Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp429-452)

Aggregating Subjective & Objective Measures of Web Search Quality using Modified Shimura Technique

Rashid Ali
Department of Computer Engineering
A. M. U. Aligarh
UP - 202002
India
rashidaliamu@rediffmail.com

M. M. Sufyan Beg
Department of Computer Science
University of California, Berkeley
CA 94720
USA
mmsbeg@hotmail.com

Abstract – Web Searching is perhaps the second most popular activity on Internet. Millions of users search the web daily for their purpose. But as there are a number of search engines available, there must be some procedure to evaluate them. In this paper, we try to present an effort in this regard. We are making an attempt to get a comprehensive evaluation system for web search results. We are taking into the consideration the "satisfaction " a user gets when presented with search results The feedback of the user is inferred from watching the actions of the user on the search results presented before him in response to his query, rather than by form filling method. This gives an implicit ranking of documents by the user. Then, classical vector space model for information retrieval is used for computing the similarity of documents selected by the user to that of query. The documents from the search results presented in response to a query are represented by term vectors in vector space. The query is also represented by a term vector. The similarity of a document with the query is thus obtained by computing the dot (scalar) product. Sorting the documents in decreasing order of dot products of their term vectors with that of query, gives a new ranking of the documents on the basis of vector space model. Then, Boolean similarity measure is used to compute the similarity of the documents selected by the user to that of the query and thus another ranking of the documents based on the similarity measure is obtained. We propose a simplified version of a well known Boolean similarity measure and use it for our purpose. All the three rankings obtained in the process is then aggregated using Modified Shimura technique of Rank aggregation. The aggregated ranking is then compared with the original ranking given by the search engine. The correlation coefficient thus obtained is averaged for a set of queries. We show our experimental results pertaining to seven public search engines and fifteen queries.
Keywords – web search evaluation, user feedback, vector space model, boolean similarity measure, rank aggregation

1. INTRODUCTION

Internet has been very popular since its inception. Everyday, a number of Internet users search the web for some data and information using some query. A number of public search engines are available for this purpose. In an Internet search, the user writes some query to describe the nature of documents, and the search engine responds by returning a number of web pages that match the description. The results are ranked by the search engines and returned in the order of ranks. Since different search engines use different search algorithms and indexing techniques, they return different web pages in response to same query. Also, same web pages is ranked differently by

different search engines and returned at different positions in the list of search results. Then the question arises, which search engine one should use for web searching? A better search engine is one, which gives relevance results in response to a query and also returns them in proper order of relevance. For this, search results need to be evaluated

The evaluation procedure may be subjective or objective. In the present work, we propose a comprehensive web search evaluation system, which combines both the subjective as well as objective techniques. For subjective evaluation, the users' vote is to be counted. For objective evaluation, different similarity measures based approaches such as Boolean similarity measures based; vector space model based approaches are used.

How are the users rating the results of a search engine should be taken into account to evaluate that search engine subjectively. Thus, it becomes imperative to obtain the feedback from the users. This feedback may either be explicit or implicit. The explicit feedback is the one in which the user is asked to fill up a feedback form after he has finished searching. This form is easy to analyze as the user may be asked directly to rank the documents as per the relevance according to his evaluation. But the problem is to obtain a correct feedback. The problem with the form-based approach is that it is too demanding from the user. In this approach, there is a lot of work for a casual user who might either fill it carelessly or not fill it at all. We, therefore, felt a need to devise a method to obtain the implicit feedback from the users. We watch the actions of the user on the search results presented before him in response to his query, and infer the feedback of the user there from.

We augment the subjective evaluation technique based on implicit user feedback as mentioned in the preceding paragraph with objective evaluation based on Vector Space Model and Boolean similarity measures. For that, we need to do the text processing first, which includes removal of stop words and stemming operations.

1.1 Related Work

In the past, some efforts have been made to evaluate search results from different search engines. In most of the cases, a uniform sample of the web pages is collected by carrying out random walks on the web. The size of indices, which indirectly estimates the performance of a search engine, is then measured using this uniform sample. A search engine having larger index size has higher probability to give good search results. In [1], [2] and [3], some attempts involving this is easily visible. In [4] also, the relative size and overlap of search engines is found but by using random queries, which are generated from a lexicon of about 400,000 words, built from a broad crawl of roughly 300,000 documents in the Yahoo hierarchy. In [5] and [6], the search engines are compared using a standard query log like that of NEC research institute. In [7], a frozen 18.5 million page snapshots of part of the web is created for proper evaluation of web search systems. In [8], for two different sets of ad-hoc queries, the results from AltaVista, Google and InfoSeek are obtained. These results are automatically evaluated for relevance on the basis of vector space model. These results are found to agree with the manual evaluation of relevance based on precision. Precision scores are given as 0, 1 or 2. But then this precision evaluation is similar to the form-filling exercise, already discussed for its demerits in section 1. Precision evaluation of search engines is reported in [9]. But then, "precision" being just the ratio of retrieved documents that are judged relevant, it doesn't say anything about the ranking of the relevant documents in the search results. Upon just the precision evaluation, other important aspects of web search evaluation such as recall, coverage, response time and web coverage etc. are also missed out.

