UTILISATION DES SIMILARITES STRUCTURELLES POUR L'EVALUATION DE LA PERTINENCE EN RECHERCHE D'INFORMATION

Yaël Champclaux (*), Taoufiq Dkaki (*), Josiane Mothe (*)
champ@irit.fr, dkaki@irit.fr, mothe@irit.fr

(*)IRIT, Université Paul Sabatier, 118 Route de Narbonne Toulouse, FRANCE

Mots clés :
Recherche d'information, similarité structurelle, graphes.
Keywords:
Information Retrieval, structural similarity, graphs.
Palabras clave :
Búsqueda de información, semejanza estructural, gráficos.

Résumé

Notre travail se situe dans le domaine de la recherche d'information (RI), un système de recherche d'information vise à restituer les documents supposés pertinents pour l'utilisateur par rapport à sa requête. Le choix des documents à restituer est basé sur la similarité entre les documents et la requête. La notion de similarité et par conséquence la construction de la mesure qui permet de l'apprécier est donc un point central d'un SRI. Les approches traditionnelles de comparaison document / requête abordent le problème de la comparaison sous l'angle des similarités de surfaces. L'approche originale présentée dans cet article consiste à comparer les documents et la requête sur la base de leur système de relation. Il s'agit d'exploiter l'information représentée par les liens entre termes et documents et ceux qu'ils induisent de manière interne entre les documents et entre les termes.

Notre article fait suite à [3] où les auteurs motivés par la fouille de l'espace des propriétés des graphes, proposent une méthode générale pour mesurer la similarité structurelle dans un graphe et de ce fait étudier la ressemblance entre les nœuds qui composent ce graphe. Comme nous le verrons plus loin, les informations gérées par un SRI peuvent être modélisée par un graphe. L'objet de cet article est d'adapter la méthode proposée par [3] pour évaluer sa pertinence dans le domaine de la RI et proposer un modèle de recherche d'information basé sur les graphes. Notre postulat est qu'une méthode utilisant les similarités structurelles améliorera les performances des SRI par rapport à l'utilisation des seules similarités de surface.

Abstract

Within the framework of information retrieval, one interest is to select relevant documents related to a user request among a predefined collection of document. In this context, defining a “similarity measure” between documents and users’ requests is a central issue that shapes the core component of information retrieval systems (IRS) namely the comparison engine. This measure also highly influences the performance of IRS. Traditional approaches for document/request comparison use surface similarity, i.e. the comparator engine uses surface attributes to compare two elements (common words); the original approach presented in this article consists in comparing document and request on the basis of their system of relation. This approach exploits the information conveyed by the relationships between terms and documents, those between terms and these between documents. Our article follows upon [3] where the authors motivated by mining the space of graph properties, propose a general method for measuring the structural similarity of a graph and then for studying the resemblance between the nodes which compose this graph. We adapt this method, we propose a new graph based information retrieval model and evaluate the relevance of this proposal and of the approach it induces. Our postulate is that a method using structural similarities improves the performances of the IRS in comparison with traditional methods that use of surface similarities.

1 Introduction

De nombreuses applications nécessitent l'utilisation d'une mesure de similarité parmi lesquelles la tâche de déterminer quels sont les documents similaires à une requête utilisateur que se soit sur un corpus traditionnel ou bien sur le web. Déterminer si oui ou non une information (un document) correspond aux attentes d'un utilisateur est une tâche complexe. L'utilisateur exprimant son besoin d'information sous forme d'une requête à un système d'information (SRI) gérant une base de documents, la question centrale est : comment détermine t'il si un document et une requête sont suffisamment similaires ?

