A Hybrid Model to Improve Relevance in Document Retrieval

Tanveer J. Siddiqui
Department of Electronics & Communication,
University of Allahabad, Allahabad, India
tjs@jkinstitute.org

Umashanker Tiwary
Indian Institute of Information Technology, Allahabad, India
ust@iiita.ac.in

***ABSTRACT**: In information retrieval community a lot of work is focused on increasing efficiency by capturing statistical features. The other dominant approach is to improve the relevance by capturing the semantic and contextual information which is invariably inefficient. Generally the two approaches are assumed to be diametrically opposite. In this paper we have tried to combine the two approaches by proposing a hybrid information retrieval model. The model works in two stages. The first stage is a statistical model and the second stage is based on semantics. We have first downsized the document collection for a given query using vector model and then used a conceptual graph (CG) based representation to rank the documents. Our main objective is to investigate the use of conceptual graphs as a precision tool in the second stage. The use of CGs brings semantic in the ranking process resulting in improved relevance. Three experiments have been conducted to demonstrate the feasibility and usefulness of our model. A test run is made on CACM-3204 collection. We observed 34.8% increase in precision for a subset of CACM queries. The second experiment is performed on a test collection specifically designed to test the strength of our model in situation where the same terms are being used in different context. Improved relevance has been observed in this case also. The application of this approach on results retrieved from LYCOS shown significant improvement. The proposed model is both efficient, scalable and domain independent.*

Categories and Subject Descriptors
H.3.1 [Content Analysis and Indexing]**; H.3.3 [Information Search and Retrieval]: I.7** [Document and Text Processing]

General Terms
Conceptual Graphs, Semantic retrieval

Keywords: High precision information retrieval model, Intelligent retrieval, CG-based retrieval model, Two-stage retrieval model.

Received 12 May 2005; Revised and accepted 27 Sep. 2005

1. Introduction

Traditionally Information retrieval (IR) has been concerned with retrieving relevant information from a collection of documents based on a set of terms provided by the user. Users enter a query and information retrieval system responds by returning a list of documents that seem relevant to the request. On a very large document collection thousands of documents might contain query terms therefore it is important that information retrieval systems rank the retrieved documents according to relevance. Users are not interested in huge amount of information, but in precise, accurate and relevant information. But relevance can't be judged simply on the basis of term occurrence. Most of the existing retrieval system still rely on standard retrieval models (e.g. Boolean, Standard vector and probabilistic) that treat both document and queries as a set of unrelated terms. These statistical models have the advantage of being simple, scalable and computationally feasible but they do not offer accurate and complete representation. These models ignore semantic and contextual information in the retrieval process (Watters 1989). It is difficult to identify useful documents simply on the basis of words used by the author of the document. This is also because of polysemy and synonymy in natural language. Polysemy designates the phenomenon of a lexeme with multiple meaning. The ambiguity makes it difficult for a computer to automatically determine the conceptual content of documents. Synonymy creates a problem when document is indexed with one term and the users' query contains a different term and the two terms share a common meaning. Another important problem owing to syntactic terms representation is that contextual information is lost in the extraction of keywords from the text. The context can not be recovered. We attempt to handle these problems through the inclusion of conceptual graph based representation. Our concern in this paper is to use semantics through Conceptual Graph (CG) to improve relevance in document retrieval. Martin and Eklund (2000) argued in favor of general knowledge representation languages for indexing web documents and suggested the use of concise and easy comprehensible CGs. They argued that CG representation has advantage over metadata language based on extensible markup language (XML). Rama and Srinivasan (1993) underlined the utility of conceptual roles for information retrieval. Their work provides strong evidence in support of use of conceptual roles in information retrieval. Conceptual graphs are very closely related to natural language and hence can be used for representing text. Such a representation holds the promise of extracting more information from documents by explicitly capturing logical relationship between terms; unlike word-statistical approaches that merely count nouns and noun phrases. This fact suggests the use of conceptual graphs in information retrieval, database interfaces and natural language processing. With such representation we will be able to improve precision in information retrieval. For example, Statistical model will fail to distinguish between "library school" and "school library". CG-based model keeps the contextual information between document and/or query terms intact, hence is able to differentiate documents in such cases. Let us consider two documents having fragments like:

1. … genetic algorithm for information retrieval
2. …genetic algorithm, neural network, ….

Information retrieval…

and a query "genetic algorithm for information retrieval".

Both the documents contain the terms "genetic", "algorithm", "information" and "retrieval". If we represent these two document fragments as a set of terms then traditional vector space models will return both of them as equally relevant, though document containing fragment 1 is clearly more relevant to the query. This is the local context that helps us to distinguish the two document fragments in this case. A model that considers the relationship existing between the words in two documents will find fragment 1 more relevant as compared to fragment 2. We have attempted to make distinction possible in such cases through the use of conceptual graphs. The variants of CG model have been used in information retrieval. Marega and Pazienza (1994) also emphasized the use of contextual role of words in CoDHir system and concluded that this results in an improvement in retrieval precision over traditional IR technologies. They used conceptual graphs in CoDHir system. Their work consists in identifying contextual roles of words and to extend vector model to consider compound descriptor (contextual role – word). DR-LINK (Liddy and Myaeng, 1993, 1994) uses conceptual graphs to extract and use semantic relation for information retrieval. Cgkat (Martin 1997) and WebKB (Martin and Eklund 2000) uses CGs to index document elements (chapters, paragraphs, etc.). The retrieval mechanism is based on the projection. Relief(ounis, 1998) uses CGs for indexing images and uses properties of relations such as transitivity for querying.

Researchers have also proposed statistical approaches to include contextual information in the retrieval process through the use of multi-term phrases and proximity search algorithm on top of the first stage. Multi-word phrase matching is simpler and existing methods for single term matching can be applied to multi-words. However, it fails to capture variations in syntactic structure unless phrases have been normalized. For example "extraction of roots" might be transformed into "root extraction". Further, the use of multi-word phrases has not yielded significant improvement (Mittendorf et al, 2000) Proximity search may also fail. Consider following two sentences:

The Allahabad bank is situated near river Ganga.
and I stayed in a hotel situated near the bank of river Ganga.
and a query "Bank near Ganga"

Though it is clear from the context that the first sentence is more relevant to the query but this distinction is lost when treated as a bag of words. A proximity search algorithm will not help in this case. It is the relationship existing among words that make this distinction clear and not just their relative positions. The keyword matching methods ignore relations that are expressed in the query (Khoo, 1997). Because a CG aims to capture the relationships between concepts it holds the promise to improve ranking in such situations. Further the model has the potential to handle ambiguities implicitly.

