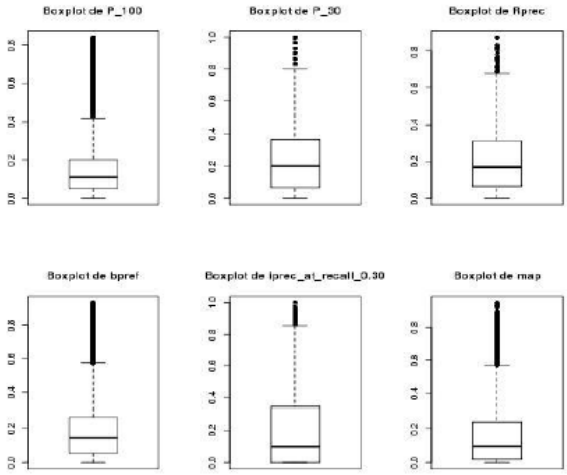

FIG. 4 – *Distribution des variables d'évaluation de la performance - Boîte à moustaches.*

	P_100	P_30	Rprec	bpref	iprec_at_recall_0.30	map
P_100	1					
P_30	0.902	1				
Rprec	0.594	0.768	1			
Bpref	0.609	0.782	0.981	1		
iprec_at_recall_0 .30	0.526	0.700	0.951	0.953	1	
Map	0.517	0.716	0.968	0.977	0.969	1

FIG. 5 – *Corrélation entre les mesures de performance.*

paramètres ont un effet significatif. Ces trois paramètres sans effet pourront donc être fixés dans la génération éventuelle d'autres données. Pour être réellement exhaustif, il faudrait vérifier que ces paramètres n'ont pas d'influence lorsque l'on change de collection, mais compte tenu de leur nature, la probabilité que cela soit le cas est forte.

Effet significatif	Effet non significatif
TrecQueryTagsProcess, Topic RetrievingModel, Stemmer	BlocSize, IgnoreEmptyDocuments IgnoreLowIdfTerms

TAB. 2 – *Effets significatifs et non significatifs pour l'ensemble des mesures de performance.*

3.4 Variables ayant le plus d'influence

Nous avons utilisé la méthode *Stepwise* qui est une régression linéaire multiple Bendel et Afifi (1977) pour étudier l'influence relative des différentes variables paramètres. Cette mé-

thode ajoute les variables les plus significatives du modèle et retire les moins significatives pas à pas. Dans le cadre de notre étude, elle a pour objectif de sélectionner les paramètres qui ont le plus d'influence sur les mesures de performance. C'est une combinaison de la méthode Forward et de la méthode Backward. La première méthode part du modèle vide et ajoute les variables les plus significatives du modèle progressivement, tandis que la seconde part du modèle complet et élimine progressivement les variables les moins significatives du modèle.

Après avoir réalisé cette analyse sur les différentes mesures de performance, nous observons que les trois variables supprimées sont $BlocsSize$, $IgnoreEmptyDocuments$ et $IgnoreLowIdfTerms$, comme dans le cas de l'étude des corrélations précédente.

Au final cette méthode sélectionne donc les paramètres $Topic$, $TrecQueryTagsProcess$, $RetrievingModel$ et $Stemmer$. La variable $Topic$ est la plus significative du modèle, suivi de $TrecQueryTagsProcess$, puis de $RetrievingModel$ et de $Stemmer$. Le besoin d'information considéré est le paramètre dont dépend le plus les résultats. Cela est un résultat important concernant la variabilité des résultats. On aurait pu penser que le modèle de recherche utilisé pouvait être le plus important des paramètres. La formulation du besoin d'information est également importante puisque le paramètre $TrecQueryTagsProcess$ correspond aux parties du besoin d'information pris en compte lors du traitement. Lorsque seul le titre est considéré, il correspond à quelques mots, taille typique des requêtes sur le web. Lorsque les autres champs sont également considérés, il peut s'agir de requêtes plus longues, donnant un contexte précis du besoin d'information. Cette première analyse avait pour objet de déterminer les paramètres du moteur les plus importants ou qui influencent le plus la performance d'une recherche. Dans la section suivante, nous déterminons quelles sont les valeurs de paramètres les plus susceptibles de conduire à de bons résultats.

4 Paramètres des SRI pour des classes de précision

L'objectif de cette analyse est d'étudier les valeurs des paramètres du moteur de recherche qui peuvent être associées à des valeurs de précision. Nous nous sommes appuyés dans cette étude sur la AP qui est la mesure consensuelle lorsqu'il s'agit de comparer globalement plusieurs systèmes et qui est utilisée en particulier dans la campagne d'évaluation TREC (trec.nist.gov) (Voorhees, 2007).