La similarité est un concept difficile à définir dans le cadre de la RI, comme il l'est dans tous les domaines relatifs à la cognition, tels que la classification, la catégorisation, la généralisation [4,6], ou encore en psychologie cognitive, dans le raisonnement à partir d'exemple, dans le processus de découverte [9] où elle intervient comme une contrainte. Dans le cadre d'un SRI, la similarité est une notion dépendante du modèle de recherche associé au système : pour générer la liste des documents restitués à l'utilisateur par rapport à sa requête, le SRI procède à la comparaison de la représentation de la requête et de celles des documents. C'est dans le modèle que l'on définit la méthode de représentation ainsi que la méthode de comparaison des représentations. Notre modèle à base de graphe appartient à la famille des modèles à espace vectoriel dans lequel un document est représenté comme un vecteur dans l'espace des termes. Chaque coordonnée d'un vecteur désigne l'importance d'un terme dans le document ou la requête que le vecteur représente. L'espace vectoriel est définit par l'ensemble de termes que le système a rencontré durant l'indexation. Pour mesurer la ressemblance entre un document et une requête dans cet espace on définit une mesure comme par exemple la mesure Cosinus, la mesure de Jaccard, le coefficient de Dice… de telles mesures déterminent la ressemblance entre un document et une requête sur la base de la comparaison locale des termes qu'ils ont en commun. Notre objectif est d'exploiter d'autres types de similarités dites structurelles. Ces similarités identifient les ressemblances au niveau des relations entre éléments [9]. La relation structurelle que nous exploitons ici est la contenance : les documents contiennent des mots, les mots sont contenus par des documents. L'idée est donc de comparer des documents entre eux au travers des ressemblances entre les mots qu'ils contiennent; les ressemblances entre les mots dépendant elles-mêmes des ressemblances entre les documents qui les contiennent. On peut remarquer le caractère itératif de cette approche. Nous exploitons une méthode générale de mesure de la similarité structurelle proposée dans [3]. Dans son adaptation à la RI, cette méthode présente l'avantage de propager les scores de similarité au travers des relations terme/terme, document/terme et document/document. Ce qui présente l'avantage de retrouver les documents possédant une relation indirecte avec la requête et donc de générer de l'information nouvelle sans utilisation de source extérieure. Cela est rendu possible par la structure de graphe qui selon [5,1] nous permet d'appliquer des algorithmes itératifs convergents à nos données et bénéficier ainsi des propriétés d'une telle structure.

2 La méthode initiale

La méthode de mesure de la similarité structurelle a été proposée dans [3]. Elle est définie comme suit :

Soit un graphe biparti composé de nœuds de type 1 reliés à des nœuds de type 2 ; c'est-à-dire que des nœuds de type 1 pointent vers des noeuds de type 2, ou bien des nœuds de type 2 sont pointés par des nœuds de type 1.

On s'intéresse aux ressemblances entre les nœuds de type 1 et réciproquement à ceux entre les nœuds de type 2, sont alors définis :

-un score de similarité s1 pour les nœuds de type 1. Deux objets de type 1 seront considérés comme similaires s'ils référencent (ou pointent vers) des nœuds de type 2 similaires.
-un score de similarité s2 pour les nœuds de type 2. Deux objets de type 2 seront similaires si ils sont référencés par des objets de type 1 similaires.

Ces notions peuvent être formalisées par les deux fonctions suivantes :
Soit un graphe G(*V* ;*E*),
O(v) est défini comme l'ensemble des successeurs (outputs) d'un nœud *v* et *I(v)* l'ensemble de ces prédécesseurs (inputs).
|*O(v)*| représente le cardinal de l'ensemble des successeurs et |*I(v)*| est le cardinal de l'ensemble des prédécesseurs.
s_i *(a,b)* correspond au Score SimRank entre deux objets *a* et *b* de type *i* .

$$s_1(a,b) = \frac{C_1}{|O(a)||O(b)|} \sum_{i \in O(a)} \sum_{j \in O(b)} s_2(O_i(a), O_j(b)) \qquad (1)$$

$$s_2(a,b) = \frac{C_2}{|I(a)||I(b)|} \sum_{i \in I(a)} \sum_{j \in I(b)} s_1(I_i(a), I_j(b)) \qquad (2)$$

