Middle-East Journal of Scientific Research 24 (7): 2385-2394, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.07.23370

Modernized Information System Using Semantic Web Through the Experts Ranking Process

[1]K. Sridharan and [2]M. Chitra

[1]Department of Information Technology,
Panimalar Engineering College, Chennai, Tamilnadu, India
[2]Department of Information Technology, Sona College of Technology, Salem, Tamilnadu, India

Abstract: Searching for pages on the www is challenging today and people dump a huge amount of data either authorized or unauthorized by every day. In addition every hunter hopes they don't retrieve a lot of junk. Unfortunately getting "everything" while avoiding "junk" is very complex, if not impossible, to accomplish. It becomes difficult for the people to get the content what they seek. However, it is possible to measure how well a search performed with respect to precision and recall. Many Information retrieval techniques are introduced consecutively in order to make people's task ease and they are working pretty well but still people are getting junk other than the content what they expect to seek and more time. This paper mainly focuses on the task overcome the above problem such as what the information retrieval performs and refines them so that the accuracy of the data that is being searched is achieved to the maximum . For this we use the concepts of cache-based semantic checking which includes data extraction, clustering and identification of semantic similarity between the entities to refine the searching process. Finally, in order to rank and to rate, an experts' system is used.

Key words: Search Engines · Data Extraction · Data Clustering · Semantic Similarity · Expert Rank System · Web Intelligence

INTRODUCTION

World Wide Web acts as a major source of information in the world today. As people rely more on internet, the contents get increased so much that they could not be properly handled. Spotting the content on the internet that we look for is something strenuous. According to a recent research, there are nearly 40 billion active Web sites on the Internet today. You also remember that search engines classify the content on the page by using reference points like the site's title, paragraph headers and other usability information. As a whole, the way a site links to its own pages is also a big hint for a search engine, so your website navigation should be easy for people and their robot counterparts. The Google uses an algorithm called Page Rank, it assigns each web page a relevancy score based on frequency of keywords, webpage expires and link of other webpage. But it does not bother about content accuracy as well as frequency of exact answer and its page. Even though the process of information retrieval is done by automatic system with web, it should require small manual verification by the expert when the user content uploads in the web at least once. Because of maintaining accuracy of content which leads to avoid the unauthorized and false content. The task of reviewing all those websites to find the content what we are looking for is something imaginary. As a result, a lot of research is under process in order to handle this bulk amount of documents. The offshoots of these researches are the enhanced information retrieval. 90% of the content that are displayed after the search is not relevant to the query that is searched. For example if we want to search information about definition of service oriented architecture, the current Google search engine displayed the results as follows:

Corresponding Author: K. Sridharan, Department of Information Technology, Panimalar Engineering College, Chennai, Tamilnadu, India.

Table 1: Keyword Based Information Retrieval Sample from Top Two Search Engines

Aim	Possible Sample Keyword	Search Engine	Results	Time
Definition	Define Service Oriented Architecture	Google	18, 00, 000 results	0.37 seconds
	Define Service Oriented Architecture	Bing	469, 00, 000 results	-
	Define Of Service Oriented Architecture	Google	22, 70, 000 results	0.24 seconds
	Define Of Service Oriented Architecture	Bing	545, 00, 000 results	-
	Define Service Oriented Architecture By Thomas Erl	Google	29, 700 results	0.36 seconds
	Define Service Oriented Architecture By Thomas Erl	Bing	432, 00, 000 results	-
	What is meant by service oriented architecture by thomas ERL	Google	49, 900 results	0.37 seconds
	What is meant by service oriented architecture by thomas ERL	Bing	454, 00, 000 results	-

The above table shows the sample results for the given keyword by top search engines for the same objective like definition of Service oriented architecture with different way. But the answer is not what the user expect properly and still it is also displayed the junk information related to the keyword. Therefore understanding the context is extremely important during the information search process.

This paper undergoes many filter process in order to refine the search results from the search engines. Initially they use word mapping database in order to find the various synonyms and matching for the keyword that is entered which is discussed in preliminary processing A. Once the preliminary processing is over, many algorithms are applied on the search results in order to filter the contents that are extracted from the search engines. Initially the distribution hypothesis is used to identify the matching documents in the web and help in clustering of the documents which is mentioned as lexical pattern clustering after which the comparison is done using the genetic algorithm. They are discussed in measuring the semantic similarity. After which the initial ranking is based on the similarities found using the genetic algorithm and then the experts' ranking is considered. The final ranking of the document is consolidation of the results of the genetic algorithm and the experts' ranking. The obtained result is expected to be more accurate than the results obtained from the normal search engines. The verification of the enhanced results is provided with the experimental results.

