Evaluation de la précision pour un système hypertexte

Idir Chibane, Bich-Liên Doan

SUPELEC, plateau de Moulon, 3 rue Joliot Curie
91192 Gif sur Yvette
France
Idir.Chibane@supelec.fr, Bich-Lien.Doan@supelec.fr

RÉSUMÉ. *Certains moteurs de recherche, par exemple Google, utilisent les liens hypertextes dans le processus de sélection des documents en réponse à une requête. Dans ce papier, nous présentons une nouvelle fonction de correspondance qui effectue un classement des réponses à partir d'une mesure d'appariement entre les mots clés d'une requête et le texte ancre associé aux liens hypertextes des pages. Nous avons évalué cette fonction de correspondance par des expérimentations sur la collection TREC-9 et nous concluons que pour certains types de requêtes, notre système fournit de meilleures réponses en terme de précision.*

ABSTRACT. *Several search tools, for instance Google, use the hypertext links to select the matching documents against a query. In this paper, we propose a new matching function that select and sort the responses through a similarity measure. This measure is computed with the keywords of a query and the anchor text of pages. We assessed our matching function with experiments on the TREC-9 collection. We conclude that for certain types of queries, our system provides better results in terms of precision.*

MOTS-CLÉS : *recherche d'information sur le Web, systèmes hypertextes.*

KEYWORDS: *information retrieval on the Web, hypertext systems.*

1. Introduction

Les systèmes de recherche d'information (SRI) sont composés essentiellement de deux modules. Un module d'indexation qui représente les documents, et un module d'interrogation qui représente la requête. La fonction de correspondance permet de calculer le degré d'appariement entre les termes de la requête et les termes d'indexation des documents afin d'évaluer la pertinence des documents par rapport à la requête. Avec le succès grandissant du Web (Google recense plus de 8 milliards de pages Web) le classement des réponses devient critique. Aussi des fonctions de correspondance prenant en compte les liens hypertextes ont vu le jour. Ces dernières années, plusieurs méthodes de recherche d'information employant des informations sur la structure des liens ont été développées et se sont avérées en pratique plus efficace sur le web que les méthodes exploitant uniquement le contenu des pages web. En réalité, la plupart des fonctions de correspondance utilisées par les systèmes de recherche hypertextes combinent une mesure de pertinence calculée en fonction du contenu de la page et de la requête utilisateur avec une mesure de popularité de la page qui elle, est indépendante de la requête. Cette dernière mesure repose sur la structure du Web, considéré comme un graphe orienté de pages et de liens. Le PageRank (Brin *et al., 1998*) de Google et le HITS (Kleinberg, 1998) de Clever sont deux algorithmes fondamentaux utilisant la structure d'hyperlien entre les pages Web. Un certain nombre d'extension de ces deux algorithmes ont été proposés, comme (Lempel *et al.*, 2000) (Haveliwala, 2003) (Kamvar *et al.,* 2003) (Jeh *et al.,* 2003) (Deng *et al.,* 2004) et (Xue-Mei *et al.,* 2004). Tous ces algorithmes reposent sur la règle suivante : Si une page A cite une page B, alors c'est que la page A juge que la page B est suffisamment importante pour mériter d'être citée et d'être proposée aux visiteurs. Dans ces systèmes, on parle plus de la fonction de classement des résultats que de la fonction de correspondance car la fonction de classement ne dépend pas de la requête tandis que la fonction de correspondance relie la requête aux documents. L'étude des systèmes existants nous a permis de conclure que toutes les fonctions de correspondance utilisant les liens hypertextes ne dépendent pas des termes de la requête. Cela a diminué considérablement la précision des résultats retrouvés. Dans cet article, nous évoquons d'abord quelques méthodes d'utilisations de liens pour la recherche d'information. Ensuite, nous présentons notre système avec la nouvelle fonction de correspondance et nous décrivons les tests effectués. Nous concluons par l'analyse des résultats et des perspectives.

2. Utilisation des liens pour la RI

Différentes études ont suggéré de tenir compte des liens entre documents afin d'augmenter la qualité de la recherche d'information. Le PageRank (Brin *et al., 1998*) de Google et le HITS (Kleinberg, 1998) de Clever sont deux algorithmes fondamentaux utilisant la structure d'hyperlien entre les pages Web. Généralement, ces systèmes fonctionnent en deux temps. Dans une première étape, un moteur de recherche classique retrouve une liste de pages répondant au mieux à la requête posée, en fonction des termes de la requête et des termes d'indexation des

documents. Dans une seconde étape, ces systèmes tiennent compte des liens pour classer les documents. Dans ce qui suit, on présentera quelques-uns des systèmes de recherche d'information utilisant les liens hypertextes.

