Information retrieval model based on graph comparison

Quoc-Dinh Truong[1,2,3], Taoufiq Dkaki[1,3], Josiane Mothe[1], Pierre-Jean Charrel[1,3]

[1]IRIT, Institut de Recherche en Informatique de Toulouse, 118 Route de Narbonne,
31062 Toulouse Cedex 9, France

[2]Cantho University, 1 Rue Ly Tu Trong, Cantho, Vietnam

[3]University Toulouse II, 5, Allées A. Machado 31058 Toulouse, France

Abstract

We propose a new method for Information Retrieval (IR) based on graph vertices comparison. The main goal of this method is to enhance the core IR-process of finding relevant documents in a collection of documents according to a user's needs. The method we propose is based on graph comparison and involves recursive computation of similarity. In the framework this approach, documents, queries and indexing terms are viewed as vertices of a bipartite graph where edges go from a document or a query – first node type- to an indexing term – second node type-. Edges reflect the link that exists between documents or queries on the one hand and indexing terms on the other hand. In our model, graph edge settings reflect the tf-ifd paradigm. The proposed similarity measure instantiates and extends this principle, stipulating that the resemblance of two items or objects can be computed using the similarities of the items to which they are related. Our method also takes into account the concept of similarity propagation over graph edges.

Experiments conducted using four small sized IR test collections (TREC 2004 Novelty Track, CISI, Cranfield & Medline) demonstrate the effectiveness of our approach and its feasibility as long as the graph size does not exceed a few thousand nodes. The experiment's results show that our method outperforms the vector-based model. Our method actually highly outperforms the vector-based cosine model, sometimes by more than doubling the precision, up to the top sixty returned documents. The computational complexity issue is resituated in the context of MAC-FAC approaches – many are called but few are chosen. More precisely, we suggest that our method can be successfully used as a FAC stage combined with a fast and computationally cheap method used as a MAC stage.

Keywords: graph, graph comparison, information retrieval.

1. Introduction & Related works

Searching for relevant information is a difficult task, and deciding whether or not a piece of information can fulfill a user's need is somewhat complex. In IR as in many other fields, especially those related to cognition (Gentner et al, 1993), (Medin et al, 1990), such as recognition, clustering and categorization, case-based reasoning and generalization, similarity acts as a key element.

This concept of similarity is difficult to circumvent and there is no universal similarity assessment that can be measured straightforwardly (Goodman, 1972): it is always needed to define in what regard two objects or items are similar. Any similarity measurement is, therefore, model – conceptual space and representation – dependent (Gärdenfors, 2004). Many authors define similarity on two levels: the surface level and the structural level. Surface similarity is defined as an attribute-related function while structural similarity is defined as a relation-oriented function. Surprisingly, several cognitive psychology studies (Bracke, 1998) suggest that structural similarity favors precision while surface similarity – as used in most IR models – favors recall. For that reason and because nowadays IR systems generally handle huge amounts of information and thus are more expected to perform well with respect to precision, we focus our interest on structural similarity.

Relations are the main features used in structural similarity computation. Since graphs are common representations that can capture structure and thus model a wide range of relational data and knowledge, we consider graph theory. Moreover, graph theory – sometimes, more precisely, graph comparison – already plays a major role in many specific information-related domains such as Web Information Retrieval (Henzinger, 2000; Sahami et al, 2004; Page et al, 1998), Text Information Retrieval (Gómez et al, 2000; Quintana et al, 1992; Siddiqui et al, 2005), Social Networks Analysis (Freeman, 1979; Newman, 2003) and science citation, and co-citation networks analysis (Jeh et Widom, 2002). Computing similarity based on graph structure has also been explored in the specific context of database schema matching (Melnik et al, 2002; Blondel et al, 2004) shows that the Web, as a citation graph, is structurally similar to the two-node graph (hub → authority) and expresses the approach of hub and authority analysis (Kleinberg, 1999) as a graph mapping issue. Basic approaches which employ graph models generally use immediate neighboring nodes of two vertices to compute their similarity, while more sophisticated approaches such as SimRank (Jeh et Widom, 2002) are based on the entire graph structure.

In this paper, we further investigate these aspects and propose a precision-driven method for Information Retrieval (IR). This method is based on graph vertices comparison. Consequently, it is inspired from previous works in graph matching in discreet mathematics, and similarity studies in cognitive psychology. The overall goal of this method is to enhance the core IR process of matching documents against queries in order to retrieve relevant information from a set of documents. Relevance is defined as an end-users' satisfaction measurement with respect to the needs they express in queries. In our approach, documents and queries are represented as nodes of a directed bipartite graph. In such a representation, graph vertices are either documents/queries (first type of nodes) or indexation terms (second type of nodes). Graph edges connect indexing terms to the documents and queries they represent. The resulting IR graph model facilitates the use of structural similarities in the process of matching documents and queries, which we consider as a graph vertices comparison.

