

# Variations autour de *tf idf* et du moteur Lucene

Jacques Savoy, Ljiljana Dolamic

Institut d'informatique  
Université de Neuchâtel – rue Emile Argand 11 - 2 009 Neuchâtel - Suisse  
{Jacques.Savoy, Ljiljana.Dolamic}@unine.ch

## Abstract

This paper evaluates and compares the retrieval effectiveness resulting from various models derived from the classical *tf idf* paradigm when searching into a test-collection written in the French language (CLEF, 299 queries). We show that the simple paradigm “*tf idf*” may hide various formulations providing different retrieval effectiveness measured either by the mean average precision (MAP) or the mean reciprocal rank (MRR). Our analysis demonstrates that the best retrieval performance can be obtained from applying the Okapi probabilistic model. However, when faced with particular contexts (e.g. distributed IR) where the *idf* value cannot be obtained during the indexing process, we demonstrated that a simple indexing scheme (based only the frequency of occurrence or *tf*) may produce a significantly better performance than the classical « *tf idf* » model. Using the Lucene search engine, we also analyze and evaluate two particular features of this open-source system (namely the boost and coordinate level match).

## Résumé

A l'aide d'un corpus écrit en langue française et composé de 299 requêtes, cet article analyse et compare l'efficacité du dépistage de diverses stratégies d'indexation et de recherche basées sur le modèle classique « *tf idf* ». Cette dernière formulation demeure ambiguë et cache diverses variantes possédant des performances différentes, performance mesurée soit par la précision moyenne (MAP) soit par le rang moyen de la première bonne réponse (MRR). Notre analyse confirme que la meilleure efficacité s'obtient par le modèle Okapi. Mais lorsque nous sommes dans des contextes particuliers (e.g., systèmes distribués) dans lesquels la valeur de l'*idf* n'est pas connue lors de l'indexation des documents, nous démontrons que des stratégies simples, basées uniquement sur la fréquence d'occurrence (ou *tf*) permettent d'obtenir une performance significativement meilleure que le modèle classique « *tf idf* ». En utilisant le moteur Lucene (logiciel libre), nous avons également évalué deux de ses facettes, à savoir l'accroissement d'importance attachée aux mots des titres et la prise en compte du nombre de termes en commun entre le document dépisté et la requête.

**Mots-clés :** évaluation, recherche d'information, *tf idf*, Lucene, langue française.

## 1. Introduction

Le domaine de la recherche d'information (RI) (Baeza-Yates & Ribeiro-Neto 1999) s'est développé en alliant théorie et expérience contrôlée. Dès les années 60, nous avons désiré évaluer et comparer les diverses stratégies proposées afin, par exemple, de savoir si une stratégie d'indexation s'avèrerait meilleure qu'une autre (Cleverdon 1967) ou pour connaître si un système de dépistage proposait, en moyenne, de meilleurs résultats qu'un autre. Cette tradition a vu son influence grandir grâce à la diffusion de collections tests. Cette tendance empirique a conservé toute sa vigueur également grâce à la mise à disposition des logiciels nécessaires pour créer et développer son propre modèle de recherche d'information. Le logiciel SMART (Salton 1971) développé par l'équipe du professeur G. Salton durant les

années 60 constitue certainement le meilleur exemple de ce point de vue. Actuellement plusieurs autres moteurs de recherche sont mis à disposition du public comme Lemur<sup>1</sup>, un logiciel libre basé sur les modèles de langue en RI, Terrier ou MG (*Managing Gigabytes*). Dans notre cas, nous avons opté pour le logiciel Lucene (Gospodnetić & Hatcher 2004) qui nous semble plus apte à pouvoir traiter de grands volumes de documents.

Ces outils permettent à d'autres chercheurs de reproduire des expériences, avec les mêmes outils et les mêmes données respectant ainsi un des trois critères imposés par Popper (Popper 1968) pour définir ce qu'est une science. En informatique nous avons souvent tendance à construire un prototype, à faire une démonstration qui s'assimile souvent à une preuve formelle. En RI on dispose donc d'une situation privilégiée car nous sommes capable de répliquer un modèle de recherche pour en vérifier l'exactitude. Mais encore faut-il pouvoir dupliquer une telle expérience dans ses moindres détails. Malheureusement, nous nous retrouvons de plus en plus devant des descriptions fort lacunaires du système utilisé : « c'est Lucene », « on a pris le moteur standard de Lucene » ou « c'est le modèle classique *tf idf* ».

Dans cet article nous désirons présenter et évaluer quelques variations possibles autour du modèle classique « *tf idf* » de même que de comparer ces dernières avec les possibilités offertes par le modèle « standard » de Lucene. De plus, la RI distribuée se basant par exemple sur les réseaux pair-à-pair (Skobeltsyn *et al.* 2007) propose de nouvelles solutions afin de faciliter l'accès à l'information (robustesse, meilleure répartition de la charge sur le réseau ou pour le stockage de l'information, etc.). Mais dans ce cas, la statistique *idf* ou son approximation demeurent difficile à obtenir de manière précise. Dans ce cas, nous proposons d'indexer les documents en tenant uniquement compte de la fréquence d'occurrence (ou *tf*) et nous démontrons que, dans ce cas, les performances pouvant être obtenues peuvent dépasser celle du modèle classique « *tf idf* » (cosinus).

La suite de cette communication est organisée de la manière suivante. La section 2 décrit diverses variantes du paradigme « *tf idf* » tandis que la section 3 les évalue au moyen de la précision moyenne (MAP) ou du rang de la première bonne réponse dépistée (MRR). La section 4 présente deux caractéristiques additionnelles du moteur Lucene et analyse leurs performances. La section 5 résume nos principales contributions.