Related Work: Ontology's define a common vocabulary to share domain information. Many researchers have proved the importance of ontologies as a main technology for knowledge modeling. The definition of a Semantic Virtual Learning Environment (SVLE) whose purpose is to provide customized and contextualized learning experiences [1]. A framework for semi-automatically learning ontologies from domain specific texts by applying machine learning techniques. The TEXT-TO-ONTO framework integrates manual engineering facilities to follow a *balanced cooperative modeling* paradigm [2]. Without pondering any specific application scenarios with human actors in SOA, a trust management structure has been conferred in [3-5]. Task-based platforms on the Web permit users to divide their competence [6] or users offer their expertise by helping other users in forums or answer communities [7, 8]. Site restructuring or modification can be done by humans based on the users navigational behavior, which can be achieved through extracting useful patterns and rules using data mining techniques[9]. The ranks are calculated dynamically based on success and failure. A skill model is also proposed as a classification system [10]. In [11], a method to rank semantic web services is proposed. Finally, in [12], the authors propose a method to diversify Web service search results in order to deal with that have different, but unknown, preferences. Service Rank [13] considers the QoS aspects as well as the social perspectives of services. Services that have good QoSs and are frequently invoked by others are more trusted by the community and will be assigned high ranks. In [14], Web service combinations can be compared with each other and ranked according to the user preferences.

The World Wide Web Consortium (W3C) developed formal specifications such as RDF, RDFS, OWL and SWRL in order to provide an accurate description of the concepts, terms and relationships within knowledge domain [15, 16] .Multi-meaning words; words that share the similar spelling and pronunciation but have different meanings.

Proposed System: In retrieval strategies, the knowledge / expectation / behavior of the searcher need to be anticipated. Here we are considering the one issues of reader's behavior reflected by the usage which is very different from what the author would like. Need to work harder to hold our readers' attention like be topical, timely, visual, informative, Succinct, yet precise, more relevant, etc. because of dynamic changing behavior of readers.