The remainder of this paper is organized as follows: Section 2 presents the GVC information retrieval model based on graph comparison and discusses the proposed approach in light of those presented in (Kleinberg, 1999) and (Blondel et al, 2004). Section 3 presents primary tests and puts forward some improvements. Section 4 deals with implementation issues. Section 5 explains the experimental results. Finally, Section 6 provides some perspectives.

2. Graph vertices comparison

2.1. Overview

A text IR system is a computing tool which stores textual information (documents or chunks of text) and provides efficient means to retrieve them at request. It combines two major processes: the indexing process and the matching process. The main objective of the indexing process is to provide a representation of the contents of both documents and queries. The matching process is often based on a similarity measure used to compare users' queries to the indexed documents.

A traditional way to represent documents is to associate an m-dimensional vector to each document and to the query, m being the number for indexing terms. We use this representation as a starting point to build the initial graph. More precisely, we consider documents/query and indexation terms to be the vertices of a bipartite graph whose edges connect documents and queries to the indexation terms they contain. The adjacency matrix of this bipartite graph can be deduced from the documents-terms matrix, built during the indexing process (see figure 1).

The matching process, which is at the core of our concern and contribution, ranks the documents so that those most likely to be relevant (those with the higher similarity score with the query) are placed at the top of the retrieved document list. The matching process within the GVC model is based on the computation of the similarity scores between the graph vertices. Similarity scores are not locally computed, but rather take into account the whole graph structure.

The principle of our information retrieval system (IRS) is represented as in figure 1 at the right.

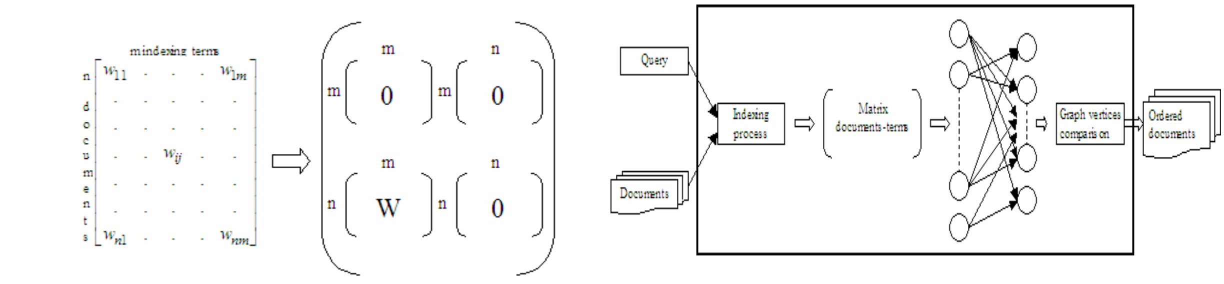


Figure 1: Building the bipartite graph starting from document-indexing term matrix. W=[wij] where wij represents the weight of the jth term in the ith document (left) and IRS based on graph comparison (right)

The next section addresses the theoretical basis of our proposal. This section introduces graph comparison in a way that goes beyond the solely IR considerations and that lies within the field of graph mining and analysis (Dkaki et al, 2006).

2.2. Background

Let G = (V, E) be a citation graph. In the context of hyperlinked web pages, V is a set of vertices representing web pages and E is a set of directed edges representing web hyperlinks. Let h_j and a_j be, respectively, the hub score and the authority score of vertex j (Kleinberg, 1999). The computation of hub and authority scores of vertex j is as follows:

$$h_j = \sum_{i(j,i)\in E} a_i \quad \text{and} \quad a_j = \sum_{i(i,j)\in E} h_i \qquad (1)$$

In our approach, the computation of hub and authority scores is seen as a graph comparison problem where the web, as a citation graph, is compared to the two-node-directed graph hub → authority. Indeed (1) can be expressed as follows:

$$\begin{bmatrix} h_{p_1} & a_{p_1} \\ \\ h_{p_n} & a_{p_n} \end{bmatrix}_{k+1} = B \begin{bmatrix} h_{p_1} & a_{p_1} \\ \\ h_{p_n} & a_{p_n} \end{bmatrix}_{k} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T + B^T \begin{bmatrix} h_{p_1} & a_{p_1} \\ \\ h_{p_n} & a_{p_n} \end{bmatrix}_{k} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad (2)$$