La mesure de PageRank (PR) proposée par L. Page et S.Brin (Brin *et al., 1998*) est une distribution de probabilité sur les pages. Elle mesure en effet la probabilité PR, pour un internaute navigant au hasard, d'atteindre une page donnée. Elle se base sur un concept très simple : un lien émis par une page A vers une page B est assimilé à un « vote » de A pour B. Plus une page reçoit de « votes », plus cette page est considérée comme importante. Le calcul du coefficient PR s'avère long et nécessite de balayer tout le Web. Afin de diminuer le temps de calcul du PR, l'idée de calculer un coefficient PR personnalisé a fait aboutir trois solutions radicalement différentes (Haveliwala et al., 2003)

– Le coefficient PR modulaire ("Modular PageRank")

– Le coefficient PR calculé par blocs ("BlockRank").

– Le coefficient PR sensible à la thématique ("Topic sensitive PageRank")

Chacune de ces trois approches calcule une certaine approximation du coefficient PR. Cependant, elles diffèrent considérablement dans leurs conditions de calcul et dans la granularité de la personnalisation réalisée.

Dans (Kleinberg, 1998), Kleinberg a présenté une procédure pour identifier les pages web qui sont des bons pivots (hubs) ou des bonnes autorités (authorities), en réponse à une requête utilisateur donnée. Il s'avère utile de distinguer les pages pivots des pages qui font autorité. Les premières correspondent aux pages possédant un nombre important de liens sortants (comme, par exemple, la page d'accueil de Yahoo !). Généralement, ces pages pivots sont des index. Tandis que les secondes correspondent aux pages beaucoup citées par d'autres pages. En bibliothéconomie, les documents qui font autorité (donc cités par de nombreux auteurs) correspondent souvent à une description de l'état de l'art, à un article fondamental ou à une méthodologie largement utilisée. L'hypothèse stimule : "*Une page qui pointe vers beaucoup de bonnes Autorités est un bon pivot, et une page pointée par beaucoup de bons pivots est une bonne Autorité*". Pour identifier ces bonnes pages pivots et autorités, l'algorithme (HITS) exploite la structure du Web vu comme un graphe orienté. Etant donné une requête Q, le procédé de Kleinberg construit d'abord un sous-graphe G constitué de pages contenant les termes de la requête, et calcule alors un poids pivot et un poids autorité de chaque nœud de G. L'algorithme HITS distingue deux types de liens (inter site et intra site). Les liens inter site établissent des relations entre des pages appartenant à des sites différents et pouvant être vus comme des liens de proximité sémantique entre les pages Web. Tandis que les liens intra site établissent des relations entre des pages d'un même site dont le premier but est de faciliter la navigation à l'intérieur d'un site (liens de navigation définissant la structure d'un site) Tous les liens intra site sont supprimés du graphe. On ne garde que les liens inter site.

Une extension de l'algorithme HITS (Ji-Rong *et al.,* 2004) a été proposée par un group de chercheur de Microsoft. Il repose sur un calcul des blocs sémantiques. L'idée de base de cet algorithme est de segmenter chaque page Web en multiples blocs sémantiques en utilisant un algorithme de segmentation de page (Vision-based Page Segmentation VIPS) qui détecte la structure du contenu sémantique dans la page Web (ex. couleur du fond, ligne, font size, paragraphe, etc). Les étapes de l'algorithme HITS sont exécutées au niveau bloc ainsi qu'au niveau page. On note BlockRank (BR) le poids pivot d'un bloc. Pour chaque page, le bloc avec un haut poids BR (BRMax) est sélectionné, puis combiné avec le poids pivot de la page de la façon suivante :

$$H(P) = \alpha \cdot rank_{PR}(P) + \beta \cdot rank_{PR\_BRMax}(P)$$

Ou H(P) le poids pivot de la page P, $rank_{PR}(P)$ le poids pivot de la page P calculé par l'algorithme HITS et $rank_{PR\_BRMax}(P)$ le poids pivot maximum d'un bloc de la page P. $\alpha et \beta$ sont deux facteur compris entre 0 et 1. Ces deux facteurs permettent de voir l'impact du poids pivot d'un bloc sur le poids pivot de la page.