On peut remarquer le caractère itératif de ces deux fonctions, en effet, la similarité des objets de type 1 est exprimée en fonction des similarités des objets de types 2. La similarité entre objets de type 2 étant elle-même définie en fonction des similarités de type1.
Une itération consiste à calculer la ressemblance entre deux objets *a* et *b* de type 1 grâce à la ressemblance initiale des objets de type 2 auxquels ils sont reliés, puis à calculer la nouvelle inter ressemblance des objets de type 2 grâce à l'inter ressemblance des objets de type 1 calculée en début d'itération.
A la 1ere itération, on aura la similarité entre deux objets de type 1 en fonction des objets de type 2 directement reliés, à l'itération suivante, on aura la similarité entre deux objets de type 1 en fonction des objets de type 2 directement reliés et aussi en fonction des objets de type 2 relié indirectement à un pas d'indirection. A chaque nouvelle itération, on étend l'indirection d'un pas supplémentaire tout en gardant égale à 1 la similarité d'un objet (document) à lui même.
Le Simrank est le score obtenu après convergence, une fois que tous les chemins ont été parcourus, elle est traditionnellement observée au bout de quelques itérations.

3 Transposition à la RI : Notre méthode

L'algorithme SimRank a pour intérêt de pouvoir rapprocher des objets en fonction de leurs ressemblances « directes » et aussi en fonction des ressemblances entre les relations qu'ils entretiennent entre eux [3].
En RI, l'objectif est de trouver parmi un ensemble de documents ceux et seulement ceux qui sont pertinents par rapport à une requête. La tâche d'un tel système est alors d'effectuer une comparaison document/requête afin de ne retenir que les plus ressemblants. On peut considérer que ce problème est une application particulière de la méthode décrite précédemment, prévue à l'origine pour identifier les ressemblances de chaque

document avec l'ensemble des autres documents de la collection (y compris la requête). Nous ne nous intéressons qu'à la ressemblance des documents à la requête.
Adapter cet algorithme à la RI revient à considérer deux types d'objets : les documents (la requête est considérée comme un document) et les termes qui les indexent; et à appliquer sur ces objets un calcul itératif qui doit permettre d'attribuer à chaque document un score de ressemblance avec la requête. Un score nul signifie que le document n'a aucun rapport direct avec la requête (aucun terme de la requête) ni aucun rapport indirect (aucun terme commun à un document qui posséderait un lien direct ou indirect avec la requête). Cet algorithme a donc pour propriété le fait de propager les scores de liens en liens de façon à retrouver des documents n'ayant pas forcément de rapport direct avec la requête.
Nous avons traduit les formules (1) et (2) de la façon suivante :

Soient d_i et d_j deux documents.
Soient t_i et t_j deux termes.

$|T_{di}|$ Le nombre de termes composant le document i.

$|T_{dj}|$ Le nombre de termes composant le document j.

$|D_{ti}|$ Le nombre de documents dans lesquels apparaît t_i.

$|D_{tj}|$ Le nombre de documents dans lesquels apparaît t_j.

C1 et *C2* sont deux coefficients de propagation dans [0,1].

$$s_d(d_1,d_2) = \frac{C_1}{|T_{d_1}||T_{d_2}|} \sum_{i \in T_{d1}} \sum_{i \in T_{d2}} s_t(t_i(d_1), t_j(d_2)) \qquad (3)$$

$$s_t(t_1,t_2) = \frac{C_2}{|D_{t_1}||D_{t_2}|} \sum_{i \in D_{t1}} \sum_{j \in D_{t2}} s_d(d_i(t_1), d_j(t_2)) \qquad (4)$$

Les formules traduisent le fait que la similarité de deux documents dépend de la similarité des termes qui les indexent; et, réciproquement la similarité de deux termes dépend de la similarité entre les documents dans lesquels ils apparaissent.