B is the adjacency matrix of the graph of the web. Matrix $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ is obviously the adjacency matrix of the two-node graph hub → authority. (2) is exactly the formulation of graph vertices comparison in (Blondel et al, 2004)

More generally, when considering two structurally similar graphs (same kinds of nodes and relationships), one to be analyzed (the target) and the other serving as the model (the source), we can map the first onto the second in a transfer-like approach (Bracke, 1998) by identifying a context-sensitive similarity measure between their sets of nodes. The similarity between two vertices i from the first graph and j from the second one is computed using the similarity scores between their related vertices (vertices pointing to i and j and vertices pointed by i and j in the two graphs). The similarity matrix S representing similarity scores between vertices of target graph and vertices of source graph can be expressed as follows

$$S_{k+1} = BS_k A^T + B^T S_k A \tag{3}$$

S_k and S_{k+1} are the similarity matrix at iteration k and k+1. B and A are the adjacency matrices of target graph and source graph.

(3) defines the similarity between nodes as a reflexive and recursive function. This triggers two fundamental questions: one related to the algorithm convergence and the other to the best initial choice for similarity values (S_0).

- Convergence

The convergence property of (3) is essential if we want to achieve the computation of similarity scores between vertices. Unfortunately, the convergence of (3) is uncertain. This convergence problem can be partially overcome by normalizing the similarity matrix S at each iteration step. (3) is then rewritten as follows:

$$S_{k+1} = \frac{BS_k A^T + B^T S_k A}{\left\| BS_k A^T + B^T S_k A \right\|_F} \tag{4}$$

The matrix norm we use is the square root of the sum of all squared entries (known as Euclidean or Frobenius norm). In this case, the series convergence is not entirely assured but at least, whatever the initial similarity values, the sequence admits two adherence values: one as a limit for S_{2k} series and the other as a limit for S_{2k+1} series (see (Blondel et al, 2004) for proof). The limit of sub-series S_{2k} can be used as the similarity matrix between vertices of source and target graphs.

- Initialization

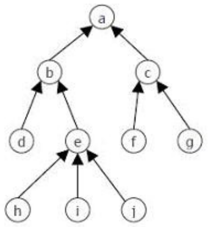
There are two possible ways to choose an initial similarity matrix S_0. They are related to the awareness of a priori resemblance between vertices of the two graphs. If there is no previously known information, then it seems natural that all node pairs must be associated to the same score of similarity (e.g. 1). Thus S_0 is a matrix full of ones and – the chance of being similar is equal for all pairs of vertices.

3. Model

3.1. Preliminary test

Initial tests using the above approach present roughly good results. The method provides a satisfactory mapping of target graph onto source graph. However, in specific cases, the results we obtain, although they are fully explainable (see figure 2), seem somewhat unnatural. This calls for an improvement of the method.

In the following example, we compare the graph represented in figure 2 to the reference graph hub→authority. The initial similarity matrix is the matrix full of ones. The obtained results are unsatisfactory. Indeed, upon examining the graph in figure 2, we easily apprehend the role of each vertex of this graph: vertices *d, f, g, h, i, j* assume the role of hub while vertices *b, c, e* take on the dual role of hub and authority, and vertex *a* adopts an authority role. When examining table in figure 2 which shows the similarity matrix result, we notice that vertices a, d, f, g, h, i and j obtain the same authority score.



Vertex	a	b	c	d	e	f	g	h	i	j
Hub score	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.2
Authority score	0.0	0.0	0.0	0.0	0.866	0.0	0.0	0.0	0.0	0.0

Figure 2: A graph used for preliminary test purposes and Results from the comparison with the two-nodes graph hub→authority

3.2. Model enhancement

There exist several reasons that can explain the unrealistic results obtained above. Perhaps the most important one is that our method does not take into account the notion of similarity inheritance (flooding similarity within a graph). When considering a feature such as authority, the inheritance must somehow play a role in similarity evaluation over a graph's set of nodes. In other words, relation to an authoritative node is likely to confer some kind of authority. (3) must be modified to comply with the similarity inheritance principle. The propagation and the retro-propagation of similarities within a graph are likened to a "flooding" similarity (Melnik et al, 2002) which can be expressed as follows:

$$S_{k+1}(i,j) = S_k(i,j) + \sum_{a:(a,i)\in A, b:(b,j)\in B} S_k(a,b)w((a,b),(i,j)) + \sum_{a:(i,a)\in A, b:(j,b)\in B} S_k(a,b)w((a,b),(i,j)) \qquad (5)$$