Un algorithme alternatif, SALSA, est proposé par Lempel & Moran (Lempel *et al.*, 2000) qui combine les deux idées de HITS et PR. Comme dans le cas de HITS, le graphe du Web est visualisé comme un graphe biparti où les pages centrales pointent celle des autorités. Dans la méthode SALSA, l'algorithme est appliqué pour identifier des structures dans les pages pivots et autorités détectées par HITS. Cette méthode permet de filtrer les pages pivots et autorités pour ne garder que les plus pertinentes. Ensuite intervient l'approche stochastique. Cette approche n'est autre que celle des chaînes de Markov ou « promeneur aléatoire » qui constitue les bases théoriques du coefficient PR de Google. La formule de SALSA est définie comme suit :

$$A_i = \sum_{j \in B(i)} \frac{1}{|F(j)|} H_j \qquad et \qquad H_i = \sum_{j \in F(i)} \frac{1}{|B(i)|} A_j$$

3. Synthèse

Dans les approches présentées, L'utilisation des liens sert à classer les pages résultantes par rapport à une requête utilisateur. Le point commun entre ces méthodes est que le classement s'effectue indépendamment des termes de la requête. On peut classer ces méthodes en deux catégories : les méthodes utilisant les liens lors de la phase d'indexation (PR) et les méthodes utilisant les liens lors de la phase d'interrogation (HITS). Dans le premier cas, les calculs se font « off line » au moment de l'indexation. Tandis que dans le deuxième cas, les calculs se font « on

line » en réduisant le graphe du web en un sous graphe contenant les pages résultantes d'une requête. De nombreuses expériences ont montré qu'il n'y a pas de gain significatif par rapport aux méthodes de recherche reposant seulement sur le contenu. Intuitivement, les résultats doivent être classés par rapport à la requête utilisateur. Dans la suite, nous détaillons notre système et notre fonction de correspondance.

4. Modélisation de la fonction de correspondance

Notre système est composé de deux modules : le module d'indexation qui représente les documents et le module d'interrogation qui représente les requêtes. La fonction de correspondance qui permet de calculer le degré d'appariement entre les termes de la requête et les termes d'indexation des documents afin d'évaluer la pertinence des documents par rapport à la requête repose sur le contenu et la popularité de la page par rapport au termes de la requête. Nous allons décrire cette fonction ultérieurement. Tout d'abord, nous commençons par le module d'indexation.

4.1. *Module d'indexation*

La manière la plus simple pour construire un index est d'identifier tous les mots uniques dans le document entier. Puis d'exclure certain mots très fréquents (i.e, mots vides) non significatifs. Les différentes variantes du mot et de ses dérivés sont considérées comme des termes différents (par exemple informer, information). Beaucoup de SRI adoptent une stratégie appelée lemmatisation (stemming) qui opère par réduction des mots en une entité première (lemme), appelée aussi forme canonique. Par exemple le lemme de "cheval" et celui de "chevaux" sont les mêmes. Il y a plusieurs variétés d'algorithmes de lemmatisation reposant sur la suppression des suffixes ou l'utilisation d'un dictionnaire. Les deux algorithmes de lemmatisation les plus connus reposent sur la suppression des suffixes [(Lovins, 1968), (Porter, 1980)]. Nous avons opté pour l'algorithme de *Porter* (Porter, 1980) comme algorithme de lemmatisation. Ce choix est dû à son utilisation par plusieurs moteurs de recherche actuels et de sa popularité dans la communauté web. De plus on a utilisé une liste des mots vides pour éliminer certains mots très fréquents et non significatifs. Notre choix du modèle de représentation des documents et requêtes s'est porté sur le modèle vectoriel (Salton *et al.,* 1975). Ce choix est motivé par son succès dans la communauté web et les résultats très satisfaisants qu'il engendre. Nous définissons ici notre modèle de recherche d'information. Par définition, un modèle de recherche d'information consiste en une représentation des documents et des requêtes, ainsi qu'une fonction de correspondance. Nous présentons ces éléments du modèle en nous focalisant notamment sur le nombre de dimensions de l'espace vectoriel et la pondération.

4.2. *Représentation des documents et des requêtes*

La dimension de notre espace vectoriel est égale au nombre de termes distincts dans toute la collection. Chaque document et chaque requête sont représentés par un vecteur de poids des termes. Pour calculer ces poids, nous avons choisi la combinaison des pondérations locales et globales de la manière suivante :

$$W(t,d)=wl(t,d)*wg(t)$$

Où ***W (t,d)*** représente le poids du terme ***t*** qui apparaît dans un document ***d***, **w*l(d,t)*** est la pondération locale du terme ***t*** dans le document ***d*** et ***wg (t)*** la pondération globale du terme ***t*** . Les pondérations locale et globale du terme t sont définies comme suit :

Pondération locale :

$$Wl\ (d,ti)=0.5+0.5*tf\ /\ max\ (tf)$$

Pondération globale :

$$Wg\ (ti)=idf\ (ti) = log\ (|D|/df\ (ti))$$

Où ***tf*** est la fréquence d'occurrence du terme t dans le document (term frequency), |***D***| représente le nombre de documents de la collection et ***df (ti)*** le nombre de documents de la collection qui contiennent le terme ***ti*** (le terme considéré).

