GVC: a graph-based Information Retrieval Model

Quoc Dinh Truong[*,**,***] **– Taoufiq Dkaki**[*,**] **– Josiane Mothe**[*] **– Pierre-Jean Charrel**[*,**]

** IRIT, Institut de Recherche en Informatique de Toulouse, 118 Route de Narbonne, F-31062 Toulouse Cedex 9 France*
{dkaki, mothe}@irit.fr
*** Université de Toulouse, Toulouse, France*
{truong, charrel}@univ-tlse2.fr
**** Cantho University, 1 rue Ly tu Trong, Cantho, Vietnam*

ABSTRACT. *GVC is a new information retrieval model that is based on Graph Vertices Comparison (GVC). It implements a new similarity measure to compare documents and users' queries based on graph matching. In this model, graphs are composed of two types of nodes. Documents, queries and indexing terms are viewed as vertices of this bipartite graph where each edge goes from a document or a query –first type of nodes- to an indexing term –second type of nodes-. Edges reflect the relationship that exists between documents or queries on the one hand and indexing terms on the other hand; they are set according to the tf.idf principal. Our method implements similarity propagation over graph edges using an iterative process. We evaluate the model using 4 different collections (TREC 2004 Novelty Track, CISI, Cranfield and Medline). We show that considering precision at 5 documents, GVC outperforms Okapi model from 9% to 62%, depending on the collections.*

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' need is somewhat complex. Winning this challenge mainly implies finding out how well a given document or chunk of text matches a user's query. Therefore, the main question to be answered is how similar - a document and a query are. 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.

Still, this concept of similarity remains 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, concept 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 top-precision, we focus our interest on structural similarity.

Since relations are the main features used in structural similarity computation and since graphs are common representations that easily capture the structure of a wide range of relational data and knowledge, we consider graph theory. Moreover, graph theory 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*, 1990][Siddiqui *et al*, 2005], Social Networks Analysis [Freeman, 1979][Newman, 2003] and science citation, and co-citation networks analysis [Jeh *et al*, 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 $\rightarrow$ authority) and expresses the approach of hub and authority analysis depicted in [Kleinberg, 1999] as a graph mapping issue. Approaches using graph models can be split into two categories: basic approaches that use immediate neighboring nodes to compute the similarity of two vertices and sophisticated approaches such as SimRank [Jeh *et al*, 2002] that use 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. 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, queries and indexing terms are represented as nodes of a directed bipartite graph. In such a representation, graph vertices are either documents/queries (first type of nodes) or indexing 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. This process is apprehended as a graph vertices comparison. Thus, retrieving relevant documents is likened to searching for document nodes similar to a given query node in the IR graph.

The remainder of this paper is organized as follows: Section 2 presents the GVC information retrieval model based on graph comparison. Section 3 discusses the proposed approach in light of those presented in [Kleinberg, 1999] and [Blondel *et al*, 2004]. It also 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 software that manages the storage of 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.

We use the vector based model as a starting point to build our graph based IR system. 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 concerns and contribution, ranks the documents so that those most likely to be relevant (those with the higher similarity score in comparison to the query) are placed at the top of the retrieved document list. Considering the GVC model, the matching process computes the similarity scores between vertices of the modeling graph. Similarity scores are not locally computed, but rather take into account the whole graph structure.

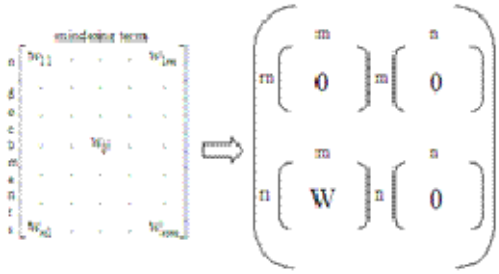


Figure 1. *Building the bipartite graph starting from document-indexing term matrix.W=[w_{ij}] where w_{ij} represents the weight of the j^{th} term in the i^{th} document*

The next section addresses the theoretical basis of our proposal. it 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].

2.1 Background

The starting point of our matching process approach is a graph vertices comparison method proposed by Blondel et al. [Blondel *et al*, 2004]. We improved this method that consider the comparison of the vertices of two different graphs (see below) in order to properly take into consideration the case of self-similarity needed in IR systems where only one bipartite graph is considered and where this graph nodes have to be compared to each other. The purpose of such method is to determine a similarity measure for computing the resemblance between graph vertices of two graphs.