With the present effort of combining the subjective and objective techniques of web search evaluation, we aspire to get a complete and comprehensive picture of web search evaluation.

1.2 Useful Definitions

Here we have some definitions that are useful while evaluating search results.

Definition 1. *Given a universe U and $S \subseteq U$, an ordered list (or simply, a list) l with respect to U is given as $l = [e_1,e_2,.....,e_{|s|}]$, with each $e_i \in S$, and $e_1 \succ e_2 \succ .... \succ e_{|s|}$, where "$\succ$" is some ordering relation on S. Also, for $j \in U \wedge j \in l$, let l(j) denote the position or rank of j, with a higher rank having a lower numbered position in the list. We may assign a unique identifier to each element in U and thus, without loss of generality we may get $U = \{1,2 ,....,|U|\}$.*

Definition 2. Full List: *If a list contains all the elements in U, then it is said to be a full list.*

Example 1. A full list l_f given as [e,a,d,c,b] has the ordering relation $e \succ a \succ d \succ c \succ b$. The Universe U may be taken as {1,2,3,4,5} with say a ≡ 1, b ≡ 2, c ≡ 3, d ≡ 4, e ≡ 5. With such an assumption, we have $l_f = [5,1,4,3,2]$. Here $l_f(5) \equiv l(e) = 1$, $l_f(1) \equiv l(a) = 2$, $l_f(4) \equiv l_f(d) = 3$, $l_f(3) \equiv l_f(c) = 4$, $l_f(2) \equiv l_f(b) = 5$.

Definition 3. Kendall Tau distance: The ***Kendall Tau distance*** between two full lists l_1 *and* l_2, each of cardinality |l|, is given as follows.

$$K(l_1, l_2) = \frac{\left|\{(i,j) \mid \forall\, l_1(i) < l_1(j), l_2(i) > l_2(j)\right|}{(1/2)|l|(|l|-1)}$$

Definition 4. Spearman footrule distance: The *Spearman footrule distance* (SFD) between two full lists l_1 *and* l_2, eachof cardinality |l|, is given as follows.

$$F(l_1, l_2) = \frac{\sum_{\forall\, i} |l_1(i) - l_2(i)|}{\lfloor (1/2)\,|l|^2 \rfloor}$$

Definition 5. Given a set of k full lists as $L=\{ l_1, l_2,..., l_k \}$, the normalized aggregated Kendall distance of a full list l to the set of full lists L is given as $K(l,L) = \frac{\sum_{i=1}^{k} K(l,l_i)}{k}$, *while the normalized aggregated footrule distance of l to L is given as* $F(l,L) = \frac{\sum_{i=1}^{k} F(l,l_i)}{k}$

Definition 6. Rank Aggregation: Given a set of lists $L=\{ l_1, l_2,..., l_k \}$, Rank Aggregation is the task of coming up with a list l such that either K(l,L) or F(l,L) is minimized.

***Definition 7. Partial List*:** *A list l_p containing elements, which are a strict subset of universe U, is called a partial list. We have a strict inequality $|l_p| < |U|$.*

Definition 8. Spearman Rank Order Correlation coefficient [10]**:** *Let the full lists $[u_1,u_2,.....u_n]$ and $[v_1,v_2,.....v_n]$ be the two rankings for some query Q. Spearman rank-order correlation coefficient (r_s) between these two rankings is defined as follows-*

$$r_s = 1 - \frac{6\sum_{i=1}^{n}[l_f(u_i) - l_f(v_i)]^2}{n(n^2-1)} \qquad (1)$$

The Spearman rank-order correlation coefficient (r_s) is a measure of closeness of two rankings. The coefficient r_s ranges between –1 and 1. When the two rankings are identical $r_s = 1$, and when one of the rankings is the inverse of the other then the $r_s = -1$.

***Definition 9. Modified Spearman Rank Order Correlation coefficient*:** *Without loss of generality, assume that full list be given as [1,2,……,n].Let the partial list be given as [v_1,v_2,……,v_m]. Modified Spearman rank-order correlation coefficient (r_s') between these two rankings is defined as follows-*

$$r_s' = 1 - \frac{\sum_{i=1}^{m}(i - v_i)^2}{m\left([\max_{j=1}^{m}\{v_j\}]^2 - 1\right)} \qquad (2)$$

Example 2. For |U|=5, let the full list be $l_f = \{1,2,3,4,5\}$ and the partial list l_p with |lp| = m =3 be $l_p = \{40,35,100\}$.

$$r_s' = 1 - \frac{(1-40)^2 + (2-35)^2 + (3-100)^2}{3\times\left([\max\{40,35,100\}]^2 - 1\right)} = 0.401$$

2 WEB SEARCH EVALUATION USING USER FEEDBACK VECTOR MODEL

2.1 User Feedback Vector

The underlying principle of our approach [11] of subjective evaluation of search engines is to measure the "satisfaction" a user gets when presented with the search results. For this, we need to monitor the response of the user to the search results presented before him. We characterize the feedback of the user by a vector (*V, T, P, S, B, E, C*), which consists of the following.

(a) The sequence *V* in which the user visits the documents, $V = (v_1, v_2, \ldots v_N)$. If document i is the k^{th} document visited by the user, then we set v_i = k. If a document i is not visited by the user at all before the next query is submitted, the corresponding value of v_i is set to -1.