## 2. Quelques variantes du modèle *tf idf* et leurs évaluations

Nous nous sommes intéressés à évaluer diverses formulations pouvant être dérivées du modèle « *tf idf* » tout en posant comme référence auxiliaire la performance obtenue par le modèle probabiliste Okapi. Ces modèles sont décrits dans la section 2.2 qui sera précédée par une brève description de notre corpus d'évaluation (section 2.1).

### 2.1. La collection-test

Dans le cadre des campagnes d'évaluation CLEF (<http://www.clef-campaign.org>) (Peters *et al.* 2006), différentes équipes collaborent afin de créer des collections-tests. De tels corpus comprennent un ensemble de documents, des requêtes et pour chacune d'entre elles la référence aux documents jugés pertinents. Dans le but d'avoir un nombre important de requêtes, nous avons retenu les corpus rédigés en français durant les campagnes CLEF de

---

<sup>1</sup> Lemur a été développé à l'University of Massachusetts à Amherst (<http://www.lemurproject.org/>), Terrier à l'University of Glasgow (<http://ir.dcs.gla.ac.uk/terrier/>) et le système MG à l'University of Melbourne (<http://www.cs.mu.oz.au/mg/>). Lucene a été conçu par D. Cutting (<http://lucene.apache.org/>).

2001 à 2006 (voir table 1). Regrouper ces différents corpus nous permet de pouvoir évaluer les différentes stratégies de dépistage sur un nombre élevé de requêtes, soit 299 pour être précis. Grâce à ce nombre relativement élevé, nous pouvons plus facilement détecter des différences statistiquement significatives.

	2001	2002	2003	2004	2005	2006
Source	<i>Le Monde</i> 94 ATS 94	<i>Le Monde</i> 94 ATS 94	<i>Le Monde</i> 94 ATS 94-95	<i>Le Monde</i> 95 ATS 95	<i>Le Monde</i> 94-95 ATS 94-95	<i>Le Monde</i> 94-95 ATS 94-95
Taille	243 MB	243 MB	331 MB	244 MB	487 MB	487 MB
Doc.	87 191	87 191	129 806	90 261	177 452	177 452
Requête	49	50	52	49	50	49
n° à n°	41 à 90	91 à 140	141 à 200	201 à 250	251 à 300	301 à 350

Table 1 : Quelques informations sur les différentes parties de notre corpus

Les documents de notre corpus d'évaluation proviennent de deux sources, soit des articles de presse du journal *Le Monde* et des dépêches de l'Agence Télégraphique Suisse (ATS). Dans ces deux cas, notre corpus couvre les années 1994-1995. Chaque document comprend habituellement un titre suivi de un à quatre paragraphes de texte ne possédant que peu de fautes d'orthographe. Plusieurs informations concernant les diverses campagnes d'évaluation ayant produit ce corpus sont indiquées dans la table 1. A l'aide de cette dernière, on constate par exemple que la campagne d'évaluation de CLEF 2002 utilisait uniquement l'année 1994 du journal *Le Monde* et de l'ATS. Par contre, pour la campagne 2004 on a eu recours uniquement à l'année 1995 de ces deux mêmes sources représentant 244 MB de texte pour 90 261 documents. Les requêtes concernant cette partie sont numérotées de 201 à 250. Pour 49 d'entre elles, on peut dépister au moins une bonne réponse en consultant soit *Le Monde* ou l'ATS de l'année 1995.

Les besoins d'information exprimés couvrent des sujets divers (« Embargo sur l'Iraq », « Le snowboard », « Langues officielles de l'Union Européenne » ou « Films de James Bond »), touchant parfois des sujets plutôt nationaux voire régionaux (« Référendums en Suisse ») ou, inversement, des thèmes possédant une couverture internationale (« Variations du prix du pétrole »). Lors de la formulation de ces requêtes, on s'intéressait non pas à des événements temporels précis mais plutôt à des thématiques récurrentes.

La formulation complète de ces demandes ne se limite pas à un bref titre comme l'illustrent les exemples de la table 2. Chaque requête se subdivise en quatre champs, à savoir l'identificateur (<num>) suivi par le titre (<title> ou T) exprimant par deux ou trois mots le thème central de la requête, la partie descriptive (<desc> ou D) indiquant par une phrase le besoin d'information de l'utilisateur et, finalement, la partie narrative (<narr> ou N) précisant les critères de pertinence. On remarque que les subdivisions logiques « titre » et « description » comprennent parfois des formulations très similaires ou, dans d'autres cas, ces deux parties se complètent, l'une apportant des synonymes à l'autre ou des formulations différentes du même concept (voir la table 2 où le mot « affaires » voit son sens précisé, c'est-à-dire « dans la politique », « corruption » et « financement des partis »). Comme lors de l'évaluation officielle des campagnes CLEF, nous allons construire nos requêtes sur la base de ces champs (TD). D'autre part, afin de refléter plus fidèlement la longueur des requêtes soumises aux

moteurs de recherche sur le Web (Witten *et al.* 2007), nous avons également procédé à des évaluations basées uniquement sur le titre (T) des requêtes. Limitées à la partie « titre » (T), les requêtes comprennent, en moyenne, 2,86 mots pleins (écart-type : 0,84, min : 1, max : 6) tandis qu'avec la formulation plus longue (TD), nous rencontrons, en moyenne, 10,77 termes par requête (écart-type : 3,5, min : 2, max : 25).