4.1 Classification mixte

La classification mixte est une méthode de classification qui a pour objectif d'obtenir, à partir des facteurs issus d'une analyse des correspondances multiples (ACM), des classes d'individus les plus cohérentes possibles en constituant les groupes les plus homogènes.

Dans notre cas d'étude, nous utilisons cette méthode afin d'associer à une classe de valeur de AP les paramètres de moteurs. L'idée sous-jacente est de favoriser les combinaisons de paramètres qui sont plutôt associées à des valeurs fortes de AP et d'éviter les combinaisons de paramètres plutôt associées à des valeurs faibles de AP.

Cette étude nécessite d'appliquer une ACM, méthode qui s'applique sur des variables qualitatives. Afin de transformer le paramètre d'évaluation qualitatif considéré (l'AP) en valeurs qualitatives, nous avons créé des classes de valeurs de AP. Les classes ont été définies de sorte d'avoir des effectifs comparables. Nous noterons dans la suite la classe $map1$ la classe ayant

	Classe 1	Classe 2	Classe 3	Classe 4
blocs_size	/	/	/	/
ignore_empty_documents	FALSE	FALSE	FALSE	TRUE
Stemmer	**Crop**, FSS	ESS, FSS, TRPS, TRWPS, **WPS**	**Crop**, FSS, TRWPS, WPS	ESS, **PS**, TRPS, TRWPS, WPS
retrieving_model	DFI0, DFRee, DirichletLM, DLH, **DLH13**, DPH, InB2, InexpC2, JsKLs, LGD, PL2, XSqrAM	BM25, DFRBM25, DirichletLM, DLH, **HiemstraLM**, InL2, LGD, PL2, TFIDF	**DFI0**, DFRee, DPH, JsKLs, LGD, XSqrAM	BB2, BM25, DFRBM25, DLH, HiemstraLM, IFB2, InB2, InexpB2, InexpC2, InL2, **LemurTFIDF**
TrecQueryTags_process	DESC-NARR, TITLE-DESC-NARR, **TITLE-NARR**	DESC, **TITLE**, TITLE-DESC	DESC-NARR, **NARR**, TITLE-DESC-NARR, TITLE-NARR	DESC, **TITLE**, TITLE-DESC
ignore_low_idf_terms	TRUE	FALSE	FALSE	TRUE
P_100	**P_100_2**, P_100_3	P_100_3, **P_100_4**	**P_100_1**	P_100_3, **P_100_4**
P_30	**P_30_2**	**P_30_3**	**P_30_1**	**P_30_4**
Rprec	**Rprec_2**	**Rprec_3**	**Rprec_1**	**Rprec_4**
Bpref	**bpref_2**	**bpref_3**	**bpref_1**	**bpref_4**
iprec_at_recall_0.30	iprec_0.30_1, **iprec_0.30_2**	**iprec_0.30_2**	**iprec_0.30_1**	**iprec_0.30_3**
Map	**map_2**	**map_3**	**map_1**	**map_4**

FIG. 6 – *Valeur des paramètres pour les classes de AP.*

les valeurs de AP les plus faibles jusqu'à $map4$ la classe ayant les valeurs de AP les plus fortes. La figure 6 présente les résultats du croisement entre les classes de AP et les variables paramètres selon la méthode de classification mixte.

Nous pouvons observer dans la figure 6 les modalités présentes dans chacune de ces classes. La classe 1 contient des mesures de performance à valeurs faibles, la classe 2 des mesures de performance à valeurs moyennes, la classe 3 des mesures de performance à valeurs très faibles et la classe 4 des mesures de performance à valeurs élevées.

La combinaison de paramètres la plus représentative pour la classe 4, c'est-à-dire pour la classe ayant des mesures de performance à valeurs élevées est :

- IgnoreEmptyDocuments = TRUE
- Stemmer = PS
- RetrievingModel = LemurTFIDF
- TrecQueryTagsProcess = TITLE
- IgnoreLowIdfTerms = TRUE
- IgnoreEmptyDocuments = TRUE
- Stemmer = PS
- RetrievingModel = LemurTFIDF
- TrecQueryTagsProcess = TITLE
- IgnoreLowIdfTerms = TRUE

La combinaison de paramètres la plus représentative pour la classe 3, c'est-à-dire pour la classe ayant des mesures de performance à valeurs très faibles (map1) est :

- IgnoreEmptyDocuments = FALSE
- Stemmer = Crop
- RetrievingModel = DFI0
- TrecQueryTagsProcess = NARR
- IgnoreLowIdfTerms = FALSE

Cette combinaison de paramètres est donc à éviter, de même, que la combinaison de paramètres la plus représentative pour la classe 1, c'est-à-dire pour la classe ayant des mesures de performance à valeurs faibles (map2).