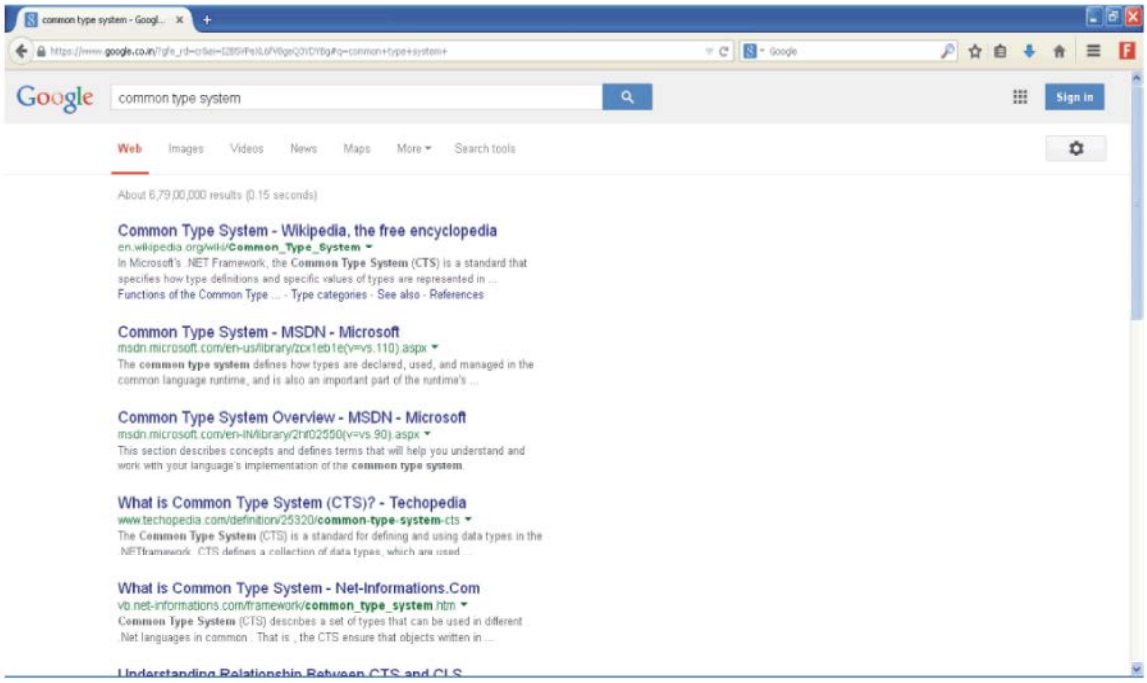

Fig. 1: Show the Sample Result Screen for the Given Query by Learner

For a search query, retrieved documents ought to be concise with 80% - 90% of above mentioned reader factors. Another issue taken here is retrieval information from web is by commonly searched keywords. That is we have used a keyword which is very frequently searched then you get a higher chance of your website being visited. Simply called as more visitors are equal to better page rank. In our approach provide better process for problem stated above through recommended procedure to write and extract content with simple template established by the corresponding expert each other's after resulted pages for the search query now for the first one. In second case, we addressed the problem like more visitors equal to better page rank which leads crucial process in searching now. Because some visitors are clicked the page without knowing the proper content. For example, learner or trainee to search the particular topic like "common type system " in web the result may be as follows.

Analysis of this page is About 6, 79, 00, 000 results (0.17 seconds) but page content quality is not exact what the learner significant. The first link in the above figure is from Wikipedia it is based on more visitors. Here what we address is how much visitors are satisfied their content. If they are satisfied why they go for another search with same keyword. Most of the visitors doing this for current search now. We are introduced the concept to give the weight of keyword based on their domain additions of the current process. For example, the keyword "common type system " is mostly related. Net Framework. So our result will be based on this with more accuracy.

So we are mainly concentrated to rank the page based on the quality content Page Rank is calculated by various algorithms(by the number of links) made by search engines.

The proposed system has two main parts, the preliminary processing and the core processing. The preliminary processing consists of processing of the entered query with a standard database and retrieval of relevant words from the database whereas the actual processing consists of the major processing with the documents that are retrieved from the web which uses the results from the preliminary processing.

Preliminary Processing: Preliminary processing of web data especially link data has been carried out for some application, the most suitable being google style web search. In this proposed system, the initial processing deals with the search query that is provided by the user to the search engines. These query words undergo series of steps that are explained below.

Synonymous Identification: Initially, when the query is passed to the search engine, the search words are extracted by the system and as a first step, the various synonyms of the word is extracted. The usage of

synonymy may have some importance. Since there is a possibility to have variety of words that have the same meaning in different languages. For this process we use a lexical database for the acquiring the various synonyms of a particular term. Those retrieved words are related to the search word and they are used during the data processing.

Another important issue that has to noted is the ambiguity in the sense of a particular term. There is always an necessity to identify the sense of the word in a particular context. This can be commonly observed in the natural languages and for different parts of speech which is called as polysemy. This is a potential source for mistakes in content extraction. To resolve this, we use automatic determination of the most appropriate meaning of the word relying on the context in which it resides.

Terminology Extraction: All the aforementioned sub phases are performed to extract the relevant terminology related to a particular domain. We refer to terminology as the set of words or word strings which convey a single possibly complex shared meaning within a community. Since they have low ambiguity and high specificity, they can be used to address a unique domain.

Reduction of Input Dataset: The earlier algorithm proposed found the U-Skyline from an uncertain dataset efficiently whereas here we propose additional two techniques which are Input dataset reduction (SR) and reduced dataset partition (SP) to increase the pace of the U-skyline search. In addition, the divide and conquer strategy is used for query processing.

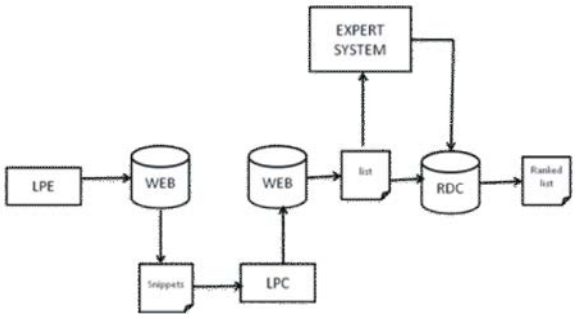

Fig. 2: Proposed system architecture

Reduction is an important concept which is used for reducing dimensions to decrease the computation complexity and time of classification. Since now many approaches have been proposed for solving this problem, but almost all of them just presented a fix output for each input dataset that some of them aren't satisfied cases for classification. In this we propose an approach as processing input dataset to increase accuracy rate of each feature extraction methods. This method is used to process input dataset of the feature reduction approaches to decrease the misclassification error rate of their outputs more than when output is achieved without any processing. The noises that are based on adapting datasets with feature reduction approaches can be handled better with this approach.

