Concept based Ranking of Results using an Ontology and Fuzzy Network for a Personalized Web Search Engine

B.Bhaskara Rao Dept. of IT, G I T GITAM University Vishakapatnam-45 Valli Kumari vatsavayi Dept. of CS&SE Andhra University Vishakapatnam-03

ABSTRACT

For a given end user query, a personalized search engine returns an enormous set of related results. The results pertinent to a user are not regularly put on the top. The most fretting issue for the user would be to quickly find the related information in the first few. An efficient personalized search engine should be able to rank the search results and display more relevant ones as first few on the top. It is much more convenient for any user to find their required related result with lesser effort to search for it in the wide and huge list of information produced from the search results. The ranking of personalized web search results is a process of finding small number of highly relevant documents from large number of search results. The relevance is dependent on the user query and context of the subject. Ranking reflects the most relevant results to the user. These are very few and to be placed on top. In this paper, we proposed a method for ranking of search results using fuzzy networks that have been developed using enriched extended user profile. Our approach learns the user profile and constructs fuzzy net by calculating togetherness between concepts, documents or both. This can be done in two phases. In the first phase, we construct the fuzzy nets with enriched extended user profile. In second phase, we evaluate the rank of each document by using clustering algorithm.

General Terms

Architecture of Work Flow (AWF), Core Concepts (CC), Enriched Extended Concepts (EEC), Document Affinity Matrix (DAM) and Document Cluster Matrix (DCM).

Keywords

Personalized Search Engine, Ranking, Fuzzy Networks, Document, Ontology, Concepts, Relevance, Enriched Extended User Profile.

1. INTRODUCTION

Information on the Internet is growing swiftly i.e., the information and data from various sources are updated every second on the Internet. As the Internet source grows along with it, the congestion and complexity of similar things grow and the possibility of finding the relevant results decreases. Simultaneously it contains huge information of complicated hyper-texts and documents. The most frequent and big worried issue of the user will be how to collect the related and useful information from the huge Internet effectively and promptly. A lot of work has been done on the search engine, but some way that has not gratified the users. For instance, take a specific search request where the web search engines return a large number of documents. In that the results relevant to a user are not often among the top few. The user is forced to scan a long list of documents linearly or one by one. This process consumes more users' time, patience and effort. This paper aims to provide an interactive and friendly tool which is efficient enough and provides appropriate results. The relevance of the results depends on the perception of the user, context and query. Today most of the search engines return the same set of results without considering the users perception and need as important. As the information on the Internet is growing exponentially need to serve an end user by the search engine by taking into account the interests and needs.

Unlike other text documents, web documents have certain specifications like anchors, hyper links and internal metadata. The technique that finds the relevance of different documents increases the retrieval performance. Some web search engines rely on link analysis techniques that take advantage of the structure of web to determine the degree of relevance. This analysis is might be static and is analyzed based on assumptions or predefined data and ignore the semantics of the entities or words. Moreover the limitations of traditional retrieval mechanisms, the keyword based information retrieval technology cannot meet the need of users with semantic knowledge. The semantic web aims to achieve better results by applying semantic web operations such as interoperability, data automation and annotation [1]. The Semantic Web uses ontology to enhance the search mechanisms. Ontology is defined as collection of all concepts and their relationships to assesses and describe the things. The Resource Description Framework and Web Ontology Languages are data representation models recommended by W3C [2]. The basic technique for creating the semantic web is to use the words defined in ontology as metadata to mark up the web's material. Most of the information models do not exploit the semantics of the user queries. The problem of improving relevance in search and ranking of documents requires techniques that consider the semantic annotation.

Moreover in traditional searching methods, the ranking is calculated based on the document content; however the huge data of Internet would be a great challenge to the traditional information searching technology. The simplest search engine is keyword-based search engine that exploit the literal match between the index terms of a query and document to be retrieved from the Internet. Some documents are relevant to the given query may not be returned by the search engine unless the document includes few or more of those terms in the given query. So, it is very difficult to retrieve and determine their order of priorities or ranks. On the other hand the web document is different from the general text since web document is semi-structured text includes structured information and moreover the web page does not exist independently because the links indicate the interrelation between the web pages. Document rank method uses the link characteristic to evaluate compactness of the pages, which determine the ranks of results. Marchiori [3] believes that the reference of the web page is up to the amounts of linking to it.