(b) The time t_i that a user spends examining the document i. We denote the vector $(t_1, t_2, \ldots, t_N)$ by .*T*. For a document that is not visited, the corresponding entry in the array *T* is 0.

(c) Whether or not the user prints the document i. This is denoted by the Boolean p_i. We denote the vector $(p_1, p_2, \ldots, p_N)$ by *P*.

(d) Whether or not the user saves the document i. This is denoted by the Boolean s_i. We denote the vector •$(s_1, s_2, \ldots, s_N)$ by .*S*

(e) Whether or not the user book-marked the document i. This is denoted by the Boolean b_i We denote the vector $(b_1, b_2, \ldots, b_N)$ by *B*.

(f) Whether or not the user e-mailed the document v to someone. This is denoted by the Boolean e_i. We denote the vector $(e_1, e_2 \ldots, e_N)$ by *E*.

(g) The number of words that the user copied and pasted elsewhere. We denote the vector $(c_1, c_2, \ldots, c_N)$by *C*.

The motivation behind collecting this feedback is the belief that a well-educated user is likely to select the more appropriate documents early in the resource discovery process. Similarly, the time that a user spends examining a document, and whether or not he prints, saves, bookmarks, e-mails it to someone else or copies & pastes a portion of the document, indicate the level of importance that document holds for the specified query.

2.2 Search Quality Measure (SQM) using User Feedback Vector

When feedback recovery is complete, we propose to compute the following weighted sum σ_j for each document j selected by the user.

$$\sigma_j = \left(w_V \frac{1}{2^{(v_j - 1)}} + w_T \frac{t_j}{t_j^{\max}} + w_P p_j + w_S s_j + w_B b_j + w_E e_j + w_C \frac{c_j}{c_j^{total}} \right) \tag{3}$$

Where $t_j^{\max}$ represents the maximum time a user is expected to spend in examining the document j, and c_j^{total} is the total number of words in the document j. Here, w_V, w_T, w_P, w_S, w_B, w_E and w_C, all lying between 0 and 1, give the respective weightages we want to give to each of the seven components of the feedback vector

The sum σ_j represents the importance of document j. The intuition behind this formulation is as follows. The importance of the document should decrease monotonically with the postponement being afforded by the user in picking it up. More the time spent by the user in glancing through the document, more important that must be for him. If the user is printing the document, or saving it, or book-marking it, or e-mailing it to someone else, or copying and pasting a portion of the document, it must be having some importance in the eyes of the user. A combination of the above seven factors by simply taking their weighted sum gives the overall importance the document holds in the eyes of the user.

As regards the maximum time a user is expected to spend in examining the document j, we clarify that this is taken to be directly proportional to the size of the document. We assume that an average user reads at a speed of about 10 bytes per second. This includes the pages containing text as well as images. So a document of size 1 kB is expected to take a minute and 40 seconds to go through. The above mentioned default reading speed of 10 bytes per second may be set differently by the user, if he wishes so.

It may be noted that depending on his preferences and practice, the user would set the importance of the different components of the feedback vector. For instance, if a user does not have a printer at his disposal, then there is no sense in setting up the importance weight (w_P) corresponding to the printing feedback component (P). Similarly, if a user has a dial-up network connection, and so he is in a habit of saving the relevant documents rather than spending time on it while online, it would be better to give a higher value to w_S, and a lower value to w_T. In such a case, lower values may also be given to w_P, w_E and w_C, as he would not usually be printing or e-mailing or copying and pasting a document at a stretch while online. So, after explaining the modalities to him, the user is to be requested to modify the otherwise default values of 1 for all these weights. It may, however, be noted that the component of the feedback vector corresponding to the sequence of clicking, always remains to be the prime one and so w_V must always be 1.

Now, sorting the documents on the descending values of their weighted sum will yield a sequence $\Re_A$, which is a ranking of documents based on user feedback.

3. WEB SEARCH EVALUATION USING VECTOR SPACE MODEL &BOOLEAN SIMILARITY MEASURES

Before we proceed, we must have a look at the text pre-processing operations, which are a pre-requisite for the application of any objective evaluation procedure.

3.1 Text Pre-Processing

First of all, we need to remove the stop-words, the words that have very little semantic meaning and are frequent homogeneously across the whole collection of documents. They are generally prepositions or articles like the, an, for etc. Over a period of time people have come up with a list of stop-words pertaining to a general domain. However, it may be argued that a stop-word is very much context dependant. A word like web may be treated as a stop-word in a collection of web-related articles, but not so in a set of literary documents.

The text pre-processing also includes an important step of word stemming, wherein all the words with same root are reduced to a single form. This is achieved by stripping each word of suffix, prefix or infix. This is to say that all words are reduced to their canonical form. For instance, the words like drive, driver and driving, all would be reduced to the stem word drive. This way the burden of being very specific while forming the query, is taken off from the shoulders of the user. A well- known algorithm for carrying out word stemming is Porter Stemmer algorithm [12].

It may be noted that the text pre-processing techniques are very much dependant on the language of the document. For instance, just the removal of suffixes may usually suffice as the stemming technique in the case of English language, but not necessarily so with other languages.