Earlier attempts to include semantics were based on latent semantic indexing (Deerwester et al. 1990, Foltz 1990) and natural language processing (Strzalkowski 1995, Smeaton, 1995) techniques. Use of Latent Semantic Indexing (LSI) in information retrieval is based on the assumption that there is some underlying "hidden" semantic structure in pattern of word-usage across documents. LSI attempts to identify this hidden semantic structure through statistical techniques and then uses this structure for representing and retrieving information. However it is costly in terms of computation and requires re-computation if many new terms are added. Natural language processing has also been used in information retrieval for resolving anaphoric references, discourse-level processing, proper name identification, etc. The direct application of natural language processing results in lack of robustness and efficiency (Strzalkowski, 1995). Hence it is often used with other existing system for indexing, retrieval, query expansion and query modification through relevance feedback.

Dick(1992) pointed out that conceptual retrieval is a precision tool not an all purpose device. Agreeing with Dick we propose a two stage information retrieval model to handle the problems associated with pure statistical models. Retrieval in the first stage is performed using vector space model. The second stage ranks the documents retrieved in first stage based on CG representation. As conceptual graphs make relevance judgment based on semantics, it more closely correspond to users' intent, as expressed through queries. In traditional information retrieval system relevancy simply refers to the degree to which the query terms are present or absent in a document. The model builds notion of "relevancy" based on an understanding of semantics of terms. It ranks documents on the basis of their relational and conceptual similarity to query. This form of relevancy more closely corresponds to users' mental model as can be verified through the improved observed precision or improved acceptance from user.

For CG based ranking, relationships existing among concepts in a sentence have been captured. The corresponding conceptual graphs have been stored for each document during preprocessing stage. The model thus takes the advantage of the simplicity of traditional retrieval models and the versatility of semantic approaches. Unlike LSI, the CG-based model has the advantage of being scalable. Vector space model has been used first to quickly retrieve set of potentially relevant documents and then relevance judgment has been made based on conceptual graphs. A quantified measure proposed by Montes-y-Gomez et al (2000) to match conceptual graphs has been used in this work instead of graph derivation. This measure combines conceptual and relational similarity. The CG representation of documents is prepared along with vector representation. Relatively small number of relations has been captured and some simple heuristics (rules) have been used during matching to allow an exact match in case of semantically similar but structurally dissimilar CG fragments. All these factors contribute in keeping the model computationally simple and add generality to it.

Rest of the paper is organized as follows. Section 2 introduces Conceptual Graphs. Section 3 discusses the retrieval model. This includes a discussion of both vector space and CG-based representation to be used during first and second stage of retrieval. CG similarity measure has also been discussed. Experimental investigations have been made in section 4. Finally conclusions have been made in section 5.

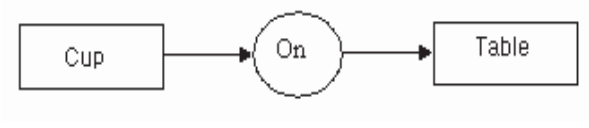
2. Conceptual Graphs

Conceptual graphs are highly expressive form of logic and were originally designed for representing natural language semantics. They have been evolved out of conceptual structure theory as set down by Sowa (1984). A conceptual graph is a network of concept nodes and relation nodes. Concept nodes represent entities, attributes, states and events and relation nodes show how the concepts are interconnected".

In the graph boxes represent concept nodes and the circles are called conceptual relations. Consider the following sentence:

"A cup is on the table" (1)

The conceptual graph of the above sentence is:



Concept nodes have two types of field – type field and referent field. Two fields are separated with a colon. In the box type field is shown on the left and referent field is shown on the right separated by a colon. Concepts that do not identify a specific individual are called generic concepts. The referent part of these concepts is omitted. The existential quantifier ($\exists$) is assumed to apply on concepts with blank referent field. For individual concepts the referent field is a specific entity such as a name.

A CG can be represented in three different forms. There is a graphic notation called the display form (DF), a more compact notation called the linear form (LF) and a concrete syntax called the conceptual graph interchange form (CGIF) which has a simplified syntax and a restricted character set designed for compact storage.

The LF representation of the sentence (1) is as:

$$[\text{Cup}] \rightarrow (\text{On}) \rightarrow [\text{Table}]$$

In CGIF the sentence (1) can be represented as:

[Cup: *x] [Table: *y] (On ? x ? y)

The symbols *x and *y are called defining labels. The matching symbols ?x and ?y are the bound labels that indicate references to the same instance of a cup x or a table y. CGIF also permits concepts to be nested inside the relation nodes. Nesting of concepts helps in reducing number of co reference labels.

(On [Cup] [Table])

The above representations can be translated to a statement of the following form in typed predicate calculus:

$$(\exists x : \text{cup})(\exists y : \text{Table})\text{on}(x, y)$$

Formally a CG is defined over a support S. A support is a 3-tuple S = (T_C, T_R, I). T_C, and T_R are two partially ordered finite sets, respectively, of concept types and relation types and I is a countable set of individual markers. T_C, T_R, I are pair wise disjoint. Each support also possesses the generic marker *, which does not belong to I.

The set $I \cup \{*\}$ is partially ordered with * being the greatest element.

A truly conceptual representation of text is difficult to achieve automatically hence we propose a simplified form of conceptual graph model that can be easily extracted from the text without requiring much of the domain knowledge for the IR task. This model

relies on a small set of basic relations (T_R), as shown in figure 1, that can be identified based on syntactic patterns. The terms appearing in a document are all considered individual marker and all are of the same type "concept" which is the only type in the set of concept types. The support is thus ({ "concept"}, T_R, T) where T is the set of terms used to represent documents. The focus mainly is on capturing semantics through the inclusion of relationships between terms as it goes in line with the cognitive way of understanding. Most of the research work on CG has focused on graph theory and graph algorithms (Mugnier, 1995). However, it seems unreasonable to use non-deterministic graph derivation algorithms for IR applications, hence a simplified form of conceptual graph matching function proposed by Gomez and others (Gomez et al. 2000; Gomez et al. 2001) has been used in this work.

AGNT. Links [ACT] to the concept representing actor of the act.
Example: *Tuba bit an apple.*
[Tuba] ←(agnt) ←[Bite] →(ptnt) → [Apple]
ATTR. Links an entity to its attribute. That is to say, (ATTR) links [Entity: *x] to [Entity: *y] where y is an attribute of x.
Example: *The water is cool.*
[water]→(attr)→[cool]
MOD. Links a concept to its modifier(qualifier).
Example: *Information retrieval is discussed*
[discuss]→(obj)→[retrieval] →(mod)→[information]
QTY. Links a set of [ENTITY: (*)] to a [NUMBER] that indicates the number of entities in that set.
Example: *There are 10 students in the class.*
[class]←(loc) ←[student: (*)] →(qty) → [number: 50]
PTNT. Links an [ACT] to an [ENTITY], which is acted upon.
PMOD. We have used PMOD relation to represent all the prepositional relations [PMOD_FOR, PMOD_AT, PMOD_IN,PMOD_OF ettc.]
MNR. Links an [ACT] to its attribute.
Example: *Shereen drives slowly.*
[Shereen] ← (agnt) ← [Drive] →(mnr) →[Slowly]

Figure 1. A subset of conceptual relations

3. Retrieval Model

Figure 2 shows our retrieval model. Two models are used in conjunction.