In our approach, similarity propagation can be considered as a graph transitive-closure of target and/or source graphs in a more sophisticated approach than the one in (Melnik et al, 2002). Indeed, we provide a more wide-ranging function of gradual attenuation of inheritance "over generations". Note that in (Melnik et al, 2002) this attenuation is solely depth-dependent. In our approach, adjacency matrices A and B of source and target graphs are modified to take into account a depth-attenuated flooding similarity. Modifications are expressed as follows:

$$A \leftarrow A + \sum_{n=2}^{\infty} f_A(n) g_A\left(\frac{A^n}{\|A^n\|}\right) \qquad B \leftarrow B + \sum_{n=2}^{\infty} f_B(n) g_B\left(\frac{B^n}{\|B^n\|}\right) \qquad (6)$$

$f_A(n) = \alpha^n$ and $f_B(n) = \beta^n$ where α and β are positive constants lower than 1, appear to be good instantiation of attenuation functions. These two functions establish the fact that the influence of 'generations' exponentially decreases with path depth (A^n and B^n are matrices whose respective entries a^n_{ij} and b^n_{ij} equal the number of paths of length n from vertex i to vertex j).

g_A and g_B are monotonically increasing functions from [0, 1] onto [0,1]. They can be exponential or stair functions.

The use of (6) overcomes the disadvantages depicted above (table in figure 2) as shown in the following results (table 1).

Vertex	a	b	c	d	e	f	g	h	i	j
Hub score	0.0	0.12	0.12	0.15	0.15	0.05	0.05	0.27	0.27	0.27
Authority score	0.36	0.37	0.09	0.0	0.65	0.0	0.0	0.0	0.0	0.0

Table 1: Enhanced results from the comparison of graph in figure 2 with the graph hub→authority

3.3. Graph self comparison

The core IR process is the retrieval of relevant information in a set of documents. Relevance is defined as a measurement of document concordance with the user's needs expressed in a query. As we mentioned in section 2, documents, queries and indexing terms are viewed as vertices of a bipartite graph where edges represent the relationship that exists between the documents/query on the one hand and indexing terms on the other hand. Using graph comparison for IR assumes that we look for document vertices that are similar to a query node. This is a search for similar nodes inside the same graph which implies a graph self-comparison. This graph vertices self comparison can be achieved by assuming that target and source graphs are the same. When experimenting with our first proposals (6) in this context of graph self-comparison, some cases show situations where the similarity of a given node to itself can be less than its similarity to another node: s(i, j) > s(i, i) (see figure 3). This opposes a condition that a similarity measure must fulfill. Full satisfactory measures must satisfy the following properties:

- $(i, j), s(i, j) \geq 0$
- $(i, j), s(i, j) = s(j, i)$
- $(i, j), s(i, i) = s(j, j) \geq s(i, j)$

The measure we obtain when applying (4) and (6) satisfies the two first properties, but not the third.

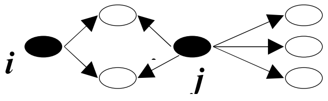


Figure 3: A test graph for self-comparison issue: the analysis of this graph shows that s(i, j) ≥ s(i, i).

Satisfying this condition is, of course, not mandatory since very common "similarity" measures in IR, such as the dot product, does not fulfill this prerequisite. Nevertheless, to avoid this weakness, we normalized similarity matrix S_{AB} by dividing each value $S_{AB}(i, j)$ by the product of self-similarity $S_{AA}(i, i)$ of vertex i in graph A and $S_{BB}(j, j)$ of vertex j in graph B. The final algorithm for graph vertices comparison (the one we use in the case of graph self-comparison) is described in paragraph 3.4.