3.2 Vector Space Model

An n-dimensional vector space is taken with one dimension for each possible word or term. Thereore, n would be the number of words in a natural language. Each document or query is represented as a *term vector* in this vector space. Each component of *term vector* is either 0 or 1, depending on whether the term corresponding to that axis is absent or present in the text. Alternatively, each component of the term vector is taken as a function that increases with the frequency of the term (corresponding to that axis) with in the document and decreases with the number of documents in the collection that contain this term. This is to take into account the TF-IDF factor.

The frequency of a word in a given document is taken as a good measure of importance of that word in the given document [13]. This, of course, holds true only after the text pre-processing has been carried out.

An improved version of the term frequency is the *term frequency-inverse document frequency* (TF-IDF). In this, uniqueness of a keyword in a document, and hence the relevance, is measured by the fact that the word should be frequent in the current document but not so frequent in the rest of the documents of the given collection. The TF-IDF value for a given keyword w is given as

$$f_{TF-IDF} = \frac{f_w}{f_{w\max}} \log \frac{\rho}{\rho_w} \quad (4)$$

where, f_w is the frequency of the keyword w in the document, is $f_{w\max}$ the maximum frequency of any word in the document , ρ_w is the number of documents in which this keyword occurs and ρ is total number of documents in the collection.

As web is very dynamic with respect to nature of its content , the vector space model can not be used directly in the performance evaluation of search engines[14].The value $\log(\rho/\rho_w)$ is not available because we don't know the total number of documents ρ and documents containing the given keyword ρ_w .So, we use a simplified version of vector space model. Here, we assume $\log(\rho/\rho_w)$ to be constant with the argument that all keywords in our queries are technical terms which appear approximately the same number of times. This simplified version may favor long documents and give documents with many appearances of same keywords a higher score. The term vectors of documents are normalized to one to compensate for different document lengths and a many occurrences of a keyword in a document indicates relevance of the document to the query.

Once the term vectors are obtained the similarity between a document and a query is obtained by computing the dot product of their term vectors. Larger the dot product, the greater would be the similarity. The document with larger dot product is more relevant to query.

Now, sorting the documents with the decreasing values of their respective dot products with that of the query, will yield a sequence $\Re_B$, which is a ranking of documents based on vector space model

3.3 Boolean similarity measures

There are a number of *Boolean similarity measures*[14] that can be used to compute the similarities of one document to another and documents to queries. Some of such well-known similarity measures are Dice's Coefficient, Jaccard's Coefficient, Cosine coefficient and overlap Coefficient. In order to use these measures, documents and queries are to be represented as sets of keywords

Radecki proposed two similarity measures, S and S*, based on Jaccard's coefficient .We assume that each query is transformed to a Boolean expression and denote Ð (Q) and Ð (C) as sets of documents in the response to query Q and in the cluster of documents represented by C . The similarity value S between Q and C is defined as the ratio of common documents to total number of documents in Ð (Q) and Ð (C). .

$$S(Q,C) = \frac{|Ð(Q) \cap Ð(C)|}{|Ð(Q) \cup Ð(C)|} \quad (5)$$

but since all the documents in response to query belong to the cluster represented by C (i.e Ð (Q)$\subseteq$ Ð (C)) we can have

$$S(Q,C) = \frac{|Ð(Q)|}{|Ð(C)|} \quad (6)$$

This measure requires the actual results to the query and is mainly useful as index of comparison

In similarity measure S*, Boolean expression Q is transformed into its reduced disjunctive normal form (RDNF), denoted as $\tilde{Q}$, which is disjunction of a list of reduced atomic descriptors. If set T is the union of all the descriptors that appear in the to be compared Boolean expression pair, then a reduced atomic descriptor is defined as a conjunction of all the elements in T in either their true or negated form. Let T_Q and T_C be the set of descriptors that appear in Q and C respectively. Suppose $T_Q \cup T_C = \{t_1, t_2, t_3, \ldots . t_n\}$ where n is the set size of $T_Q \cup T_C$ then the RDNF of Q and C are:

$$\left(\tilde{Q}\right)_{T_Q \cup T_c} = \left(\tilde{q}_{1,1} \wedge \tilde{q}_{1,2} \wedge \ldots \wedge \tilde{q}_{1,n}\right) \vee \left(\tilde{q}_{2,1} \wedge \tilde{q}_{2,2} \wedge \ldots \wedge \tilde{q}_{2,n}\right) \vee \ldots . \vee \left(\tilde{q}_{l,1} \wedge \tilde{q}_{l,2} \wedge \ldots \wedge \tilde{q}_{l,n}\right) \tag{7}$$

and

$$\left(\tilde{C}\right)_{T_Q \cup T_c} = \left(\tilde{c}_{1,1} \wedge \tilde{c}_{1,2} \wedge \ldots \wedge \tilde{c}_{1,n}\right) \vee \left(\tilde{c}_{2,1} \wedge \tilde{c}_{2,2} \wedge \ldots \wedge \tilde{c}_{2,n}\right) \vee \ldots . \vee \left(\tilde{c}_{m,1} \wedge \tilde{c}_{m,2} \wedge \ldots \wedge \tilde{c}_{m,n}\right) \tag{8}$$

where l and m are number of reduced atomic descriptors in $\left(\tilde{Q}\right)_{T_Q \cup T_c}$ and $\left(\tilde{C}\right)_{T_Q \cup T_c}$ respectively, and

$$\tilde{q}_{i,j} = \begin{cases} t_j \ldots\ldots\ldots\ldots \textit{true} \\ \neg t_j \ldots . \textit{negated} \end{cases} \ldots . 1 \le i \le l, 1 \le j \le n \tag{9}$$

$$\tilde{c}_{i,j} = \begin{cases} t_j \ldots\ldots\ldots\ldots \textit{true} \\ \neg t_j \ldots . \textit{negated} \end{cases} \ldots . 1 \le i \le m, 1 \le j \le n \tag{10}$$