For example, the problem of computing hub and authority scores of a Web search engine to increase top accuracy as proposed in [Kleinberg, 1999] can be considered as a graph vertices comparison problem [Blondel *et al*, 2004] where the Web, as a citation graph, is compared to the two-node directed-graph hub $\rightarrow$ authority. The formula for computing hub and authority scores of Web pages can be expressed as follows:

$$\begin{bmatrix} h_{p_1} & a_{p_1} \\ \vdots & \vdots \\ h_{p_n} & a_{p_n} \end{bmatrix}_{k+1} = B \begin{bmatrix} h_{p_1} & a_{p_1} \\ \vdots & \vdots \\ h_{p_n} & a_{p_n} \end{bmatrix}_k \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T + B^T \begin{bmatrix} h_{p_1} & a_{p_1} \\ \vdots & \vdots \\ h_{p_n} & a_{p_n} \end{bmatrix}_k \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad [1]$$

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*$\rightarrow$*authority.*

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 and j respectively from target graph and source graph is computed by examining the similarity scores between their related vertices (vertices pointing to i or j and vertices pointed by i or j in the analyzed and model graphs). The subjacent idea is a similarity mutual reinforcement: a score is associated to each couple of nodes of the graphs and, at each step of the iterative process, the value of this score is updated by the sum of the similarity scores of the predecessors and the similarity scores of the successors in the two graphs. The similarity score S_{ij} between vertex i of target graph and vertex j of source graph can be expressed as follows:

$$S_{ij} = \sum_{r:(r,i)\in E_B, t:(t,j)\in E_A} S_{rt} + \sum_{r:(i,r)\in E_B, t:(j,t)\in E_A} S_{rt} \qquad [2]$$

E_A and E_B are, respectively, the edge sets of target and source graphs.

[2] can be written in the more compact matrix form:

$$S_{k+1} = BS_kA^T + B^TS_kA \qquad [3]$$

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 choice for similarity initial values (S_0).

- Convergence

Convergence of [3] is uncertain but this problem can be overcome by normalizing the similarity matrix S at each iteration step. [3] is then rewritten as follows:

$$S_{k+1} = \frac{BS_kA^T + B^TS_kA}{\left\| BS_kA^T + B^TS_kA \right\|_F} \qquad [4]$$

In this case, the S_k series convergence is not entirely assured but at least, whatever the initial similarity values, the sequence admits two adherence values: one limit for S_{2k} series and another 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.

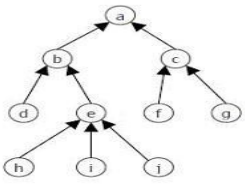
- Initialization

There are two possible ways to choose an initial similarity matrix S_0. These cases are related to the awareness of a priori resemblance between vertices of the two graphs. If there is no 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. Such initial similarity for matrix S_0 produces quite good results as mentioned in [Blondel *et al*, 2004]. Otherwise, previously known similarity scores (e.g. some attributes-dependent surface similarity) can be used to build matrix S_0. This second case will be discussed in more detail in the implementation section.

3. GVC model

3.1. Preliminary test

In the following example, the graph represented in fig. 2 is compared to the graph hub→authority. Initial similarity matrix is the matrix full of ones. The obtained results are unsatisfactory if not to say odd. They oppose commonsense -or at least the results of an in-degrees/out-degrees analysis. Indeed, examining table in fig. 2, we notice that vertices a, d, f, g, h, i and j get the same authority score. This calls for a method enhancement



Vertex	A	B	C	D	E	F	G	H	I	J
Hub	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.2
Authority	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0

Figure 2. *A graph used for preliminary test purposes and the results of its comparison with 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 is that the method does not take into account the notion of similarity inheritance or flooding similarity within a graph. When considering a concept 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 authority likely confers some kind of authority.

[4] has to be modified to comply with this similarity inheritance principle. The propagation and the retro-propagation of similarities within a graph are likened to a "flooding" of similarities [Melnik *et al*, 2002]

In our approach, similarity propagation is considered as a graph transitive-closure of target and/or source graphs. This is a more sophisticated approach than the one in [Melnik *et al*, 2002] that leads to a more wide-ranging function of gradual attenuation of inheritance "over generations".

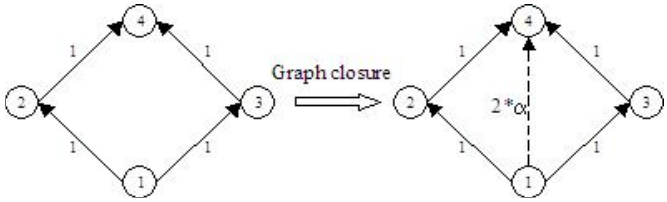


Figure 3. *Example of graph transitive closure. The graph to the left is transformed into the graph to the right. There is only one added edge from node 1 to node 4 because there are only two indirect paths of length 2 from 1 to 4.*

