Sādhanā Vol. 40, Part 6, September 2015, pp. 1793–1819. © Indian Academy of Sciences

Semantic association ranking schemes for information retrieval applications using term association graph representation

K VENINGSTON*, R SHANMUGALAKSHMI and V NIRMALA

Department of Computer Science and Engineering, Government College of Technology, Coimbatore, India
e-mail: veningstonk@gct.ac.in; cseit.gct@gmail.com

MS received 31 August 2014; revised 30 March 2015; accepted 6 June 2015

Abstract. Most of the Information Retrieval (IR) techniques are based on representing the documents using the traditional vector space and probabilistic language model i.e., bag-of- words model. In this paper, associations among words in the documents are assessed and it is expressed in Term Association Graph model to represent the document content and the relationship among the keywords. Earlier attempt on exploiting term association graph was done for non-personalized document re-ranking task. This paper experiments improved non-personalized and personalized re-ranking strategy which exploits term association graph data structure to assess the importance of a document for the user query and thus documents are re-ranked according to the association and similarity exists among the documents. This paper proposes various approaches under two models namely, Term Rank based Approach (TRA) and Path Traversal based Approaches (PTA1, PTA2, and PTA3). These approaches employ term association graph and has been evaluated using manually prepared real dataset and benchmark OHSUMED dataset. The results obtained are reasonably promising.

Keywords. Information retrieval; personalization; term association graph; ad-hoc retrieval; re-ranking.

1. Introduction

Typically, Information Retrieval (IR) systems (Salton & McGill 1986; Baeza-Yates & Ribeiro-Neto 1999; Manning *et al* 2008) have been used for the task of ranking documents for a given query from the vast collection. Most IR system uses keywords to match with retrieve documents in order to find relevant document to be presented to the user. As the cost of storage devices continues to decrease, there is a tremendous growth in databases of all sorts including relational, graphical, textual and multimedia contents. IR has become one of the dominant area of research in web mining due to the growth and evolution of web documents. IR process faces the problems

*For correspondence

of information mismatching and overcapacity. Besides, re-ranking problem in IR is assumed as a large scale problem in order to organize these documents automatically as World Wide Web accumulate documents rapidly. As the amount of information on the Web increases speedily, it creates many new challenges for large scale ad-hoc retrieval system such as web search. When the query is submitted by a user, a typical search engine returns a large set of results. Users may be expecting relevant documents in the first few pages of search results for the query. Most of the state-of-the-art retrieval techniques adopt the approach of transforming the document retrieval problem into machine learning problem. Typically, the documents are represented using the popular Vector Space Model (VSM) (Wong & Raghavan 1984). Intuitively, the documents are preprocessed in order to prepare a list of terms with corresponding term frequencies.

As the key issue with the abundance of online information is to find relevant documents in the higher order, this paper exploits term association graph model for document representation in order to re-rank the documents so as to bring the documents in the order of relevance. The proposed approach also exploits personalization features to prepare personalized ranking of documents.

2. Background

2.1 *Baseline models*

Existing web search engines rank Web pages mainly based on keyword matching and hyperlink structures (e.g., authorities and hubs) (Kleinberg 1999; Brin & Page 1998; Page *et al* 1998; Eirinaki & Vazirgiannis 2005; Haveliwala 2003). Not much importance has been paid to measure the informative values of Web pages. The following are the techniques to represent web documents for the application of web IR. BM25 (Best Matching) (Robertson *et al* 1992) is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document. The probabilistic retrieval system (Sparck *et al* 2000) responds to each search request in order to rank the documents in the collections in the order of decreasing probability of usefulness to the user who submitted the request. The system estimates the probabilities as accurately as possible on the basis of whatever data made available to the system for this purpose. Language model (Berger & Lafferty 1999; Ponte & Croft 1998; Croft & Lafferty 2010) is built for each document. The base of the language model is probabilistic model. In language modeling approach, a separate language model is estimated for each document. The LM for document d is a probability distribution $p(w|d)$ over the words w in the vocabulary. The query likelihood $p(q|d)$ is calculated by assuming that the query terms are independent, and then multiplying the probabilities for the individual terms. The smoothing techniques for LM applied to IR are discussed in (Lafferty & Zhai 2001). The smoothing methods combine the document model $p(w|d)$ with the collection model $p(w|C)$ in order to smoothen the probabilities. Two major problems that arise using the VSM are synonymy i.e., many ways to refer to the same object which leads to poor recall, e.g., car and automobile and polysemy i.e., most words have more than one distinct meaning which leads to poor precision, e.g., model, python, and chip. LSI finds the hidden semantic meaning of terms based on their occurrences in documents. LSI is a technique that maps query terms and documents to a latent semantic space. Comparing terms in this space would make synonymous terms look more similar. In the latent semantic dimension, a query and documents can have high cosine similarity even if they do not share any terms. Co-occurring terms are projected onto the same dimensions. Dimensions of the reduced space correspond to the axes of greatest variation. pLSA has been evolved from LSA with probabilistic model. LSA (Scott *et al* 1990) and pLSA (Hofmann 1999) are representative probabilistic topic modelling for identifying word/document

relationship. Typically, document is a mixture of topics as in pLSA, but according to a Dirichlet prior θ. When a uniform Dirichlet prior is used, pLSA and LDA become same. A word is also generated according to another variable β. α and β are corpus level parameters sampled once per corpus. α is the parameter of the Dirichlet prior on the per-document topic distributions i.e., α tells how much Dirichlet prior scatters around the different topics z. β is the parameter of the Dirichlet prior on the per-topic word distribution i.e., distribution over topics. θ is the topic distribution for document. In order to compute topics of a given document, it is essential to compute the posterior distribution of the hidden variables given in a document.

2.2 *Term graph model*

In this model, text documents are modeled as a graph whose vertices represent words, and whose edges denote meaningful statistical (e.g., co-occurrence) or linguistic (e.g., grammatical) relationship (Pado & Lapata 2007) between the words. There are different types of text graph which includes thesaurus graph (Leicht *et al* 2006), concept graph (Montes-y-Gomez *et al* 2000), **Syntactic-semantic association graphs** (Nastase *et al* 2006), **Co-occurrence graph** presented in Masucci & Rodgers (2006), and so on. *Thesaurus graph* denotes terms as vertices and sense relations e.g., synonymy or antonomy as edges. *Concept graph* denotes concepts as vertices and conceptual relations e.g., hypernymy or hyponymy as edges. The graph model proposed in (Blanco & Lioma 2012) incorporates topological properties of graph such as degree distribution, average path length, and clustering coefficient. Degree distribution gives the probability that a randomly selected vertex v_i will have degree k =1,2,3... edges i.e., degree of a vertex v_i is defined as the number of edges adjacent to v_i. Average path length is the average number of edges in the shortest path between any two vertices in a graph. Average clustering coefficient of the graph is computed by averaging the clustering coefficients of all vertices in a graph. Clustering coefficient of a vertex measures the proportion of its neighbors that are themselves neighbors. The average clustering coefficient can be used to identify connected graph partitions i.e., it defines the strength of connectivity within the graph. The two variants of text graphs have been presented in Blanco & Lioma (2012) namely, undirected co-occurrence text graph and directed co-occurrence text graph with grammatical constraints. The major limitation of this approach is that it does not preprocess the documents before constructing the text graph. Text graph denotes document based graphs and not collection based graphs. Therefore, this approach constructs individual graph for every text document which may increase the computational complexity when using it for re-ranking process. Thus, the proposed approach constructs collection based graph which is more appropriate for the document ranking process.

2.3 *Concept graph model*

Graph model has been adapted to a concept representation of documents which captures the dependencies between concepts found in medical document text. Concept graph proposed in Koopman *et al* (2012) integrates graph-based term-weighting model into concept-based approaches to medical IR. The concept graph is constructed similar to that of the construction of term graph (Blanco & Lioma 2012). A fixed length context window is moved across a document and thus the concepts which co-occur within the context window are connected with an edge in the concept graph. Even though the process of building the term graph and concept graph is similar, the resultant concept graph differs significantly. As a single term can map to multiple concepts, there are many more concepts in the concept graph than terms in the term graph.

Typical graph-based retrieval functions estimate the relevance between a document and a query as shown in Eq. (1).

$$R(d, q) \approx \sum_{t \in q} w(t, q) * w(t, d), \tag{1}$$

$w(t, d) = idf(t) * S(v_i)$, where $w(t, q)$ is the weight of the term in query, $w(t, d)$ is the weight of the term in the document, $S(v_i)$ is the vertex score computed using Page Rank algorithm (Page *et al* 1998) and $idf(t)$ is the inverse document frequency of the term. The graph-based score provides a means of estimating $w(t, d)$. The graph-based term weighing method is then applied in order to use concept-based representation. Thus, the typical term weighting function is updated to weight a concept c within document d_c as shown in Eq. (2).

$$w(c, d_c) = idf(c) * S(v_i). \tag{2}$$

Furthermore, background smoothing is applied similar to the language models based on term frequency within the corpus. A concept weight is therefore adjusted based on its background weight within the corpus of medical domain.

$$w'(c, d_c) = idf(c) * S(v_i) * \log(|V_s(c)|) \tag{3}$$

$$R(d_c, q_c) = \sum_{c \in q_c} w(c, d_c), \tag{4}$$

where $V_s(c)$ is the set of edges adjacent to concept c in the ontology graph, d_c is the document converted to concepts and q_c is query converted to concepts. The retrieval function is updated as shown in Eq. (4) and then employed for ranking documents in the corpus.