Google search engine uses document characteristics and documents relevance to help the keyword-based search provide more related results [4]. The relevance of the web document can be identified by taking the number of incoming and outgoing links in that page or web document. A page has more number of links to the related documents means that the page can be more related or close to the referring pages or user query. If a document is highly referenced by other related documents then the document is treated as highly ranked with respect to the given query of the user. Personalized search engine [5] is one that has been proposed for retrieving the documents by understanding the query and also considering the user interests that's why it retrieves the documents which are relevant to the user query. We aim to improve the results by assigning ranks using fuzzy nets with extraction of semantics on user query and mapped to non linked documents which are retrieved by the personalized web search engines.

2. RELATED WORK

Many techniques are used in personalized search engines to provide the significant or relevant set of results for a given user query. A lot of research in information retrieval investigates matching user query to a set of collections for to get suitable results. All the above methods return the same results approximately for given query regardless of who submitted the query. Some works on this contour have been summarized. Early search engines having traditional information retrieval analysis technology attracted more attention, such as multi-substring test technique, vector model, natural language processing, neural network and so on. For example, one classical method is the weighted frequency and location algorithm [6]. The main idea is to find out the relevance between query and document based on the query keywords weighted frequency. The disadvantage of this method is over confidence on the importance of the words, without thinking expectation and need of the user and relevance of information in the web page. Sometimes malicious keywords are to be processed in order to improve the search results.

In Page-Rank algorithm [7], the web page linked by many page is surely a high quality page because links from the other pages could be considered as the recommendations to the first page. The backward linking of high web page evaluation will receive high evaluation i.e., a few links will receive high evaluation and the page with many links will receive low evaluation. From the analysis of page rank, the value is related to the situation of, but not to the searching. Moreover the page rank is off-line calculation and is acquired by the matched key words. It ignores the relativity.

Sugiyama et.al [8] performed personalization by constructing a term matrix using memory based collaborative filtering algorithm and applied traditional collaborative filtering predictive algorithms to predict a term weight in user profile. The ranking of search results is done using the use profile and profile of other users in the community as selected by the user. Several other works have been published, but in most of the works at least one of user query or document are not used to retrieve the context based information. Moreover, they focused on providing search results to users, they do not consider the semantics of queries and document. On the other hand, our approach focuses on providing results based on the conceptual matching. So, we believe that the result will be depending on the context or concept of the user query which we use in our work.

Pitkow et al [9] have introduced two approaches for personalized search: query expansion and result processing. Query expansion is to map a user's query to the user's preferences or preferred concepts in order to eliminate the ambiguity of the given query. Result processing is to re-rank search results according to the user's preferences. In this approach, the web directory is broadly used to represent the user's preferences which are predefined. Some time it might not retrieve the relevant information and the same result set may be obtained for more than one user's query. Our approach justifies that concepts are co-related even though some concepts are not in user query.

In [10], semantic annotation is about assigning to the entities in the text links to their semantic description. The usage of domain ontology's are employed for the annotations. To improve the recognition of indexing terms, it is possible to weight the concepts of a document. For an instance, the concepts that form semantically related terms gain more weights. The manual annotation of document is expensive and error prone task. This can be avoided by automating the annotation of document that requires predefined ontology. In our approach we extract the concepts from existing ontology WordNet. In [11], the author uses relationships for finding relative results and ranking documents that exploit the semantic. Thread-activation method has been applied for searching related documents. The main difference from our work is that their method puts weights on entities as part of the search process. In our method, only the concepts of query are used during the document retrieval as well as ranking. The main reason for which we do not use other literals of entities is that there might be a chance of ambiguity because of large variety of information in literals of entities that is not relevant for search process and ranking process.