<pre>&lt;top&gt; &lt;num&gt; C60 &lt;/num&gt; &lt;title&gt; Les affaires en France &lt;/title&gt; &lt;desc&gt; Rechercher des documents concernant les affaires de corruption dans la vie politique française, en particulier celles concernant le financement des partis &lt;/desc&gt; &lt;narr&gt; En France, de nombreuses personnalités de premier plan, y compris des politiques et des dirigeants d'entreprise, sont touchées par des affaires de corruption. Les documents pertinents doivent donner des informations sur ces affaires. Les informations sur l'ouverture d'enquêtes policières et judiciaires en rapport avec des affaires de corruption dans la vie politique sont également à prendre en considération. &lt;/narr&gt; &lt;/top&gt;</pre>	<pre>&lt;top&gt; &lt;num&gt; C52 &lt;/num&gt; &lt;title&gt; Dévaluation de la monnaie chinoise &lt;/title&gt; &lt;desc&gt; Trouvez des documents décrivant les raisons et les effets de la dévaluation de la monnaie chinoise. &lt;/desc&gt; &lt;narr&gt; Des documents pertinents discuteront des arguments en faveur ou contre la dévaluation officielle de la valeur de la monnaie chinoise et les conséquences sociales et économiques de la dévaluation. &lt;/narr&gt; &lt;/top&gt; ...</pre>
---	--

Table 2 : Exemples de requêtes de notre corpus

## 2.2. Stratégies d'indexation et modèles de dépistage

Dans le but de représenter un document ou une requête par un ensemble de mots pondérés, on admet que trois facteurs devraient être pris en compte. D'abord, si un mot apparaît souvent dans un document, son importance devrait croître dans la représentation de celui-ci. La manière habituelle d'inclure cette composante consiste à retenir la fréquence d'occurrence (ou fréquence lexicale, notée  $tf_{ij}$  pour le terme  $t_j$  dans le document  $i$ ).

Ensuite, afin de mieux discerner les termes les uns des autres, on a recours à la fréquence documentaire d'un terme  $t_j$  (ou  $df_j$ ) correspondant au nombre de documents dans lesquels ce terme  $t_j$  apparaît. Cette statistique n'est pas utilisée telle quelle mais on recourt au logarithme de son inverse (noté  $idf_j = \log(n/df_j)$ , avec  $n$  le nombre de documents dans le corpus) (Sparck Jones 1972). Ainsi si un terme apparaît dans tous les documents d'une collection, sa valeur  $idf$  sera nulle ( $\log(n/n) = 0$ ). Pour combiner les deux premières composantes, on procède le plus souvent à une multiplication des deux parties produisant une première interprétation possible du fameux modèle «  $tfidf$  ».

Enfin, on remarque que ces pondérations ne sont pas normalisées. Ainsi, selon le contexte une valeur de  $tfidf = 0,4$  peut refléter une valeur importante ou signaler qu'une valeur dans la moyenne. Afin de disposer d'une pondération ayant des valeurs uniquement dans l'intervalle  $[0; 1]$ , nous pouvons recourir à la normalisation par le cosinus calculée selon la formule suivante (Salton & Buckley 1988).

$$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^t [tf_{ik} \cdot idf_k]^2}} \quad (1)$$

dans laquelle  $w_{ij}$  indique la pondération associée au terme  $t_j$  dans le document  $i$ . Cette équation constitue donc une deuxième interprétation de la formule «  $tfidf$  » que nous distinguerons en ajoutant le terme « (cos) ».

Nous pouvons également proposer une troisième version. Jusqu'à présent on a admis implicitement que l'on appliquait la même pondération aux termes des documents et à ceux de la requête. Or, lors de l'indexation des documents nous pouvons retenir uniquement la valeur  $tf$  comme pondération. Une telle approche s'avère particulièrement intéressante dans un environnement distribué dans laquelle les différentes sources de documents sont gérées par des serveurs distincts. Dans notre cas, on peut imaginer qu'un site Web gère la collection du Monde et un autre celle de l'ATS ou que plusieurs sites sont nécessaires pour une gestion efficace du corpus *Le Monde*. Dans un tel cas, il s'avère difficile de connaître, lors de l'indexation d'un document, le nombre d'articles qui seront finalement indexés par tel ou tel terme. Par contre la fréquence lexicale des mots s'obtient simplement en consultant l'article. Mais comme la statistique  $idf$  permet de bien discerner le pouvoir discriminant des divers termes, nous pouvons l'obtenir plus facilement lors de l'interrogation. Dès lors, l'indexation des documents s'appuie exclusivement sur des informations locales (le document lui-même et ses diverses subdivisions logiques). Ensuite, lors de l'indexation de la requête, les termes de celle-ci sont pondérés selon la formule «  $tfidf$  » (sans normalisation). Remarquons que la composante  $tf$  ne joue pas un grand rôle car les termes des requêtes présentent très souvent une fréquence lexicale unitaire en particulier si l'on considère les requêtes courtes (ou T).

De multiples variantes peuvent être proposées comme, par exemple, pour imposer que la première occurrence d'un terme doit posséder plus d'influence que les autres. Ainsi à la simple fréquence d'occurrence  $tf_{ij}$ , on peut substituer  $\log(tf_{ij})$  (approche que nous dénoterons «  $\log(tf)$  » (Buckley *et al.* 1996)) ou encore  $0,5 + (0,5 \cdot tf_{ij} / \max tf_{ik})$  (ou «  $\max tf$  »). Cette dernière possède l'avantage de retourner une valeur entre 0 et 1. Dans ce cas, l'idée sous-jacente consiste à attribuer un poids important à la première occurrence avec l'ajout de la constante 0,5 puis de normaliser les fréquences des différents termes d'un document par la fréquence maximale (le terme le plus fréquent dans ce document).