2.4 *Types of text graphs*

The goal of IR is to effectively retrieve documents relevant to users' queries. Graph-based document ranking algorithms have been widely used in calculating term weights to represent the contribution of a term in search context. In this research, the variants of IR techniques utilizing graph representation have been investigated. **TextRank** (Mihalcea & Tarau 2004) is a graph-based ranking model for natural language text processing. The algorithm takes into account edge weights while computing the score associated with a vertex in the text graph. **Random walk algorithm** proposed in Blanco & Lioma (2007) used to weight terms in the *tf-idf* weighting scheme by adapting the TextRank algorithm (Mihalcea & Tarau 2004). This method uses term co-occurrence as a measure of dependency between words to determine how a word contributes to a given context and this model does not consider edge weights. **Click graph** presented in Craswell & Szummer (2007) models query-document pairs as relevance judgement to produce a probabilistic ranking of documents for a given query. The graph is bipartite with two types of nodes namely, queries and documents. An edge connects a query and a document if there is a click for that query-document pair by any user. The edge is weighted according to the total number of clicks from all users. This model attempted to retrieve relevant documents that have not yet been clicked for that query and rank those documents. This model does not employ document content or query content instead it uses click data alone. The random walk approach is employed on click graph to find association between queries and documents in order to provide query reformulation. **Affinity graph model** (Zhang *et al* 2005) presents affinity ranking mechanism based on the content link structure constructed for entire documents collection. It then employs to link analysis algorithm on affinity graph to compute information richness of

each document to obtain the affinity rank score so as to re-rank the top returned document list. **Graph diffusion model** (Ma *et al* 2012) presents a framework on mining web graphs for recommendation tasks such as web query suggestions, tag recommendations, expert finding, image recommendations, image annotations and so on. **Keyword-URL bipartite graph** (Mei & Church 2008) has been widely used in IR applications wherein every query is connected with a number of URLs on which the users clicked when submitting this query to the search engine. The weights on the edges present how many times the users used this query to access this URL and there are no edge connecting two queries or two URLs. Keyword-URL Bipartite graph has been used to identify implicit topics or facets for web query suggestion and expansion (Jain & Mishne 2010). **Click-through graph** represents the query and click-through page relationships by a directed bipartite graph that consists of a set of queries, a set of web page URLs, and a set of edge. Model presented in (Yi & Maghoul 2009) extracts all maximal bipartite cliques termed bicliques from a click-through graph and computes a query cluster. **Query flow graph** (Boldi *et al* 2008) is an outcome of query-log mining which is built by mining time and textual information as well as aggregating queries from different users. **Ontology model** termed as semantic-based information retrieval technique such as lexical chain has been used in (Sim & Wong 2004) for processing user queries and filtering relevant information to capture the search context of concepts. Semantic-based information retrieval techniques understand the meanings of the concepts that users specify in their queries. **Collocation graph model** is defined as expression of lexical semantic affinities apart from grammatical restrictions. This model allows discovering semantically related words based on their co-occurrences. **Association graph** employs human perceptual and cognitive processes in terms of the organization of the human lexicon e.g., human spoken word associations. **Dictionary graph** is a text graph typically employed for automatic synonym extraction or word sense disambiguation process (Blondel *et al* 2004). In this graph, topological properties are interpreted as indicators of dictionary quality or consistency e.g., WordNet, Wikipedia. **Opinion graph** represents opinions or sentiments linked by lexical similarities. **Resource Description Framework (RDF) Graph** is a directed labeled graph to represent the relationship between entities. The meaningful semantic association paths can be discovered in a RDF graph. In Viswanathan & Ilango (2012), the important path connecting entities are identified depending on user's perspective with varying results for the same query and thus rank the semantic relationship paths.

3. Term association graph model

The term association graph model is an enhanced model of the VSM. The traditional weighting schemes such as Boolean weighting, term frequency (*tf*) weighting and term frequency-inverse document frequency (*tf-idf*) weighting assign weight value to a term according to its importance without considering the term associations. The Boolean weights, *tf* weights and *tf-idf* weights determine the weight of each term in a document independently. Thus, rich information such as relationships existing among the terms in a document is not considered by traditional term weighting schemes. Hence, the term graph model proposed in Wang *et al* (2005) for text classification has been adopted in this approach in order to solve the problem of retrieving relevant document for a user query in IR.

A graph is a pair $G = (V, E)$. The elements of V are the vertices (or nodes) of the graph G and the elements of E are its edges. Usually a graph is represented by drawing a dot for each node and joining two of these dots by a line if they are connected by an edge and labeling of

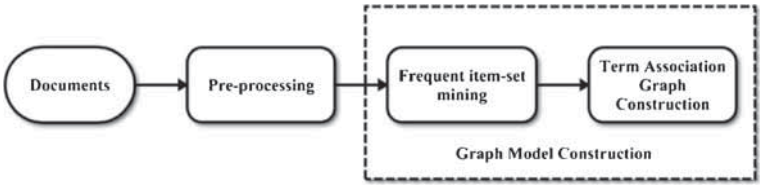

Figure 1. Process involved in term association graph model.

nodes and edges are marked if necessary. Figure 1 shows the process involved in construction of Term Association Graph model.

3.1 *Preprocessing*

Extract all the terms from the collection of documents. Each text document in the corpus is treated as a transaction in which each word is an item similar to that of a transaction in Association Rule Mining (ARM) approach described in (Han & Kamber 2006). However, not all the terms in document are important to be retained in the collection. Typically, there are two-step processes to preprocess natural language text documents. Firstly, stop words removal (e.g., articles, prepositions, etc.). The terms that appear frequently in the document but have no significant meanings may be removed. Secondly, stemming of words is done to keep only the root form of words. The most widely used stemming algorithms are porter and Lancaster algorithm.

3.2 *Graph model construction*

The graph data structure reveals the important semantic relationship among the words of the document when the terms in corpus are expressed as graph model. The features about the documents are extracted using a data mining technique and it is represented in terms of term association graph model.

3.2a *Frequent item-set mining*: The idea here is to capture the relationships among the terms in document using the frequent item-set mining algorithm which is based on Apriori algorithm in Agrawal & Srikant (1994). In this context, an item-set is a set of words that occur together. Each resulting frequent item-set is a document description which is a possible terms or concepts. In this work, an item-set is defined as frequent if it appears in more than two documents.

A widely used ARM called the Apriori algorithm works in two steps. In the first step, it finds all frequent item-sets from a set of transactions that satisfy a user-specified minimum support. In the second step, it generates rules from the discovered frequent item-sets. In this work, the first step alone needs to be performed. Since each transaction denotes a document, it is needed to find only the frequent item-sets as that of the number of documents. After preprocessing, each document in the corpus has been stored as a transaction (item-set) in which each term/concept (item) is represented numerically by a non-negative integer. Then the first step of Apriori algorithm alone has been used to find all the subset of items that appeared more than a user specified threshold (minimum support threshold) in the corpus.

3.2b *Term association graph construction*: The goal is to explore and discover the relationships among terms of the text document in the corpus and to devise a strategy to make use of

these semantic associations in the re-ranking task. The traditional VSM lack in expressing such rich relationship exists among terms. Typically, each term may be associated with more than one term in the corpus. Thus, graph is found to be most appropriate data structure in this context. In order to construct the graph from the set of frequent item-sets mined from the text collections, firstly create a node for each unique term that appear at least once in the frequent item-sets. Secondly, create edges between two node a and b if and only if they are both contained in one frequent item-set. Besides this, the weight of the edge between a and b is the largest support value among all the frequent item-sets that contain both terms i.e., a and b.

Typically, after a query is submitted to a medical information database or search engine, a list of documents is returned to the user. In this work, document denotes the title and abstract of a medical journal article in the corpus. It is assumed that if a query keyword exists frequently in the document, it represents that there is closeness in terms of similarity between the query and the documents that are retrieved. Thus, the support metric defined in Han & Kamber (2006) is employed to measure the interestingness of a particular keyword t extracted from the document for the given query.

$$Support_d = \frac{\sum_{i=1}^{n} f_d(t_i)}{\sum_{j=1}^{N}\sum_{i=1}^{n} f_{d_j}(t_i)} \tag{5}$$

$$f_d(t_i) = \frac{term\_frequency_d(t_i)}{\underset{i}{MAX}(term\_frequency_d(t_i))}, \tag{6}$$

where $f_d(t_i)$ is the *support* of the term t_i defined as the frequency of the term/concept t_i in the document d, n is the number of terms in item-set, N is the number of frequent item-sets i.e., number of documents returned for the query. The minimum support threshold of the term is assumed as 0.05. Table 1 shows a sample frequent item-sets and its corresponding term graph for the query "Ribonuclease". Before the items are extracted, stop words, such as "the", "of" "to", etc., are removed from the documents. The maximum length of an item-set is limited to twenty five words. Thereby the extraction of meaningless terms is avoided and also the computational time is reduced. The length of the item-set may be increased for large scale systems. The term association graph model has been constructed using a benchmark dataset Text REtrieval Conference (TREC-9) Filtering track, which is a medical information database (Hersh *et al* 1994) called the Oregon Health and Science University MEDline (OHSUMED) test collection. The detailed description on the synthetic dataset has been given in section 5 (figure 2).

4. Ranking schemes based on semantic association for re-ranking

The term association graph model reveals richer information i.e., association between the terms exist across different documents. The term association graph model presented in this work is different from the one which is proposed in Wang *et al* (2005). The terms i.e., nodes are considered

Table 1. Frequent item-sets and its corresponding document support value.