Where, $\neg$ is not operator

The similarity value S* between the Boolean expressions (Q and C) is defined as the ratio of the number of common reduced atomic descriptors in $\tilde{Q}$ and $\tilde{C}$ to the total number of reduced atomic descriptors in them,

$$S^*(Q,C) = \frac{\left|\left(\tilde{Q}\right)_{T_Q \cup T_c} \cap \left(\tilde{C}\right)_{T_Q \cup T_c}\right|}{\left|\left(\tilde{Q}\right)_{T_Q \cup T_c} \cup \left(\tilde{C}\right)_{T_Q \cup T_c}\right|} \tag{11}$$

A new similarity measure $S^{\oplus}$, based on Radecki similarity measure S*, was proposed by Li Danzig For this, a Boolean expression Q is transformed to its compact disjunctive normal

form(CDNF) denoted as $\hat{Q}$,which is a disjunction of compact atomic descriptors. Each compact atomic descriptor itself is in turn the a conjuction of subsets of descriptors present in its own Boolean expression The CDNFs of Q and C are

$$\hat{Q} = \left(\hat{q}_{1,1} \wedge \hat{q}_{1,2} \wedge \ldots \wedge \hat{q}_{1,x_1}\right) \vee \left(\hat{q}_{2,1} \wedge \hat{q}_{2,2} \wedge \ldots \wedge \hat{q}_{2,x_2}\right) \vee \ldots \vee \left(\hat{q}_{l,1} \wedge \hat{q}_{l,2} \wedge \ldots \wedge \hat{q}_{l,x_l}\right), \quad (12)$$

and

$$\hat{C} = \left(\hat{c}_{1,1} \wedge \hat{c}_{1,2} \wedge \ldots \wedge \hat{c}_{1,y_1}\right) \vee, \left(\hat{c}_{2,1} \wedge \hat{c}_{2,2} \wedge \ldots \wedge \hat{c}_{2,y_2}\right) \vee \ldots \vee \left(\hat{c}_{m,1} \wedge \hat{c}_{m,2} \wedge \ldots \wedge \hat{c}_{m,x_m}\right), \quad (13)$$

Where, l and m are the numbers of compact atomic descriptors in $\hat{Q}$ and $\hat{C}$,

x_i is the number of descriptors in the i^{th} $(1 \le i \le l)$compact atomic descriptor of $\hat{Q}$, and

y_j is the number of descriptors in the j^{th} $(1 \le j \le m)$ compact atomic descriptor of $\hat{C}$.

Each $\hat{q}_{i,u}$ and $\hat{c}_{j,v}$ in the CDNFs represents a descriptor in T_Q and T_C respectively.

Specifically, we have $\hat{q}_{i,u} \varepsilon T_Q$, where $1 \le i \le l$ and $1 \le u \le x_i$ and $\hat{c}_{j,v} \varepsilon T_C$, where $1 \le j \le m$ and $1 \le v \le y_j$.

The individual similarity measure is defined as

$$s^{\oplus}\left(\hat{Q}^i, \hat{C}^j\right) = \begin{cases} 0 \ldots\ldots\ldots\ldots if T_Q^i \cap T_C^j = 0 . or . \exists t \in T_Q^i, \neg t \in T_C^j \\ \dfrac{1}{2^{\left|T_C^j - T_Q^i\right|} + 2^{\left|T_Q^i - T_C^j\right|} - 1} \ldots\ldots\ldots\ldots\ldots\ldots otherwise \end{cases} \quad (14)$$

Where, $\hat{Q}^i$ indicates the i^{th} atomic descriptors of CDNF $\hat{Q}$,

$\hat{C}^j$ indicates the j^{th} compact atomic descriptor of CDNF $\hat{C}$.

T_Q^i and T_C^j are the set of descriptors in $\hat{Q}^i$ and $\hat{C}^j$ respectively.

The similarity of two expressions, $S^{\oplus}$ defined as the average value of the individual similarity measures ($s^{\oplus}$) between each atomic descriptor is given by

$$S^{\oplus}(Q,C) = \frac{\sum_{i=1}^{\left|\hat{Q}\right|} \sum_{j=1}^{\left|\hat{C}\right|} s^{\oplus}\left(\hat{Q}^i, \hat{C}^j\right)}{\left|\hat{Q}\right| \times \left|\hat{C}\right|} \quad (15)$$

Example 3

Suppose Q be the query represented by Boolean Expression Q = $(t_1 \vee t_2) \wedge t_3$ and the three to be compared documents or servers descriptions say C_1, C_2 and C_3 be represented by Boolean expressions $C_1 = t_1 \wedge t_2 \wedge t_4 \wedge t_5$, $C_2 = (t_1 \vee t_3) \wedge t_4$ and $C_3 = t_2 \wedge t_3 \wedge t_5$ respectively.