Pour calculer le score d'un document par rapport à une requête, nous avons choisi le produit interne (voir équation 2). Dans cette formule, le poids attaché au terme  $t_j$  dans le document  $i$  est noté par  $w_{ij}$  tandis que  $w_{qj}$  indique la pondération de ce même terme dans la requête.

$$Score(Q, D_i) = \sum_{t_j \in Q} w_{ij} \cdot w_{qj} \quad (2)$$

En plus de ces solutions basées sur la vision géométrique dérivée du modèle vectoriel, nous avons considéré le modèle probabiliste Okapi (ou BM25) (Robertson *et al.* 2000) définit par la formule suivante :

$$Score(Q, D_i) = \sum_{t_j \in Q} tf_{qj} \cdot \log\left(\frac{n - df_j}{df_j}\right) \cdot \frac{(k_1 + 1) \cdot tf_{ij}}{K + tf_{ij}} \quad \text{et } K = k_1 \cdot [(1 - b) + b \cdot (l_i / avdl)] \quad (3)$$

dans laquelle  $l_i$  indique le nombre de termes d'indexation inclus dans la représentation du document  $i$  et  $n$  le nombre de documents dans le corpus. Dans nos expériences, la constante  $avdl$  a été fixée à 190 (ce qui correspond à la longueur moyenne, en nombre de termes, des articles de la collection),  $b = 0,4$ , et  $k_1 = 1,2$ .

Finalement, dans la représentation des documents et des requêtes, les formes très fréquentes et peu ou pas porteuses d'information ont été éliminées (soit 463 formes disponibles à l'adresse <http://www.unine.ch/info/clef/>). Les majuscules et les accents sont ignorés. De même, nous avons procédé à l'élimination automatique de certaines séquences terminales (par exemple les « -s » ou « -es » selon l'algorithme donnée à <http://www.unine.ch/info/clef/>, (Savoy 2002)).

Grâce à ces techniques, une requête comme « le jeu de Nim » permet de dépister tant un document ayant la forme « jeu » ou « jeux ». De même, les formes « le » ou « de » n'ajoutant aucune information pertinente à la requête seront ignorées. Il faut reconnaître que ces deux stratégies ne sont pas exemptes d'erreurs. Ainsi notre requête précédente soumise à Google dépiste en premier lieu le document « Jeux à proximité de Nîmes » avec un appariement erroné entre les mots « Nim » et « Nîmes ».

### 3. Evaluation

Afin de mesurer la performance de ces différents modèles de dépistage, nous avons utilisé la précision moyenne (*Mean Average Precision* ou MAP) (Buckley & Voorhees 2005 ; Voorhees 2007) calculée par le logiciel `trec_eval` sur la base des 1 000 premières réponses. Cette mesure a été adoptée par les diverses campagnes d'évaluation pour évaluer la qualité de la réponse à des interrogations en ligne. La meilleure valeur possible serait 1,0 obtenue si le modèle classe toujours tous les documents pertinents au début de sa liste de réponse. A l'inverse, si aucun document pertinent n'est retourné, la performance s'élèverait à 0,0. Enfin, pour déterminer si une différence de performance entre deux modèles s'avère statistiquement significative, nous avons opté pour un test bilatéral non-paramétrique (basé sur le ré-échantillonnage aléatoire ou *bootstrap* (Savoy 1997), avec un seuil de signification  $\alpha = 5 \%$ ).

Modèle de recherche \ Type de requête	MAP	
	T	TD
1. Okapi	<b>0,4008 *</b>	<b>0,4588 *</b>
2. doc= <i>tf idf</i> (cos), requête = <i>tf idf</i> (cos) Eq. 1	0,2591	0,2985
3. doc= <i>tf idf</i> , requête = <i>tf idf</i>	0,2562	0,2789 *
4. doc= <i>tf</i> , requête = <i>tf idf</i>	0,2229 *	0,2341 *
5. doc=log( <i>tf</i> ), requête = <i>tf idf</i>	0,3586 *	0,3943 *
6. doc=0,5*(0,5 <i>tf</i> /max <i>tf</i> ), requête = <i>tf idf</i>	0,3790 *	0,4274 *

Table 3 : Précision moyenne (MAP) de divers modèles de dépistage (299 requêtes)

Dans la table 3, nous avons repris l'évaluation avec des requêtes construites uniquement à l'aide du champ « titre » (T) ou avec les champs « titre & description » (TD). En première ligne, nous avons repris le modèle probabiliste Okapi connu pour apporter une performance élevée. Les trois lignes suivantes indiquent la précision moyenne obtenue par trois variations du schéma « *tf idf* ». Enfin les deux dernières présentent deux autres approches dans lesquelles la statistique *idf* n'est pas utilisée lors de l'indexation des documents mais uniquement lors de l'analyse de la requête. On notera également que si formellement la pondération pour les requêtes est de type « *tf idf* », en présence uniquement du titre des besoins d'informations (voir exemples dans la table 2), un tel formalisme se limite à *idf*.