Une amélioration est néanmoins nécessaire : dans l'état actuel des formules (3) et (4), seule la présence ou l'absence de mots dans un document est considérée, or il faut prendre en compte la pondération des mots dans les documents. [8]

Pour ajouter la prise en compte de la pondération, les formules 3 et 4 sont définies comme suit :
Soit p(i,j) le poids du mot i dans le document j,

$$s_d(D_i, D_j) = C_1 * \frac{\sum_{l=1..T} \sum_{k=1..T} p(t_l, D_i).p(t_k, D_j).s(t_l, t_k)}{\sum_{l=1..T} p(t_l, D_i) \sum_{k=1..T} p(t_k, D_j)} \qquad (3')$$

$$s_t(t_i, t_j) = C_2 * \frac{\sum_{l=1..T} \sum_{k=1..T} p(D_l, t_i).p(D_k, t_j).s(D_l, D_k)}{\sum_{l=1..T} p(t_i, D_l) \sum_{k=1..T} p(t_j, D_k)} \qquad (4')$$

(3') et (4') peuvent être reformulées de manière matricielle comme suit:

Nous disposons de 3 structures de données :

-***M*** la self matrice des Mots de taille 1..M où ***M*** [i, j] contient s_t (t_i, t_j)

-***D*** la self matrice des Documents de taille 1..N où ***D*** [i, j] contient $s_d(d_i, d_j)$

-***W*** la matrice documents * termes de taille N * M où ***W*** [i, j] contient le poids du mot i dans le document j.

En prenant en compte ces structures, (3') et (4') s'écrivent :

$$s_d(D_i, D_j) = C_1 W M^t W^t \,/\, \sum_{l=1..m} W(l,i). \sum_{k=1..m} W(k, j) \qquad (5)$$

$$s_t(t_i, t_j) = C_2 W^t D W \,/\, \sum_{l=1..m} W(i,l). \sum_{k=1..m} W(j,k) \qquad (6)$$

4 Evaluation de la méthode

4.1 Collection de test

Nous avons utilisé le corpus CISI pour nos tests. En effet, comme on peut aisément le concevoir, la complexité les calculs dans (5) et (6), réduit le champ d'application de cette approche à des corpus documentaire de petite taille.
Ce corpus compte 1460 documents et 112 requêtes.
L'indexation des documents est basé sur les différentes étapes classiques en RI. Les termes sont extraits de chaque document. Les termes issus d'une liste de mots vides (mots outils de la langue comme les pronoms, les déterminants, etc.) sont supprimés. Nous avons utilisé une liste comportant 640 termes que nous avons déjà considérée dans le cadre de TREC[2]. Les termes restants sont radicalisés afin de limiter les variations

syntaxiques. Nous utilisons pour cette étape l'algorithme de Porter [7]. Enfin les termes sont pondérés par leur fréquence dans le document.
Les requêtes sont considérées et traitées comme des documents.
Après indexation, le corpus CISI compte 2215 termes pour 1460 documents.
Le corpus CISI compte de nombreuses requêtes qui n'ont pas ou peu de documents jugés réellement pertinents. Pour cette raison, nous avons limité notre étude à l'ensemble des requêtes ayant au moins 10 documents pertinents soit 67 requêtes.

4.2 4.2 Critères d'évaluation

Pour évaluer notre algorithme nous nous sommes basés sur :

- La précision moyenne qui est définie comme suit :
  Soient un ensemble de N documents et une requête Q,
  Soient N_r, l'ensemble des documents retournés, N_{Pert} l'ensemble des documents pertinents pour Q et $N_{non\ Pert}$ l'ensemble des documents non pertinents pour Q
  La précision p est définie par $p = \frac{\left|N_{Pert} \bigcap N_r\right|}{\left|N_r\right|}$
  Notons p(n) la précision à n documents retournés.
  $P_{moy} = \sum_{i=1}^{i=N} p(i) * R(i)$ *R(i)* = 1 si le i^{eme} document retourné est pertinent
  R(i) = 0 si le i^{eme} document retourné est non pertinent
- La Mesure F :
  Le rappel R étant définit par $R = \frac{\left|N_{Pert} \bigcap N_r\right|}{\left|N_{Pert}\right|}$
  La Mesure F est définie par : $F = \frac{2 \cdot P \cdot R}{P + R}$