(i) Vector space model
and (ii) Conceptual Graph model

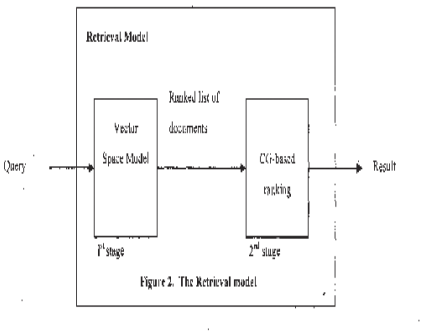


Figure 2. The Retrieval model

Information retrieval systems retrieve documents for a query and present a ranked list of retrieved documents to users. This ranking is based on system's understanding of users' relevance model. Vector model is quite efficient in retrieving documents satisfying users' need. However it ranks documents on the basis of statistical data. Because of the inherent ambiguities and vagueness of natural language text the ranking offered by vector model may not correspond to users' relevance model. Including semantics in ranking bridges the gap between the systems' and users' relevance model. Hence we have used two stage retrieval model.

3.1 Vector Space model

Our base model is vector space model. It is a mathematical model (Dominich, 2001) that forms the basis of many of the existing search engines.

3.1.1 Document Representation

In vector model both the query and the document is represented as a vector. The values of elements in these vectors represent weights assigned to terms occurring in the documents. These weights denote relative importance of terms in documents. These vectors can be represented as:

$$d_j = (w_{1j}, w_{2j}, w_{3j}, \ldots, w_{ij}, \ldots, w_{mj})$$

where d_j is a vector representing document j. If there are n documents in the collection and a total of m index terms have been identified to represent, then the document collection as a whole is represented by a m x n matrix, called term-document matrix. An entry, say $(i,j)^{th}$, in this matrix represents weight of term i in document j. The steps involved in preparing term-document matrix automatically are -

(i) Find individual words and their frequencies in each document
(ii) Use stop list to remove common words.
(iii) Reduce words to their stems. We have used Porter's stemmer for this purpose.
(iv) Assign weights to each term and prepare term-document matrix.

3.1.2 Term weighting

In the information retrieval field, different term weights have been proposed over the years to represent importance of a term in the retrieval process. Most of the weighting schemes consider the following three factors to assign weights:

- Within-document frequency or term frequency
- Collection frequency or inverse document frequency
- Document length

According to Salton (Salton et al., 1996), a high performance weighting scheme should assign more weight to a term that occur frequently in a document and rarely outside, as these term will be more useful for discriminating among documents. The tf-idf weighting scheme fulfils this criterion. The weight assigned to terms according to this scheme is product of tf (term frequency) and idf (inverse document frequency). The first component considers term frequency within document and second component captures distribution of terms across documents. Inverse document frequency for term t_i is calculated as :

$$idfi = \log \frac{n}{ni}$$

Where n is number of document in the collection and n_i is the number of documents containing term t_i . Many variations of tf-idf measure have been reported (Salton and Buckley, 1988). A number of recent weighting schemes has been proposed by researchers, particularly the BM25 which has reportedly best performance in TREC (Robertson et al. 1994, Robertson *et al.* 1995) environment. Robertson and Walker motivated the best match (BM) algorithms by the probabilistic model and by some simple approximations to 2-poisson model (Robertson and Walker, 1994)l, but indicated that their result was as much guided by experimentation as by theory (Hiemstra and de Vries, 2000). For CACM-3204 collection we have compared the performance of BM25 and tf-idf weighting function. The following approximation of BM25 term weighting function has been used in this comparison.

Document term weights :

$$wij = \frac{tf_{ij} \times (k1+1)}{k1 \times ((1-b) + b \times \frac{dl}{avdl}) + tf_{ij}} \times \log\left(\frac{n - n_i + 0.5}{n_i + 0.5}\right)$$

Query term weights: $w_{ik} = tf_{ij}$
Query-document similarity : " w_{ij} w_{ik}

Where, tf_{ij} = frequency of term t_i in document d_j
n = number of documents in the collection
n_i = number of documents in the collection containing the term t_i
dl = document length, and
avdl = average document length in the collection
The two coefficients k1 and b were empirically set to 2 and 0.1.

The form of tf-idf weighting scheme used by us in the first stage is doc= "atn" and query = "ntc" (Savoy and Picard, 2001). In the rest of the discussion we will call this scheme as "modified tf-idf" scheme. This selection was based after evaluating the performance of the CACM collection on seven different combination of "atn", "ltc", "ntc" , "nnn" and "atc" weighting scheme. We observed maximum precision (36.1%) with doc= "atn' and query = "ntc" weighting scheme. This scheme uses following expression to compute term weights:

Document-term weight ("atn"): $wij = (0.5 + 0.5 \times \frac{tfij}{\max tfj}) \times (\log \frac{n}{ni})$

Query-term weight ("ntc") : $w_{ik} = \frac{w1_{ik}}{\sqrt{\sum_{i=1}^{m} w1_{ik}^2}}$

where $w1_{ik} = tf_{ik} \times \log(\frac{n}{n_i})$

Query-document similarity : $\sum$ w ij X w ik

3.2 Conceptual Graph Model

A conceptual graph representation of document is prepared which will be used in the second stage of the retrieval.

3.2.1 Document Representation

The CG representation of documents has been obtained by identifying relationships among concepts occurring in a sentence. For constructing conceptual graph syntactic patterns in a sentence are identified. The relations captured include the relations between the constituent nouns of complex nominal, relations between verb and other constituents surrounding the verb, mainly AGNT(agent or subject) and PTNT(patient or object) relationship, and prepositional relations. A set of terms that can be substituted for each other has been maintained and utilized in the search process.

The CGs are stored in the form of triplets as:

(rel c1 c2)

Where 'rel' is the relation and 'c1' and 'c2' are concepts participating in this relation.

Example: The tagged representation of the fragment "iterative procedure for solution of equation" will be:

iterative {JJ} procedure {NN} for {IN} solution {NN} of {IN} equation {NN}.