If set T_Z be the union of all the descriptors in Boolean expression Z (Z=Q, C_1, C_2 or C_3) We have $T_Q = \{t_1, t_2, t_3\}$, $T_{C1} = \{t_1, t_2, t_4, t_5\}$, $T_{C2} = \{t_1, t_3, t_4\}$, $T_{C3} = \{t_2, t_3, t_5\}$.

The CDNF of Q, C_1, C_2 and C_3 are

$Q = (t_1 \wedge t_3) \vee (t_2 \wedge t_3)$, $C_1 = t_1 \wedge t_2 \wedge t_4 \wedge t_5$, $C_2 = (t_1 \wedge t_4) \vee (t_3 \wedge t_4)$ and $C_3 = t_2 \wedge t_3 \wedge t_5$ respectively.

From above, it is clear that

(i) CDNF of Q contains two compact atomic descriptors $\hat{Q}^1 = t_1 \wedge t_3$ and $\hat{Q}^2 = t_2 \wedge t_3$ and set of descriptors in them being $T_Q^1 = \{t_1, t_3\}$ and $T_Q^2 = \{t_2, t_3\}$ respectively. Similarly,

(ii) CDNF of C_1 contains only one compact atomic descriptors $\hat{C}_1^1 = t_1 \wedge t_2 \wedge t_4 \wedge t_5$ and set of descriptor in that being $T_{C_1}^1 = \{t_1, t_2, t_4, t_5\}$.

(iii) CDNF of C_2 contains two compact atomic descriptors $\hat{C}_2^1 = t_1 \wedge t_4$ and $\hat{C}_2^2 = t_3 \wedge t_4$ and set of descriptor in them being $T_{C_2}^1 = \{t_1, t_4\}$ and $T_{C_2}^2 = \{t_3, t_4\}$ respectively.

(iv) CDNF of C_3 contains only one compact atomic descriptors $\hat{C}_3^1 = t_2 \wedge t_3 \wedge t_5$ and set of descriptor in that being $T_{C_3}^1 = \{t_2, t_3, t_5\}$

Thus,

$$s^{\oplus}\left(\hat{Q}^1, \hat{C}_1^1\right) = \frac{1}{2^{|T_{C1}^1 - T_Q^1|} + 2^{|T_Q^1 - T_{C1}^1|} - 1} = \frac{1}{2^3 + 2^1 - 1} = 0.1111$$

$$s^{\oplus}\left(\hat{Q}^2, \hat{C}_1^1\right) = \frac{1}{2^{|T_{C1}^1 - T_Q^2|} + 2^{|T_Q^2 - T_{C1}^1|} - 1} = \frac{1}{2^3 + 2^1 - 1} = 0.1111$$

Hence,

$$S^{\oplus}(Q, C_1) = \frac{s^{\oplus}\left(\hat{Q}^1, \hat{C}_1^1\right) + s^{\oplus}\left(\hat{Q}^2, \hat{C}_1^1\right)}{2} = 0.1111$$

and

$$s^{\oplus}\left(\hat{Q}^1, \hat{C}_2^1\right) = \frac{1}{2^{|T_{C2}^1 - T_Q^1|} + 2^{|T_Q^1 - T_{C2}^1|} - 1} = \frac{1}{2^1 + 2^1 - 1} = 0.3333$$

$$s^{\oplus}\left(\hat{Q}^1, \hat{C}_2^2\right) = \frac{1}{2^{|T_{C2}^2 - T_Q^1|} + 2^{|T_Q^1 - T_{C2}^2|} - 1} = \frac{1}{2^1 + 2^1 - 1} = 0.3333$$

$$s^{\oplus}\left(\hat{Q}^2, \hat{C_2}^1\right) = 0 \text{ (because } T_Q^2 \cap T_{C_2}^1 = \phi)$$

$$s^{\oplus}\left(\hat{Q}^2, \hat{C_2}^2\right) = \frac{1}{2^{|T_{C_2}^2 - T_Q^2|} + 2^{|T_Q^2 - T_{C_2}^2|} - 1} = \frac{1}{2^1 + 2^1 - 1} = 0.3333$$

Hence,

$$S^{\oplus}(Q, C_2) = \frac{s^{\oplus}\left(\hat{Q}^1, \hat{C_2}^1\right) + s^{\oplus}\left(\hat{Q}^1, \hat{C_2}^2\right) + s^{\oplus}\left(\hat{Q}^2, \hat{C_2}^1\right) + s^{\oplus}\left(\hat{Q}^2, \hat{C_2}^2\right)}{4} = 0.2500$$

and

$$s^{\oplus}\left(\hat{Q}^1, \hat{C_3}^1\right) = \frac{1}{2^{|T_{C_3}^1 - T_Q^1|} + 2^{|T_Q^1 - T_{C_3}^1|} - 1} = \frac{1}{2^2 + 2^1 - 1} = 0.2000$$

$$s^{\oplus}\left(\hat{Q}^2, \hat{C_3}^1\right) = \frac{1}{2^{|T_{C_3}^1 - T_Q^2|} + 2^{|T_Q^2 - T_{C_3}^1|} - 1} = \frac{1}{2^1 + 2^0 - 1} = 0.5000$$