Nous avons choisi de comparer notre mesure à la mesure cosinus suivante :

Soient un document $d = < t_1, t_2, t_3, \ldots, t_n >$; et une requête $q = < q_1, q_2, q_3, \ldots, q_n >$
Les t_i et les q_i représentent respectivement les poids des mots t dans *d*, et les poids des mots q_n dans *q*. Le cosinus entre le document *d* et la requête *q* est obtenu de la façon suivante :

$$\textit{Cos (d, q)} = \sum_i (t_i * q_i) / [\sum_i (t_i)^2 * \sum_i (q_i)^2]^{½}$$

Concernant la mesure décrite par (1) et (2), nous avons choisi dans un premier temps de l'utiliser telle que suggéré dans [2], c'est à dire en fixant les paramètres C1 et C2 (coefficient de propagation) à 0.8. Dans ces conditions la convergence de l'algorithme de calcul des similarités est atteinte au bout de la 10^{eme} itération.

Résultats

La méthode SimRank donne un résultat comparable à la méthode cosinus :

	SimRank	Cosinus	Gain %
Moyenne des precisions n	0,05044214	0,05303501	-5,14028548
Moyenne des precisions moyennes	0,06137202	0,06468538	-5,39880584
Mesure F	0,11204846	0,11691011	-4,33888159

Tableau comparatif du Simrank par rapport au Cosinus

Les précisons ainsi que les mesures F vérifiées sont du même ordre de grandeur que celle calculée pour le cosinus, tout en restant légèrement inférieures. Les conclusions immédiates à tirer sont que le SimRank ne fournit pas une mesure suffisamment efficace dans le cadre de la RI : elle est légèrement inférieure à la mesure cosinus, pour un temps de calcul très supérieur (le temps de calcul du cosinus correspond au temps d'une itération du SimRank, or il a fallu 10 itérations pour que l'algorithme converge vers les valeurs finales soit un coût temporel 10 fois supérieur) et une place mémoire également supérieure.
Sur le graphique suivant on peut voir l'évolution des moyennes des précisions à n documents retournés (n variant de 1 à 200) pour l'ensemble des requêtes traitées.

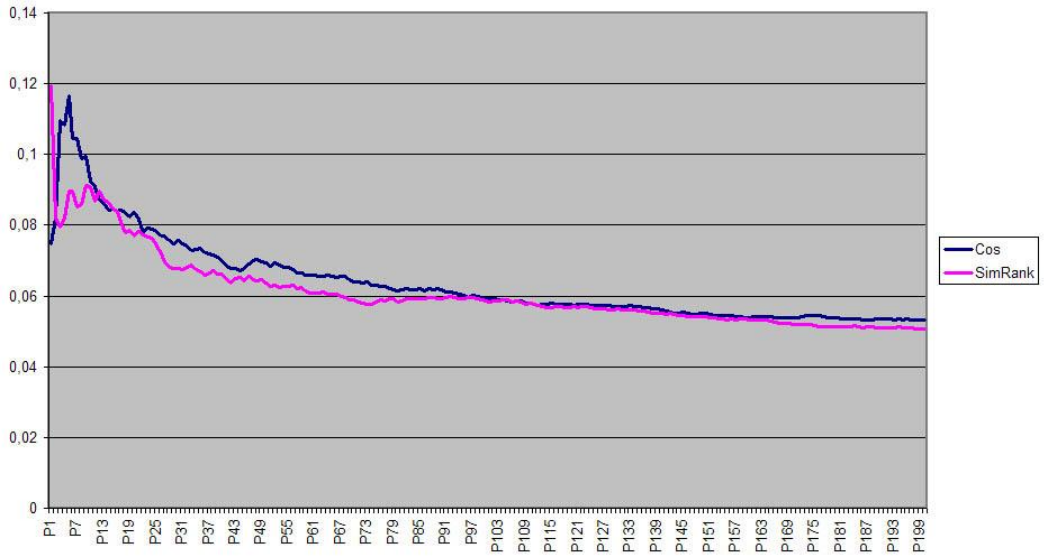

Figure 1 : évolution de la moyenne des précisions à n documents retournés (0 < n < 200)