Its CG representation will be stored as:

(attr procedure iterative) (pmod procedure solution) (pmod solution equation)

Unlike Liddy and Myaeng (1994) we have not maintained alternate representation of nominalized verb for documents. This together with a small number of relations identified by us helps in keeping our model computationally simple and thereby making it useful to be used on top of first stage. In order to capture variations in grammatical structures we have used simple heuristics during matching. These heuristics allow exact match for structurally different but semantically similar relations. For instance if the root form of a query concept involved in a 'mod' relation is a verb and it matches with a concept appearing in a 'ptnt' relation in a sentence CG then this will be considered an exact match if second participating concept in both the relation is same. This heuristic will give an exact match for 'task {NN} scheduling {NN} and 'schedule {VB} task {NN}'. The CG for these two fragments is:

[schedule] → (type) → [task] and [schedule] → (ptnt) → [task]

Steps involved to get Conceptual graph representation are as follows:

1. Tag documents in the collection. We have used TnT[1] tagger for this purpose.
2. Tagged representation of document is processed to make certain substitution and modifications, such as, removal of modal words, wh-determiner, wh-pronoun and few patterns like 'such that' , 'so that', 'as well as' (replaced with 'and') etc.
3. Tagged and processed document is input to a sentence extractor which extracts sentences. Each extracted sentence of the tagged text is then passed through four modules. Each of these modules is devoted to identify certain types of relationships between concepts. These modules correspond to:
   - Preposition and Adverb handler: to extract prepositional and adverbial relations.
   - Noun handler: Extracts relations between noun sequences and cardinals.
   - Adjective handler: To extract adjectival relations.
   - Verb handler: To extract relations between verb and its subject and object.

Similar steps are followed for query CG construction except that alternate representations are prepared for query using the set of replaceable terms. For example suppose "article", "literature", "paper", "information" etc. are in the set of replaceable terms for "document" then additional CGs will be added corresponding to each of these while processing the query "Role of computers in retrieval of scientific documents".

Document 1
Abstract
The American society has begun an analysis of the role of computer in the reproduction, distribution, and retreival of scientific information.

Figure 3. Sample Document

Example: The conceptual graph representation of sample document shown in figure 3 is explained below:

The tagged representation of the document is :

%% document 1

%% abstract

the	DT		
american	JJ	computer	NN
chemical	JJ	in	IN
society	NN	the	DT
has	VBZ	reproduction	NN
begun	VBN	,	,
an	DT	distribution	NN
analysis	NN	,	,
of	IN	and	CC
the	DT	retrieval	NN
role	NN	of	IN
of	IN	scientific	JJ
the	DT	information	NN

The tagged text is then passed through the four modules discussed above. The linear form of the CG produced by module 1 is :

[Analysis] → (PMOD_OF) →[role: #] →(PMOD_OF) →[computer: #]

-

→ (PMOD_IN)→[reproduction]→(PMOD_OF)→ [information:#]

→ (PMOD_IN)→[distribution]→(PMOD_OF)→ [information: #]

→ (PMOD_IN)→[retrieval]→(PMOD_OF)→ [information: #]

[1] Use of TnT was possible through an evaluation license agreement. Tagset used is Penn TreeBank.

The CGs produced by module 3 are:

[society : #]-

→(attr)→[american]

→(attr)→[chemical]

[information] →(attr) →[scientific]

Module 4 adds following CG:

[analysis] ← (ptnt) ← [begin]→(agnt) →[society]

3.2.2 Conceptual Graph Similarity Measure

In the second stage of the retrieval process the query CG is compared with conceptual graphs of those sentences that contain query concepts. We have not used graph derivation for matching query and sentence CG, rather these conceptual graphs have been compared using the similarity measure proposed by Gomez et al (2000). Given two texts represented by the conceptual graphs *G*1 and *G*2 respectively and their intersection graphs G_c, the similarity *s* between them will be a combination of their conceptual similarity s_c and relational similarity s_r:

$$s = sc \times (a + b \times sr) \qquad \text{(i)}$$

The intersection graphs G_c, consists of the following elements:

- All concept nodes that appear in both the conceptual graphs G1 and G2.
- All relation nodes that appear in both the initial conceptual graphs and relate the same concept nodes.

The conceptual similarity s_c and relational similarity s_r are obtained using the following expression:

$$sc = \frac{2 \times n(Gc)}{n(G1) + n(G2)} \qquad sr = \frac{2 \times m(Gc)}{mGc(G1) + mGc(G2)}$$

The values of coefficients a and b depend on the structure of document and are computed as:

$$a = \frac{2 \times n(Gc)}{2 \times n(Gc) + mGc(G1) + mGc(G2)}$$

and b = 1- a

where, $m(G_c)$ is the number of the arcs in the graph G_c, $mG_c(G)$ is the number of the arcs in the immediate neighborhood of the graph G_c in the graph G.

Various similarity values obtained using equation (i) have been combined to get a single score for document as:

$$S = \alpha \times \frac{\sum si}{c} + \beta \times \max(si),$$

where, c is the number of CGs in a document having one or more matching concept with query CG.

The first factor in this expression ensures that a document having a number of identical repetitions of query concepts will score more than one having a single identical and many different occurrences.The second factor ensures that a document having a single exact match and many partially matching fragments have fair chances of being ranked better. A high value of α improves ranking of a document having many repeated occurrences of partially matched fragments. A high value of β improves ranking of a document having a single matching CG fragment. We have given equal weight to average and maximum similarity values ($\alpha = \beta = 0.5$). These values have been set empirically.

Retrieval Algorithm

```
Algorithm Retrieval (FILE *dfreq, *doccg, int N) //dfreq -
document frequency file
1. Start                    // N- collection size
2. read query
3. extract query terms and prepare query vector(qv).
4. Prepare query CGs.// Alternate representation possible
5. for i = 1 to N do
     read term vector for document i
    s_i.score = "dv_i x qv_i
    s_i.doc = i
   end for
6. sort(s)// Sort on score to get a ranked list of documents
7.cut_off= get_maxrecall(s, qrels)//For experiment 2 & 3 cut
off is specified as no. of documents
  8.c = count of documents up to cut-off value // i.e. No. of
documents retrieved for highest recall value
 9.  for i = 1 to c do
       for each CG (j) in document s_i.doc having a matching
concept
      cgf_j = CG similarity value between query CG and j^th CG.
     end for
     savg = " cgf_j/k            // k = total number of matching CGs
     smax = max( cgf_j )
    cgs_i.score = a . savg. + b . smax //  a=b=0.5
    cgs_i.doc = s_i.doc
   end { for}
10. sort(cgs_i )// To get a ranked list of document
11. end {of algorithm}
```

Figure 4. Retrieval Algorithm

4. Experimental Design

In order to test the effectiveness of our model and to compare the improvement with other models we have performed three experiments. A test run is made on CACM-3204 collection. To test the capability of our retrieval model to capture context information and improve ranking we performed another experiment on a document collection specifically designed to contain documents having similar terms being used in different contexts. In an attempt to investigate a possible application of our approach on top of existing retrieval systems we performed a third experiment using the top 10 results returned by LYCOS browser. The retrieval algorithm shown in fig.4 explains the approach followed in these experiments.