Doc ID	Item-set	Support
54711	{Ribonuclease, catalytic, lysine, phosphate, enzymatic, ethylation}	0.12
55199	{Ribonuclease, Adx, glucocorticoids, chymotrypsin, mRNA}	0.2
62920	{Ribonuclease, anticodon, alanine, tRNA}	0.1
64711	{Cl- channels, catalytic, Monophosphate, cells}	0.072
65118	{isozyme, enzyme, aldehyde, catalytic}	0.096

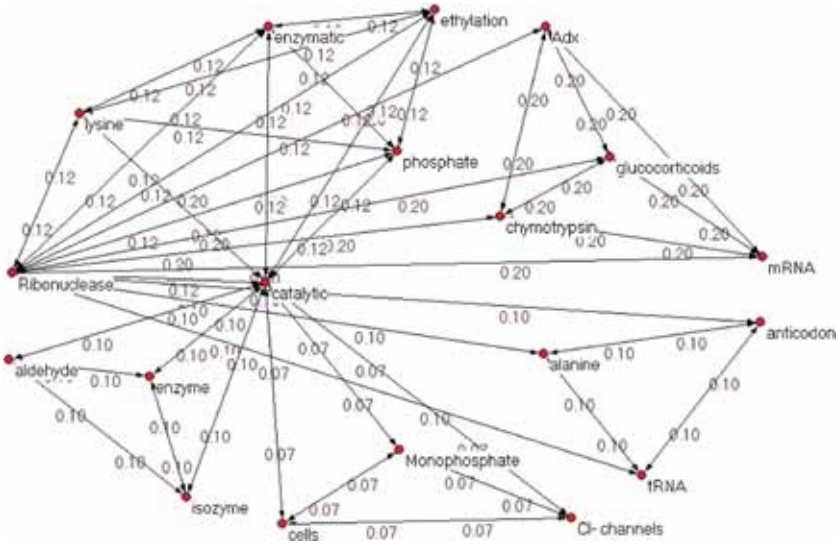

Figure 2. The term association graph representation for frequent item-sets shown in table 1.

based on a novel *support metric* similar to the one presented in Leung & Lee (2010) shown in Eq. (5). The process flow of the proposed schemes has been shown in figure 3.

This work proposes a graph based information retrieval model which employs a graph structure that captures occurrences of terms in documents as well as correlations among terms, and calculate the similarity between a document and the query by the systematic way of graph walk approach. The model has also been extended to incorporate personalization features. Two approaches have been presented. Firstly, TermRank based document re-ranking method (TRA) and secondly, Graph traversal based document re-ranking method (PTA).

4.1 *TermRank based Approach (TRA)*

The notion of PageRank has been employed in this approach. The PageRank score for the nodes in the term association graph is computed. PageRank is a link analysis algorithm which assigns a numerical weighting to each term with the purpose of measuring its relative importance. It is found to be an excellent way to prioritize the results of keyword searches. For example, if a word that appears frequently with many other words in the corpus, it is an important word; words that appear together with some important words may also be important. Since the PageRank algorithm assumes directed un-weighted graph as an input, the term association graph shown in figure 2 has been transformed to a directed graph structure. Then the rank of each term is computed using Eq. (7).

$$Rank(t_a) = c \sum_{t_b \in T_a} \frac{Rank(t_b)}{N_{t_b}}, \tag{7}$$

where, t_a, t_b are terms (nodes), T_b is a set of terms t_a points to, T_a is a set of terms that point to t_a, $N_{t_b} = |T_b|$is the number of links from t_a, and c is a normalization factor. The Eq. (7) is processed recursively to assign rank to each node in the graph. It is computed by starting with set of initial ranks and iterating the computation until it converges. If two terms point to each other but to no

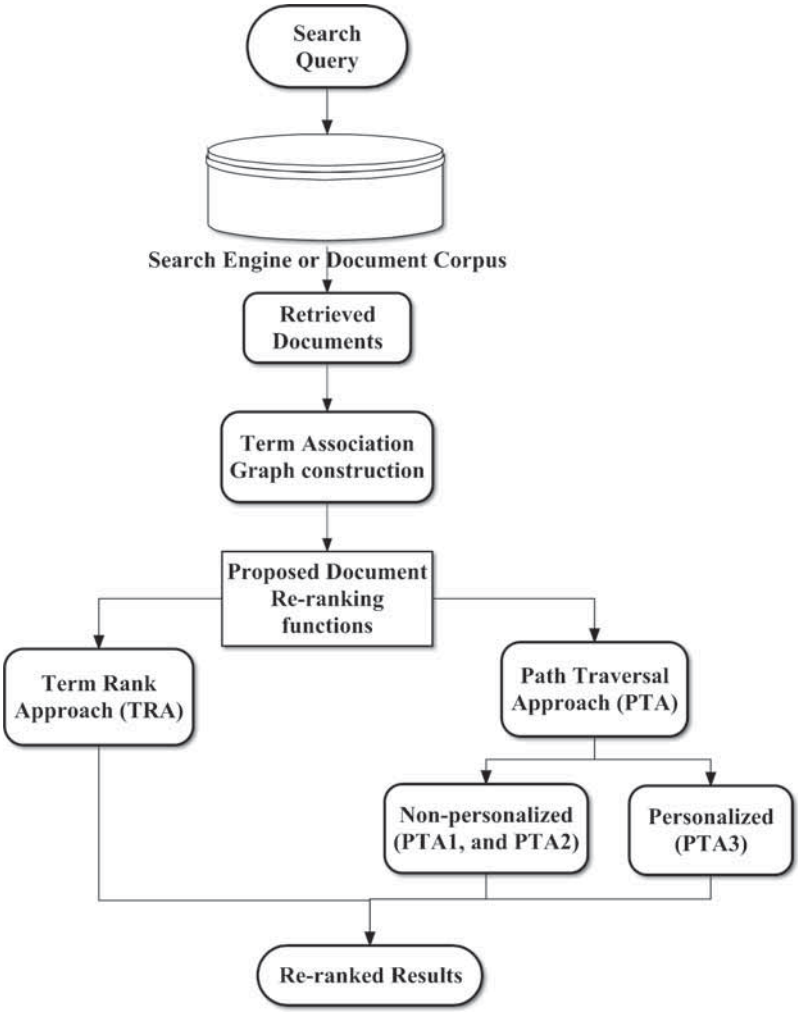

Figure 3. Process flow of the proposed schemes.

other terms, this loop will accumulate rank but never distribute rank to other terms during the iteration. This approach has been described in detail in Veningston & Shanmugalakshmi (2014).

4.2 *Path traversal based approach (PTA)*

In this approach, Depth First Search (DFS) graph traversal algorithm has been employed so as to find the paths originating from the query node in order to re-order the documents. After visiting a query term node q, which is adjacent to w_1, w_2, w_3, ...; next visit one of q's adjacent term nodes. Subsequently visit all term nodes adjacent to w_1 before coming back to w_2, and so on. It is essential to keep track of term nodes already visited in order to avoid cycles. The recursive implementation of DFS, based path traversal algorithm for generating dfs_paths of specified depth has been shown in Algorithm 1. The reason behind adopting DFS algorithm for traversing potential relevance path is that it goes in to specified depth. Due to this, the related

documents linked with latent semantic space could be identified. Alternatively, if Breadth First Search traversal (BFS) algorithm is used, terms belong to same documents would be visited and other semantically related terms may not be visited/traversed. Since, the terms present in same document would be linked in close proximity in term association graph; DFS would be a good choice to explore semantically related documents that are far off.

Algorithm 1. DFS_Path_Traversal(Query_term q, depth).

Input: Term_graph T_G
Output: dfs_paths p_1, p_2, p_3,...

```
(1)  visit q;
(2)  while (depth!=0)
(3)      depth=depth-1
(4)      for each neighbor_term_node w of q
(5)          if w has not been visited then
(6)              DFS_Path_Traversal(w, depth);
```

As the result of Algorithm 1, the different dfs_paths p_1, p_2, p_3,... have been returned. Then find optimized path from these paths traversed using DFS for the re-ranking process.

Table 2 shows the different paths obtained from the query node while traversing using Algorithm 1. The format (T_j/D_i) denotes the presence of terms T_j in document D_i. From the observed dfs_ paths shown in table 2, different approaches have been implemented and evaluated.

4.2a *PTA 1: Naïve approach*: Each *dfs_path* is a possible set of relevant documents based on the link structure exists among the terms in Term Association Graph T_G. The solution path can be chosen in such a way that its total cost is as higher as other possible paths. The cost is defined as the *support value* between two terms in T_G. These two terms may be present either in same document or different documents. For example, if the cumulative support of paths $(p_{i=1,2,...,6})$ shown in table 2 are 0.61, 0.55, 0.72, 0.38, 0.24, and 0.32 respectively, then the sequence of relevant documents are chosen from p_3 i.e., D_1, D_{11}, D_{37}, D_{17}, D_{22}, and D_5. D_{11} could also be the top ranked document because D_{11} occurs twice in this sequence, whereas other documents occur only once. Hence, documents could be re-ordered based on the frequency of occurrences. Alternatively, the documents which occurs frequently in different paths could also be ranked higher from the possible set of dfs_paths i.e., set of ordered documents. For example, the document D_{11} occurs frequently in different dfs_paths p_i for the query_term T_1 with depth = 6 from T_G.