Le graphique montre la grande ressemblance entre ces deux courbes, confirmant le fait que les mesures étudiées sont de même ordre. Néanmoins, les mesures moyennes sont le reflet du comportement général, en y regardant de plus prêt, pour certaines requêtes (un tiers), le SimRank majore le cosinus. Il semble donc présenter un intérêt que nous avons essayé de mettre en évidence :

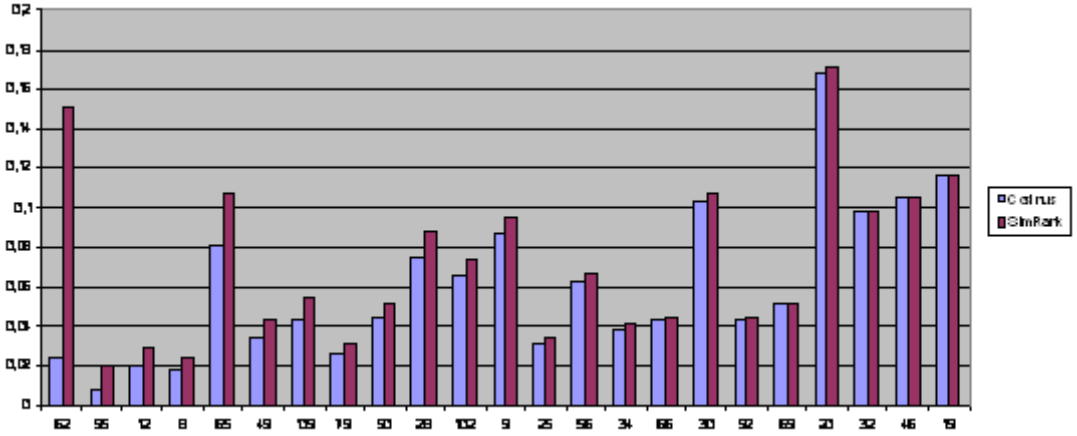

Figure 2 : Comparaison Cosinus/Simrank des moyennes des précisions n par requête

Ce graphique montre 23 requêtes (sur 67 traitées) pour lesquelles le SimRank majore le cosinus. Cela confirme que le SimRank semble avoir un effet positif dans certains cas.
Intéressons nous maintenant à la requête 62 pour laquelle notre méthode majore le cosinus de 600% :