4.1 Experiment 1

The document collection used in the first experiment is CACM-3204. This collection has 3204 documents and 64 queries with known relevance judgment. A list of 429 stop words is also provided. The objective of this experiment was to (i) compare the performance of the weighting function used by us with one of the recent weighting scheme, the BM25 and (ii) see the performance of our model on an existing document collection with known relevance judgments.

4.1.1 The Experiment

To see the performance of the two weighting functions, the two coefficients, k1 and b, of BM25 has been set empirically to 2 and 0.1 respectively. Following Robertson and Jones (1997) we set k1 to 2 and empirically set b to 0.1. Figure 5 shows recall-precision curve for the CACM collection for the two weighting schemes used. 11-point average precision (averaged over 52 queries of the collection) has been used.

To test the effectiveness of our two stage retrieval model we have compared the precision after first and second stage of retrieval. Wefirst retrieved documents using vector model in stage 1 and then used CG-based based representation to re- rank documents. We have considered documents up to highest recall point. The queries considered are query 12, 19, 23, 30 and 63 of the CACM-3204 collection. The query 23 was selected as it experienced lowest precision with "modified tf-idf" model used in first stage. The observed precision for query 30 was quite close to average precision. The performance of remaining three queries was better than average. The ranking of relevant documents after first and second stage for query 12 ("portable operating system") and query 23 ("Distrib

uted computing structures and algorithms") is shown in Table 1. The total number of relevant documents for query 12 and 23 is 5 and 4 respectively. Table 2 lists the precision after first and second stage and % improvement for queries considered in this paper. The average recall-precision graph for the subset of four queries has been shown in figure 6. Query 23 has been excluded from this, as this single query might contribute a lot to the average performance.

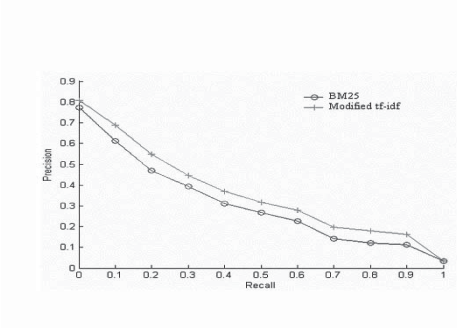


Figure 5. Recall-Precision curve (averaged over 52 queries) for BM25 and modified weighting tf-idf scheme

4.1.2 Results and Discussions

As shown in figure 5, the average performance of "modified tf-idf scheme" was found better than BM25 scheme. The observed mean average precision (averaged over 52 queries) for "modified tf-idf" and bm25 scheme was 35.9% and 30.5 % respectively. So we have used "modified tf-idf" scheme in our experiment.

Document #	Rank		Document #	Rank	
Relevant	First Stage	Second Stage	Relevant	First Stage	Second Stage
1523	391	60	2578	64	43
2080	34	30	2849	29	2
2246	4	3	3137	36	33
2629	26	25	3148	47	1
3127	1	1			

Table 1. Ranking of relevant documents for CACM query 12 and 23

Table 1 show that the ranking of the relevant documents for query 12 and 23 of the CACM-3204 collection has been improved after second stage. This improvement is because of the consideration of semantics through the conceptual graph in the second stage. The vector model gives high rank to documents containing frequent occurrences of terms like 'operations' and 'operator' or including a discussion like 'portable random generator'. Similarly for query 23 ("Distributed computing structures and algorithms") vector model gives high ranks to documents pertaining to "probability distribution", "tree structure" etc. CG-based model improves ranking by eliminating these documents resulting in improved precision.

Figure 6 shows 11-point standard recall-precision curve for query19 and query 63 of the CACM-3204 collection after first and second stage of retrieval. Extrapolated precision is used for all recall points that exceed the maximum observed recall value. As shown in Table 2, we experienced increase in precision after second stage in all the five queries. The maximum increase in precision of 979.6% was observed for query #23. During the first stage of retrieval the first relevant document for this query was obtained after retrieving 29 documents. With CG based ranking this figure was reduced to 1. The minimum improvement of 3.6% was observed for query 12. For query 19 these figures were 48.6% and 59.5%. This represents 22.3% increase in precision. The mean average precision for four queries after first and second stage of retrieval was observed as 46.4% and 62.6%, resulting in an improvement of 34.8%. This improvement is more than significant.

Query #	Precision (%) Stage 1	Precision (%) Stage 2	Increase in Precision (%)
12	42.7	44.3	3.7
19	48.6	59.5	22.4
23	5.4	57.8	970.4
30	35.1	51.5	46.7
63	44.8	74.5	66.3

Table 2. Percentage increase in precision

4.2 Experiment 2

4.2.1 Document collection (CGDOC)

In the second experiment we have considered our own document collection specifically designed so as to contain identical terms being used in different areas to further investigate the fruitfulness of our approach. This is a small collection consisting of abstracts of 70 documents and 10 queries. We however do not provide enforce our own relevance judgment, instead list the titles of the documents returned for the queries considered in the paper so as to enable understanding of the improvements being made. The objective was to test the retrieval effectiveness of our model in an environment where differentiation among documents may be difficult solely by statistical means. The average length of the document in the collection is 71.9 words excluding stop words. The size of smallest and largest document in the collection is 12 words and 136 words respectively. Fig.8 lists titles of few of the documents (titles only) and sample queries in this collection.

4.2.2 The experiment

When a query is made, query vector and query CG is prepared to be used in the first and the second stage of retrieval process respectively. In the first stage of retrieval a ranked list of the documents is returned using modified vector model to be used in second stage. The last observed recall point is used as cut-off.