Partition and Conquest of Reduced Dataset: The size of the input should be reduced further and so a new technique of divide and conquer is used in order to split the reduced set into a number of disjoint subsets. The U-Skyline is processed for each subset independently and merged to achieve the final U-skyline answer.

Partitioning addresses key issues in supporting very large tables and indexes by decomposing them into smaller and more manageable pieces called partitions, which are entirely transparent to an application. SQL queries and Data Manipulation Language (DML) statements do not need to be modified to access partitioned tables. However, after partitions are defined, Data Definition Language (DDL) statements can access and manipulate individual partitions rather than entire tables or indexes. This is how partitioning can simplify the manageability of large database objects. Each partition of a table or index must have the same logical attributes, such as column names, data types and constraints, but each partition can have separate physical attributes, such as compression enabled or disabled, physical storage settings and table spaces. Partitioning is useful for many different types of applications, particularly applications that manage large volumes of data. OLTP systems often benefit from improvements in manageability and availability, while data warehousing systems benefit from performance and manageability.

Partitioning Offers These Advantages:

- It enables data management operations such as data loads, index creation and rebuilding and backup and recovery at the partition level, rather than on the entire table. This results in significantly reduced times for these operations.
- It improves query performance. Often the results of a query can be achieved by accessing a subset of partitions, rather than the entire table. For some queries, this technique (called partition pruning) can provide order-of-magnitude gains in performance.

Core Processing System: The proposed system has various processes that are executed in order to acquire the expected refined result. These are executed with 5 major processes.

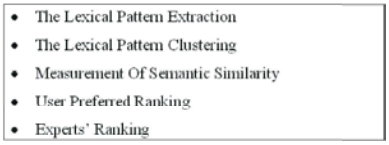

Fig. 2: Architecture of the proposed system

F.1 Lexical Pattern Extraction: Manually constructed linguistics has very high precision entries but has very limited coverage and so, their advantages cannot be still used in many applications or domains. A new approach is done to extract synonyms and antonyms and hyponyms from free text documents. Here the extraction of hyponyms from free text is much concentrated. Here we use a pattern based approach and text mining technique for the extraction of lexical patterns from the web. This lexical pattern extraction has three major process, pattern discovery, instance extraction and instance ranking. This process adopts some ideas from elsewhere but revise the computations of initial weights of the obtained pattern.

F.2 Pattern Discovery: This module discovers a set of lexical patterns from the web. This helps in capturing the common written conventions which is used in introducing a hyponym relation between the words. They mainly use a small set of seed instances for example Lamborghini-car, to collect similar matches from web. Now a mining method is adopted where all the collected results from the web are compared with the given pattern and the most relevant and maximal frequent sequences are stored and the rest are discarded. It retains only the patterns that matches the sequence

<left-string>specific<center-string>general
General<center-string>specific<right-string>

F.3 Lexical Pattern Clustering: Typically, a semantic relation can be expressed using more than one pattern. For example, consider the two distinct patterns, X is a Y and X is a large Y. Both these patterns indicate that there exists and is-a relation between X and Y. Identifying the different patterns that express the same semantic relation enables us to represent the relation between two words accurately. According to the distributional hypothesis, words that occur in the same context have similar meanings. The distributional hypothesis has been used in various related tasks, such as identifying related words and extracting paraphrases. If we consider the word pairs that satisfy (i.e., co-occur with) a particular lexical pattern as the context of that lexical pair, then from the distributional hypothesis, it follows that the lexical patterns which are similarly distributed over word pairs must be semantically similar.

F.3.1 Distributional Hypothesis: Law: The words that have similar meanings occur in same context and words with similar meanings will occur in similar neighbors if enough text material is available [Sch¨utze & Pedersen, 1995]. The distributed hypothesis is constructed based on the assumptions on the kind of language that is processed.

This can construct the word profiles based on two factors:

- The type of Relationship that is identified. This is done by building distributional profiles for words based on the words that surround them and the other approach is to build the profiles for words based on which area of text they exist.
- In what sense the meaning is conveyed by the distribution patterns

Relations That Exist Between the Texts: Syntagmatic relationship are related to the positions of the texts in the sentence or the entities that co-occur. It is a relation with present. These combinations are linear in nature. One example is with a sequence of word that states “The lion is hungry”. They are combinational relations. Here the word that occurs in the sentence can be combined and thus stated as syntagms are combinational relations.