4.3. *Fonction de correspondance*

La nouveauté dans notre modèle est l'utilisation d'une fonction de correspondance qui dépend en plus du contenu textuel des pages, de la popularité de celle-ci par rapport aux termes de la requête. Cette dépendance permet une meilleure adéquation des résultats retrouvés par un modèle classique de RI avec les besoins utilisateurs. Notre fonction de correspondance repose sur deux mesures : l'une est classique et utilisée dans les systèmes actuels. C'est la mesure cosinus qui calcule le cosinus de l'angle entrant entre le vecteur représentant la requête et celui représentant le document. Cette mesure est définie comme suit :

$$Sim_{\cos ine}\left(\overrightarrow{D},\overrightarrow{Q}\right)=\frac{\overrightarrow{D}\cdot\overrightarrow{Q}}{\left\|\overrightarrow{D}\right\|^2\cdot\left\|\overrightarrow{Q}\right\|^2}=\frac{\sum_{t_i\in D\cap Q} w_{t_i,D}\cdot w_{t_i,Q}}{\sqrt{\sum_{t_i\in D} w_{t_i,D}^2\cdot\sum_{t_i} w_{t_i,Q}^2}}$$

Où $W_{t,D}$ et $W_{t,Q}$ représentent respectivement les poids du terme t dans le document D et dans la requête Q.

La deuxième mesure est celle qui tient compte de la structure du Web composée des liens hypertextes. Afin de comprendre notre démarche, nous partons de l'hypothèse suivante : *on considère qu'une page est bien connue pour un terme t si celle-ci contient des liens entrants et/ou sortants qui possèdent le terme t dans leur texte ancre ou autour de l'ancre*. Cette mesure peut être calculée de deux façons différentes.

Supposons un document D (ou page) retrouvé par un système classique de RI et Q{t} l'ensemble des termes de la requête utilisateur Q. Soit I(D) l'ensemble des documents citant le document D, C(t) l'ensemble des documents contenant les termes de la requête Q et soit IC(D,t) l'ensemble des documents contenant des termes de Q et citant le document D. Notre fonction structurelle est définie de la manière suivante

La première mesure correspond à la fraction du nombre de documents citant D et contenant les termes de la requête Q par rapport au nombre de documents citant D. Cette mesure favorise les documents qui ont peu de liens entrants.

$$Sim_{struct1}\left(\vec{D},\vec{Q}\right)=\log\left(1+\frac{\left|IC(D,t)\right|}{\left|I(D)\right|}\right)$$

Tandis que la seconde mesure correspond au nombre de documents citant D et contenant les termes de la requête Q par rapport au nombre de documents contenant les termes de la requête Q. Cette deuxième mesure favorise les documents dont les termes sont peu fréquents dans toute la collection.

$$Sim_{struct2}\left(\vec{D},\vec{Q}\right)=\log\left(1+\frac{\left|IC(D,t)\right|}{\left|C(t)\right|}\right)$$

Plusieurs variantes de cette fonction peuvent être envisageables, par exemple en ne tenant compte que des liens entrants, des liens sortants ou les deux ensemble, ainsi qu'une combinaison des deux mesures :

$$Sim_{struct}=Max\left(Sim_{struct1},Sim_{struct2}\right)$$

Dans notre système, Nous avons utilisé la mesure $Sim_{struct1}$ pour évaluer notre système. Nous l'avons utilisée pour les liens entrants et sortants. Voici donc notre fonction de correspondance entre document et requête. C'est une combinaison des deux mesures cosinus et structurelle :

$$Sim(\vec{D},\vec{Q}) = \alpha \cdot Sim_{\cos ine} + \beta \cdot Sim_{struct1}$$

Où α et β sont deux facteurs tel que α + β = 1. Ces deux facteurs nous permettent d'accorder de l'importance à l'une ou l'autre des mesures selon la structure du Web. Ils nous permettent aussi de comparer les résultats et de déterminer l'importance d'une mesure par rapport à l'autre. Dans notre évaluation ces deux facteurs sont fixés à 0.5. Nous pourrions tester notre système avec différentes valeurs de α et β.

Apres avoir décrit notre système qui intègre une nouvelle fonction de correspondance, nous allons maintenant passer à l'expérimentation et à l'évaluation de notre système.