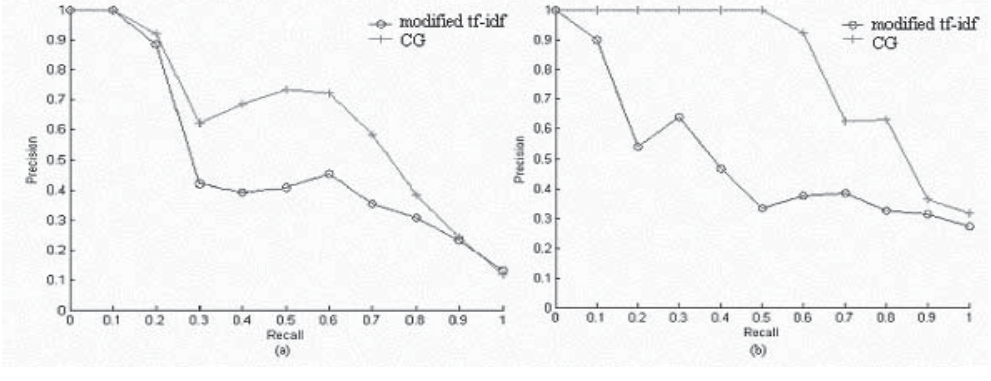


Figure 6. Recall-Precision curve for (a) CACM query 19 (b) CACM query 63

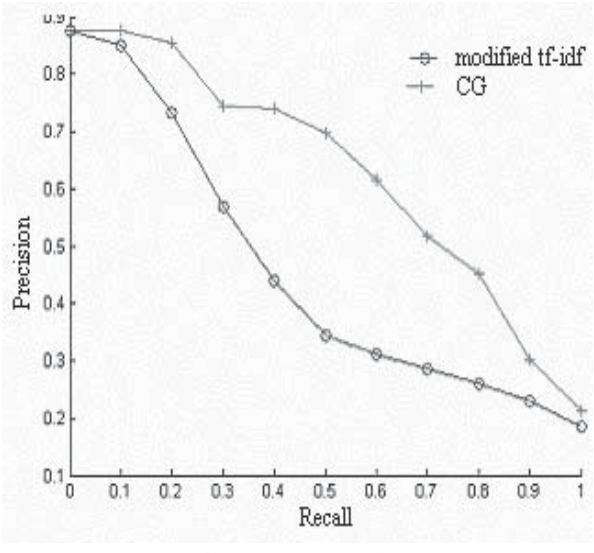


Figure 7. Average Recall-Precision curve for CACM queries 12, 19, 30 and 63

We compare query CG with sentence CGs of each document returned by first stage. As there can be many sentences in a document containing query terms, a combination of average and maximum similarity value has been used for ranking purpose. To better explain the effect of our CG-based approach we have made a comparison of the ranking obtained after first and second stage of retrieval instead of forcing our own relevance judgments. The titles of documents under consideration for query 1 have been listed in fig.8.

Query #1. Genetic algorithm for information retrieval
Query #2. Fuzzy information retrieval
Query #3. Information retrieval using conceptual graph
Doc#2 An Efficient Information Retrieval Method in WWW Using Genetic Algorithms
Doc #9 An extended inverted file approach for information retrieval
Doc #11 Genetic algorithm based redundancy resolution of robot manipulators
Doc #13 Genetic algorithms: A Survey
Doc #14 Genetic Algorithm and Graph Partitioning
Doc #21 An Image Retrieval Method Based on a Genetic Algorithm
Doc #22 Evolutionary Reinforcement of User Models in an Adaptive Search Engine
Doc #24 Genetic algorithm approach to image segmentation using morphological operations
Doc #25 Multiprocessor Document Allocation: A Genetic Algorithm Approach
Doc #27 Probabilistic and genetic algorithm for document retrieval
Doc #42 Intelligent Agents and its Applications in Information Retrieval

Figure 8. Sample queries and Document titles

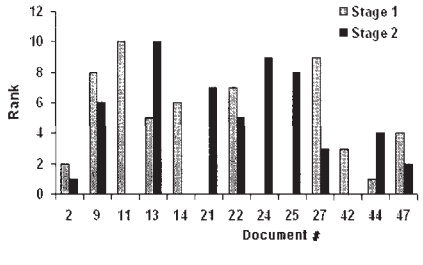


Figure 9. Ranks of documents after first and second stage

4.2.3 Results and Discussions

The effect of CG application is that relevant documents that were ranked low by vector model were shifted up in the ranking. The ranking of top 10 documents after first and second stage of retrieval for the query "genetic algorithm in information retrieval" is shown in fig. 9. It can be verified easily from the titles that documents ranked 3 and 5 (Doc#13 and #42) returned by first stage are not as much useful for the query under consideration as doc#22 and doc#27 which are ranked 7 and 9 respectively. Application of CG improves ranking by including doc#22 and doc#27 in top five documents, resulting in improved precision. The vector model fails to identify the similarity between "document retrieval" and "information retrieval" and gives a low rank to doc#27. Our CG-based ranking is able to detect this similarity and improves the ranking of this document. Doc# 14 is ranked 6 after stage 1 but is excluded from the list of top 10 documents after stage 2. This is because of the relatively high frequency of the terms "genetic" and "algorithm", all of which do not appear together. The occurrence of "genetic" and "algorithm" in close context results in high CG similarity value as compared to the case when these terms appear in a different context, such as "simulated annealing algorithm". These occurrences of "algorithm" contribute to low average similarity value. As the second stage of ranking is a combination of average and maximum CG similarity values, this document is ranked low. In contrast to this doc#21 has more relation match (as it talks about use of genetic algorithm in retrieval) resulting in a high CG similarity value, even though the frequency of these terms is not as high as in doc#14. Similar arguments hold for doc#24 and doc#25. Fig. 10 shows the ranking after first and second stage of retrieval for query 2 and 3 listed in fig. 8. The titles of the documents being considered can be found in Appendix II. Similar improvements in ranking have been achieved for other queries also.

4.3 Experiment 3

In order to investigate a possible application of our approach in web search environment we have conducted a small experiment. In this experiment we have simply considered first ten results returned by LYCOS search engine for query#1 of our collection (See Appendix I) and then constructed conceptual graphs for fragments of sentences returned by search engine, involving query terms and compared it with query graph. LYCOS has been used because it is based on simple keyword search. Unlike other search engines it does not consider the position of keyword in the document, number of inbound hyperlinks etc. in the ranking process. This more closely corresponds to the retrieval model used in the first stage of retrieval. The result of this comparison is then used for ranking. The result is shown in figure 11. We have achieved significant improvement in this case also, even though we have just used very little piece of context information (highlighted in Appendix I), as we do not have access to actual document representation.

5. Conclusions

We presented a hybrid two stage information retrieval model. This model first retrieves a set of potentially relevant documents to query using modified vector model and then ranks documents based on conceptual understanding of terms. We have observed an increase of 34.8% in precision for a subset of CACM queries. The experiment performed by us on our own document collection also shown significant improvement in the ranking of retrieved documents. This is because of the semantic considerations in second stage of retrieval through conceptual graphs. A specific query gives more useful relationships among query terms that will help in improving ranking.

The syntactic model proposed by us helps in quickly short listing the relevant documents from a large document set without hampering efficiency. Application of CG-based model in the second stage brings conceptual understanding in making relevance judgment. Relevance based on semantics more closely correspond to users' mental model resulting in improved acceptance for user.