3.4. Algorithm

Our final proposal for general graph vertices comparison is described in the algorithm below. This algorithm compares graph vertices from two graphs. It will be later rewritten and refined for the purpose of IR which involves a single sparse graph. As we argued above, this iterative algorithm converges to the similarity matrix S_{AB} between vertices of graph A and those of graph B.

```
$S_0 \leftarrow 1 \quad k \leftarrow 0$
$A \leftarrow A + \sum_{n=2}^{\infty} f_2(n) g_2(\frac{A^n}{\|A^n\|})$   ;   $B \leftarrow B + \sum_{n=2}^{\infty} f_1(n) g_1(\frac{B^n}{\|B^n\|})$
Repeat
    $S_{AA_{k+1}} \leftarrow \frac{AS_{AA_k}A^T + A^T S_{AA_k} A}{\|AS_{AA_k}A^T + A^T S_{AA_k} A\|_F}$   $S_{BB_{k+1}} \leftarrow \frac{BS_{BB_k}B^T + B^T S_{BB_k} B}{\|BS_{BB_k}B^T + B^T S_{BB_k} B\|_F}$   $S_{AB_{k+1}} \leftarrow \frac{BS_{AB_k}A^T + B^T S_{AB_k} A}{\|BS_{AB_k}A^T + B^T S_{AB_k} A\|_F}$
    $k \leftarrow k+1$
Until convergence is achieved for k even
$S_{AB} \leftarrow \bullet \frac{S_{AB} \bullet * S_{AB}}{diag(S_{AA}) \bullet * diag(S_{BB})^T}$
Output   $S_k$ ($k$ is even) as similarity matrix
```

Figure 4: Algorithm for two graphs similarity matrix computation. • and $\overset{\bullet}{\text{—}}$ are term-to-term matrix multiplication and division*

4. Implementation

In this section, we give a brief description of how we implemented our graph comparison algorithm for IR purposes as a special instance of the algorithm in figure 4.

- First, we construct a directed bipartite graph representing documents and query as described in section 2. Edges go from document/query nodes to term nodes and are weighted following a tf-idf approach. An English common stop words list is used to filter the set of terms. Porter's algorithm is used, for stemming purposes, to remove the more common morphological endings from selected indexing terms.

- Secondly, we compute initial values for the similarity scores between document and query nodes, and between term nodes (matrix S_0). We choose the cosine function to set the initial values of similarity scores. S_0 is then a (m+n)*(m+n) matrix where m is the total number of terms and n is the total number of documents (this includes the query which is seen as a document). Let G be the (m+n)*(m+n) adjacency matrix of the bipartite graph representing the sets of documents/query and indexing terms, the initial value of similarity score between node i and j is computed as follows:

$$\begin{cases} S_0(i,j) = \dfrac{\sum_{k=1\to n+m} G(i,k)*G(j,k)}{\sqrt{\sum_{k=1\to n+m} G(i,k)*G(i,k)}*\sqrt{\sum_{k=1\to n+m} G(j,k)*G(j,k)}} \\ S_0(i,i) = 1 \end{cases}$$

S_0 can be written as $\begin{bmatrix} S_T & 0 \\ 0 & S_D \end{bmatrix}$ *(6a)* where S_T is the m*m term similarity matrix and S_D is the n*n document/query similarity matrix.

- At each iteration of the similarity computing process, the similarity matrix S is updated as described in figure 4, with A=B=G. In this context of IR, we deal with large but sparse graphs. We take advantage of property (6a) to simplify our algorithm in order to reduce the size of the matrices, thus decreasing the memory load and increasing the computing speed. Using the algorithm of figure 4, we can write:

$$S_{2k+2} = \frac{\begin{bmatrix} W^T W S_{T_{2k}} W^T W & 0 \\ 0 & W W^T S_{D_{2k}} W W^T \end{bmatrix}}{\sqrt{\left\|W^T W S_{T_{2k}} W^T W\right\|_F^2 + \left\|W W^T S_{D_{2k}} W W^T\right\|_F^2}} \tag{7}$$

We can then rewrite the graph vertices comparison algorithm in figure 4 as follows:

```
$S_0 \leftarrow \begin{bmatrix} S_{T_0} & \\ & S_{D_0} \end{bmatrix}$
$k \leftarrow 0$
$G \leftarrow G + \sum_{n=2}^{\infty} f_2(n) g_2(\frac{G^n}{\|G^n\|})$
Repeat
    $S_{k+1} = \frac{\begin{bmatrix} W^T W S_{T_k} W^T W & 0 \\ 0 & W W^T S_{D_k} W W^T \end{bmatrix}}{\sqrt{\|W^T W S_{T_k} W^T W\|_F^2 + \|W W^T S_{D_k} W W^T\|_F^2}}$
    $k \leftarrow k+1$
Until convergence isachieved
$S_k \leftarrow \bullet \frac{S_k \bullet * S_k}{diag(S_k) \bullet * diag(S_k)^T}$
Output    $S_k$ as similarity matrix
```

Figure 5: Graph vertices comparison algorithm for Information Retrieval System

This algorithm converges for the same reasons the algorithm in figure 4 does. Conducted experiments (see below) show that few iterations are required to achieve the convergence (less than 10 iterations are needed).