Table 2. Possible *dfs_paths* for the Query_term T_1 with depth = 6.

dfs_paths	*Term/Document*$_i$ (T/D_i)					
p_1	T_2/D_{11}	T_{26}/D_1D_{48}	T_3/D_9 D_{62}	$T_{37}/$ $D_3D_5D_{32}$	$T_{29}/$ $D_1D_6D_{22}$	T_9/D_9D_{11}
p_2	$T_6/D_{11}D_1$	$T_7/$ $D_{14}D_{23}$	T_{13}/D_9D_7	T_{31}/D_6D_{17}	$T_{23}/D_6D_{41}D_1$	T_{19}/D_{41}
p_3	T_{21}/D_1	T_{17}/D_{11}	T_2/D_{11}	T_{12}/D_{37}	T_{21}/D_{17}	$T_{59}/D_{22}D_5$
p_4	T_{61}/D_{16}	$T_{29}/$ $D_1D_6D_{22}$	T_{14}/D_{11}	$T_7/D_{14}D_{23}$	T_{32}/D_{52}	T_2/D_{11}
p_5	T_{34}/D_{71}	T_{43}/D_{31}	$T_4/D_{58}D_{17}$	T_{11}/D_{16}	$T_8/D_{48}D_{12}$	T_{30}/D_1
p_6	T_{13}/D_9D_7	$T_9/D_{57}D_{41}$	T_{44}/D_9	T_{71}/D_{11}	$T_8/$ $D_{48}D_{12}$	$T_{37}/$ $D_3D_5D_{32}$

Thus, the document D_{11} ranked higher for the query T_1 according to the paths shown in table 2. Subsequently, other documents will be ranked in the order of higher frequency.

4.2b *PTA 2: Paired similarity ranking*: In order to find the closest term for the query word from the keywords extracted from medical documents that are retrieved from the corpus, similarity measure was proposed (Wu & Palmer 1994; Viswanathan & Ilango 2012) has been employed.

$$sim(T_1, T_2) = 2 \times \frac{depth(LCS)}{depth(T_1) + depth(T_2)}, \tag{8}$$

where, T_1 and T_2 denote the term nodes in T_G to be compared, *LCS* denote the Least Common Sub-Sumer or Longest Common Subsequence of T_1 and T_2, and $depth(T)$ is the shortest distance from the query node q to a node T on T_G. The possible *dfs_paths* shown in table 2 have been ranked according to the Eq. (8) by computing the similarity between all pairs of T_9, T_{19}, T_{59}, T_2, T_{30}, T_{37}. For example, $sim(T_9, T_{19})$ is computed as 2*(5/6+6)=0.83 as shown in figure 4, thereby the depth based similarity matrix D_{sim} is constructed as shown in table 3. Then the documents are re-ranked according to the paired similarity obtained using the Algorithm 2.

Algorithm 2. Paired_Similarity_Re-ranking (Documents in *dfs_path*, query_term q).

Input: Depth based Similarity matrix D_{sim}
Output: Re-ordered documents

(1) k =number of pairs on$N$terms
(2) Map the upper or lower triangle of the symmetric matrix D_{sim} into an array A
(3) Sort the array A in descending order
(4) Compare the D_{sim} with sorted array A in order to identify the indices of the pair of words possesses higher depth based similarity
(5) From the list of term indices with duplication, Generate ranked list of terms by removing duplicate entry of term indices.
(6) Order the documents according to the term sequence in the ranked list
(7) Display the re-ranked list of documents

For the depth based similarity matrix shown in table 3, the Paired_Similarity_Re-ranking algorithm given in Algorithm 2 prepares the terms in the following sequence as T_9, T_{19}, T_2, T_{37}, T_{59}, T_{30}. Accordingly, documents possess these terms has been re-ranked as shown in table 4. After removing duplicates the following sequence of documents D_9, D_{11}, D_{41}, D_3, D_5, D_{32}, D_{22}, D_{62} are retrieved.

4.2c *PTA 3: Personalized path selection*: In order to incorporate advantage of personalization into term association graph model for document re-ranking, the personalized path selection algorithm has been proposed. Since the notion of personalization has been employed, this approach could not be evaluated using benchmark OHSUMED dataset. Thus, this approach focuses on general document search.

Document topics. In order to present personalized search results, personalization concept proposed by Viswanathan & Ilango (2012) has been adopted in this method. In this approach, the

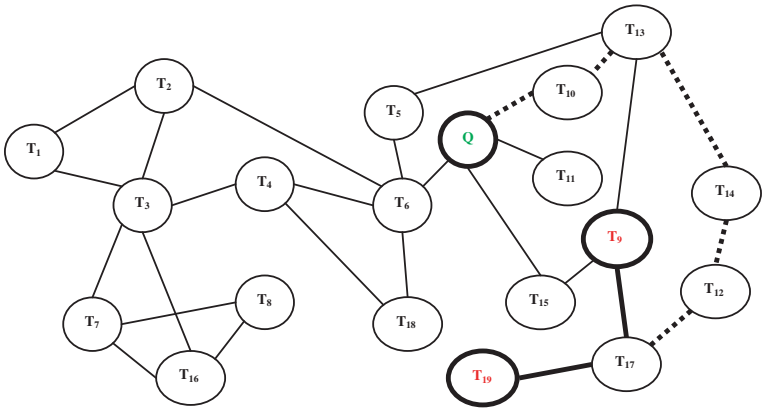

Figure 4. LCS computation on T_G.

Table 3. Depth based similarity matrix.

***sim*(*T*,*T*)**	T_9	T_{19}	T_{59}	T_2	T_{30}	T_{37}
T_9	1	0.83	0.66	0	0.17	0.33
T_{19}	0.83	1	0.5	0.83	0.17	0
T_{59}	0.66	0.5	1	0.66	0.5	0.17
T_2	0	0.83	0.66	1	0.33	0.83
T_{30}	0.17	0.17	0.5	0.33	1	0
T_{37}	0.33	0	0.17	0.83	0	1

Table 4. Ranked list of terms and its associated documents.

T_9	T_{19}	T_2	T_{37}	T_{59}	T_{30}
D_9, D_{11}	D_{41}	D_{11}	D_3, D_5, D_{32}	D_{22}, D_5	D_9, D_{62}

user's dynamic search interests on various topics are captured from their web browser search history. The topics are trained on the Open Directory Project (ODP) corpus (Carpineto & Romano 2009). The ODP home page has been shown in figure 5 (http://www.dmoz.org/). The ODP corpus has documents covering almost all of the dominant categories existing on the World Wide Web. Figures 6 and 7 show the sample main topics and sub topics of ODP dataset. In this work, topical categories from the top most level of the ODP are used to represent user interested topics. The ODP corpus includes 15 broad categories such as arts, games, home, health, etc. and its subcategories. The weight value of the user's interest topics is maintained in a table. Subsequently, these weight values are incorporated in order to calculate the context weights of the *dfs_paths* identified on real dataset during the re-ranking process. The detailed description on real dataset has been given in section 5. The documents possess terms in *dfs_path* are ranked according to the users' search need without their intervention in search context specification. The architecture of the proposed personalization model has been shown in figure 9.

Table 5 shows the sample set of documents retrieved for the query 'Web' from the search engine. In this way, real data has been prepared and used to evaluate the proposed personalized approach. Figure 8 shows the term association graph constructed over the initial set of 9 documents retrieved for the query 'Web' with 50 nodes. Nodes in the graph denote the unique terms

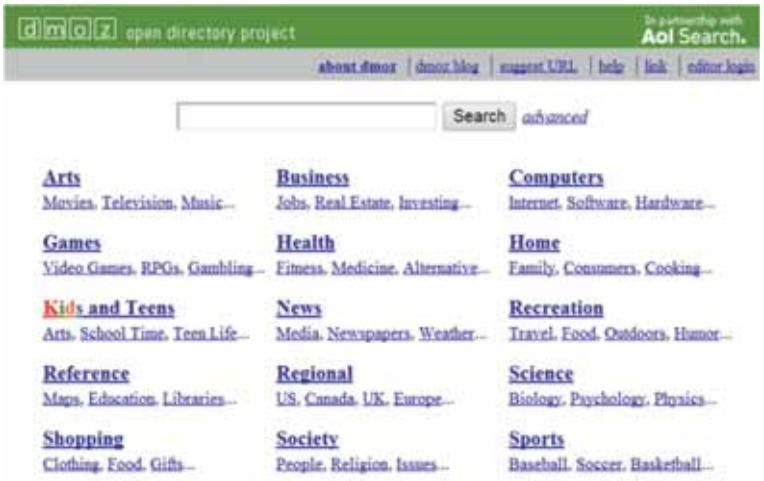

Figure 5. ODP home page [http://www.dmoz.org/].

1. Computers > Algorithms
2. Computers > Companies
3. Computers > Programming
4. Computers > Software
5. Computers > Graphics
6. Computers > Hacking
7. Computers > Hardware
8. Computers > Internet
9. Computers > Mobile_Computing
10. Computers > Multimedia

Figure 6. ODP main topics on 'computers'.

1.1. Computers > Algorithms > Compression
1.2. Computers > Algorithms > Conferences
1.3. Computers > Algorithms > Research_Groups
3.1. Computers > Programming > Languages
3.2. Computers > Programming > Software_Testing
3.3. Computers > Programming > Compilers
3.4. Computers > Programming > Games
8.1. Computers > Internet > E-mail
8.2. Computers > Internet > Protocols
8.3. Computers > Internet > Chat

Figure 7. ODP sub topics on 'computers'.

in the set of documents collection. The graph structure has been created using Pajek visualization tool (http://vlado.fmf.uni-lj.si/pub/networks/pajek/).

Table 5. Sample real data on the query 'Web'.