5. Expérimentations sur la collection TREC-9 (WT10g)

Nous abordons la partie expérimentale de notre étude. Nous commençons par décrire d'une manière générale la collection de tests que nous avons utilisée dans nos expérimentations. Ensuite nous expliquons la façon dont nous avons exploité les données composant la collection pour les adapter à nos tests. Puis nous détaillons les tests effectués. Nous avons exécuté 50 requêtes sur trois systèmes différents selon le mode de sélection des documents répondant à la requête et comparé trois catégories d'algorithmes (contenu, liens et notre algorithme). Nous concluons avec une analyse des résultats.

5.1. *La collection WT10g*

Une collection de tests dans le cadre de la recherche d'information consiste traditionnellement dans un ensemble de documents, un ensemble de besoins d'information et un ensemble de jugements. Un jugement indique si un document donné est pertinent ou non à un besoin d'information donné. Dans le cadre de nos expérimentations, nous avons choisi comme collection de tests la collection WT10g issue du corpus de la conférence TREC-9 ayant eu lieu en 2000. Nous l'avons choisie en raison de la notoriété des collections issues de TREC et par conséquent, leur statut de collections standard dans le domaine de la recherche d'information. Dans (Bailey *et al.,* 2001), le processus de construction de la collection WT10g est décrit. D'une façon générale, la collection a été conçue dans l'objectif de modéliser une recherche réelle dans le Web et de permettre une évaluation fiable des méthodes

de recherche d'information basées sur l'analyse des liens. La collection est composée de 1692096 documents (pages Web) totalisant environ 11 gigaoctets de données. D'après (Bailey *et al.,* 2001), pour ce qui est de la structure du graphe sous-jacent à ces données. Il existe 1532012 pages avec des liens entrants et 1295841 pages avec des liens sortants. Ces liens entrants et sortants ne relient que les pages de la collection.

5.2. *Les tests effectués*

Dans cette section nous décrivons les tests que nous avons effectués. Nous avons testé notre système sur un ensemble significatif de sites présents dans la collection WT10g qui sont pertinents à au moins un topique de l'ensemble des topiques que nous avons utilisés. Avant de détailler notre méthodologie expérimentale nous rappelons quelques chiffres sur la collection utilisée. L'idée de départ consistait à tester la collection TREC. Cependant, pour une question d'espace mémoire et de calcul, nous avons sélectionné pour les tests les sites contenant au moins deux pages pertinentes à l'une des requêtes exécuté sur notre système. Avec cette limite sur le nombre de pages, nous sommes passés de 870 sites à 490. Notre collection de tests contient 546423 documents et 2544746 liens hypertextes, en moyenne 4.66 liens par pages. Nous avons comparé les fonctions de correspondance suivantes :

– ***Celles qui tiennent compte du contenu textuel de la page seulement :*** calcul des fréquences des termes (simple modèle vectoriel). La fonction de correspondance calcule le cosinus de l'angle entre le vecteur représentant le document et celui représentant la requête.

– ***Celles qui tiennent compte de la popularité de la page indépendamment de la requête***

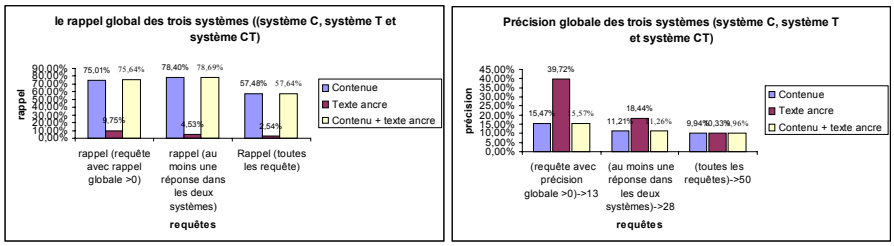
– ***Celles qui tiennent compte de la popularité de la page par rapport aux termes de la requête***.

5.3. *Evaluation et analyse des résultats*

Le processus d'évaluation est le suivant : nous avons exécuté 50 requêtes sur chaque système et nous avons calculé, pour chaque fonction de correspondance, la précision aux 11 niveaux standards du rappel : 0%, 10%, 20%,…,100%. En première étape, nous avons comparé trois systèmes de sélection de documents. Puis, nous avons évalué chaque fonction de correspondance sur les 50 requêtes exécutées dans notre système. L'analyse des résultats de nos expérimentations montre que les algorithmes reposant sur le texte ancre dans la sélection et le classement des résultats sont meilleurs par rapport à ceux qui ne le considèrent pas.