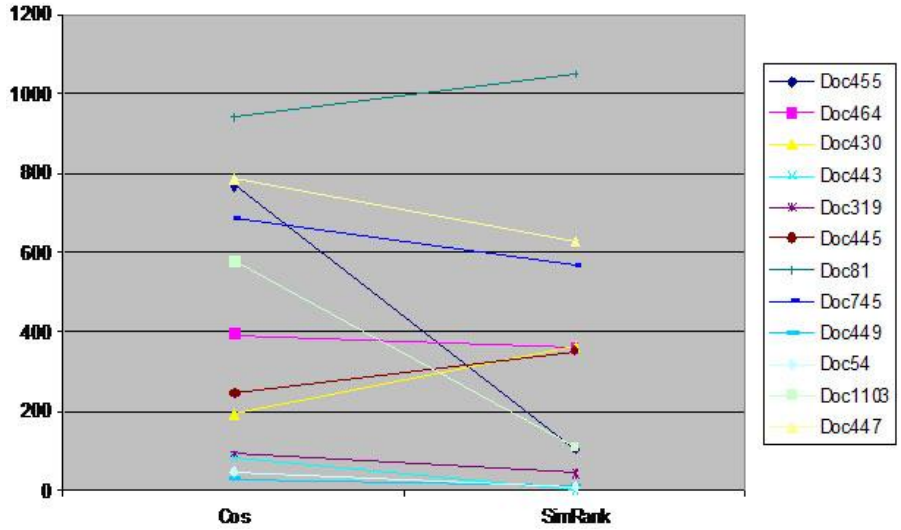

Figure 3: Comparaison Cosinus/Simrank des rangs des documents pertinents pour la requête 62

Ce graphique permet de visualiser la position des documents pertinents retrouvés et ainsi l'évolution de cette position d'une méthode à l'autre.
Commençons par étudier les documents pertinents retrouvés en tête de liste : les documents 449 et 54, retrouvés en 45eme et 31eme position par la méthode cosinus, progressent de quelques rangs grâce à la méthode SimRank (9eme et 12eme). De même les deux documents pertinents suivants 319 et 443 sont retrouvés en 90eme et 82eme position et se trouvent améliorés par le SimRank qui les classe en 43eme et 1ere position. Cela est dû au fait que ces documents ont un fort rapport direct avec la requête et une inter ressemblance importante qui augmente leur similarité à la requête.
Les deux documents pertinents suivants, 430, et 445, respectivement positionnés 191eme, 247eme se trouvent groupés autour de 350eme position par le SimRank, cela indique une ressemblance indirecte à des documents ne ressemblant pas à la requête, d'où leur léger éloignement de la tête de liste.
Le gain le plus marquant est obtenu par les documents 1103 et 455 positionnés 579eme et 768eme par le Cosinus qui se trouvent propulsé 110eme et 103eme par le SimRank.
Nous supposons que cela est du au faible cosinus de ces documents avec la requête et un rapport indirect fort avec celle-ci.
Sur 12 documents pertinents, 3 régressent et 9 progressent en rang.
Les documents ayant un cosinus élevé progressent légèrement avec le SimRank, c'est ce à quoi on pouvait s'attendre : en effet cette mesure indique une ressemblance à la fois directe (comme le cosinus) à laquelle une ressemblance indirecte est ajoutée, et il est probable que des documents se ressemblant fortement directement ont aussi des forts rapports indirects, ce qui se traduit dans le SimRank.

Concernant les deux documents à très forte progression (1103 et 455), nous sommes agréablement surpris de voir que les ressemblances indirectes peuvent ramener aussi significativement des documents devant des documents ayant au départ une plus grande ressemblance directe.
Considérons maintenant les trois documents pertinents qui régressent (430, 445, 81) : on peut remarquer que leur diminution de classement est faible et que le cosinus de ces trois documents est faible. Comme dit plus haut, cela traduit une relation de ses documents avec des documents ne ressemblant pas à la requête, l'impact de cette relation est modéré, mais néanmoins négatif car il fait régresser le classement de documents pertinents. C'est pourquoi nous n'envisageons pas d'utiliser notre mesure telle qu'elle, mais plutôt d'extraire la quantité d'information utile et de la combiner à une mesure déjà fonctionnelle.