	Input Query: Web
Doc 1	{WWW, interlink, hypertext, access, browser, internet, multimedia, navigate, text}
Doc 2	{structure, spider, WWW, hypertext, internet, browser, host, client, online}
Doc 3	{design, business, website, online, market, services}
Doc 4	{WWW, community, organization, standard, develop}
Doc 5	{telegram, message, support, computer, file, send, client}
Doc 6	{information, health, medical, news, community, drug, doctor, symptom, webMD}
Doc 7	{video, tutorial, design, create, website, course, online}
Doc 8	{yahoo, search, engine, relevance, multimedia, information, video, image, answer, text}
Doc 9	{google, search, engine, personalization, information, text, multimedia}

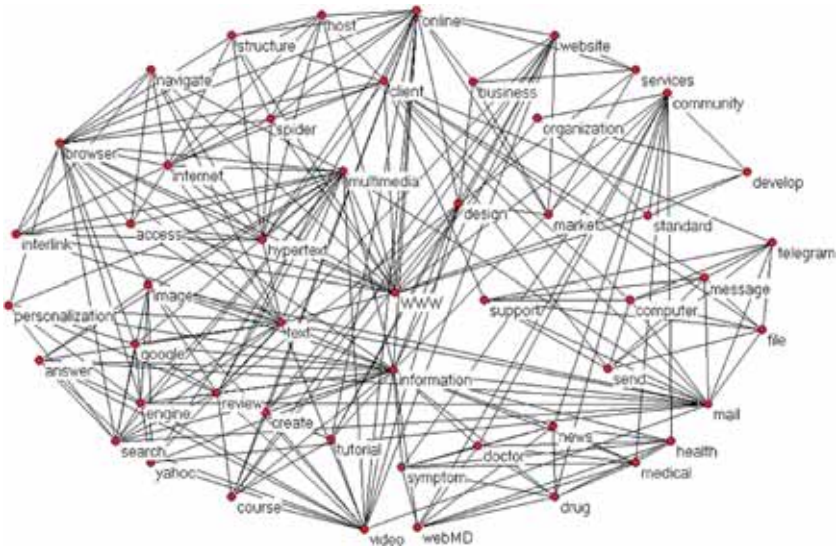

Figure 8. Term association graph on real data with 50 nodes.

Table 6. User search interest value table.

Session ID	Software	Algorithms	Healthcare	Sports	Movies	Music
S1	0.312	0.671	0.090	0.232	0.001	0.030
S2	0.134	0.245	0.322	0.301	0.023	0.010
S3	0.472	0.107	0.024	0.149	0.200	0.174
S4	0.048	0.110	0.261	0.642	0.098	0.145
S5	0.076	0.093	0.047	0.184	0.594	0.611

User search interest value table. The user search interest value table keeps track of user's current topical interest. The values shown in table 6 have been periodically updated in order to maintain user's current search interest (figure 9).

Session identification. In order to recognize the end of a previous session and the beginning of the current session, the Kullback-Leibler Divergence (KLD) has been employed to compute

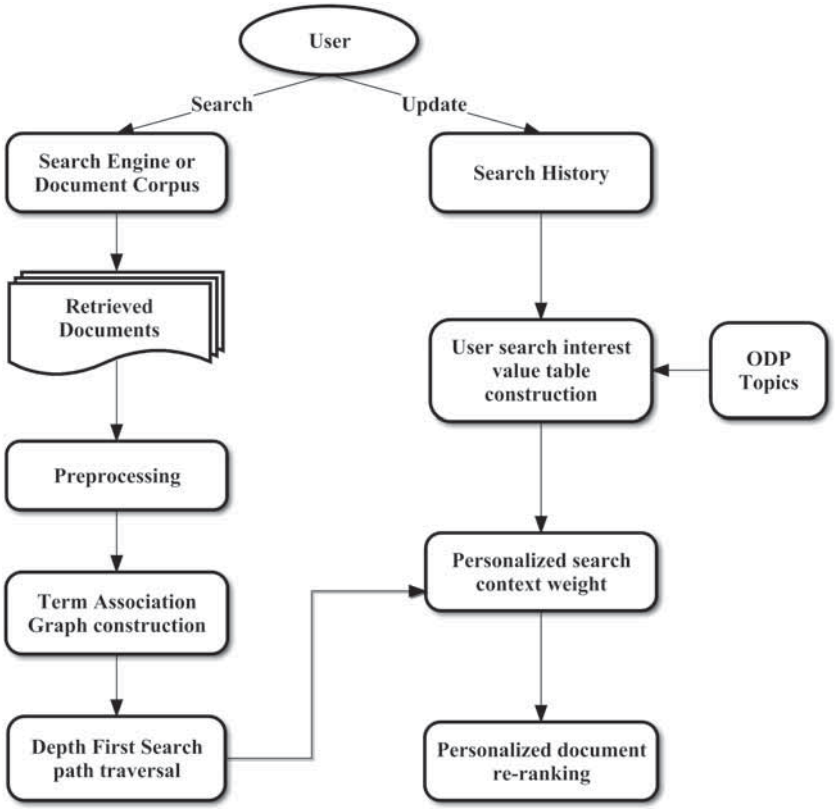

Figure 9. Architecture of the proposed personalization model (PTA3).

the similarity between two query terms. KLD compares the probability distribution of two query terms over the set of web pages in the collection. This difference measures the similarity between two queries, thus helps in separating two search sessions. The separation threshold is set to 0.5 and it is assumed that the queries are from the same session if the similarity is below the threshold. The similarity between two queries is computed as shown in Eq. (9).

$$KLD(q_1||q_2) = \sum_{t \in D_1 \cap D_2} P(t) \log \frac{P(t \in D_1)}{P(t \in D_2)}, \tag{9}$$

where $P(t)$ is the probability of term t in documents retrieved for query q_1 and q_2 i.e., D_1 and D_2. D_1 and D_2 is the set of documents retrieved for q_1 and q_2, respectively.

Search context weight. Search context weight is one of the semantic metrics that is used to determine the relevancy based on user's preference. Assume the scenario in which user is interested in retrieving documents related to their search interest. For example, if the user's current search interest is on 'algorithms', search contexts such as 'Compression', 'Sorting_and_Searching', and 'Research_Groups' may be of relevance to the user; whereas, 'composition', 'instruments', and 'lyrics' may be less relevant or irrelevant to the user. Thus, using the

identified search context, it is more likely to re-rank *dfs_path*s according to its relevance with a user's search interest.

Algorithm 3. Personalized_similarity_Re-ranking (documents, query_term q).

Input: *dfs_paths*, Depth based Similarity matrix D_{sim}
Output: Re-ranked documents

(1) $siv_i = 0$;
(2) $count = 0$;
(3) MAX= 0;
(4) for each dfs_path_j
(5) for each term t_i in $dfs\_path \in$ topic T in ODP
(6) $siv_i = siv_i + siv_i(T)$;
(7) for each term t_i in $dfs\_path \notin$ topic T in ODP
(8) $count = count + 1$;
(9) $PSC_{weight} = (siv_i \times (1\text{-}(\text{count}/|t|)))/|t|$
(10) if PSC_{weight} >MAX then
(11) MAX= j
(12) Order the documents that possesses the terms in dfs_path_j

Here, $|t|$ is the total number of terms in path including the query term. T is the set of user interested topics, siv_i is the search interest value of the i^{th} topic of specified user. This value is taken from user search interest value table given in table 6. The Personalized Search Context weight (PSC_{weight}) has been calculated using the Algorithm 3 and it has been used to calculate the weight of *dfs_path* including query term. Thus, the documents possesses terms in *dfs_path* are re-ranked incorporating personalization features.

5. Experimental evaluation

The approaches proposed in this paper focuses on the query-centric re-ranking of search results. Typically, input query issued by the user are keywords in which user does not have a special page in mind intends to find out documents related to a topic/concept. Experimentation of the proposed approaches has been evaluated using both synthetic and real dataset. In re-ranking task, the documents with low scores are not eliminated; instead it is moved to lower in the ranking. Presumably, documents with extreme scores were moved more than others.

5.1 *Dataset description*

5.1a *Synthetic dataset*: The Oregan Health and Science University MEDline (OHSUMED) document collection described (Hersh *et al* 1994; Qin *et al* 2010) used for the Text Retrieval Conference (TREC-9) Filtering Track (http://trec.nist.gov/) has been employed in this work. The OHSUMED test collection is a set of 348,566 document references from MEDLINE on-line medical information database, consisting of titles and/or abstracts from 270 medical journals over a period of 5 years from 1987 to 1991. The available fields are title, abstract, indexing terms, and author.

Queries were issued to retrieve initial set of results from the OHSUMED document collection and then the proposed methods have been applied to re-rank documents. The dataset contains

Table 7. Sample test queries and the expected documents to be retrieved.

Test query	Expected documents
Phosphate	Catalytic activity
Alcolmeter	Blood ethanol concentration measures
Ethanol	Alcohol, Breath analysis and tests
Morphine	Narcotic Syndromes
Endorphin	Blood-Brain Barrier due to Alcoholism
Alcoholism	Brain syndromes, Treatments
Erythrocyte	State of Hemoglobin on narcotic consumption
Serotonin	Brain test during sleep and depression
Platelets	Measure of platelet affinity, Treatments
Abstinent	Appetite for food or drink, insulin tests

associated documents of over 106 queries. A query is about a medical search need. The relevance degrees of documents with respect to the queries are judged by humans on three point scale such as highly relevant, moderately relevant, not relevant. The dataset consists of around 16,140 query-document pairs with relevance judgements. The sample set of queries used for experimentation is given in table 7.

5.1b *Real dataset*: Typically, a search engine retrieves thousands of web documents for a given query. Users were asked to input search queries related to their professional knowledge and other information including business, health, computers, movies, news, etc. from a commercial search engine and to review top 50 results for the relevance in terms of information richness. The same results were taken as input to the proposed methods in order to further re-order the results based on the personal preferences in addition to the query. The sample list of queries issued by the users is given in table 8.

Table 8. Sample queries issued to Google search engine for real dataset construction.

S. No.	Test queries
1	Process
2	Algorithm
3	Data structure
4	Synchronization
5	Information
6	Categorization
7	Web
8	Scheduling
9	Threading
10	Computing
11	Marathon
12	Hotspot
13	Nutrition
14	Hollywood
15	Lyrics
16	Foreign exchange
17	War
18	Currency
19	Jobs
20	Stock market

Table 9. Statistics about the corpus considered for experimentation.