3. PROPOSED APPROACH

The proposed approach constitutes of three major steps. The first step is preprocessing the given user's query, extract the related and nearest concepts for further use. The second step creates the fuzzy net by using an ontology which is a concept descriptor in English. The last one evaluates the ranks for the search [12, 13] results. Preprocessing the given user's query is nothing but removing the unnecessary concepts and adding more precise concepts to the user's query. Extracting concepts from the given user's query means collect related concepts or entities from ontology. Here we use the WordNet as resource and expand the given user's query by adding the synonyms, hyponyms and hypernyms. The number of applications where WordNet is being used as ontology rather than as a mere lexical resource seems to be growing eternally. Indeed, WordNet contains a good coverage of both the lexical and conceptual palettes of the English language. The expanding of query is done by using WordNet as a set of entities called core concepts (CC). Fuzzy net is constructed by using existing user profiles and core concepts and represented in the form of matrix. Calculate and assign a value f $(c_i, c_j) \in [0,1]$ to the edge that connect the given two concepts in the fuzzy net. The work flow diagram of the proposed approach is given in figure 1. It consists of components, open source data, and outputs. The user query is the input to the search engine that retrieves the documents as output. Open source Ontology and user query are two inputs to the component preprocessor and

produce enriched concepts. Document affinity matrix and Document cluster matrix are outputs used by the evaluation component to produce the ranks for the documents.

3.1 Core Concepts and User Profile

A user's query is a representation of his/her needs to be extracted from the Internet and represents one category of his interests. Documents are relevant to the user interest with respect to the context carried by the query. Some documents might not be relevant to the user may be relevant to the other user for the same query. It is always better to preprocess the query before giving it to the search engine. Preprocessing means removal of irrelevant concepts and adding precise concepts or words based on the past history. The query is left as it is, if it is first time to enter into the profile. The assumption is that the result of the query not only depends on the query and elements or concepts of the context but also depends on the concepts of previous retrievals called as user profile. The user's interests are extracted using ontology and kept in a set C called core concepts of the given user's query. The extraction is done by determining the synonyms, hyponyms and hypernyms of the each and every word in the user's query.

$$C = \{ \forall w_i \in Q \} \cup \{ c_j \in O \mid \exists w_i \in Q, \sigma (c_j, w_i) = 1 \text{ or } c_j \leq w_i \text{ or } w_i \leq c_j \}$$

where wi is the concept in a user query Q
O is the used ontology
 c_j is the related concept in O

A user profile is a log file that describes his or her past history. A user can be interested in different kinds of results. We here consider a user profile as a collection of discrete sub profiles. Each sub profile corresponds to the user's interest in a particular context. Each sub profile is understood from the queries that belong to a particular kind of the relevant documents. The core concepts and user profiles are used to create the fuzzy net.



Fig. 1: Architecture of the Work Flow Diagram



Fig. 2: Structure of the fuzzy Concept model



Fig. 3: Fuzzy concept network

Ę

3.2 Fuzzy concept Net

The information systems based on the Boolean logic model cannot represent uncertain information. The query processing of these systems does not handle uncertain information properly. Fuzzy net theory has been proposed to provide a reverberation mathematical model to deal uncertain information. A fuzzy net is used in personalized search engines includes nodes and direct links, where each node represents a concept c_i , each direct link connects two concepts or concept to document and is labeled with real value belongs to [0,1]. Initially, we begin with query concepts, enriched query concepts and user enriched profile are treated as nodes. The relevance between the two concepts is calculated as a function f (c_i , c_j) \in [0, 1]. The degree of the relevance between two concepts f(c_i , c_j) is defined below.

$$f(c_i, c_j) = \frac{2\left|F_{c_i} \cap F_{c_j}\right|}{\left(\left|F_{c_i}\right| + \left|F_{c_j}\right|\right)} \quad c_i, c_j \in C$$