La phase de la sélection est très importante dans tout système de recherche d'information. Cette phase permet de déterminer quels sont les documents qui

répondent au mieux à une requête utilisateur. Nous avons testé trois systèmes en fonction de la manière de sélectionner les documents. On distingue trois catégories de systèmes réputés dans la littérature : ceux qui tiennent compte du contenu des documents (système C), du texte ancre des liens (système T) et les deux ensemble (système CT). Les résultats montrent qu'avec le texte ancre seulement, nous avons une grande précision. Nous avons calculée la précision globale moyenne pour les 50 requêtes et pour les 13 requêtes adaptées à notre système. Ces 13 requêtes contiennent au moins un terme présent dans le texte ancre d'un lien entre deux pages de notre sous-ensemble de la collection TREC. Ce choix des 13 requêtes est motivé par le fait que notre fonction de correspondance tient compte des termes issus du texte ancre des liens. En effet, la précision globale moyenne dans le système T est largement au dessous du système C avec une précision de 39,72% pour les 13 requêtes contre 15,45% pour le système C (10,33% contre 9,94% pour les 50 requête utilisée) (figure 1.b). Ces chiffres montre que l'existence des termes de la requête dans les liens joue un rôle important dans la sélection des documents par rapport à la requête utilisateur. Malheureusement, le rappel global moyen dans le système T est minime par rapport au rappel dans le système C avec 9,75% contre 75,01% pour les 13 requêtes décrites précédemment (2,54% contre 57,48% pour les 50 requêtes utilisées) (figure 1.a). Cette différence peut être justifiée par le nombre moyen de liens par page qui ne dépasse pas 5 et par le fait que pour certains types de requête que nous verrons dans la suite, les termes se trouvent plutôt dans le corps du document que dans le texte ancre des liens. Cette évaluation nous a amené à fusionner la liste des résultats obtenus par les deux systèmes. Nous avons remarqué que le rappel global moyen est passé de 75,01% à 75,64 %. De plus la précision est passée, elle aussi, de 15,47 à 15,57 malgré le bruit engendré par la combinaison des deux systèmes pour certaines requêtes.



(a) (b)

(

Figure 1. *Comparaison des systèmes (C, T, CT) : rappel global moyen et la précision globale moyenne*

Le but d'une telle comparaison des différents systèmes est de voir l'impact du texte ancre dans la sélection des documents pertinents. Nous avons opté pour la sélection selon le contenu et le texte ancre dans notre implémentation. Ce choix est motivé par son utilisation par les moteurs de recherche actuels. Dans la suite, nous

détaillons les résultats satisfaisants que nous avons obtenus en appliquant notre fonction de correspondance. Le tableau suivant présente un extrait des résultats obtenus pour les 50 requêtes exécutées sur notre système.

Topique	Les 11 niveaux du rappel standard					
	0%	**10%**	**20%**	**30%**	**40%**	**50 à 100%**
451	100,00%	100,00%	85,71%	88,88%	_83,33%	
452	100,00%	32,72%	0,00%	0,00%	0,00%	
453	100,00%	22,00%	6,75%	5,11%	43,43%	…
454	100,00%	46,87%	35,00%	31,73%	32,20%	
455	14,28%	19,04%	22,72%	21,95%	20%	
:						

Tableau 1. *Extrait de la table récapitulative des résultats obtenus des 50 requêtes exécutées*

La figure suivante présente un tableau comparatif des résultats obtenus selon les différentes fonctions de correspondance pour les 50 requêtes exécutées et pour les 13 requêtes dont la précision globale est supérieure à 0.

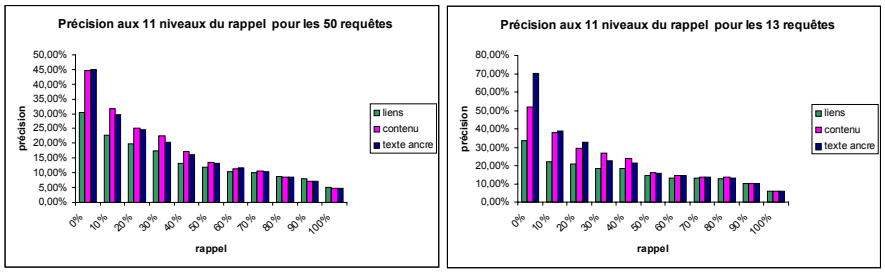
Les 11 niveaux du rappel	**50 requêtes**			**13 requêtes**		
	Contenu et texte ancre	***Contenu***	***Nombre de liens***	***Contenu et texte ancre***	***Contenu***	***Nombre de liens***
0%	44,88%	44,80%	30,53%	70,01%	52,06%	33,42%
10%	29,71%	31,65%	22,81%	38,57%	37,69%	22,10%
20%	24,58%	25,19%	19,75%	32,64%	29,50%	20,87%
30%	20,44%	22,60%	17,46%	22,66%	26,63%	18,19%
40%	16,14%	17,33%	13,22%	21,37%	23,64%	18,20%
50%	13,26%	13,49%	12,01%	15,59%	16,29%	14,45%
60%	11,53%	11,50%	10,30%	14,39%	14,51%	13,31%
70%	10,37%	10,47%	10,11%	13,49%	13,79%	13,05%
80%	8,34%	8,49%	8,75%	13,16%	13,51%	12,84%
90%	7,23%	7,25%	7,92%	10,36%	10,42%	10,21%
100%	4,80%	4,80%	5,09%	6,05%	6,06%	5,89%