Les valeurs de performance de la table 3 indiquent clairement que le modèle probabiliste Okapi présente, pour les deux formulations de requête, la meilleure précision moyenne. En recourant à notre test statistique, nous avons constaté que la différence s'avère toujours statistiquement significative avec les autres approches. L'écart relatif entre le modèle Okapi et

le modèle « *tfidf* » indiqué par l'équation 1 (normalisation du cosinus) s'élève à 35 % (T : 0,4008 vs. 0,2591, soit -35,4 % ; TD : 0,4588 vs. 0,2985, soit -34,9 %).

Pourtant l'écart de performance entre Okapi et un modèle plus simple n'est pas toujours aussi considérable. Ainsi, si l'on reprend la dernière ligne de notre tableau (modèle « max *tf* » en 6<sup>e</sup> ligne), on constate que l'écart relatif est assez réduit, de l'ordre de 6 % (T : 0,4008 vs. 0,3790, soit -5,4 % ; TD : 0,4588 vs. 0,4274, soit -6,8 %). Ce résultat s'avère particulièrement intéressant dans le cas de la recherche distribuée. En effet, lors de l'indexation des documents, nous n'avons pas recours à la valeur *idf*. Cette dernière n'est utilisée que pour pondérer les termes de la requête.

Si nous fixons comme performance de référence celle obtenue par le modèle « *tfidf* » de la deuxième ligne (avec normalisation), les différences de précision moyenne sont souvent jugées statistiquement significatives (signalée par un astérisque « \* »). Le seul écart jugé non significatif se situe entre le modèle « *tfidf* » (cos) (2<sup>e</sup> ligne) et « *tfidf* » sans normalisation (3<sup>e</sup> ligne) et ceci uniquement pour les requêtes courtes.

L'évaluation indiquée dans le tableau 3 s'appuie sur le rang de tous les documents pertinents à la requête sous-jacente. Or, sur le Web en particulier (Witten *et al.* 2007), les usagers sont souvent intéressés à obtenir une seule bonne réponse à leur interrogation. Afin d'évaluer les diverses stratégies d'indexation et de dépistage dans de telles conditions, nous pouvons utiliser le rang moyen (*Mean Reciprocal Rank* ou MRR) de la première réponse exacte retournée par le système (Buckley & Voorhees 2005). Ainsi, si le premier document dépisté est pertinent, le score obtenu sera de 1,0 et si c'est le deuxième, nous obtenons la valeur  $\frac{1}{2} = 0,5$ .

Modèle de recherche \ Type de requête	MRR	
	T	TD
1. Okapi	0,6631 *	<b>0,7360 *</b>
2. doc= <i>tfidf</i> (cos), requête = <i>tfidf</i> (cos) Eq. 1	0,5072	0,5678
3. doc= <i>tfidf</i> , requête = <i>tfidf</i>	0,5044	0,5260
4. doc= <i>tf</i> , requête = <i>tfidf</i>	0,4586 *	0,4716 *
5. doc=log( <i>tf</i> ), requête = <i>tfidf</i>	<b>0,6658 *</b>	0,6896 *
6. doc=0,5*(0,5 <i>tf</i> /max <i>tf</i> ), requête = <i>tfidf</i>	0,6308 *	0,7034 *

Table 4 : Inverse du rang moyen de la première bonne réponse selon nos différents modèles de dépistage (299 requêtes)

L'évaluation des différents modèles de recherche selon cette mesure est reprise dans la table 4. Ces mesures indiquent que le modèle Okapi propose une performance statistiquement meilleure que toutes les autres approches. La seule exception à cette conclusion est la différence entre le modèle Okapi et l'approche « log(*tf*) » (5<sup>e</sup> ligne) et uniquement pour les requêtes courtes (T : 0,6631 vs. 0,6658). Dans ce cas, la différence entre les deux modèles n'est pas jugée significative. De même, l'écart de performance entre Okapi et le modèle « max *tf* » (6<sup>e</sup> ligne) s'avère peu important (T : 0,6631 vs. 0,6308, soit -4,9 %) mais dans ce cas, la différence est tout de même jugée statistiquement significative.

Fixons comme performance de référence celle obtenue par le modèle « *tfidf* » (cos) de la deuxième ligne. Dans ce cas, les différences de performance sont souvent statistiquement

significatives (signalée par un astérisque « \* »). Le seul écart jugé non significatif se situe entre le modèle « *tf idf* » (cos) (2<sup>e</sup> ligne) et le modèle sans normalisation (3<sup>e</sup> ligne).

Si la moyenne possède l'avantage de résumer par une valeur une distribution, elle cache aussi les irrégularités dans les performances individuelles de chaque requête. Afin de mieux comprendre quelques valeurs indiquées dans la table 4, nous avons analysé en détail quelques performances obtenues avec les requêtes courtes (T). Avec le modèle Okapi, pour 166 requêtes sur 299, le premier document dépisté est pertinent. Si l'on consulte les premiers dix références (soit le premier écran de résultat d'un moteur commercial sur le Web), on trouve 257 requêtes pour lesquelles au moins une bonne réponse apparaît. Ces valeurs sont similaires pour le modèle « *log(tf)* » (5<sup>e</sup> ligne) avec 168 requêtes avec une bonne réponse en tête ou 264 interrogations avec une bonne réponse parmi les dix premières références.