The F_{c_i} is calculated from the enriched user's query as a set of related and closed concepts of c_i at a specific threshold

value that has appeared in the user's profile. Similarly F_{c_j} is a set of enriched concepts as similar as before. If there is no such concept in the user's CC, then the value of 'f' is zero. The value zero indicates that the node is dead end. The above formula is applied for all concepts in core concepts and enriched user's profile. If there is no concept to insert into the fuzzy net then all these values are tabulated as a matrix (R) of size n by n where n describes the number of concepts in C is called degree of relevance [5] and calculate R* that describe the transitive closure of the relevance matrix. For a specific value $t \in N$, $R^* = R^t = R^{t+1}$. Here we have extended fuzzy concept network by adding the result set $D = \{d_1, d_2, d_3, \ldots, d_n\}$ d_{p} of the personalized search engine. The size of the result set is 'p'. The fuzzy concept network consists of three parts: ranked document layer, fuzzy net and core query concept layer. Here we find the degree of the relevance between the concepts to each and every document in the result set of the personalized search engine. The degree of the relevance will be calculated as a function of $g(c_i, d_i) \in [0,1]$.

$$g(c_i, d_j) = \frac{\alpha |g_{d_j}(c_i)|}{|d_j|}$$

where $n = |C|, \forall c_i \in C, \forall d_i \in D, \alpha \in [0, 1]$

The value of $g_{d_i}(c_i) \in [0,1]$ describes set of related and closed concepts of concept c_i in the document d_i i.e. the number of times the concept (c_i) has appeared in the given document (d_i) . If the document is not related to the concept then the value of $g_{d_i}(c_i)$ is zero and the fuzzy concept network has not included such links and documents. The number of concepts is 'n' and number of documents is 'p'. The figure 2 describes the context diagram of the fuzzy concept model. Figure 3 illustrates the fuzzy concept network for personalized search engine. It consists of nodes and links. Each node represents either a document or a concept. All documents are closed to upper layer and concepts are in between the documents and above the query concept layer. Each link bridge the gap with value between two concepts or concept and document by the functions $f(c_i, c_i)$ and $g(c_i, d_i)$ respectively. Function f will calculate the togetherness or closeness between the concept and document. It measures that at what level the document related to the given concept in quantitative figures. Similarly $g(c_i, d_i)$. Fuzzy concept model exploits concept as hierarchical knowledge base. Each concept within a knowledgebase has description that is allowed to differentiate the concept. The degree of the relevance of a document (d_1) with respect to the concepts is a vector (v_1) . Similarly for the document d_i we will get the vector v_i . The vector v_i is called document relevance vector. The general form of the v_i is given below.

$$v_{i} = \begin{pmatrix} dw_{1,i} \\ dw_{2,i} \\ \vdots \\ \vdots \\ dw_{n,i} \end{pmatrix} \text{ where } i <= |D|, n <= |N|$$
$$dw_{b,i} = g(c_{i},d_{b}) b=1..n$$

by tabulating all the relevance vectors of the documents a matrix (V) of size n by p is obtained. The matrix V is called document relevance matrix which is a dynamically calculated matrix. The relevance of the document can be improved by

calculating $U=R^* \odot V$ as a Boolean max of min matrix multiplication. The operator Θ has been defined as below.

$$u_{iOj} = \vee^n_{k=1} (r_{i,k} \wedge v_{k,j}) \forall i = 1..., n, j = 1..., p$$

The dynamic relevance of the document with respect to the given user query is as simple as by using concept fuzzy network that includes user's profile concepts, query concepts and documents. Document relevance matrix shows relevance of documents with respect to the semantics of the user query. If any column contains zeros mean that the document is not related to the user query. Fuzzy net can determine the irrelevant documents. Now we can assign ranks to the documents. User's profile is not a static data structure, it will be updated dynamically. During every execution of user's query the user's profile will be increased by adding the query concepts. So, user profile is a dynamic data structure. Relevance and rank of the documents are depending on the user's profile; therefore these values are not static. Every time the execution of the user's query will produce new values for both rank and relevance of the documents. By putting the threshold value we will limit the user's profile size.