In order to make our CG model more efficient we have used a new scoring function to get the combined document score. The form of graph matching function used by us keeps the computational cost low. The CG representation used by us is easily scalable. Further we have also proposed a tabular representation of CGs to reduce matching time.

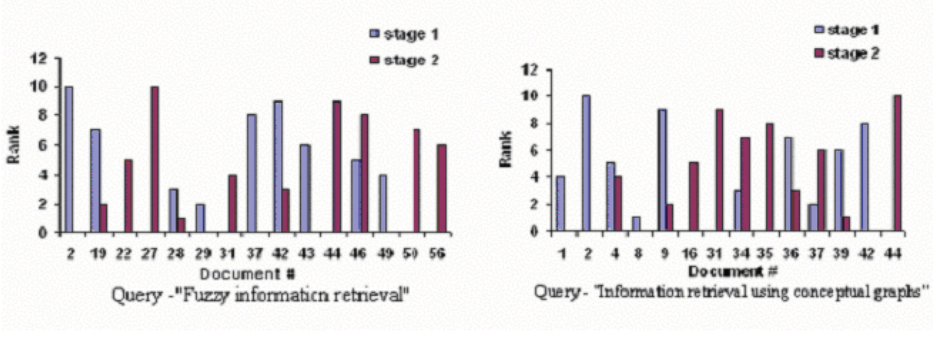


Figure 10. Ranking after first and second stage of retrieval for query 2 and 3

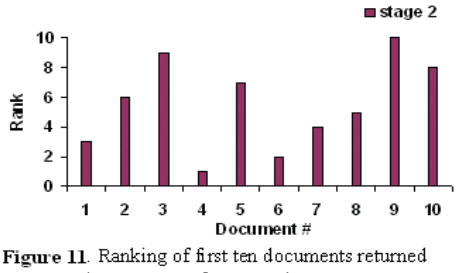


Figure 11. Ranking of first ten documents returned by LYCOS after second stage.

The proposed model takes the advantages of both the efficiency of the syntactical approach and the accuracy of the semantic approach. The improvement observed when the approach is applied on the results returned by LYCOS demonstrates the potential of our model. This suggests that CG-based model can be used as a precision tool with existing search and retrieval systems.

Acknowledgement

We would like to thank Thorsten Brants for providing an opportunity to use Trigrams'n'Tags (TnT) part of speech tagger.

References

[1] Deerwester S, Dumais Susan T, Furnas, George W and Landauer, Thomas K (1990) Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science

[2] Dick, Judith P(1992) A conceptual case relation representation of text for intelligent retrieval, Technical Report CSRI-265.

[3] Dominich Sander(2001) Mathematical foundation of Information Retrieval. In R. Lowen (Ed.) Mathematical modeling: Theory and applications, Kluwer Academic Pub.

[4] Foltz, Peter W(1990) Using Latent Semantic Indexing for Information Filtering. The ACM conference on office information system (COSIS'90)

[5] Hiemstra D, de Vries Arjen P (2000) Relating the new language models of information retrieval models; published as CTIT technical report TR-CTIT-00-09, May 2000, http://www.ctit.utwente.nl

[6] Khoo, C (1997) The Use of Relation Matching in Information Retrieval, LIBRES: Library and Information Science Research, 7(2).

[7] Liddy ED, Myaeng SH(1994) DR-LINK: a system update for TREC-2. Second Text REtrieval Conference (TREC-2) (NIST-SP 500-215).NIST. pp.85-99. Washington, DC, USA.

[8] Marega R, Pazienza MT(1994) CoDHIR: an information retrieval system based on semantic document representation. Journal of Information Science, 20(6), pp.399-412. UK.

[9] Martin P (1997) CGKAT: A knowledge acquisition and retrieval tool using structured documents and ontologies. In: Lukose D, Delugach H, Keeler M, Searle L, Sowa JF (eds) Proceedings of ICCS'97 (Lecture notes in artificial intelligence 1257). Springer, Berlin Heidelberg New York, pp 581–584.

[10] Martin P and Eklund P(2000) Embedding knowledge in Web documents, Griffith University, Australia, http://decweb.ethz.ch/WWW8/data/2145/html/bindex.htm, April 24

[11] Mittendorf, E., Mateev, B. and Schäuble, P.(2000). Using the co-occurrence of words for retrieval weighting. Information Retrieval, 3, 243-251.

[12] Montes-y-Gómez M, López-López, A and Gelbukh A (2000) Comparison of conceptual graphs, Cario O, Sucar LE, Cantu FJ (Eds.) MICAI 2000: Advances in Artificial Intelligence. Lecture notes in Spriger –Verlag, pp. 548-556

[13] Montes-y-Gómez, M, Gelbukh A, López-López A and Baeza-Yates(2001) Flexible Comparison of Conceptual Graphs. Lecture notes in Computer Science 2113, Springer-Verlag

[14] Mugnier M L, On Generalization/Specialization for Conceptual Graphs. Journal of Experimental and theoretical Artificial Intelligence, 7 (1995)325-344.

[15] Oren, Nir (2002) Reexamining tf.idf based information retrieval with Genetic Programming. Proceedings of SAICIST 2002, p.224-234.

[16] Ounis I (1998) Modeling, indexing and retrieving images using conceptual graphs. In: Quirchmayr G, Schweighofer E, Bench-Capon T (eds) Proceedings of DEXA'98 (Lecture notes in computer science 1460). Springer, Berlin Heidelberg New York, pp 226–239.

[17] Rama, D. V, Srinivasan P (1993) An investigation of content representation using text grammars ACM transactions on Information Systems 11(1) pp. 51-75.

[18] Robertson et al (1994) Okapi at TREC-2. In the second Text Retrieval Conference(TREC-2) edited by D K Harman, Gaithersburg, MD: NIST, 1994.

[19] Robertson S, Walker S, Jones S, Hancock-Beaulieu M and Gatford M (1995) Okapi at TREC-3. In: Harman D (Ed.), The Third Text REtrieval Conference (TREC-3). NIST Special Publication 500-225.

[20] Robertson, S. E. & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag, New York, 232-241.

[21] Robertson SE and Jones KS (1997) Simple proven approaches to text retrieval http://www.n3labs.com/ pdf /robertson97simple.pdf

[22] Salton G and Buckley C (1988) Term Weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513–523.

[23] Savoy J and Picard P (2001) Retrieval effectiveness on the Web, Information Processing and Management, 37(4), 643-569 Smeaton Alan F(1995) Natural Language Processing and information retrieval. A tutorial presented at EACL'95 .

[24] Sowa JF (1984) Conceptual structures – Information processing in mind and machine, Addison –Wesley.