Tableau 2. *La précision moyenne aux 11 niveaux standard du rappel pour les 50 requêtes exécutées et les 13 requêtes dont la précision globale >0*

La plupart des résultats obtenus avec notre formule pour le top du rappel sont au au-dessus des autres algorithmes. Notre fonction est au-dessous de la fonction

utilisant le contenu seulement avec une précision moyenne à 10% du rappel pour les 50 requêtes exécutées de 44,88% contre 44,80% pour la fonction basée sur le contenu seulement (29,71% contre 31,65% pour 20% du rappel). Les mauvaises performances ont été observées pour les fonctions qui tiennent compte du nombre de liens entrants et/ou sortants indépendamment de la requête avec 30,53% de précision moyenne (22,81 pour 20% du rappel) (voir tableau 2. et figure 2.a).

Cependant, pour les 13 requêtes décrites auparavant dont la précision globale est supérieure à 0, notre fonction est largement au-dessus des autres. Avec une précision moyenne à 10% du rappel de 70% contre 52% et 33% pour les fonctions reposant sur le contenu et le nombre des liens respectivement (Tableau 2. et Figure 2.b). Intuitivement, l'existence des termes de la requête dans le texte ancre des liens favorise notre système vu que notre formule tient compte du texte ancre des liens dans l'évaluation de la pertinence du document par rapport à la requête utilisateur.



(a) (b)

Figure 2. *La moyenne de la précision en fonction du rappel pour les requêtes exécutées*

En plus, les résultats obtenus par notre fonction de correspondance pour certains types requêtes sont meilleurs par rapport aux résultats obtenus par le modèle vectoriel. Un extrait des résultats des requêtes pour lesquelles notre système produit de bons résultats est illustré dans la figure suivante.

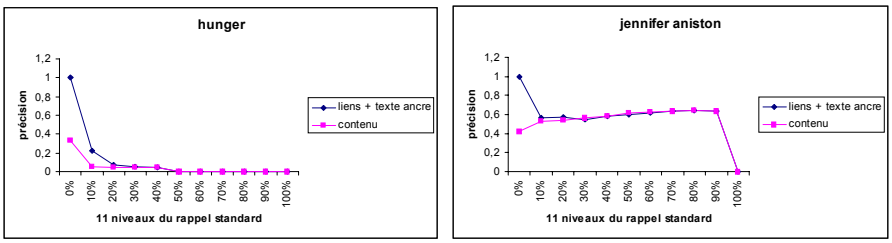


Figure 3. *Comparaison des requêtes : contenu vs liens +texte ancre*

Un extrait des mauvais résultats est illustré dans la Figure 4

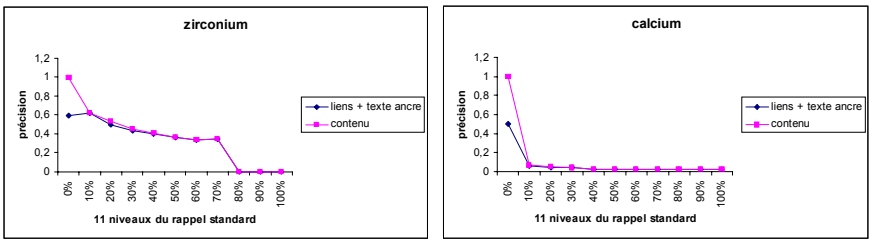


Figure 4. *Comparaison des requêtes : contenu vs liens +texte ancre*

Nous pensons que selon certains types de requête, notre système prenant compte des ancres des liens dans la fonction de correspondance peut s'avérer plus efficace. C'est le cas pour les requêtes « Jennifer Anniston », « hunger », « kappa alpha psi » et « peer gynt suite». Ce résultat peut s'expliquer pour :

✓ Les requêtes dont les termes sont des syntagmes. Un syntagme est un groupe de mots qui, ensemble, produisent un sens unique. Ce groupe de mots ne peut être fractionné en unités plus petites, ex. « kappa alpha psi » et « peer gynt suite». Un contre exemple, avec la requête « Mexican food culture», les résultats sont mauvais.