3.3 Rank of Document

The documents required are related to query given by the user. The following argument is exclusively on what needs to be determined for ranks that will run against the result set of the personalized search engine with respect to the user query. The default rank of each document is the serial number or the order in which they are. The re-rank [14, 15] arranges the documents in an order when usually accessed together and there is a need for some measure that would define more precisely the notation of togetherness. This measure is the document association for re-ranking, which indicates how closely, related the documents are. The personalized search engine takes user's query as input and produces set of documents (D) as result. The set D consists of word documents, text documents or both. The set C consists of core query concepts. Re-ranking of the search results will be a three step procedure. First step determines the document usage matrix with respective to the concepts. Since, for each concept (c_i) and document (d_i) there exists a document usage value, denoted by use (c_i, d_j) and defined as follow.

$$use(c_i,d_j) = \begin{cases} g_{d_j}(c_i) & k/use(c_i,d_j) > 0\\ 0 & 0 \end{cases}$$

Document relevance matrix [16] U is used to pre-process the documents at its first stage. Documents have zero concepts i.e., no concepts are included in the document, are not related to the user query. So, carrying such unrelated documents makes inefficiency, slowdown the process and time. Such documents are identified and avoided before going to the second step. The U is not sufficient to form the documents in ranked order. This is because these values do not represent the ranks of document but represents set of related concepts of the query. The closeness between documents can be measured as documents affinity aff(d_i, d_j), which measure the togetherness/strength between the two documents according to how they are accessed by the concepts. The document affinity measures variance between two documents d_i, d_i with respect to the set of concepts $C = \{c_1, c_2, c_3, \ldots, c_n\}$. The aff(d_i,d_i) measures dependencies between two entities and it is defined as

$$aff(d_i, d_j) = \frac{\sum_{k=1}^{n} (d_{i_k} - \overline{d_i}) (d_{j_k} - \overline{d_j})}{n-1}$$

where $d_{i_k} = use(c_i, d_j)$

The result of this computation is p by p matrix called Document Affinity Matrix (DAM). Each element of which is calculation of the variance between two entities here we called documents. The document affinity matrix will be used by Clustering Algorithm BEA that creates the Document Cluster Matrix (DCM) to guide the ranks of the documents and groups similar documents at one place with a specific threshold value. We start by placing first document at first place and order the next document based on high permutation criteria. The Bond Energy algorithm [12] takes as input the document affinity matrix, permutes its rows and columns, and generates a Document Cluster Matrix. The permutation is done in such a way as to maximize the value.

$$perm(d_i, d_k, d_i) = 2bond(d_i, d_k) + 2bond(d_k, d_i) - 2bond(d_i, d_i)$$

whre bond
$$(d_x, d_y) = \sum_{z=1}^{p} aff(d_z, d_x) aff(d_z, d_y)$$

The last set of conditions takes care of the cases where a document is being placed in DCM to the left of the leftmost attribute or to the right of the rightmost attribute during column permutations, and prior to the topmost row and following the last row during row permutations. In these cases, we assume zero to be the affinity values between the document being considered for placement and its left or right, similarly top or bottom, which do not exist in DCM since the values of concern affinities are

$$\begin{split} & aff\left(d_{0},d_{j}\right) = aff\left(d_{i},d_{0}\right) = 0\\ & aff\left(d_{p+1},d_{j}\right) = aff\left(d_{i},d_{p+1}\right) = 0 \end{split}$$

Creation of Document Cluster Matrix (DCM) is done in three steps.

- a. Initialization: Place and fix one of the column of DAM arbitrarily into DCM. Column 1 has been chosen as first column in DAM as first column of DCM by the algorithm.
- b. Iteration: Pick each of the remaining p-i columns and try to place them in the remaining i+1 positions in the DCM matrix. Choose the placement that makes the greatest permutation value. Continue this step until no more columns remain to be placed.
- c. Row ordering: once the column ordering is determined, the placement of the rows should also be changed so that their relative positions match the relative positions of the columns.

Algorithm

Input: DAM-Document Affinity Matrix is p by p matrix Output: DCM-Document Cluster Matrix begin DCM(0,1) \leftarrow DA(0,1), DCM(0,2) \leftarrow DA(0,2) While index <= p do begin for i = 1 to index-1 by 1 do $\label{eq:calculate perm} \begin{array}{l} calculate perm (d_{i-1}, d_{index}, d_i) \\ end-for \\ calculate perm((d_{index-1}, d_{index}, d_{index+1}) \\ loc & \leftarrow placement given by maximum perm value \\ for j= index to loc by -1 do \\ DCM(0,j) & \leftarrow DA(0, index) \\ Index & \leftarrow index+1 \\ End-while \\ Order the rows according to the relative ordering of \\ column \end{array}$

End.