[25] Sowa JF(1993) Relating diagram to logic. In Eds. Guy W. Mineau, Bernard Moulin and John F. Sowa, Proceedings of first International conference on conceptual structures, ICCS'93, Quebec city, Canada, August 4-7

[26] Strzalkowski Tomek(1995). Natural language information retrieval. Information Processing and Management, 31(3), pp. 397-417.

[27] Watters CR (1989) Logic framework for information retrieval. Journal of the American Society for Information Science, 40(5), 311-324.

Appendix I

First ten results returned by LYCOS for the query "**Genetic algorithm for information retrieval**"

1. Machine **Learning for Information Retrieval**: Neural Networks, Symbolic...
... Results of **genetic algorithms** testing Machine **Learning for Information Retrieval** : Neural ... **algorithms in information retrieval** . In [48 ... presented a **genetic algorithms** based approach for...
More results from: ai.bpa.arizona.edu/papers/mlir93/mlir93.html February 3, 2004 - 163 KB *3 (0.639)

2.The Art Site on the World Wide Web: McLaughlin
... electronic palettes, generated from **genetic algorithms**, or captured by digital camera ... fractal animations, **genetic algorithm** animations, morphs, ray trace ... retrieving files from the site)....
cwis.usc.edu/dept/annenberg/artfinal.html March 11, 2004 - 90 KB **6** (0.531)

3. text mining and web-based **information retrieval** reference
... Web Mining, **Information Retrieval** and ... Fast Look-up **Algorithm for Structural** ... Web Mining: Information and Pattern ... adaptive **genetic algorithm** /programming ... statistics to **information** ...
filebox.vt.edu/users/wfan/text_mining.html February 2, 2004 - 23 KB **9** (0.477)

4. Effective **Information Retrieval Using Genetic Algorithms** Based...
... Effective **Information Retrieval Using Genetic Algorithms** Based Matching ... relevant **information** from these ... improving retrieval performance ... and recall) for **retrieval** ... the area of...
www.computer.org/proceedings/hicss/0493/04932/0493201... August 22, 2002 - 10 KB **1** (0.853)

5. Machine Learning/Genetic Algorithm group - Dipartimento di...
... representation for ML, such as exploitation ... to also include **Genetic Algorithms** and Neural Networks ... Mobile **Agents for** automatic **information retrieval** , network management ... Learning Agents...
www.di.unito.it/~mluser/ February 3, 2004 - 8 KB **7** (0.522)

6. Personal **Information** Intake Filtering
... Recent work on **genetic algorithms for information retrieval** [Gordon] focused ... program uses **genetic algorithms for evolving** symmetrical ... B Cousins. "**Information Retrieval** from Hypertext...
www.baclace.net/Resources/ifilter1.html July 27, 2001 - 42 KB **2** (0.752)

7. Free C/C++ Sources for Numerical Computation
... Description : many **genetic algorithm** optimisation libraries ... Description : Objects for doing **genetic algorithm** optimization Name ... analytic statistics for the TREC IR trials Comments :...
cliodhna.cop.uop.edu/~hetrick/c-sources.html July 24, 2002 - 97 KB **4** (0.560)

8. Home Page for Haym Hirsh
... Directors: Institute for the Study of ... machine learning, **information retrieval** , data mining ... engineering design, **genetic algorithms** , knowledge representation ... Publications Information ...
www.cs.rutgers.edu/~hirsh/ March 8, 2004 - 18 KB **5** (0.560)

9. Artificial Life
... Related Topics **Genetic Algorithms** Agent technologies ... computation such as **genetic algorithms** (GAs), evolutionary ... stands for "**Adaptive Retrieval** Agents Choosing Heuristic Neighborhoods for...
www.insead.fr/CALT/Encyclopedia/ComputerSciences/AI/a... January 12, 2004 - 56 KB **10** (0.460)

10.**Information** and Communication R&D Center
Information and Communication ... Fuzzy Theory, **Genetic Algorithm** , Artificial Life ... JAPANESE) **Information Retrieval** (IR) (JAPANESE ... Service Other **Information** Abstract ... TOWN GUIDE for ...
www.ricoh.co.jp/rdc/ic/index_e.html September 30, 1999 - 3 KB **8** (0.494)

Appendix II

Doc#1 An Efficient Storage and Retrieval System for Conceptual Graphs

Doc #4 The RELIEF Retrieval System (Image retrieval system based on Conceptual graphs)

Doc #8 Compiling Conceptual Graphs

Doc #9 An extended inverted file approach for information retrieval
Doc #16 PRIME-GC A medical information retrieval prototype on the Web

Doc#19 Measuring Effectiveness in Fuzzy Information Retrieval

Doc #28 Application of fuzzy set theory to extend Boolean Information retrieval

Doc #29 Fuzzy functional dependency and its application to approximate data querying

Doc #31Improving the Performance of Existing Information Retrieval Systems Using a Software Agent

Doc #34 Flexible Comparison of Conceptual Graphs

Doc #35 CG-DESIRE: Formal Specification Using Conceptual Graphs

Doc #36 Comparison of Conceptual graphs

Doc #37 Fuzzy conceptual graphs for matching images of natural scenes

Doc #39 Text mining at detail level using conceptual graphs

Doc #43 Knowledge Representation Using Fuzzy Petri Nets - Revisited

Doc #49 fuzzy integral as a basis for the interpretation of flexible queries involving monotonic aggregates

Doc #50 Fuzzy Content-Based Retrieval In Image Databases

Doc #56 An information retrieval model based on vector space method by supervised learning

* Rank (Score) obtained through CG

Tanveer J. Siddiqui received her M.Sc. degree in Computer Science from University of Allahabad. She submitted her thesis in December 2005. Her thesis work was in the area of information retrieval. She served as a scientist in DRDO project from April 1996 to December 1998. She has been counselor at Indira Gandhi National Open University, India and guest faculty at Indian Institute of Information Technology, Allahabad. She is a faculty member at University of Allahabad since January 2000. Her research interests include information retrieval, text and data mining, document summarization and intelligent agent technology.

Uma Shanker Tiwary received his B.tech. and P.hd. degree in Electronics engineering from Banaras Hindu University in 1983 and 1992. He has 21 years of experience in research and teaching. He served as lecturer at M.M.M. Engg. College, Gorakhpur, as lecturer and reader in University of Allahabad and as a visiting scientist at IIIT, Kanpur. Currently he is associate professor at IIIT, Allahabad and visiting IT Professor, GIST, South Korea. He sponsored a number of research projects and organized many international conferences. His research interests include Medical Image Processing, Image Processing, Computer Vision, Soft Computing & Fuzzy Logic, and Language and Speech Technology.