Document corpus	Usage for evaluation	Number of documents	Number of queries	Avg. doc. length	Avg. doc. length after pre-processing
Real dataset (results from traditional search engine)	Personalized model & non-personalized models	Top 50 results for a query	100	34	21
Synthetic dataset (OHSUMED)	Non-personalized models	348,566	106	210	64

The sample set of documents retrieved for the query 'Web' has been shown in table 5. Figure 8 shows the term association graph constructed on real data with 50 nodes. Table 9 shows the statistics about the datasets considered for experimentation (table 10).

5.2 *Evaluation set-up and measures*

5.2a *Subjective evaluation set-up*: The group of 50 under graduate and post graduate students of Computer Science from Government College of Technology Coimbatore, INDIA (http://www.gct.ac.in/) performed retrieval on various queries in order to independently evaluate the experimental results. They assessed and labeled the top 50 results for each of the 100 queries according to the following steps as given in Zhang *et al* (2005). The queries vary from 1 word to 2 words pertaining to their professional knowledge and other terminologies were assumed (sample queries are shown in table 8).

(i) Assess top 50 search results for a query, and then manually group those retrieved results into non-overlapping clusters i.e., each cluster must possess at most one topic in common.
(ii) Assign a score to each document in a topic group in order to indicate the information richness for the topic. The score ranges from 0 to 3 (3 - very informative, 2 - informative, 1 - less informative, 0 - not informative).

The information richness scores are normalized into the range of 0 to 1. The labeled data served as the ground truth to evaluate the diversity, and information richness of the top N search

Table 10. Evaluation metrics and set-up employed for experimental result analysis.

Proposed approaches \ Datasets used	Real dataset	Synthetic dataset
Personalized scheme (PTA 3) with subjective evaluation	Variation in user search intents, Information Richness, and Average Information Richness	—
Non-personalized schemes (TRA, PTA1, & PTA2) with objective evaluation	Accuracy in terms of Precision at various search results positions, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG)	Accuracy in terms of precision at various search results positions, MAP, and NDCG

results ($N \leq 50$). Thus, the following measures are employed in order to assess the proposed personalized method.

Diversity. Given a set of documents retrieved R_m, Diversity is defined as Div(R_m) to denote the number of different topics contained in R. In evaluation, the number of topics contained in R is assumed to be not more than 1 because the number of topics in the top 50 search results would generally vary from user to user.

Information richness. Given a document collection $D = \{d_1, ..., d_n\}$, Information Richness is defined as InfoRich (d_i) to denote richness of information contained in the document d_i with respect to the entire collection D.

$$InfoRich(R_m) = \frac{1}{Div(R_m)} \sum_{k=1}^{Div(R_m)} \frac{1}{N} \sum_{i=1}^{N_k} InfoRich(d_k^i), \tag{10}$$

where d_k^i represent one of N_k documents associated with the k^{th} topic. The average information richness is defined as information richness of a set of documents.

5.2b *Objective evaluation set-up*: The re-ranking algorithms proposed in this paper have been evaluated using a variety of accepted IR metrics such as Precision, Recall, Interpolated precision, F-measure, MAP, MRR, NDCG (Baeza-Yates & Ribeiro-Neto 1999; Manning *et al* 2008; Jarvelin & Kekalainen 2000, 2002).

5.3 *Baselines for comparison*

The state-of-the-art bag-of-words model i.e., *tf-idf* based Okapi-BM25 has been assumed to be the baseline system. In addition to this, Clustering based document re-ranking (CA) and Affinity Graph (AG) ranking approaches have also been taken for comparing the results prepared by the proposed re-ranking approaches.

5.3a *K-means algorithm*: To re-rank top results wherein K is chosen to be 10 and the top 1 document from each cluster is used to construct the top 10 results. The k-means partition based clustering algorithms typically attempts to minimize the distance between documents in the same cluster i.e., if D(d_1, d_2,...,d_n) are the n documents and C(c_1, c_2,..., c_k) are the k clusters centroids, then k-means tries to minimize the function defined in Eq. (11).

$$\sum_{i=1}^{k} \sum_{j=1}^{n} similarity(d_j, c_i). \tag{11}$$

5.3b *Affinity graph ranking*: In this approach (Zhang *et al* 2005), the document collection is modeled as a graph by generating the link between documents. A directional link from d_i to d_j ($i \neq j$) with weight affinity(d_i,d_j) is constructed if affinity(d_i,d_j) $\geq$ affinity$_t$ where affinity$_t$ is a threshold; otherwise no link is constructed i.e., the weight of the link is regarded as zero. This defines the affinity of d_i to d_j similar to cosine similarity in order to define the similarity between

each document pair as shown in Eq. (12). Thus, each link in the graph has been assigned a weight indicating the similarity relationship between the corresponding document pair. Since all links are constructed according to the affinity value between document pairs, documents of the same topic are similar to each other. Hence, group of densely linked documents typically represents a topic group whereas documents sparsely connected or not connected belong to different topics. Thereby the affinity ranking scheme re-rank top documents.

$$affinity(d_i, d_j) = \frac{d_i . d_j}{||d_i||}. \tag{12}$$

5.4 *Experimental results*

5.4a *Evaluation criteria for statistical significance test*: In order to evaluate whether the obtained results are statistically significant essentially when comparing two methods, the statistical significance test, namely student paired t-test has been performed. This t-test assigns a confidence value (p-value) to the null hypothesis. The general form of a null hypothesis H_0 is that there is no difference between the compared systems. When the p-value is low, the null hypothesis may be rejected. The p-value is assumed to be 0.05 for the significant tests. In the improvements shown in the tables, the symbol * and ** indicate that the improvement is statistically significant according to the paired-sample t-test at the levels of $p < 0.05$ and $p < 0.01$, respectively.

The statistical significance testing of the retrieval algorithms do not provide any information about the strength of the relationship between the approaches compared. For example, achieving a value of $p = 0.001$ does not mean that the relationship is stronger than if a value of $p = 0.04$ is achieved. Thus the significance test investigates whether H_0 can be accepted or rejected. Table 11 summarizes the p-value of the two-tailed t-test against the Term Graph (TG) (Blanco & Lioma 2012), Concept Graph (CG) (Koopman *et al* 2012), and Affinity Graph (AG) (Zhang *et al* 2005) approaches for the proposed three different non-personalized variations namely, TRA, PTA1, and PTA2. The paired $t-$test has been conducted to determine the statistical significance of difference observed with respect to the baselines.

5.4b *Non-personalized evaluation on real dataset*: The proposed non-personalized approaches have been run in order to re-rank the top 20 results prepared by the baseline Okapi BM25 system on the real data collection. Only the single and two word queries were employed in order to assess the results. The reason for restricting the length of query is that short queries are inherently ambiguous.

From figure 10, it is evident that the proposed non-personalized approaches consistently perform better than baseline approaches for the precision values (P@5, P@10, P@15, P@20), especially at the top few results. Figure 11 shows the MRR and MAP obtained for 30 queries

Table 11. Summary of significance test results on real dataset.

Vs. Term Graph (TG)	Paired t-test (p-value)	Vs. Concept Graph (CG)	Paired t-test (p-value)	Vs. Affinity Graph (AG)	Paired t-test (p-value)
TRA	**0.002****	TRA	**0.047***	TRA	**0.001****
PTA1	0.619	PTA1	0.180	PTA1	**0.004****
PTA2	**0.008****	PTA2	0.825	PTA2	**0.002****

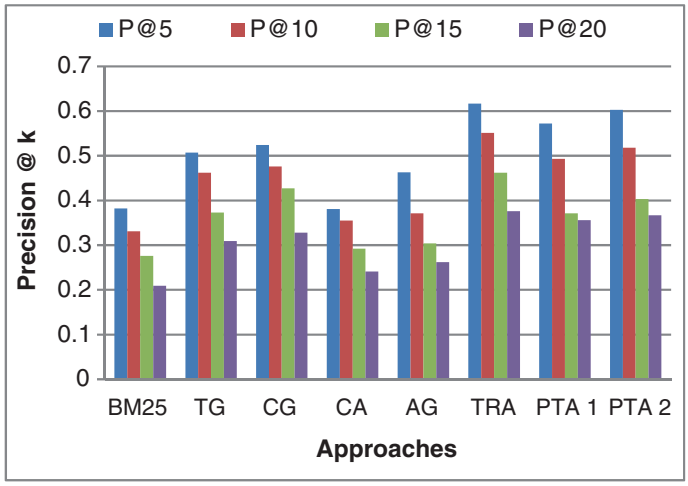

Figure 10. Precision at k result positions for 30 queries (k =5, 10, 15 & 20).

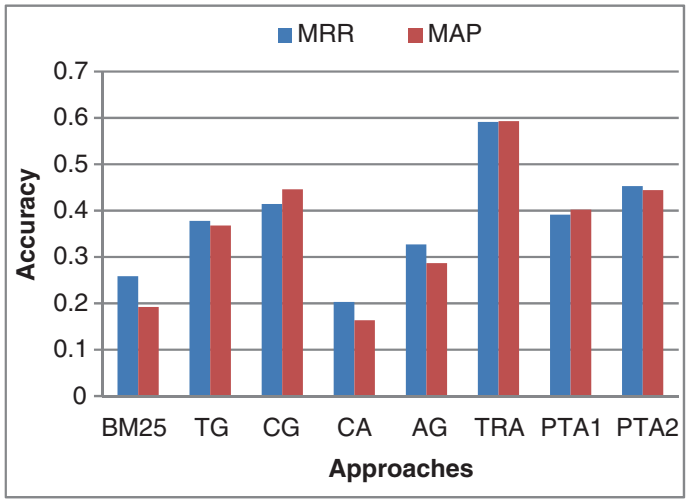

Figure 11. MRR and MAP for 30 queries at top 5 results over real dataset.

at top 5 result positions (sample queries given in table 8). The proposed TRA, PTA1, and PTA2 attain 56.41%, 3.54%, and 19.82% of improvement respectively over Term Graph approach. In comparing with Concept Graph approach, the TRA, and PTA2 achieve 42.81% and 9.41% of improvement, respectively while Concept Graph approach outperforms PTA1 by 5.45%. The MAP obtained for 30 queries at top 5 results by TRA, PTA1, and PTA2 attain 61.19%, 9.24%, and 20.76% of improvement respectively over Term Graph approach. In comparison with Concept Graph approach, the proposed TRA achieve 32.88% of improvement while Concept Graph approach outperform PTA1 by 9.94% and yield similar results as compared to PTA2.