The calculation of the bond between two documents requires the sum of multiplication of the respective elements of the two columns indexed by these documents. The sum is used in calculation of perm that will determine the strength between the documents. Repeat the same for all alternatives and choose the largest one to place the documents in right locations. Finally, the rows are organized in the same order as the columns to make it as symmetric. According to the initialization step, we copy column 1 and 2 of the DAM matrix to the DCM matrix and start with column 3. There are three alternative places in case of 4 by 4 where column 3 can be placed to the left of column 1, in between column 1 and 2 or to the right of column 2. Based on the highest value will be selected and placed according to the order in perm. The objective of the above algorithm is to find an ordered set of documents that are retrieved solely by query of the user. We split the DCM into four parts to find sets of documents that are relevant to the query. To splitting, we need to find the point in the above matrix is called fragment point. It is a location that split the matrix into upper left, lower down, upper right and lowers down. According to the need and interest of the user priorities i.e., high to low or low to high values, take the concern part of the matrix. Here we have chosen a point where all four parts of the matrix are met at s single point in somewhere middle of the matrix. The value at splitting location is the largest in concern both row and column indexes. Depending on the need, importance and time we take top 3 or more documents from the splitting point.

4. EXPERIMENTAL RESULTS

To evaluate the performance of concept based ranking of personalized search engine results, we have to examine the query that is submitted by a user since the query may be dependent on the individual preferences. We use concepts domain WordNet [17], which is open source software. User query is divided into three or more precise words from which we have extracted concepts related to the user query. From each sub-query we have extracted all related concepts from the domain. For example, a query might have related to the concepts nearly 10 to 500 depending on the fuzzy terms in the query. Extractions of such related concepts are retrieved by our tool that will be shown in the figure 4. We have extracted related concepts by using our own tool: Concept Extractor consists of methods that are producing related concepts i.e., synonyms and hyponyms at a specific threshold value. These will be stored in a specific text file called enriched concepts. Enriched concepts contain some connectors not having any meanings, so we have to remove it. It is always better to prepossess the enriched concepts before used in further methods. The preprocessed enriched concepts are shown in the figure 5. The affinities between any two documents measure the strength of it. These values are tabulated as a

matrix called Document Affinity Matrix (DAM) is in figure 6. It is a symmetric matrix of size equal to number of documents. We have taken a range of 25 to 500 documents with 1,000 to 50,000 terms in each document. We have developed a methodology to automatically extract association relations. We chose the different queries randomly and executed on different sets of documents. The Document Clustered Matrix is created by using DAM with cluster algorithm called Bond Energy Algorithm. Figure 7 describes the right lower of the DCM of 25-documents. The DCM creates four clusters: one is in the upper left corner contains the small affinity values, the second one is in the lower right corner contain large affinity values and remaining two clusters contain very low values and one is transitive to the another one. All these clusters are met at value that will be the biggest in the row and concerned column indexes. Don't consider upper left and lower right clusters since these are very low affinity values and not related to the user interests. Choose a right cluster and assign ranks. Assignment of ranks will be from large affinity value to small. The partition point will separate the first d-most related documents which are more related to the query. In our experiment results, we have tested with different values of the constant and compare results with expected results. The preliminary results seem very encouraging and provide us with the conviction that it can be completed in acceptable time with the support of semiautomatic techniques. The enriched user query is obtained by our own tool. For example, the user query is "Privacy Measurements". The output screens describe the step by step of getting the enriched user query concepts, which are given below.