✓ Les requêtes dont les termes sont des noms propres. Ex «*Jennifer aniston », «steinbach nutcrackers» et «auto skoda».* Ces termes sont souvent employés ensemble pour décrire par exemple les pages personnelles et les pages d'accueil d'entreprise. En plus, la réputation d'une personne ou d'une entreprise produit de nouvelles pages dédiées à ces personnes ou ces entreprises, et de nouveaux liens sont créés entre ces pages.

Mais, les requêtes dont les termes se trouvent généralement dans le corps d'un document, par exemple les définitions, les dates et les événements (ex. *calcium*, *zirconium* et « *dna testing* ») donnent de moins bons résultats avec notre système. Dans ce cas, il est préférable d'utiliser le modèle vectoriel.

6. Conclusion et perspectives

Plusieurs travaux ont été menés sur l'utilisation des liens dans la recherche d'information sur le WEB mais, jusqu'à maintenant de nombreuses expériences ont montré qu'il n'y a pas de gain significatif par rapport aux méthodes de recherche basées seulement sur le contenu. Ce que nous avons proposé dans ce papier est un moteur de recherche utilisant à la fois le modèle vectoriel et les liens hypertextes. La nouveauté dans notre système est l'utilisation d'une fonction de correspondance qui tient compte de la popularité d'une page par rapport aux termes de la requête. Les

résultats obtenus montrent qu'ils sont meilleurs par rapport à ceux qui reposent sur le contenu seulement. De plus, la précision pour certaines requêtes est grande au top du top par rapport aux autres systèmes testés. Donc, le texte ancre peut être un facteur déterminant pour comprendre le contenu d'une page. Les expérimentations que nous avons menées avec quelques types de requêtes montrent que notre modèle pourrait s'avérer utile. Nous poursuivons actuellement un travail pour intégrer une autre mesure sémantique reposant sur les ontologies afin d'améliorer les performances de notre système. Nous allons appliquer cette fonction à des blocs sémantiques à définir et évaluer l'impact d'un tel procédé sur les résultats répondant à une requête utilisateur.

7. Bibliographie

Brin S., Page L., The anatomy of a large-scale hypertextual Web search engine, *In Proceeding of WWW7*, 1998.

Kleinberg J., Authoritative sources in a hyperlinked environment, *In Proceeding of 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

Lempel R., Moran S., The stochastic approach for link-structure analysis (SALSA) and the TKC Effect, *In Proceeding of 9th International World Wide Web Conference*, 2000.

Savoy J., Rasolof Y., Link-Based Retrieval and Distributed Collections, Report of the TREC-9 experiment: Proceedings TREC-9, 2000, p.579-588.

Salton G., Yang C.S., Yu C.T., A theory of term importance in automatic text analysis, *Journal of the American Society for Information Science and Technology*, 1975.

Haveliwala T., Kamvar S., Jeh, G., An Analytical Comparison of Approaches to Personalizing PageRank, *rapport technique*, université de Stanford, 2003.

Haveliwala T.., Topic-Sensitive PageRank : A Context-Sensitive Ranking Algorithm for Web Search, Knowledge and Data Engineering, *IEEE Transactions on*, 2003.

Kamvar S., Haveliwala T., Manning C., Golub G., Exploiting the Block Structure of the Web for Computing PageRank, 2003.

Deng C., Shipeng Y., Ji-Rong W., Wei-Ying M., Block-based Web Search, *Microsoft research ASIA*, 2004.

Xue-Mei J., Gui-Rong X., Wen G.S., Hua-Jun Z., Zheng C., Wei-Ying M., Exploiting PageRank at Different Block Level - *International Conference on Web Information Systems Engineering*, 2004.

Jeh G., Widom J., Scaling personalized web search, *In Proceedings of the Twelfth International World Wide Web Conference*, 2003.

Porter M.F., An algorithm for suffix stripping, 1980.

Ji-Rong W., Ruihua S., Deng C., Kailhua Z., Shipeng Y., Shaozhi Y., Wei-Ying M., At the web track of TREC 2003, *Microsoft research ASIA*, 2004.

Lovins J.B., Development of a stemming algorithm, *Translation and Computational Linguistics*, 1968.

Bailey P., Craswell N., Hawking D., Engineering a multi-purpose test collection for web retrieval experiments, Information Processing and Management, 2001.