Le modèle « *max tf* » (6<sup>e</sup> ligne) propose aussi une performance proche de celle d'Okapi et s'avère bien adapté aux systèmes de recherche distribuée. Il requiert tout de même la statistique *idf* pour la pondération des requêtes. Afin de vérifier les différences de performance si l'on ignore la statistique *idf* également lors de l'analyse des requêtes nous avons construit le modèle «  $\text{document} = 0,5 * (0,5 \text{ } tf / \text{max } tf)$ , requête = *tf* » Dans un tel cas et avec des requêtes courtes, la MAP s'élève à 0,3398 vs. 0,379 (-10,3 %) et la MRR à 0,6339 vs. 0,6308 (+0,5 %). On constate que l'absence de l'information *idf* pénalise clairement la mesure MAP mais dans une moindre mesure la performance MRR. Ainsi, le modèle avec *idf* propose au premier rang une bonne réponse pour 151 requêtes sur 299, et pour l'approche sans *idf*, cette valeur est de 156 requêtes. Une constatation similaire peut être faite avec le modèle « *log(tf)* » (5<sup>e</sup> ligne) et son correspondant sans *idf* ( $\text{doc} = \text{log}(tf)$ , requête = *tf*). La MAP diminue sensiblement (0,3586 vs. 0,3077, soit -14,2 %) mais la MRR varie moins (0,6658 vs. 0,6414, soit -3,7 %). L'absence de la statistique *idf* n'implique pas une baisse considérable de la performance d'un moteur de recherche, surtout si l'on recourt à la mesure MRR.

#### 4. Quelques facettes additionnelles du moteur Lucene

Afin de pouvoir répliquer des expériences, nous nous sommes appuyés sur le moteur Lucene (<http://lucene.apache.org/>) (Gospodnetić & Hatcher 2004) faisant parti des logiciels *open-source*. Dans sa fonction de similarité, Lucene cherche à tirer parti de deux paradigmes de la RI, à savoir le modèle vectoriel et son « *tf idf* » d'une part et, d'autre part, du modèle booléen. Ainsi on accordera plus d'importance aux termes apparaissant souvent (*tf*) dans le document et qui sont relativement rares dans le corpus (*idf*). Du modèle booléen on retient que le nombre de termes communs entre le document et la requête s'avère souvent un bon indicateur de la pertinence du document. Une idée similaire se retrouve également dans le modèle basé sur la régression logistique (Gey 1994) ou dans le *quorum search* (Cleverdon 1984). Enfin, pour permettre plus de souplesse dans l'appariement, on autorise l'accroissement de l'importance d'une composante (terme ou document) en incluant des facteurs multiplicatifs (*boosting*). Ainsi un document jugé a priori important peut voir le poids associé à toutes ces composantes multiplié par une constante. Un champ particulier d'un document peut aussi faire l'objet d'une telle amplification, de même qu'une partie de la requête.

Afin d'analyser quelques facettes de Lucene, nous avons repris la possibilité d'utiliser un facteur multiplicatif pour les titres des articles de presse (section 4.1) puis nous avons évalué le modèle standard de Lucene (section 4.2).

#### 4.1. Pondération accrue des titres

Le modèle Lucene nous permet de mettre en lumière le fait que selon la partie logique du document concerné, un terme d'indexation commun avec la requête n'a pas forcément la même valeur. Ainsi, on a la possibilité d'accorder plus d'importance aux mots du titre qu'aux termes apparaissant uniquement dans le corps du document. Nous attribuons un facteur multiplicatif de trois dans ce cas. En d'autres termes, lorsque qu'un mot apparait dans le titre d'un article nous lui accordons une fréquence lexicale de trois. Dans la table 5, nous avons repris les mesures MAP et MRR uniquement avec les requêtes courtes (T) et selon nos divers modèles de recherche pour évaluer deux cas de figure, soit avec le *boost* (« 3 x TITRE ») soit en ignorant cette possibilité (« 1 x TITRE »).

Modèle de recherche	MAP		MRR	
	3 x TITRE	1 x TITRE	3 x TITRE	1 x TITRE
1. Okapi	<b>0,4008</b>	<b>0,4007</b>	<b>0,6631</b>	<b>0,6703</b>
2. doc= <i>tf idf</i> (cos), req= <i>tf idf</i> (cos)	0,2591	0,2635	0,5072	0,5087
3. doc= <i>tf idf</i> , req= <i>tf idf</i>	0,2562	0,2388 *	0,5044	0,4750 *
4. doc= <i>tf</i> , req= <i>tf idf</i>	0,2229	0,1965 *	0,4586	0,4157 *
5. doc=log( <i>tf</i> ), req= <i>tf idf</i>	0,3586	0,3345 *	0,6658	0,6280 *
6. doc=0,5*(0,5 <i>tf</i> /max <i>tf</i> ), req= <i>tf idf</i>	0,3790	0,3715 *	0,6308	0,6234

Table 5 : Précision moyenne (MAP) et rang moyen de la première bonne réponse (MRR) avec les requêtes courtes (T, 299 requêtes)

Pour le modèle Okapi, nous pouvons constater qu'une modification sensible au niveau de la MAP et une légère dégradation au niveau de la MRR (0,6703 vs. 0,6631, -1,1 %). Pour le modèle « *tf idf* » (cos), on remarque que d'accroître l'importance des mots inclus dans le titre n'a pas permis d'accroître la MAP (0,2635 vs. 0,2591, -1,7 %) ou le MRR (0,5087 vs. 0,5072, -0,3 %). Pour les approches n'appliquant pas de normalisation (ligne 3 à 5), le facteur multiplicatif associé aux titres apporte une augmentation de performance tant au niveau de la MAP que du MRR.