5 Conclusions & Perspectives

Ces travaux montrent que le SimRank utilisé comme mesure de similarité permet d'obtenir des résultats comparables à ceux obtenus par la mesure cosinus tout en restant légèrement inférieurs.
Il semble donc que notre mesure " structurelle " ne soit pas optimale pour la RI. Néanmoins, les résultats obtenus sur CISI font entrevoir l'intérêt d'une mesure structurelle en RI. En fait, il est établi que le SimRank mesure la moyenne des ressemblances inter-objet directe et indirecte. Or le cosinus fourni déjà une mesure directe (c'est-à-dire concernant les documents ayant un rapport direct avec la requête) et cette mesure semble en moyenne plus performante que la nôtre qui prend en compte la similarité directe d'une part, et aussi la similarité indirecte. Nous pensons qu'il peut être utile dans un premier temps d'isoler, de quantifier et d'évaluer la quantité d'information indirecte apportée par le SimRank ; puis, dans un second temps, essayer de déterminer de quelle façon elle pourrait être utilisée pour exploiter l'information structurelle qui est disponible au départ et non exploitée dans un modèle de type vectoriel classique. Nous pensons à combiner notre mesure avec différentes mesures existantes. En effet, les tests effectués sur un ensemble de requêtes représentatives du corpus semblent indiquer qu'utiliser la propagation du SimRank après une initialisation cosinus améliore la mesure cosinus en moyenne.
Dans nos expériences futures, nous ferons varier les paramètres de notre algorithme (coefficient de propagation et nombre d'itération) afin de déterminer leur influence sur les résultats. Puis, dans un second temps nous tenterons d'isoler l'information indirecte retournée par notre méthode. Par la suite, nous étudierons de quelle façon combiner cette mesure structurelle avec d'autres mesures directes d'usage commun en RI. Nous essayerons de voir si il est possible d'utiliser d'autres relations structurelles. Nous utiliserons notre approche sur des corpus différents (TREC, Cranfield, INEX) pour déterminer l'influence de la taille, du domaine.
Cependant, dès à présent, on peut percevoir un intérêt de la méthode que nous proposons : celui de retourner des documents n'ayant aucun terme commun avec la requête mais considérés pertinents du fait de leur rapport indirect fort avec la requête. Cela peut être utile dans le cas de requêtes imprécises où l'utilisateur pourra trouver un document en utilisant des mots proches (susceptibles de se trouver dans les mêmes documents) des mots effectivement exprimés lors de l'interrogation d'un SRI de type vectoriel. Par exemple : Imaginons qu'un utilisateur souhaite retrouver un document parlant « d'un docteur ayant sauvé un enfant blessé en Europe» mais que le document en question contienne en réalité « un chirurgien ayant soigné une fillette en Belgique », dans un tel cas l'usage des rapports indirect est évident. En effet, si la base de documents est suffisamment grande, il est probable qu'il existera un document contenant les mots « docteur » et « chirurgien » ou bien « enfant » et « fillette » ou « encore « Belgique » et « Europe ». Cette méthode pourrait donc être utilisée comme une recherche du type « dernière chance » dans un moteur de recherche ayant déjà effectué un premier filtrage.

Bibliographie

[1] Vincent D. Blondel, Paul Van Dooren, *A measure of similarity between graph and vertice. With applications to synonym extraction and web searching*. Technical Report UCL 02-50, submitted to journal, 2002.

[2] T. Dkaki, J. Mothe, *'Trec Novelty Track At IRIT-SIG*', Text Retrieval Conference, 2004.

[3.] Glen Jeh and Jennifer Widom. *SimRank: A measure of structural-context similarity*. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 2002.

[4] Jones, S. S. et Smith, L. B. (1993). *The place of perception in children's concepts*. Cognitive Development, 8, 113-139.

[5] J. Kleinberg. *Authoritative sources in a hyperlinked environment*.Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997.W.E.

[6] Medin, D., Goldstone, R. L. et Gentner, D. (1993). *Respect for similarity*. Psychological Review, *2*, 254-278.

[7] Porter, M.F., "*An algorithm for suffix stripping*", Program, vol. 14, no 3, 1980, p. 130-137.

[8] K. Spärck Jones, *A statistical interpretation of term specificity and its application in retrieval*. Journal of Documentation 28, 11-21, 1972 and 60, 493-502, 2004

[9] Vosniadou, S. et Ortony, A. (1989). *Similarity and analogical reasonning: A synthesis*. *In* S. Vosniadou et A. Ortony (dir.), *Similarity and analogical reasoning* (p. 1-17). Cambridge: Cambridge University Press.