The above sentence can be stated or occurred as:

- The lion is hungry
- The hungry lion
- The lion’s hunger

Literally they measure the co-occurance of words within a text region.

For example: For the word *knife*, the possible co-occuring words could be,
Knife-spoon
Knife-blade
Knife-cut
Knife-cutter head
Knife-noni
Knife-nimuk

The above result shows the possible combination of the knife with other words in the same entities. Noni and Nimuk are the name of a boy and his dog in a famous story where a knife plays a major role.

A table is formed that measures the number of occurrences of the co-occurring word in the document.

Table 2: No Of Co-Occurrences Of A Word In Syntagmatic Relationship

Word	Documents							
	1	2	3	4	5	6	7	8
w_1	0	1	0	0	0	0	0	0
w_2	0	0	1	0	0	3	0	0
w_3	1	0	0	2	0	0	5	0
w_4	3	0	0	1	1	0	2	0
w_5	0	1	3	0	1	2	1	0
w_6	1	2	0	0	0	0	1	0
w_7	0	1	0	1	0	1	0	1
w_8	0	0	0	0	0	7	0	0

In (Table 1) w represents the word and the numbers in the grid represents the number of times the word occur in the document.

Pragmatic relationship is relation with absent. They exist between the words that are in same entities but do not exist at the same time. One example for this pradgmatic model is that different adjective that modify the meaning of the same noun. “Good news” and “bad news”. These relationships need not share relation with the immediate neighbor but also with several other neighbors.

For example, consider the two sentences,

Had an awesome time in titlis.
Had a wonderful time in titlis.

Here awesome and wonderful are pradgmatically related words and so it would be enough to look the successive and the preceding word which is called as 1+1 context whereas when two words are considered as reference in preceding and succeeding, then its called as 2+2 context. Thus for the above example the equation could be formed as:

an awesome time -> awesome: (a 0)+time
a wonderful time -> wonderful: (a 0)+time

where 0 means the word is similar or ignored.

	Pradgmatic relation combinations			
	He	Adores	Green	Paint
Syntagmatic relation	She	Loves	Blue	Color
combinations	They	Likes	Red	Dye

Similarly like sintagmatic relationship, the pradgmatic relationship also takes count of number of co-occurrences of their words and the related words as shown in the (Table 2). They are formed by directional word by word co-occurrence matrix.

Table 3: No. Of Co-Occurrences Of A Word Extracted In Pradgmatic Relationship

Word	Co-occurrents							
	whereof	one	cannot	speak	thereof	must	be	silent
whereof	0	1	0	0	0	0	0	0
one	0	0	1	0	0	1	0	0
cannot	0	0	0	1	0	0	0	0
speak	0	0	0	0	1	0	0	0
thereof	0	1	0	0	0	0	0	0
must	0	0	0	0	0	0	1	0
be	0	0	0	0	0	0	0	1
silent	0	0	0	0	0	0	0	0

The corresponding pragmatic relationship for the word knife can be;

Knife-hammer
Knife-shovel
Knife-pencil
Knife-spoon
Knife-blanket

There are some words that occur paradigmatically for the word knife.

F.3.3 Measurement of Semantic Similarity: We defined four co-occurrence measures using page counts. We showed how to extract clusters of lexical patterns from snippets to represent numerous semantic relations that exist between two words. In this module, we describe a machine learning approach to combine both page counts-based co-occurrence measures(p_i) and snippets-based lexical pattern clusters(c_i) to construct a robust semantic similarity measure.

$$s = \sum_{i=0}^{n} (pi + ci) \quad (1)$$

S = Semantic similarity measure
P_i = Page count measured co-occurrences.
C_i = Snippet based lexical pattern clusters.

F.3.3.1: Genetic Algorithm:

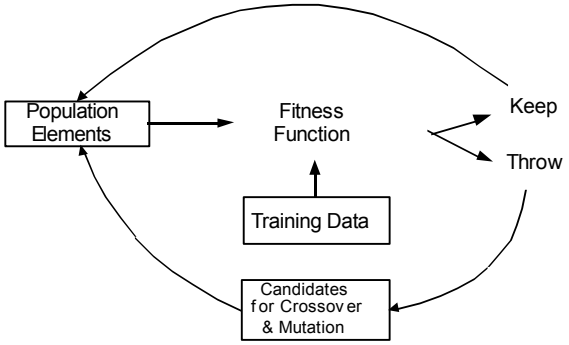

Fig. 2: Process in genetic algorithm

The genetic algorithm mainly uses a fitness function to calculate the match of the search word with the documents. They identify how far the search keyword and the words in the documents are related and in what context as represented in the (Figure 2).