Figure 12 shows the NDCG obtained at various result positions for 30 queries. The performance of the proposed approaches is impressive since the NDCG gains a notable increase. Compared with other graph based approaches namely, TG, CG, and AG, the proposed algorithms

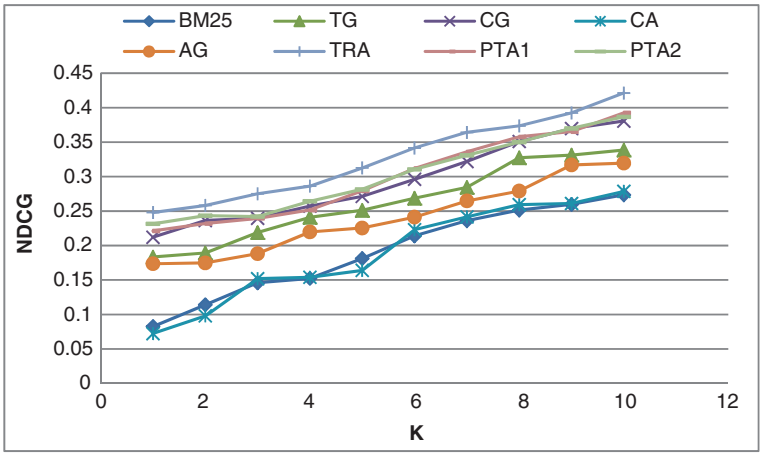

Figure 12. NDCG at K obtained for 30 queries.

improve top 10 search results. Table 12 shows the percentage of NDCG improvements achieved by the proposed methods over baseline approaches.

5.4c *Non-personalized evaluation on synthetic data set*: The precision at various positions and MAP achieved has been shown in figure 13. This demonstrates that the proposed TRA, PTA1, PTA2 perform well when comparing with baselines. Figure 14 shows the NDCG obtained at different positions which depict that the proposed methods relatively outperform baseline approaches. The evaluation on OHSUMED reports that the term association model improves the retrieval performance by identifying highly relevant documents for the query.

The performance measures reported in figures 13 and 14 show that the proposed term association graph model improves retrieval effectiveness in terms of accuracy of search results by bringing the highly relevant documents in first few results.

5.4d *Personalized evaluation on real dataset*: Since there is no standard publicly available test collection for evaluation of personalized retrieval algorithms, real dataset comprises of top 50

Table 12. NDCG improvement gain in percentage.

Result positions	NDCG improvement over TG in %			NDCG improvement over CG in %			NDCG improvement over AG in %		
	TRA	PTA1	PTA2	TRA	PTA1	PTA2	TRA	PTA1	PTA2
1	35.16	20.66	26.28	16.93	4.38	9.24	42.79	27.47	33.41
2	36.52	22.62	28.69	9.26	−1.86	3.00	47.68	32.64	39.22
3	25.61	9.31	10.54	14.81	−0.08	1.04	46.09	27.13	28.57
4	18.75	4.43	9.66	11.23	−2.17	2.72	30.20	14.51	20.24
5	24.48	11.22	12.14	15.21	2.94	3.79	38.73	23.95	24.97
6	27.13	16.11	15.59	15.36	5.36	4.89	41.62	29.35	28.77
7	28.04	18.10	16.34	13.17	4.38	2.82	37.57	26.88	25.00
8	14.14	9.28	6.99	6.49	1.96	−0.17	33.89	28.19	25.51
9	18.50	10.32	11.74	6.13	−1.18	0.08	23.85	15.30	16.78
10	24.42	15.94	14.26	10.69	3.15	1.65	31.82	22.84	21.05

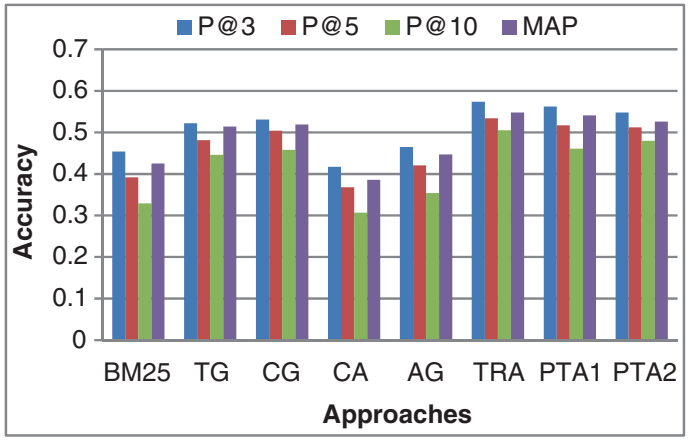

Figure 13. Precision and MAP obtained for 30 queries.

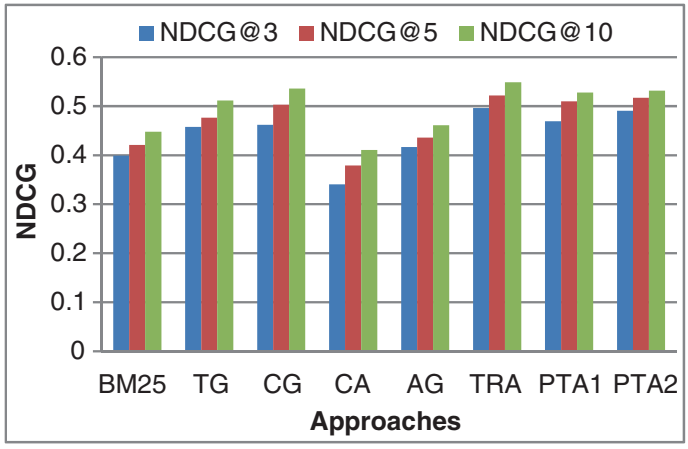

Figure 14. NDCG at $K = 3$, 5, and 10 obtained for 30 queries.

search results retrieved for 100 different queries from the traditional search engine have been prepared. Personalized evaluation has been carried out based on the subjective evaluation set-up given in section 5.2a. The documents retrieved using PTA 3 approach is assessed by information richness defined in Eq. (10).

Figure 15 shows the comparison of initial ranking and the proposed system ranking results between the results retrieved for the query 'Nutrition'. According to the proposed personalized approach named PTA 3, the x-axis represents the proposed system ranking of top 30 documents; y-axis represents initial ordering of 30 documents. The level of disagreement between the proposed systems ranking and initial ranking is evidently known from the personalized rank positions for 6 users. Figure 15 demonstrates the diverse search intent of each user in which the diagonal line indicates the typical rank ordering of documents. For instance, User 3 finds the 11^{th} document retrieved by the search engine as his/her 1^{st} document. This shows that the user interest is highly diverse and thus personalization is a potential solution to address the need of different users. As the proposed personalization approach PTA 3 prepares relevant documents

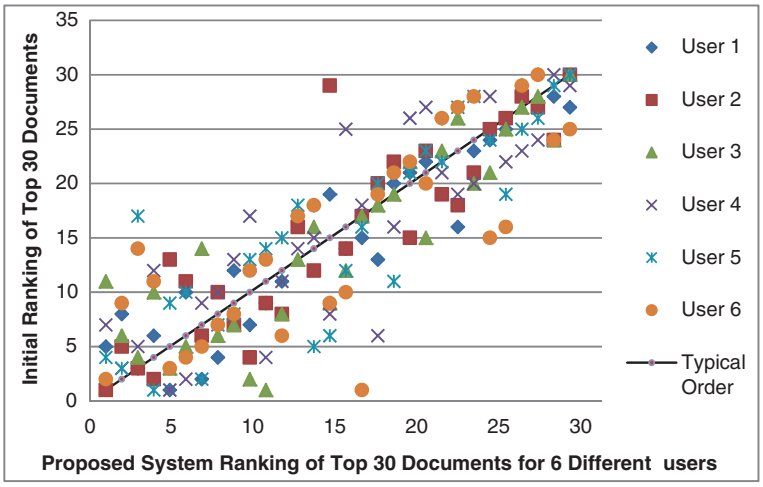

Figure 15. Initial ranking Vs. Proposed system ranking.

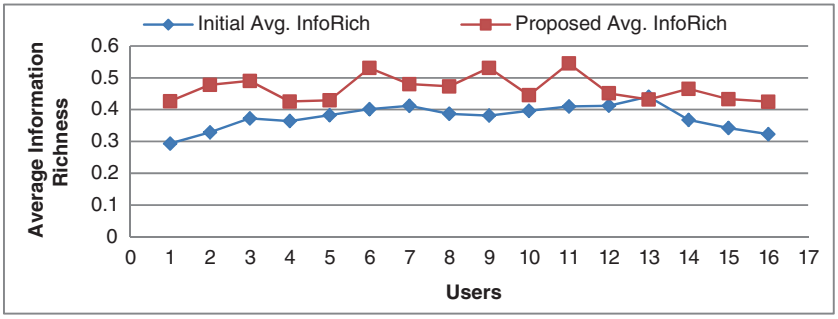

Figure 16. Information richness improvements by the proposed PTA 3 within top 30 results for 15 users.

incorporating users individual search preferences, the average information richness obtained for a sample of 30 queries are different to different users. This demonstrates that the individual preferences are essential to be captured in order to prepare relevant documents for the search query. Figure 16 observes the notable information richness improvements by the proposed PTA 3 algorithm within top 30 search results. From figure 16, it is observed that users 4, 5, 10, 12 and 13 get satisfied with the initial ranking as the difference between initial average information richness and the proposed average information richness is minor while other users prefer personalization to be performed.