5. Experiments & results

As we mentioned above, the purpose of our work is to provide a precision-oriented IR model based on graph vertices comparison. In other words, we want our system to retrieve more relevant documents within the top retrieved documents. To evaluate the performance of the GVC model, we consider four test collections (namely TREC 2004 Novelty Track[1] [TREC], CISI, Cranfield and Medline [IDOM]). Table 2 lists features about these test collections. Terms occurring in documents are stemmed, filtered using a common stop list, and weighted following the tf-idf function. Our system is compared to a cosine based system using the same indexation approach.

[1] Note that we only consider the first task of the novelty track. We are only interested in the relevance issue.

	TREC[2]	CISI	CRAN	MED
Number of documents	1057	1460	1400	1033
Number of terms	2877	2505	2167	3145
Number of evaluated queries	50	60	56	30
Average number of relevant documents[3]	166	48	14.2	23.2
Average rate relevant documents	15.70%	3.28%	1.01%	2.24%

Table 2: Statistics about the test collections

The TREC 2004 collection is different from the three other collections. While CISI, CRAN and MED have the same document set for all queries, TREC 2004 has 50 documents sets. For this collection, we computed, for each topic, the precision at the top n returned sentences. n varies from 1 to 367 which is the minimum number of sentences per topic in the TREC 2004 collection.

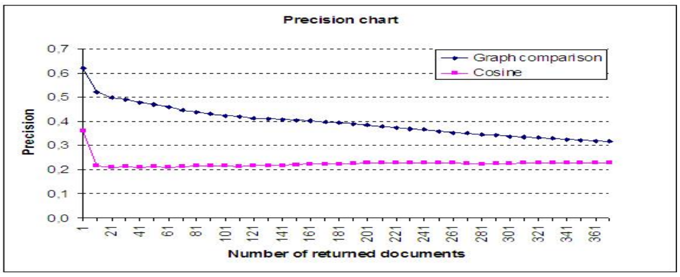


Figure 6: Average precision at top n for the TREC 50 topics

For the three remaining collections, we evaluated our model on only the queries which had at least 10 associated relevant documents. So, for the CISI collection where there are 112 queries, only 60 queries were selected. The CRANFIELD collection has 225 queries, among which 56 queries were selected. The MEDLINE collection has a total of 30 queries, all of which were selected. The average precisions at 5, 10 and R are computed for both our model and the cosine model. The table bellow shows the performance statistics of the two models on the four collections.

	Graph comp	Cosine	Enhancement
TREC	0,389	0,200	93,94%
MED	0,497	0,415	19,77%
CRAN	0,308	0,242	27,09%
CISI	0,203	0,149	35,75%

Table 3: Compared average precision for TREC, CISI, Medline and Cranfield. Comparison with table 4 shows relatively good results at top retrieved documents.

[2] TREC column shows averages over the 50 test collections

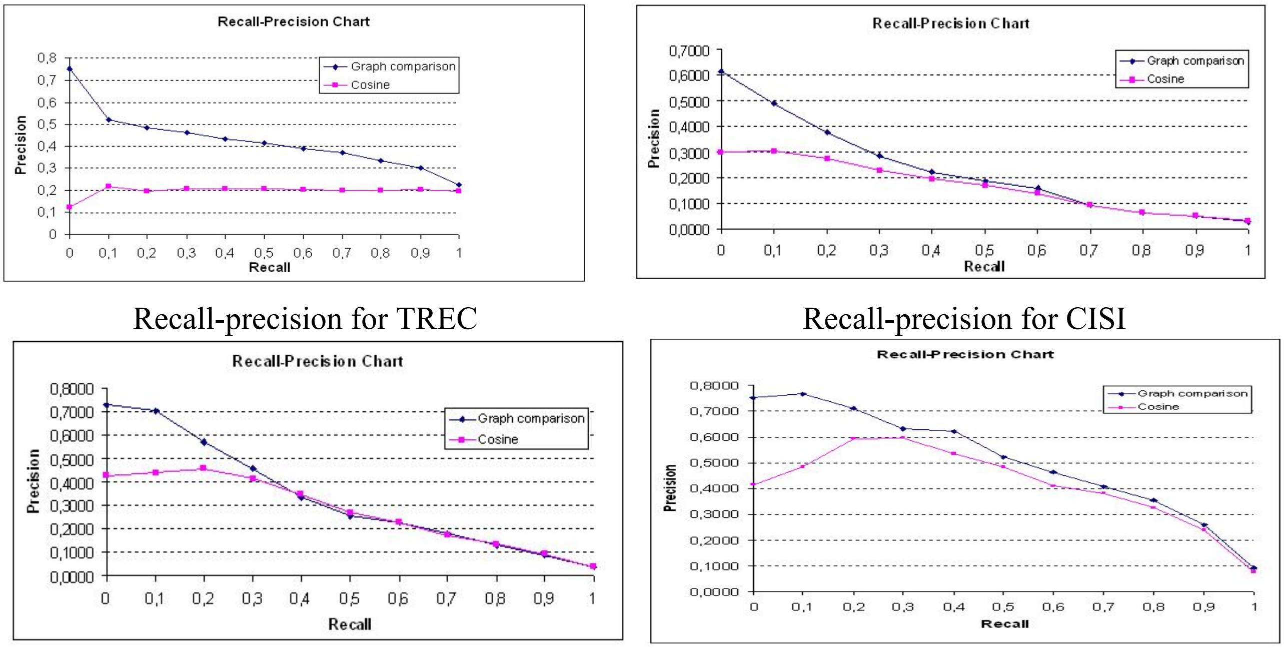
[3] Only selected queries are considered.