En analysant statistiquement les différences de performance au sein d'un même modèle, on constate que pour Okapi et « *tf idf* » (cos, 2<sup>e</sup> ligne), aucune différence n'est jugée significative. Pour les trois approches sans normalisation (ligne 3 à 5), toutes les différences sont statistiquement significatives. Pour le dernier modèle, la différence du MRR n'est pas significative ( $p\text{-value}=0,0135$ ) mais celle au niveau de la MAP l'est. Dans ce dernier cas, on constate que pour 174 requêtes la précision moyenne est supérieure avec le coefficient multiplicatif et pour 103 autres interrogations c'est l'inverse (pour 22 requêtes, la même performance est retournée). Dans un tel cas, le test du signe indiquera aussi une différence significative (au niveau  $\alpha=5\%$ ,  $p\text{-value}=0,00002$ ).

Bien que les requêtes possèdent plusieurs parties logiques (voir exemple dans la table 2), nous n'avons pas évalué la possibilité d'accorder des importances variées selon les champs. Enfin, nous n'avons pas attribué une utilité plus grande à un sous-ensemble des documents (par exemple ceux ayant une longueur plus faible que la moyenne). Cet aspect s'avère important sur le Web (Witten *et al.* 2007) où les pages d'accueil correspondent plus souvent aux

souhaits des internautes (Kraaij *et al.* 2002). Face à un corpus plus homogène comme celui des dépêches d'agence ou articles de presse, une telle distinction *a priori* entre documents n'a pas beaucoup de sens.

#### 4.2. Le modèle standard de Lucene

Dans le moteur Lucene, le score d'un document au regard d'une requête s'évalue grâce à l'équation 4. On y retrouve en premier lieu la normalisation (cosinus) des termes de la requête. Ensuite nous avons la contribution de chaque terme en commun entre la requête  $Q$  et le document  $D_i$ . Cette contribution dépend de l'*idf* de ce terme, de la racine carrée de la fréquence d'occurrence du terme  $t_j$  dans le document ( $tf_{ij}$ ) et de l'inverse de la racine carrée de la longueur du document ( $l_i$ , mesurée en nombre de termes d'indexation contenus dans le document  $D_i$ ). Chaque terme  $t_j$  peut en plus être amplifié par le facteur (*boost*) multiplicatif  $b_j$  (ou  $b_k$  pour la requête). Ce facteur multiplicatif est aussi associé au document via  $b_i$ . Finalement, on multiplie le score de similarité par un terme tenant compte du nombre de termes communs entre les représentations de la requête  $Q$  et du document  $D_i$  ( $\text{overlap}(Q, D_i)$ ).

$$\text{Score}(Q, D_i) = \frac{1}{\sqrt{\sum_{t_k \in Q} (\text{idf}_k \cdot b_k)^2}} \cdot \sum_{t_j \in Q} \left[ \sqrt{tf_{ij}} \cdot \text{idf}_j^2 \cdot b_j \cdot \frac{b_i}{\sqrt{l_i}} \right] \cdot \frac{\text{overlap}(Q, D_i)}{l_q} \quad (4)$$

Dans nos expériences, nous avons fixé toutes les valeurs  $b_i = 1$  (documents) ou  $b_k = 1$  (requête). Le facteur  $b_j$  associé au terme sera de trois si ce terme appartient au titre d'un document, et égal à 1 dans tous les autres cas. La présence dans cette formule de  $\text{idf}^2$  s'explique par le fait que l'on désire recourir à cette pondération lors de l'indexation de la requête et du document (ce qui revient au modèle «  $\text{sqrt}(tf) \text{idf}$  » pour les documents et «  $\text{idf}$  » (cos) pour les requêtes). La présence de la fonction racine carrée s'explique par le désir d'accorder une plus grande importance aux premières occurrences d'un terme. Cette caractéristique peut s'analyser comme une alternative au «  $\log(tf)$  » ou au «  $\max tf$  ». Finalement, on désire privilégier les documents courts en divisant par la racine carrée de la longueur du document considérée.

Modèle de recherche	T		TD	
		& overlap		& overlap
1. Okapi	<b>0,4008</b>		<b>0,4588</b>	
2. doc= $tf \text{idf}$ , req = $tf \text{idf}$ (cos)	0,2591	0,3511 *	0,2985	<b>0,3994 *</b>
3. doc= $\text{sqrt}(tf) \text{idf}$ , req = $\text{idf}$ (cos)	0,3421	0,3760 *	0,3618	0,3454 *

Table 6 : Précision moyenne (MAP) avec et sans le facteur de coordination (overlap) du moteur Lucene (299 requêtes)

Dans la table 6, nous avons analysé deux types de requêtes (T ou TD) avec ou sans un facteur de coordination (pourcentage de termes communs avec la requête ou *overlap*). L'ajout de ce facteur permet d'accroître la MAP pour le modèle «  $tf \text{idf}$  » (cos) (2<sup>e</sup> ligne) ou le modèle de Lucene (3<sup>e</sup> ligne) mais, dans ce dernier cas, l'augmentation se vérifie uniquement pour les requêtes courtes (T). Selon notre test statistique, toutes ces différences s'avèrent significatives.

Les résultats de la table 6 peuvent être comparés à ceux de la table 3 afin de connaître l'efficacité relative du recours à la racine carrée pour mieux distinguer les diverses valeurs  $tf$ .

Par rapport à une approche « max *tf* », la MAP diminue (T : 0,3421 vs. 0,3790 soit -9,7% en valeur relative ; TD : 0,3618 vs. 0,4274 ou -15,4 %). Finalement, si l'on compare les performances du modèle de Lucene (avec *overlap*) avec l'approche « max *tf* », nous obtenons des performances similaires avec les requêtes brèves (T : 0,3790 vs. 0,3790, ou -0,8 %) et une différence nette en présence de formulations plus longues (TD : 0,3454 vs. 0,4274, -19,2 %). En tout cas, le modèle Lucene propose une meilleure efficacité qu'une approche « *tf idf* » (avec ou sans normalisation).