5 Conclusions et perspectives

Dans cet article, nous nous sommes intéressés à une analyse massive de résultats de recherche d'information obtenus par un paramétrage du système. Ainsi, de nombreux paramètres ont été analysés, en étudiant les effets croisés de ceux-ci. Nous avons pu distinguer les requêtes en fonction de leur niveau de difficulté et définir les paramètres qui ont le plus d'influence en fonction de ces classes ainsi que leurs valeurs les plus adaptées. Un aspect qui reste à étudier est l'influence de la collection sur les résultats obtenus. En effet, nous nous sommes ici intéressés à une collection unique (TREC8).

Dans la suite de ces travaux, nous allons travailler sur des méthodes sélectives de recherche d'information, c'est à dire des méthodes qui adaptent le traitement en fonction des cas rencontrés. Ainsi, toutes les requêtes ne seront pas traitées de la même façon par le moteur, mais les paramètres du système seront au contraire différents en fonction du type de requêtes.

Le projet CAAS, financé par l'ANR dans le cadre de l'appel Contint 2010, a permis de développer le travail présenté ici. Nous remercions également Anthony Bigot et Sébastien Déjean pour leurs précieux conseils.

Références

Baccini, A., S. Déjean, J. Mothe, et L. Lafage (2012). How many performance measures to evaluate information retrieval systems ? *Knowledge and Information Systems 30*(3), 693–713.

Banks, D., P. Over, et N.-F. Zhang (1999). Blind men and elephants : Six approaches to trec data. *Information Retrieval 1*(1-2), 7–34.

Bendel, R. B. et A. A. Afifi (1977). Comparison of stopping rules in forward stepwise regression. *Journal of the American Statistical Association 357*(72), 46–53.

Bigot, A., C. Chrisment, T. Dkaki, G. Hubert, et J. Mothe (2011). Fusing different information retrieval systems according to query-topics : a study based on correlation in information retrieval systems and TREC topics. *Information Retrieval 14*(6), 617–648.

Bigot, A., S. Déjean, et J. Mothe (2014). Choisir la meilleure configuration d'un système de recherche d'information. *Document numérique 17*(2), 125–147.

Chrisment, C., T. Dkaki, J. Mothe, S. Poulain, et L. Tanguy (2005). Recherche d information - analyse des résultats de différents systèmes réalisant la même tâche. *Revue des Sciences et Technologies de l'Information 10*(1), 31–55.

Compaoré, J., S. Déjean, A. M. Gueye, J. Mothe, et J. Randriamparany (2011). Mining information retrieval results : Significant IR parameters. In *Advances in Information Mining and Management*.

Dinçer, B. T. (2007). Statistical principal components analysis for retrieval experiments : Research articles. *Journal of the Association for Information Science and Technology 58*(4), 560–574.

Laporte, L., R. Flamary, S. Canu, S. Déjean, et J. Mothe (2014). Non-convex Regularizations for Feature Selection in Ranking with Sparse SVM. *IEEE Transactions on Neural Networks and Learning Systems 25*(6), 1118–1130.

Louédec, J. et J. Mothe (2013). A massive generation of ir runs : Demonstration paper. In *Proceedings of RCIS Conference*, RCIS'13, pp. 1–2.

Mizzaro, S. et S. Robertson (2007). Hits hits TREC : exploring IR evaluation results with network analysis. In *ACM SIGIR conference on Research and development in information retrieval*, pp. 479–486.

Naini, K. D. et I. S. Altingovde (2014). Exploiting result diversification methods for feature selection in learning to rank. In *Advances in Information Retrieval*, pp. 455–461. Springer.

Ounis, I., G. Amati, V. Plachouras, B. He, C. Macdonald, et C. Lioma (2006). Terrier : A high performance and scalable information retrieval platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*.

Ponte, J. M. et W. B. Croft (1998). A language modeling approach to information retrieval. In *ACM SIGIR conference on Research and development in information retrieval*, pp. 275–281.

Qin, T., T.-Y. Liu, J. Xu, et H. Li (2010). Letor : A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval 13*(4), 346–374.

Voorhees, E. (2007). Overview of the TREC 2006. In *Proceedings of the Fifteenth Text REtrieval conference, NIST Special Publication 500-272*, pp. 1–16. NIST.

Summary

This paper presents a detailed analysis of a set of 2 million of search engine results obtained by different settings. More specifically, we used the Terrier platform and RunGeneration interface to create different versions (run in English) by changing the indexing and retrieval models. We then evaluated each result using different information retrieval performance measures. A systematic analysis was conducted on these data to determine firstly what the parameters that have the most influence were; on the other hand what the values of these parameters most likely lead to good system performance were.