- Number of genes (bits) in the genetic string l
- Population size N
- Number of generations G
- Pk = a population of n randomly generated individuals.

Algorithm:

GA(n, _, μ)
// Initialise generation 0:
k := 0;
Pk := a population of n randomly-generated individuals;
// Evaluate Pk:
Compute *fitness*(i) for each i □ Pk;
do
{ // Create generation k + 1:
// 1. Copy:
Select (1 - _) × n members of Pk and insert into Pk+1;
// 2. Crossover:
Select _ × n members of Pk; pair them up; produce offspring; insert the offspring into Pk+1;
// 3. Mutate:
Select ì × n members of Pk+1; invert a randomly-selected bit in each;
// Evaluate Pk+1:
Compute *fitness*(i) for each i ? Pk;
// Increment:
k := k + 1;
}
while fitness of fittest individual in Pk is not high enough;
return the fittest individual from Pk;

Initialize the Population: Generate N bit strings each of size l. It is easier to represent the individuals as character arrays rather than integers.

For each generation, do the following:

Calculate the Fitness of Each Individual: This can be done in several steps:

- Find the integer that the individual's bit string represents. Iterate through each bit and if the bit is a 1, then add the corresponding power of 2 to a running sum. For example, if l=20 and if the leftmost bit of the string is 1, then add 2^{19} to the sum. The power of 2 that you are currently on is a function of the loop index. After looping through all the bits, this running sum will be the integer value of the bit string.
- Compute the individual's fitness value using the fitness function $F(s)=(x/2^l)^{10}$ where x is the integer value from step a.
- When you are looping through each individual finding the fitness value, keep a running sum of the total fitness, which is the sum of fitness values for all N individuals.
- For each individual, normalize its fitness value by dividing its fitness value by the total fitness from step c. It's best to store these normalized values in a separate array than the actual fitness values since you will need to keep the original fitness values for finding the statistics in a later step.
- For each individual, compute a number which is the sum of that individual's normalized fitness value and the normalized fitness values for each of the individuals before it. Thus, a running total is kept of the normalized fitness values. This will be helpful later when probabilistically selecting parents.

The following example should make steps d and e clearer. Note that the total fitness value is 1.95. Also note that the sum of all the normalized fitness values is The following table shows the Individual Fitness value Normalized fitness value running total for the sample training data.

0 0.05 0.0256 0.0256
1 0.2 0.1026 0.1282
2 0.6 0.3077 0.4359
3 0.3 0.1538 0.5897
4 0.8 0.4103 1.0000

Table 3: Shows Individual Fitness value, Normalized fitness value, running total

0	0.05	0.0256	0.0256
1	0.2	0.1026	0.1282
2	0.6	0.3077	0.4359
3	0.3	0.1538	0.5897
4	0.8	0.4103	1.0000

Now you are set up to select parents and produce the next generation. Perform N/2 iterations doing the following:

Select Two Individuals to Be Parents: First get two random numbers between 0 and 1. Then see which range the random numbers fall into based on the running total numbers computed in step 2e above. The random number will be between two of those numbers. The individual associated with the second of those numbers will be the individual selected to be a parent. For example, suppose your two random numbers are 0.4147 and 0.7395. In the example above, the number 0.4147 is between 0.1282 and 0.4359, so individual 2 is one of the parents. In the above example, the number 0.7395 is between 0.5897 and 1.0000, so individual 4 is the other parent. Be sure that you get two distinct parents when you do this step, so that you do not result in an individual mating with itself. Keep selecting a second parent until you get one that is different from the first parent.

Mate Parents and Perform Any Crossover to Get Offspring:

- First generate a random number to determine whether crossover will be done.
- If no crossover is done, then simply copy the bit strings of the parents into new bit strings which will represent the offspring.
- If crossover is done, then first randomly select a bit to be the crossover point. Then copy the bit strings of the parents into the bit strings of the offspring up to the crossover point. After the crossover point, reverse which offspring gets the bits from which parent.

Perform Any Mutations on the Offspring: For each of the two offspring, go through all the bits in their strings. For each bit, generate a random number to indicate whether that bit will be mutated. If the bit will be mutated, then simply flip that bit, that is, change a 0 to a 1 and a 1 to a 0.