## 5. Conclusion

Regrouper les collections-tests créées lors de plusieurs campagnes d'évaluation permet d'obtenir un corpus avec un nombre très important de requêtes, soit 299 dans notre cas. Avec cet ensemble, nous pouvons plus aisément détecter des différences statistiquement significatives entre les différentes stratégies de dépistage. Dans cette étude, nous avons démontré que même si le modèle « *tf idf* » (avec ou sans normalisation du cosinus) apporte une précision moyenne nettement inférieure au modèle Okapi (différence relative de 35 %, voir table 3), ceci ne signifie pas que toutes les variantes autour de ce paradigme apporteront également de faibles performances.

En particulier, nous avons mis en évidence qu'une approche simple dans laquelle la pondération des termes dans les documents repose uniquement sur la statistique « max *tf* » (plus précisément,  $0,5 * (0,5 \text{ } tf / \text{max } tf)$ ) permet d'obtenir une précision moyenne inférieure de seulement 6 % par rapport au modèle Okapi (T : 0,4008 vs. 0,3790, soit -5,4 % ; TD : 0,4588 vs. 0,4274, soit -6,8 %). Ce faible écart se rencontre également si l'on a recours au rang du premier document pertinent dépisté (MRR, voir table 4). Un tel schéma s'adapte aisément dans un système distribué dans lequel la statistique *idf* n'est pas disponible lors de l'indexation mais s'utilise uniquement pour la pondération des termes de la requête. Si on ignore complètement la statistique *idf*, le modèle « doc = max *tf*, req = *tf* » permet d'obtenir un MRR similaire à l'approche « doc = max *tf*, req = *tf idf* ». Par contre, si l'on recourt à la MAP, la différence entre les deux approches s'élève à 10 %.

Accorder une importance plus grande aux termes apparaissant dans le titre d'un article accroît significativement la performance pour des modèles de recherche n'appliquant pas de normalisation (voir table 5). Dans le cas contraire (e.g., Okapi, « *tf idf* » (cosinus)), la variation demeure faible et non significative. Enfin, l'emploi de l'option *overlap* du moteur Lucene ne procure pas un accroissement garanti de la performance. Tenir compte du nombre de termes communs entre la requête et le document semble être une stratégie intéressante en présence d'une requête courte (deux à trois termes). Mais dès que ce nombre augmente, cette technique tend à réduire la performance moyenne (voir table 6).

## Remerciements

Cette recherche a été financée en partie par le Fonds national suisse pour la recherche scientifique (subside n° 200021-113273).

## Références

- Baeza-Yates R. & Ribeiro-Neto B. (1999). *Modern information retrieval*. The ACM Press, New York (NY).
- Buckley C., Singhal A., Mitra M. & Salton G. (1996). New retrieval approaches using SMART. In *Proceedings of TREC'4*, NIST Publication #500-236, pages 25-48.

- Buckley C. & Voorhees E. (2005). Retrieval system evaluation. In *TREC, experiment and evaluation in information retrieval* (pp. 53-75). The MIT Press, Cambridge (MA).
- Cleverdon C. (1967). The Cranfield tests on index language devices. In *Aslib Proceedings*, 19, pages 173-192.
- Cleverdon C. (1984). Optimizing convenient on-line access to bibliographic databases. *Information Service & Use*, 4, pages 37-47
- Gey F. (1994). Inferring probability of relevance using the method of logistic regression. In *Proceedings of ACM-SIGIR-1994*, pages 222-231.
- Gospodnetić O. & Hatcher E. (2004). *Lucene in action. A guide to the Java search engine*. Manning, Greenwich (UK).
- Kraaij W., Westerveld T. & Hiemstra D. (2002). The importance of prior probabilities for entry page search. In *Proceedings of ACM-SIGIR-2002*, pages 27-34.
- Peters C., Gey F.C., Gonzalo J., Müller H., Jones G.J.F., Kluck M., Magini B. & de Rijke M. (2006). *Accessing multilingual information repositories*. Springer, Lectures Notes in Computer Science #4 022. Berlin.
- Popper K. J. (1968). *The logic of scientific discovery*. Harper & Row, New York (NY).
- Robertson S. E., Walker S. & Beaulieu M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1), pages 95-108.
- Salton G. (Ed.) (1971). *The SMART retrieval system. Experiments in automatic document processing*. Prentice-Hall Inc., Englewood Cliffs (NJ).
- Salton G. & Buckley C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), pages 513-523.
- Savoy J. (1997). Statistical inference in retrieval effectiveness evaluation. *Information Processing & Management*, 33(4), pages 495-512.
- Savoy J. (2002). Recherche d'informations dans des corpus en langue française : Utilisation du référentiel Amarylis. *TSI, Technique et Science Informatiques*, 21(3), pages 345-373.
- Sparck Jones K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1), pages 11-21.
- Skobeltsyn G., Luu T, Podnar Zarko I., Rajman M., Aberer K. (2007). Web text retrieval with a P2P query-driven index. In *Proceedings of ACM-SIGIR-2007*, pages 679-686.
- Voorhees E.M. (2007). TREC: Continuing information retrieval's tradition of experimentation. *Communications of the ACM*, 50(11), pages 51-54.
- Witten I.H., Gori M., Numeric, T. (2007). *Web dragons*. Elsevier, Amsterdam.