Hence,

$$S^{\oplus}(Q, C_3) = \frac{s^{\oplus}\left(\hat{Q}^1, \hat{C_3}^1\right) + s^{\oplus}\left(\hat{Q}^2, \hat{C_3}^1\right)}{2} = 0.350$$

Now, sorting the documents in decreasing order of their Li Danzig similarity measure value, we get a ranking $C_3 \succ C_2 \succ C_1$ where '$\succ$' indicates 'is more relevant to query than'.

3.3.1 Simplified Boolean Similarity measure

It is a proved [15] fact that the Li Danzig similarity measure $S^{\oplus}$, is equivalent to Radecki similarity measure S^*, which is based on Jaccard's Coefficient and at the same time, reduces time and space complexity from exponential to polynomial in the number of Boolean terms. But as we are interested in relevant ranking of the documents rather than their individual similarity measures with the query, we strongly feel that the Li Danzig measure $S^{\oplus}$ can be further simplified. We propose a simplified Boolean similarity measure $S^{\otimes}$ based on Li Danzig measure $S^{\oplus}$.

If $\hat{Q}$ and $\hat{C}$ be the CDNF of the Boolean expressions Q and C as described above(eqns12 and13), the simplified individual similarity measure is defined as

$$s^{\otimes}\left(\hat{\boldsymbol{Q}}^{\boldsymbol{i}},\hat{\boldsymbol{C}}^{\boldsymbol{j}}\right)=\begin{cases}\mathbf{0}.....if\boldsymbol{T}_{\boldsymbol{Q}}^{\boldsymbol{i}}\cap\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{j}}=\mathbf{0}.or.\exists\boldsymbol{t}\in\boldsymbol{T}_{\boldsymbol{Q}}^{\boldsymbol{i}},\neg\boldsymbol{t}\in\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{j}}\\ \dfrac{\mathbf{1}}{\left|\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{j}}-\boldsymbol{T}_{\boldsymbol{Q}}^{\boldsymbol{i}}\right|+\left|\boldsymbol{T}_{\boldsymbol{Q}}^{\boldsymbol{i}}-\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{j}}\right|+\mathbf{1}}.....otherwise\end{cases} \quad (16)$$

Where $\hat{\boldsymbol{Q}}^{\boldsymbol{i}}$ indicates the i[th] atomic descriptors of CDNF $\hat{\boldsymbol{Q}}$, $\hat{\boldsymbol{C}}^{\boldsymbol{j}}$ indicates the j[th] compact atomic descriptor of CDNF $\hat{\boldsymbol{C}}$. $\boldsymbol{T}_{\boldsymbol{Q}}^{\boldsymbol{i}}$ and $\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{j}}$ are the set of descriptors in $\hat{\boldsymbol{Q}}^{\boldsymbol{i}}$ and $\hat{\boldsymbol{C}}^{\boldsymbol{j}}$ respectively.

The similarity of two expressions, $S^{\otimes}$ defined as the average value of the individual similarity measures ($s^{\otimes}$) between each atomic descriptor, is given by

$$S^{\otimes}(Q,C)=\frac{\sum_{i=1}^{\left|\hat{Q}\right|}\sum_{j=1}^{\left|\hat{C}\right|}s^{\otimes}\left(\hat{\boldsymbol{Q}}^{\boldsymbol{i}},\hat{\boldsymbol{C}}^{\boldsymbol{j}}\right)}{\left|\hat{\boldsymbol{Q}}\right|\times\left|\hat{\boldsymbol{C}}\right|} \quad (17)$$

The proposed simplified version reduces the computational effort substantially. Moreover, if we assume that Boolean expressions of the query Q and documents to be compared (C_1, C_2, ….., C_n), just contain only AND terms i.e their CDNF contain only a single compact descriptor and for each pair of to be compared documents C_k and C_l one of the following holds true $\left|T_{C_k}^{j}-T_{Q}^{i}\right|=\left|T_{C_l}^{j}-T_{Q}^{i}\right| or \left|T_{C_k}^{j}-T_{Q}^{i}\right|=\left|T_{Q}^{i}-T_{C_l}^{j}\right| or \left|T_{Q}^{i}-T_{C_k}^{j}\right|=\left|T_{Q}^{i}-T_{C_l}^{j}\right| or \left|T_{Q}^{i}-T_{C_k}^{j}\right|=\left|T_{C_l}^{j}-T_{Q}^{i}\right|$ then, relative rankings of documents found using the simplified version, remains same with that found using Li Danzig measure $S^{\oplus}$ as it is illustrated in following example.

Example 4

Suppose Q be the query represented by Boolean Expression $Q = t_1 \wedge t_2 \wedge t_3$ and the three to be compared documents or servers descriptions say C_1, C_2 and C_3 be represented by Boolean expressions $C_1= t_1 \wedge t_2 \wedge t_3 \wedge t_4 \wedge t_5$, $C_2=t_1 \wedge t_4 \wedge t_5$ and $C_3=t_1 \wedge t_3 \wedge t_4 \wedge t_5$ respectively.