Fig. 4: Tool Elicitation of concepts or words

angles arra	nging	arrives	assigni	ng	at	athymet	ry
between bloo	d body	by		1000	10		
can capa	cities	capacity	1	carbon	changes		- 1
changes chem	ical	chest	conceal	ed	conceal	ment	
concentration	ns conditi	on	confider	ntiality	conside	red	C
covertness	ctinome	try	data	density	depth		
depths deter	rmine	diameter	C. 85.01.091	dioxide	directi	on	
disposition	distanc	es	drometry	V	each		
eading earth	nguakes	electron	agnetic	element	emitted		
ensitometry	ephalom	etry	especia	11v	estimat	e	e1
fetus foet	ometry	function	1	gases			
gravimetry	gravity	around	groups	head			
heads hear	ing heat	hidden	hidden				
hiddenness	human	hypernyn	15	hypsogra	aphy	instrum	ent
its kept	level	light	map		1 1		
means measu	ure measure	ment	measurer	ment	measure	ments	- 1
measuring	mensura	tion	meter	meters	microph	ones	
movements	nemogra	phy	nemomet	rv	ngulati	on	nt
numbers number	ers object	oceans	optical	osimetry	v	oxvgen	pł
phenomena	phenome	na	piromet	rv	place	plotted	
plumbing	practic	e	precise	prefer	presenc	e	
presence	pressur	e	privacy	privacy	private	ness	
			p racj	P rac)			

Fig. 5: Result of preprocessed enriched concepts

1	0.080545	0.147132	0.07672	0.080626	0.078053
0.080545	1	0.057005	0.049662	0.052215	0.051603
0.147132	0.057005	1	0.021119	0.022068	0.021869
0.07672	0.049662	0.021119	1	0.548424	0.542402
0.080626	0.052215	0.022068	0.548424	1	0.573515
0.078053	0.051603	0.021869	0.542402	0.573515	1
0.07603	0.049094	0.021417	0.548353	0.546129	0.546376
0.077309	0.050619	0.021084	0.557004	0.553548	0.560408
0.074756	0.048343	0.02081	0.556107	0.539939	0.544261
0.078882	0.051616	0.02195	0.548589	0.555346	0.558004
0.077748	0.051284	0.021967	0.541082	0.556862	0.557352
0.073406	0.048806	0.02025	0.543427	0.533618	0.53172
0.076557	0.050481	0.021396	0.554859	0.54977	0.55069
0.073919	0.049629	0.020739	0.546724	0.530423	0.532682
0.075413	0.04967	0.020957	0.557911	0.543729	0.535231
0.075893	0.049528	0.02078	0.554743	0.545791	0.542587
0.087182	0.056529	0.02498	0.485976	0.494179	0.491313
0.084847	0.055976	0.024649	0.482841	0.493658	0.492523
0.077881	0.049888	0.021982	0.536215	0.544307	0.544479
0.07856	0.051238	0.021789	0.547352	0.551877	0.548813

Fig. 6: Document affinity matrix of 25 documents

Doc 3	Doc 10	Doc 1	Doc 2	Doc 9	Doc 13
0.576267	0.560719	0.557902	0.55623	0.559393	0.558624
0.562307	0.552066	0.555213	0.562129	0.558183	0.55439
0.557955	0.546205	0.547698	0.560719	0.555213	0.55439
0.55269	0.542177	0.547065	0.559393	0.55439	0.552129
0.552475	0.540104	0.540104	0.557902	0.54519	0.547698
0.54398	0.539219	0.538127	0.550272	0.54519	0.546205
0.5393	0.529966	0.536845	0.542838	0.539219	0.541254
0.53341	0.502294	0.532273	0.523084	0.538735	0.539969
0.509766	0.49447	0.474688	0.490598	0.529891	0.521234
0.49447	0.491049	0.392106	0.490598	0.477405	0.482261
0.408557	0.407247	0.392106	0.406519	0.392511	0.397855

Fig. 7: Right Lower part of the matrix 25x25

ts
t

Docs	Ranks	Closeness
Doc03	1	0.576267
Doc10	2	0.560719
Doc01	3	0.557902
Doc02	4	0.55623
Doc09	5	0.559393
Doc13	6	0.558624
Doc08	7	0.558183
Doc12	8	0.498891
Doc14	9	0.490595
Doc11	10	0.435609