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. We also consider that inheritance from a "forebears" node of nth generation of a vertex v is proportional to the number of paths of length n separating this forebear from v. This sounds reasonable: the more a vertex has forebears that share a given property, the greater its chances of possessing this same property. Still, the bare proportionality is somewhat questionable. We, therefore, break this linear proportionality relationship. Adjacency matrices of target and source graphs are formulated as follows:

$$A \leftarrow A + \sum_{n=2}^{\infty} f_A(n)\, g_A\left(\frac{A^n}{\|A^n\|}\right) \quad \text{and} \quad B \leftarrow B + \sum_{n=2}^{\infty} f_B(n)\, g_B\left(\frac{B^n}{\|B^n\|}\right) \qquad [5]$$

g_A and g_B are monotonically increasing functions from [0, 1] onto [0,1]. They can be exponential or stair functions. $f_A(n) = \alpha^n$ and $f_B(n) = \beta^n$ where α and β are positive constants lower than 1, appear to be good instantiation of attenuation functions.

The use of [5] overcomes the disadvantages depicted above (figure 2.) as shown in the following results (table 1)

Vertex	A	B	C	D	E	F	G	H	I	J
Hub	0.0	0.12	0.12	0.15	0.15	0.05	0.05	0.27	0.27	0.27
Authority	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 documents' concordance with the user's needs expressed in a query. Using graph comparison for IR assumes that we look for document vertices similar to a given query node. This is a search for similar nodes of the same graph which implies graph self-comparison where target and source graphs are the same. Unfortunately, when assessing graph self-comparison, there are cases where $s(i, j) \geq s(i, i)$. In fact, this opposes a condition that a similarity measure must fulfill. The measure we obtain is positive defined $\forall(i, j), s(i, j) \geq 0$-, symmetric -$\forall(i, j), s(i, j) = s(j, i)$- but it does not always verify $\forall(i, j), s(i, i) = s(j, j) \geq s(i, j)$.

Satisfying this condition is, of course, not mandatory since very common "similarity" measures in IR, such as the dot product, do it. 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 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.

$S_0 \leftarrow 1$; $k \leftarrow 0$;

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

Repeat

$$S_{AAk+1} \leftarrow \frac{A S_{AAk} A^T + A^T S_{AAk} A}{\left\| A S_{AAk} A^T + A^T S_{AAk} A \right\|_F} \;;\; S_{BBk+1} \leftarrow \frac{B S_{BBk} B^T + B^T S_{BBk} B}{\left\| B S_{BBk} B^T + B^T S_{BBk} B \right\|_F}$$

$$S_{ABk+1} \leftarrow \frac{B S_{ABk} A^T + B^T S_{ABk} A}{\left\| B S_{ABk} A^T + B^T S_{ABk} A \right\|_F}$$

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 as similarity matrix

Figure 4. *Algorithm for graph similarity computation.* $\bullet *$ *and* $\bullet$— *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- from the result of indexing process of the Lemur toolkit [The Lemur toolkit].

- Second, 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 and indexing terms, the initial value of similarity score between node i and j is computed as follows:

$$S_0(i,j) = \frac{\sum_{k=1 \to n+m} G(i,k) * G(j,k)}{\sqrt{\sum_{k=1 \to n+m} G(i,k) * G(i,k)} * \sqrt{\sum_{k=1 \to n+m} G(j,k) * G(j,k)}} \qquad [6]$$

S_0 can be written as $\begin{bmatrix} S_T & 0 \\ 0 & S_D \end{bmatrix}$ where S_T is the m*m term similarity matrix and S_D is the n*n document 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 = $\begin{bmatrix} 0 & 0 \\ W & 0 \end{bmatrix}$. Similarity matrix is therefore block-diagonal. Indeed, S_2 is block-diagonal

$$S_2 = \frac{\begin{bmatrix} W^T W S_{T_0} W^T W & 0 \\ 0 & W W^T S_{D_0} W W^T \end{bmatrix}}{\sqrt{\left\| W^T W S_{T_0} W^T W \right\|_F^2 + \left\| W W^T S_{D_0} W W^T \right\|_F^2}} \quad [7]$$

And if S_{2k} is block-diagonal, $S_{2k} = \begin{bmatrix} S_{T_{2k}} & 0 \\ 0 & S_{D_{2k}} \end{bmatrix}$ then S_{2k+2} is also block-diagonal

$$S_{2k+2} = \frac{\begin{bmatrix} W^T W S_{T_{ék}} W^T W & 0 \\ 0 & W W^T S_{D_{2k}} W W^T \end{bmatrix}}{\sqrt{\left\| W^T W S_{T_{2k}} W^T W \right\|_F^2 + \left\| W W^T S_{D_{2k}} W W^T \right\|_F^2}} \quad [8]$$

Because we are interested only on the similarity scores between documents and they can be computed separately with which between terms we can then rewrite the graph vertices comparison algorithm in figure 4 as follows