Update the Population: After obtaining all N new offspring in the above loop, copy all of these offspring bit string arrays into the bit string arrays of the current population. These new offspring will replace the current population. In other words, the current population arrays will be overwritten by the offspring arrays.

Find the Statistics of the Population: Find the average fitness of the population, the fitness of the best individual and the number of correct bits in the best individual. Find these three measures for each generation. You will use this data to make the required graphs. You might want to store all this data in arrays which you can then dump into a file later on.

Repeat all of the above for several different runs. Do not average over them as this will either result in loss of data for individual runs or the average will just be a straight line. Plot all of these runs on the same graph and/or choose one of the runs as a "typical" run and plot it.

Also repeat all of the above for several different combinations of the five parameters given at the beginning of this document. For each combination of parameters, do several runs as explained in the previous paragraph.

G Ranking of the Results: In this module, an automatic method to estimate the semantic similarity between words or entities using web search engines with ranking the search results occurs. Accurately measuring the semantic similarity between words is an important problem in web mining, information retrieval and natural language processing. Web mining applications such as, community extraction, relation detection and entity disambiguation; require the ability to accurately measure the semantic similarity between concepts or entities. Based on the similarity between the users given search keyword, the ranking takes place. Not only the users' keyword is used for ranking, but also, the experts' preference is used.

$$R_{at} = \Sigma (e_r + u_p)$$

R_t - Final rank value
e_r - Expert rank value
$u_{p\_}$ User preference value

Expert Ranking: To rank the results that are obtained after the refinement, the third party expert is delved. The

expert is allowed to rank the content using a special authentication. Apart from the ranking, the expert is allowed to provide the value of information in that document using the rating technique. Moreover the ranking is done even using the users preference also. But this can only be developed over a period of time as the visits of the site get increased.

$$e_r = \sum_n (er_i)$$

n – no of experts ranking document

er_i – rank by i^{th} expert (default value if rank not provided)

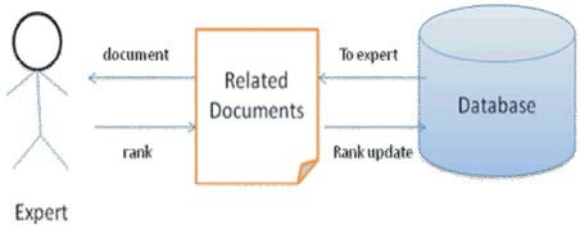

Fig. 3: Structure Of Experts' Ranking System.

Experimental Result and Discussion: In some restrictions on www, may not retrieve relevant documents because of that include conflict synonymous terms which are stated that in introduction? For some example, restaurant vs. café , apple (phone , company or fruit).we are discussed the sample data and corresponding results in the corresponding chapter accordingly. For experimental verification a comparison is done with other search engines for example say, google, yahoo and AltaVista. The comparison is performed within these search engines and the proposed system. The proposed system uses the normal search engines as API and uses the proposed system as enhanced filters. The experimental results are calculated based on the results from the distribution hypothesis for the clustering of the documents and the genetic algorithm for the identification of the similarity between the contents. Later a sample experts' preference is given and consolidated with the ranking system where the search results has enhanced to a certain level. The initial data of precision and recall is collected from reference [4].

Table 3: Shows the comparison of precision and recall value for the three search engines with our system

	Google	Yahoo	Bing	System
Precision	1.593	1.545	1.490	1.612
recall	0.882	0.058	0.059	0.132

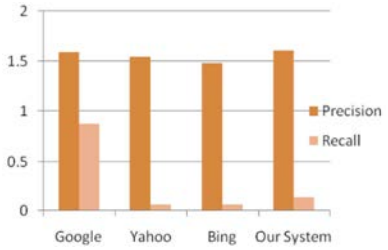

Fig. 5: The fitness function is used for calculating the precision for the system proposed

$$F(x) = \begin{cases} 0 & whenever\ f'(x) = 0 \\ f'(x) + g(x) & whenever\ f'(x) + g(x) > 0 \\ n & whenever\ f'(x) > 0\ and\ f'(x) + g(x) < 0 \end{cases}$$

where f(x) and g(x) represents the documents.

From the graph it is identified that, the results of the graph states that, the precision of the proposed system resembles the precision of the google search engines but the proposed system needs to be developed as it is based on the user's preference the recall rate of the system increases over a period of time.