5. CONCLUSION

For the re-ranking of the personalized search engine results, we have used fuzzy net to analyze the togetherness between two entities and/or documents. This process has calculated the DCM matrix that keeps the documents in an order. By using refined user's query, the result from the different queries can be expanded and relatively improved. User profile is a dynamic data structure according to that we have calculated DCM matrix of togetherness between the documents which is also be changed dynamically. Therefore, it solves the problem of static page ranking. Since page rank value is only related to the situation of , but not to the searching. Moreover, we have been considering the relative importance of the concepts. We evaluated our approach against the simulated data. We have tested our technique against the one specific document of 50 or 500 documents of each one lakh and fifty thousand words. Our evaluation has shown an improvement of performance by concept net and document togetherness algorithm.

6. REFERENCES

- Guan-yu LI, sui-ming YU and Sha-sha DAI, "Ontology based query system design and implementation", International Conference on network and prarallel computing, pp.1010-1015, 2007.
- [2] Wang Wei, Payam M.Barjaghi and Andrzej Bargiela, "Semantic enhanced information search and retrieval", Sixth International Technology, pp.218-223, 2007.
- [3] Marchiori, M.1997, "The quest for correct information on web: Hyper search engines", in proceeding of the 6th International World Wide Conference.
- [4] S.Park, D.M.Pennock, "Applying Collaborative filtering techniques to movie search for better ranking and browsing", in Proc KDD'07, 2007, pp.550-559.
- [5] Bhaskara Rao Boddu, Valli Kumari Vatsavayi, "A Modified Ontology Based Personalized Seaech Engine Using Bond Energy Algorithm, ACC(2) 2011: 296-306.
- [6] Michal Cutler, Yungming Shih, Weiyi Meng. "Using the Structures of HTML Documents to improve retrieval". Usenix symposium on Intenet Technologies and Systems, 1997.
- [7] L.Page, S.Brin, R.Motwani, and T.Winograd, "The PageRank CitationRanking: Bringing Order to the WEB", January 1998.

International Journal of Computer Applications (0975 – 8887) Volume 81 – No.13, November 2013

- [8] K.Sugiyama, K.hatano, andM.Yoshikawa. "Adaptive web search based on use profile constructed without any effort form users". In Proceedings of WWW 2004, pages 675-684, 2004.
- [9] J.Pitkow, H.Schutze, T.Cass, R.Cooley, D.Turnbull, A..Edmonds, E.Adar and T.Breuel, "Personalized search", Communication of the ACM, vol. 45, pp.50-55, 2002.
- [10] Sun Kim and Byoung-Tak Zhang, "Genetic Mining of HTML Structures for Effective Web Document Retrieval", applied Intelligence, vol.18, pp.243-256, 2003.
- [11] C.Rocha, D.Schwabe, and M.P.Aragao. "A Hybrid Approach for Searching in the Semantic web". In Proc. 13th International World Wide Web Conference, NewYork, NY, pp. 374-383, 2004.
- [12] W.T.McCormick, P.J.Schweitzer, and T.W.White "Problem Decomposition and data reorganization by a clustering Technique" Oper. Res. 1972 20(5): 993-1009.

- [13] Tian Chong "A Kind of Algorithm for PageRank Based on Classified Tree In Search Engine", International Conference on Computer Application and System Modeling, vol. 13, pp 538-540, 2010.
- [14] Rohini U, Vasudeva Varma "A Novel Approch for Reranking of Search Results using Collaborating Filtering" IEEE proceeding of ICCTA 2007, pp 491-496.
- [15] Jae-won Lee, Han-joon Kim, sang-goo Lee "Applying taxonomic Knowledge and Semantic Collaborating filtering to Personalized Search: a Bayesian belief Network Based approach" International Asia-Pacific Web Conference, 2010, pp 75-81.
- [16] Jianxiong Yang, Junzo Watada "Decomposition of Term-Document Matrix Representation for Clustering Analysis", International Conference on Fuzzy systems, pp. 976-983, 2011.
- [17] WordNet, a lexical database for English, Princeton University. www.princeton.edu/wornet/download/current-version/.