$S_0 \leftarrow S_{D_0} k \leftarrow 0$

$G \leftarrow G + \sum_{n=2}^{\infty} f_2(n) g_2 \left(\frac{G^n}{\left\| G^n \right\|} \right)$

Re *peat*

$S_{k+1} = \frac{WW^T S_{D_k} WW^T}{\left\| WW^T S_{D_k} WW^T \right\|_F}$

$k \leftarrow k + 1$

Until convergenc e is achieved

$S_k \leftarrow \bullet \frac{S_k \bullet * S_k}{diag(S_k) \bullet * diag(S_k)^T}$

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. Conducted experiments (see below) show that a few iterations are required to achieve the convergence.

The algorithm is mainly based on matrix products, since then the computational complexity of each iteration is $\Theta(Max(n,m)3)$. This reduces the applicability of our method. Our system, that uses the Colt libraries [Colt package] on a Pentium PC with 1 Go of RAM, can only handle graphs that do not exceed few thousands of nodes. For this reason, experiments use small data collection. Section 6 will give some hints of how our method can be used as a component of an IR system that can handle large document collections.

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 common test collections (namely TREC 2004 Novelty Track [TREC], CISI, Cranfield and Medline [Glasgow 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. We used Lemur toolkit [The Lemur toolkit] to index these four collections. Our similarity measure is compared to Okapi as implemented in Lemur. The different parameters are set as follows:

K1= 1.2,
B = 0.75,
K3 = 7.0,
Expanded query term TF weight = 0.5 and
Number of feedback terms = 50.

	TREC[1]	CISI	CRAN	MED
Number of documents	1057	1460	1400	1033
Number of terms	2157	5889	4614	9467
Number of evaluated queries	50	60	56	30
Average number of relevant documents[2]	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*

[1] TREC column shows averages over the 50 test collections
[2] Only selected queries are considered.

For all 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 Okapi model. The table bellow shows the performance statistics of the two models on the four collections.

	Graph comp	**Okapi**	**Enhancement**
TREC	0,307	0,243	26,34%
MED	0,543	0,32	63,39%
CRAN	0,34	0,215	57,99%
CISI	0,209	0,123	69,65%

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

These tables --especially when considering CISI, MEDLINE and CRANFIELD collection-- confirm that our model performs better than the Okapi model. The differences in terms of performance over the four test collections can be explained by the differences over the average number of terms per documents and the number of relevant documents. The fact is that average number of relevant terms per documents is quite low for TREC. This shows that our model performances are high when the average number of terms per documents and the rate of relevant document are high. This along with algorithm complexity consideration point out that the best use of our model is within the framework of a MAC/FAC [Forbus *et al*, 1995] retrieval system (see below). Confrontation of table 3 and table 4 favors the use of GVC as a precision-oriented IR model.

	Graph comp			Okapi			Enhancement		
	5	10	R	5	10	R	5	10	R
TREC	0.396	0.363	0.321	0.363	0.359	0.304	8.99%	1.14%	5.76%
CISI	0.483	0.412	0.276	0.298	0.279	0.209	61.87%	47.72%	31.68%
MEDLINE	0.733	0.672	0.569	0.527	0.484	0.369	39.24%	38.82%	54.29%
CRANFIELD	0.517	0.419	0.357	0.37	0.291	0.259	39.8%	44.16%	37.16%

Table 4. *Compared precision at top n for TREC, CISI, Medline and Cranfield*

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 al*, 2002][Melnik *et al*, 2002][Blondel *et al*, 2004][Kleinberg, 1999]. 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 and retrieved documents are nodes that are satisfactorily similar to a query node.

The experiments show that our method outperforms the Okapi model. Perhaps another 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 for example. We believe that the comparison with LSI models could turn in favour 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. Such information can be contained in classifications, thesauri or ontologies. 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 its cost. HITS isolates a relatively small citation subgraph related to a given topic before detecting the authoritative 'sources' it contains.

7. References

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

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

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

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

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

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

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

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

Gómez, M. M., Gelbukh, A., López, A. L., "Information Retrieval with Conceptual Graph", *Proc. DEXA-2000, 11th International Conference and Workshop on Database and Expert Systems Applications*, Greenwich, England, September 4-8, 2000. Lecture Notes in Computer Science, Springer.

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

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

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

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

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

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

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

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

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

The Lemur Toolkit, http://www.lemurproject.org

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

TREC, http://trec.nist.gov

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

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

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

Forbus, K., Gentner, D., Law, K., *MAC/FAC: a model of similarity-based retrieval*, Cognitive science (Cogn. sci.) ISSN 0364-0213 CODEN COGSD5, vol. 19, no2, pp. 141-205, 1995.