These results were based on three main aspects:

- Component accuracy that includes title recognition, data and location detection of the documents and list extraction
- Time performance, which includes, a maximum of 109 ms where 67 ms for html parsers, 36 ms for content processing and 6 ms for the rank allocation.
- End to end evaluation which recognizes all the relevant documents of the data and the content used.

CONCLUSION

This proposes a novel and an interesting technique for refining the search results. In this paper, we proposed systems that refines the search results that are extracted from the search engines and then they are they are passed through a series of process to refine the results that are extracted from the search engines. Here for the refinement purpose, we have used various techniques such as the distribution hypothesis for the purpose of clustering of the documents and genetic algorithm to identify the frequency match with the keyword that is provided by the user and the words in the document. They also identify the fitness match with the words in the documents to attain a conclusion whether the document is worth displaying for the user to view his relevant documents.

Later the results are ranked based on the experts' ranking system. This system cannot be attained immediately, but they could be developed over a period of time as the experts' rank the documents for the provided search query.

REFERENCES

1. Gaeta, M., F. Orciuoli, S. Paolozzi and P. Ritrovato, 2009. Effective ontology management in virtual learning environments, Int. J. Internet Enterprise Manage., 6(2): 96-123.
2. Maedche and S. Staab, 2000. The Text-To-Onto ontology learning environment, in Proc. 8th Int.Conf. Conceptual Struct., Darmstadt, Germany, pp: 14-18.
3. Conner, W., A. Iyengar, T. Mikalsen, I. Rouvellou and K. Nahrstedt, 2009. A trust management framework for service-oriented environments, in WWW '09.
4. Kovac, D. and D. Trcek, 2009. Qualitative trust modeling in soa," Journal of Systems Architecture, 55(4): 255-263.
5. Malik, Z. and A. Bouguettaya, 2009. Reputation bootstrapping for trust establishment among web services, IEEE Internet Computing, 13(1): 40-47.
6. Jurczyk, P. and E. Agichtein, 2007. Discovering authorities in question answer communities by using link analysis, in CIKM '07. New York, NY, USA: ACM, pp: 919-922.
7. Yang, J., L. Adamic and M's. Ackerman, 2008. Competing to share expertise: the tasken knowledge sharing community, submitted in Weblogs and Social Media an International Conference.
8. Ratnakumar, A.J., 2005. An Implementation of Web Personalization Using Web Mining Techniques, Journal of Theoretical and Applied Information Technology.
9. Artz, D. and Y. Gil, 2007. A Survey of Trust in Computer Science and the Semantic Web, J. Web Semantics: Science, Services and Agents on the World Wide Web, 5(2): 58-71.
10. Palmonari, M., M. Comerio and F.D. Paoli, 2009. Effective and flexible nfp-based ranking of web services, In IC-SOC/ServiceWave, pp: 546-560.
11. Skoutas, D., M. Alrifai and W. Nejdl, 2010. Re-ranking web service search results under diverse user preferences, In VLDB, Workshop on Personalized Access, Profile Management and Context Awareness in Databases, pp: 898-909.
12. Wu, Q., A. Iyengar, R. Subramanian, I. Rouvellou, I. SilvaLepe and T.A. Mikalsen, 2009. Combining quality of service and social information for ranking services. In IC-SOC/ServiceWave, pp: 561-575.
13. Agarwal, S. and S. Lamparter, 2005. User preference based automated selection of web service compositions, In K.V.A.S. M.Z.C. Bussler, editor, ICSOC Workshop on Dynamic Web Processes, Amsterdam, Netherlands, Dezember. IBM, pp: 1-12.
14. Aidan Hogan Andreas Harth, Juergen Umrich, Sheila Kinsella, Axel Polleres and Stefan Decker, 2012. Searching and Browsing Linked Data with SWSE: the Semantic Web Search Engine, Web Semantics: Science, Services and Agents on the World Wide Web, 9(4).
15. Allemang, Dean and James Hendler, 2011. Semantic Web for the Working Ontologist, 2nd Edition: Effective Modeling in RDFS and OWL, Morgan Kaufmann; 1 edition. March 1, ISBN: 978-0-12-385965-5.
16. Elkateb Sabri, William Black, Piek Vossen, David Farwell, Adam Pease and Christiane Fellbaum, 2006. Arabic WordNet and the challenges of Arabic. The Challenge of Arabic for NLP/MT, London: International conference at the British Computer Society, pp: 23.