5.5 *Result analysis and discussion*

The graph representation of text estimates the association between the texts in order to compute term ranks i.e., term importance. The computed term rank associated with each term is further used to determine the documents of relevance. From the experimental evaluation, it is observed that the proposed re-ranking approaches outperform the baseline systems over both real data and synthetic data considered for experimentation at greater extent. The term association graph model can also be used for suggesting related keywords while performing ad-hoc

retrieval in order to enhance the retrieval process even before the documents are retrieved. Thus, the same graph based term association representation model could be adapted for representing user's implicit interest in order to assist users in retrieval task by suggesting similar query terms to the users.

6. Conclusion

The various approaches that exploit term association graph model have been proposed for efficient retrieval of information from large corpus of text documents. This paper revealed the challenges that are present in the state-of-the-art IR systems, and suggested three non-personalized approaches namely, TRA, PTA1, and PTA2 and a personalized approach PTA3 to enhance the document re-ranking task in order to meet the information need of the user. The proposed work captures hidden semantic association and the results produced by the proposed algorithms demonstrate the effectiveness by improving document representation and re-ranking by incorporating more information available within the document's term association into the ranking task. The experimental results reveal that the proposed algorithms improve the retrieval performance in terms of both accuracy and coverage. It is also inferred that still there was a gap existing in the process of identifying most relevant information that is of interest for the user. Thus, the work presented in this paper has been extended in the following direction. (i) The reputation of a word measured in TRA does not guarantee the desired information to the searcher. Thus user specific relevance factor has been employed in PTA3 and (ii) the proposed non-personalized approaches TRA, PTA1, and PTA2 do not consider user's search preferences. Thus, personalized search feature has been enabled in PTA3 in order to organize search result to be more appropriate and relevant to the individual user.

Acknowledgements

The work presented in this paper was supported and funded by the Department of Science and Technology (DST), Ministry of Science and Technology, Government of India under INSPIRE scheme. Authors thank the DST and also the anonymous reviewers for their helpful comments.

References

Agrawal R and Srikant R 1994 Fast algorithm for mining association rules. In: *Proc. 20th Intl. Conf. VLDB*, 487–499, ACM

Baeza-Yates R and Ribeiro-Neto B 1999 *Modern information retrieval*. Addison Wesley: ACM Press

Berger A and Lafferty J 1999 Information retrieval as statistical translation, In: *Proc. SIGIR*, 222–229, ACM

Blanco R and Lioma C 2007 Random Walk Term Weighting for Information Retrieval. In: *Proc. SIGIR*, 829–830, ACM

Blanco R and Lioma C 2012 Graph-based term weighting for information retrieval. *Springer Information Retrieval* 15(1): 54–92

Blondel V D, Gajardo A, Heymans M, Senellart P and Dooren P V 2004 A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Rev.* 46(4): 647–666

Boldi P, Bonchi F, Castillo C, Donato D, Gionis A and Vigna S 2008 The query-flow graph: model and applications. In: *Proc. ACM CIKM*, pp. 609–618

Brin S and Page L 1998 The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: *Proc. 7th International World Wide Web Conference (WWW)*, pp. 107–117

Carpineto C and Romano G 2009 ODP239 dataset, http://credo.fub.it/odp239/

Craswell N and Szummer M 2007 Random Walks on the Click Graph, In: *ACM SIGIR*, pp. 239–246, ACM

Croft W B and Lafferty J 2010 *Language Modeling for information retrieval*. Kluwer Academic Publishers, Springer Netherlands

Eirinaki M and Vazirgiannis M 2005 UPR:Usage-based page ranking for web persoanalization. In: *Proc. 5th IEEE Intl. Conf. on Data Mining (ICDM)*, pp. 130–137

Han J and Kamber M 2006 *Data mining concepts and techniques,* Morgan Kaufmann publishers, Elsevier San Francisco*,* Second edition pp. 227–232

Haveliwala T H 2003 Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.* 15(4): 784–796

Hersh W, Buckley C, Leone T J and Hickam D 1994 OHSUMED: An interactive retrieval evaluation and new large test collection for research, In: *Proc. 17th Annual SIGIR Conference*, pp. 192–201, ACM

Hofmann T 1999 Probabilistic latent semantic indexing, In: *Proc. ACM SIGIR*, pp. 50–57

Jain A and Mishne G 2010 Organizing query completions for web search. *ACM Intl. Conf. on Information and Knowledge Management (CIKM)* 1169–1178

Jarvelin K and Kekalainen J 2000 IR evaluation methods for retrieving highly relevant documents. In: *Proc. SIGIR*, pp. 41–48, ACM

Jarvelin K and Kekalainen J 2002 Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20(4): 422–446

Kleinberg J M 1999 Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46(5): 604–632

Koopman B, Zuccon G, Bruza P, Sitbon L and Lawley M 2012 Graph-based Concept Weighting for Medical Information Retrieval. In: *Proc. 17th Australasian Document Computing Symposium (ADCS)*, pp. 80–87

Lafferty J and Zhai C 2001 Document language models, query models, and risk minimization for information retrieval. In: *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 111–119

Leicht E A, Holme P and Newman M E J 2006. *Physical Review* Vertex similarity in networks

Leung K W T and Lee D L 2010 Deriving Concept-based User profiles from Search Engine Logs. *IEEE Trans. Knowl. Data Eng.* 22(7): 969–982

Ma H, King I and Lyu M R-T 2012 Mining Web Graphs for Recommendations. *IEEE Trans. Knowl. Data Eng.* 24(6): 1051–1064

Manning C D, Raghavan P and Schutze H 2008 *Introduction to Information Retrieval*. Cambridge University Press London, pp. 151–168

Masucci A P and Rodgers G J 2006 Network properties of written human language. *Phys. Rev.* 74(2)

Mei Q, Zhou D and Church K 2008 Query suggestion using hitting time. In: *Proc. 17th ACM Conf. on Information and Knowledge Management (CIKM)*, pp. 469–478

Mihalcea R and Tarau P 2004 TextRank: Bringing Order into Texts. In: *Proc. Empirical Methods in Natural Language Processing. Association of Computational Linguistics (ACL)*, pp. 404–411

Montes-y-Gomez M, López-López A and Gelbukh A 2000 Information retrieval with conceptual graph matching. In: *Proc. 12th Intl. Conf. Database and Expert Systems Applications*, Springer LNCS, Volume 1873, 312–321

Nastase V, Sayyad-Shirabad J, Sokolova M and Szpakowicz S 2006 Learning noun-modifier semantic relations with corpus-based and wordnet-based features. In: *American Association for Artificial Intelligence*, 781–786

Pado S and Lapata M 2007 Dependency-based construction of semantic space models. *Comput. Linquist.* 33(2): 161–199

Page L, Brin S, Motwani R and Winograd T 1998 The PageRank citation ranking: Bringing order to the Web. *Technical report*, Stanford Digital Library Technologies

Ponte J M and Croft W B 1998 A language modeling approach to information retrieval, In: *Proc. SIGIR*, ACM 275-281

Qin T, Liu T-Y, Xu J and Li H 2010 LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. *J. Inf. Retrieval* 13(4): 346–374

Robertson S E, Walker S, Jones S, Hancock-Beaulieu M M, Gatford M, Gull A and Lau M 1992 Okapi at TREC, In: *Proc. Text Retrieval Conference*, pp. 21–30

Salton G and McGill M J 1986 *Introduction to modern information retrieval*. New York: McGraw-Hill, pp. 98–112

Scott C Deerwester, Susan T Dumais, Thomas K Landauer, George W Furnas and Richard A Harshman 1990 Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.* 41(6): 391–407

Sim K M and Wong P T 2004 Toward agency and ontology for web-based information retrieval. *IEEE Trans. Systems, Man, Cybern.* 34(3): 257–269

Sparck J K, Walker S and Robertson S E 2000 A Probabilistic model of Information Retrieval: Development and Comparative Experiments. *Inf. Process. Manag.* 36(6): 779–808

Veningston K and Shanmugalakshmi R 2014 Information Retrieval by Document Re-ranking using Term Association Graph In: *Proc. ACM Intl. Conf. on Interdisciplinary Advances in Applied Computing (ICONIAAC)*, Article No. 21.

Viswanathan V and Ilango K 2012 Ranking semantic relationships between two entities using personalization in context specification. *Elsevier Information Sciences* 35–49

Wang W, Do D B and Lin X 2005 Term Graph Model for Text Classification, In: *Proc. Lecture Note in Artificial Intelligence*, 3584: 19–30, Springer

Wong S K M and Raghavan V V 1984 Vector space model of Information Retrieval: A reevaluation, In: *Proc. SIGIR*, ACM 167–185

Wu Z and Palmer M 1994 Verb semantics and lexical selection. In: *Proc. Annual Meeting of the Association for Computational Linguistics*, 133–138

Yi J and Maghoul F 2009 Query Clustering using Click-Through Graph. In: *Proc. 18th Intl. Conf. on World Wide Web (WWW)*, 1055–1056

Zhang B, Li H, Liu Y, Ji L, Xi W, Fan W, Chen Z and Ma W-Y 2005 Improving web search results using affinity graph, In: *Proc. SIGIR*, ACM 504–511