	Graph comparison			Cosine			Enhancement		
	5 docs	10 docs	R docs	5 docs	10 docs	R docs	5 docs	10 docs	R docs
TREC	0.510	0.513	0.400	0.270	0.222	0.237	*88.88%*	*131.08%*	*68.78%*
CISI	0.410	0.340	0.236	0.259	0.277	0.204	*58.30%*	*22.74%*	*15.69%*
MEDLINE	0.670	0.590	0.500	0.540	0.530	0.460	*23.93%*	*10.63%*	*8.06%*
CRAN	0.460	0.360	0.310	0.390	0.330	0.280	*19.27%*	*10.20%*	*9.75%*

Table 4: Compared precision at top n for TREC, CISI, Medline, Cranfield

The above tables – especially when considering TREC and CISI collection – confirm that our model performs better than the cosine model. The differences in terms of performance over the five test collections can be explained by the differences over the number of relevant documents. The fact is that the average number of relevant documents per query is quite low for Cranfield. This shows that our model performances are high when the rate of relevant document is high. This remark along with algorithm complexity consideration point out that the best use of our model is within the framework of a MAC/FAC retrieval system (see below). Confrontation of table 3 and table 4 favors the use of GVC as a precision-oriented IR model.

We also compute the precisions at 11 different recall points for both models: graph comparison and cosine. The obtained results clearly prove that our model outperforms the cosine model.



Recall-precision for TREC

Recall-precision for CISI

Recall-precision for Cranfield

Recall-precision for Medline

Figure 7: Recall-precision for TREC, CISI, Cranfield and Medline collection

6. Conclusion

In this paper, we proposed and discussed a new model for defining similarity measures between vertices of two graphs by extending methods previously submitted in (Jeh et Widom, 2002; Melnik et al, 2002; Blondel et al, 2004). The proposed method has been designed to support the case where the two compared graphs are identical. This paved the way to our proposal for a new IR model. This model views documents and queries, along with indexation terms, as vertices of a bipartite graph. Retrieving task is achieved as a graph self-comparison process. Retrieved documents are nodes that are satisfactorily similar to the query node.

The experiments show that our method highly outperformed the vector-based cosine model especially in the case of passage retrieval (e.g. TREC Novelty Track). Perhaps a most decisive performance test – which we intend to carry out – would be this that compares our method to similar methods that try to capture and use indirect relationships between documents and indexing terms as it is the case in Latent Semantic Indexing (LSI) approaches. We believe that the comparison with LSI models could turn in favor of our methods as suggested by published results in (Cherukuri et al, 2006), (Srinivas et al, 2006).

We also strongly believe that we can further enhance the obtained results by taking into account previously known information about existing similarities between documents and/or indexation terms. Such information can be contained in classifications, thesauri or ontologies. For example, WordNet (Richardson et al, 1994) can be used to initialize the similarity scores between terms. Also, there are indications that our method will offer new perspectives for XML retrieval, which can be achieved by using multipartite labeled graphs.

The main drawback to our method is its high computational complexity which makes it unaffordable in the context of large document collections. Still, we can use it in a MAC/FAC (Forbus et al, 1995) architecture. MAC/FAC is a two-stage process in which a computationally cheap filter (MAC) is used to select a restricted subset of likely good candidates that are conveyed to a more accurate and computationally expensive filtering process (FAC). Our graph vertices comparison method can be used as a FAC filter in association with a MAC method which will easily and quickly eliminate unnecessary documents. This is roughly what Kleinberg's HITS algorithm (Kleinberg, 1999) does in order to reduce the computational cost. HITS isolates a relatively small citation subgraph related to a given topic before detecting the authoritative 'sources' it contains.

References

Benjamin P. (2000). Learning in Information Retrieval: a Probabilistic Differential Approach, *Proceeding of the BCS-IRSG, 22nd annual Colloquium on Information Retrieval Research.* London.

Blondel V.D., Gajardo A., Heymans M., Senellart P. and Van Dooren P. (2004). A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. *SIAM Rev*. 46(4):647-666.

Bracke D. (1998). Vers un modèle théorique du transfert: les contraintes à respecter. *Revue des sciences de l'éducation*, XXIV(2) : 235-266.

Cherukuri A.K. and Suripeddi S. (2006). Latent Semantic Indexing using eigenvalue analysis for information retrieval. *Int. J. Appl. Math. Comput. Sci.*, Vol. 16, No. 4, 551-558.

Colt : http://dsd.lbl.gov/~hoschek/colt/

Dkaki T., Truong Q. D. and Charrel P-J. (2006). Visualisation interactive et comparaison de graphes pour l'analyse des réseaux, in *Les cahiers de l'INRIA*.

Forbus K., Gentner D. and Law K. (1995). MAC/FAC: a model of similarity-based retrieval. *Cognitive science* (Cogn. sci.) ISSN 0364-0213 CODEN COGSD5, Vol. 19, No. 2, 141-205.

Freeman L.C. (1979). Centrality in Networks: Conceptual Clarification. *Social NetWork 1*, pp.215-239.

Gärdenfors P. (2004). *Conceptual Spaces. The Geometry of Thought*. Cambridge, Mass.: MIT Press.

Gentner D., Ratterman M. J., Forbus K. D. (1993). The roles of similarity in transfer: Separating retrievability From inferential soundness. *Cognitive Psychology*, 25, 524-575.

Glasgow IDOM – Test Collection : http://www.dcs.gla.ac.uk/idom/

Gómez M. M., Gelbukh A. and López A. L. (2000). Information Retrieval with Conceptual Graph. *Proc. DEXA-2000, 11th International Conference and Workshop on Database and Expert Systems Applications.* Greenwich, England. Lecture Notes in Computer Science, Springer.

Goodman N. (1972). Seven strictures on similarity. In Goodman, N. (ed.) *Problems and Projects*, pp.437-447. Indianapolis and New York: Bobbs-Merrill.

Henzinger M. (2000). Link Analysis in Web Information Retrieval. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.

Jeh G. and Widom J. (2002). SimRank: a measure of structural-context similarity. *Proc. of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.538-543.

Kleinberg J.M. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604-632.

Medin D. L., Goldstone R. L., and Gentner D. (1990). Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, 1(1): 64-69.

Melnik S., Garcia-Molina H. and Rahm E. (2002). Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Scheme Matching. *Proc. of the 18th ICDE Conference*.

Newman M.E.J. (2003). Random graphs as models of networks. In *Handbook of Graphs and Networks*, S. Bornholdt and H. G. Schuster (eds.). Wiley-VCH, Berlin.

Page L., Brin S., Motwani R. and Winograd T. (1998).The PageRank citation ranking: Bringing order to the web. *Stanford Digital Libraries Technologies Project Working Paper*.

Quintana Y., Kamel M. and Lo A. (1992). Graph-based retrieval of information in hypertext systems. *Proc. of the 10th annual international conference on Systems documentation*, pp.157-168.

Richardson R., Smeaton A. and Murphy J. (1994). Using Wordnet as a Knowledge Base for Measuring Semantic Similarity between Words. *Working Paper CA-1294, School of Computer Applications*. Dublin City University.

Sahami M., Mittal V., Baluja S. and Rowley H. (2004). The Happy Searcher: Challenge in Web Information Retrieval. In *Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligent (PRICAI)*.

Siddiqui T. J., Shanker T., Khosla R., Howlett R.J. and Lakhmi C. (2005). Integrating relation and keyword matching in information retrieval. *International Conference on Knowledge-Based Intelligent Information and Engineering Systems No9*. Melbourne, Australia.

Srinivas S. and AswaniKumar Ch. (2006). Optimising the Heuristics in Latent Semantic Indexing for Effective Information Retrieval. *Journal of Information & Knowledge Management*, Vol. 5, No. 2, 97-105.

TREC : http://trec.nist.gov