The CDNF of Q, C_1, C_2 and C_3 are

$Q= t_1 \wedge t_2 \wedge t_3$, $C_1= t_1 \wedge t_2 \wedge t_3 \wedge t_4 \wedge t_5$, $C_2= t_1 \wedge t_4 \wedge t_5$ and $C_3= t_1 \wedge t_3 \wedge t_4 \wedge t_5$ respectively.

From above, it is clear that

(i)CDNF of Q contains one compact atomic descriptor t_2 $\hat{\boldsymbol{Q}}^{\boldsymbol{1}} = t_1 \wedge t_2 \wedge t_3$ and set of descriptors in that being $\boldsymbol{T}_{\boldsymbol{Q}}^{\boldsymbol{1}}=\{t_1, t_2, t_3\}$ Similarly,

(ii) CDNF of C_1 contains one compact atomic descriptor $\hat{\boldsymbol{C}}_{\boldsymbol{1}}^{\boldsymbol{1}} = t_1 \wedge t_2 \wedge t_3 \wedge t_4 \wedge t_5$ and set of descriptor in that being $\boldsymbol{T}_{\boldsymbol{C_1}}^{\boldsymbol{1}}=\{t_1, t_2, t_3, t_4, t_5\}$.

(iii) CDNF of C_2 contains one compact atomic descriptor $\hat{\boldsymbol{C}}_{\boldsymbol{2}}^{\boldsymbol{1}} = t_1 \wedge t_4 \wedge t_5$ and set of descriptor in that being $\boldsymbol{T}_{\boldsymbol{C_2}}^{\boldsymbol{1}}=\{t_1, t_4, t_5\}$.

(iv) CDNF of C_3 contains one compact atomic descriptor $\hat{C}_3^1 = t_1 \wedge t_3 \wedge t_4 \wedge t_5$ and set of descriptor in that being $T_{C_3}^1 = \{t_1, t_3, t_4, t_5\}$

Thus,

$$s^{\oplus}\left(\hat{Q}^1, \hat{C}_1^1\right) = \frac{1}{2^{\left|T_{C1}^1 - T_Q^1\right|} + 2^{\left|T_Q^1 - T_{C1}^1\right|} - 1} = \frac{1}{2^2 + 2^0 - 1} = 0.2500$$

Hence,

$$S^{\oplus}(Q, C_1) = s^{\oplus}\left(\hat{Q}^1, \hat{C}_1^1\right) = 0.2500$$

and

$$s^{\oplus}\left(\hat{Q}^1, \hat{C}_2^1\right) = \frac{1}{2^{\left|T_{C2}^1 - T_Q^1\right|} + 2^{\left|T_Q^1 - T_{C2}^1\right|} - 1} = \frac{1}{2^2 + 2^2 - 1} = 0.1429$$

Hence,

$$S^{\oplus}(Q, C_2) = s^{\oplus}\left(\hat{Q}^1, \hat{C}_2^1\right) = 0.1429$$

and

$$s^{\oplus}\left(\hat{Q}^1, \hat{C}_3^1\right) = \frac{1}{2^{\left|T_{C3}^1 - T_Q^1\right|} + 2^{\left|T_Q^1 - T_{C3}^1\right|} - 1} = \frac{1}{2^2 + 2^1 - 1} = 0.2000$$

Hence,

$$S^{\oplus}(Q, C_3) = s^{\oplus}\left(\hat{Q}^1, \hat{C}_3^1\right) = 0.2000$$

Now, sorting the documents in decreasing order of their Li Danzig similarity measure value, we get a ranking $C_1 \succ C_3 \succ C_2$ where '$\succ$' indicates 'is more relevant to query than'.
Now, in the similar way, we compute the simplified Similarity measures $S^{\otimes}$ for the same expressions

Thus,

$$s^{\otimes}\left(\hat{Q}^1, \hat{C}_1^1\right) = \frac{1}{\left|T_{C1}^1 - T_Q^1\right| + \left|T_Q^1 - T_{C1}^1\right| + 1} = \frac{1}{2 + 0 + 1} = 0.3333$$

Hence,

$$S^{\otimes}(Q, C_1)= s^{\otimes}\left(\hat{\mathbf{Q}}^{\mathbf{1}},\hat{\mathbf{C}}_{\mathbf{1}}^{\mathbf{1}}\right)=0.3333$$

and

$$s^{\otimes}\left(\hat{\mathbf{Q}}^{\mathbf{1}},\hat{\mathbf{C}}_{\mathbf{2}}^{\mathbf{1}}\right)= \frac{\mathbf{1}}{\left|\boldsymbol{T}_{\boldsymbol{C2}}^{\mathbf{1}}-\boldsymbol{T}_{\boldsymbol{Q}}^{\mathbf{1}}\right|+\left|\boldsymbol{T}_{\boldsymbol{Q}}^{\mathbf{1}}-\boldsymbol{T}_{\boldsymbol{C2}}^{\mathbf{1}}\right|+\mathbf{1}}= \frac{\mathbf{1}}{\mathbf{2+2+1}}=0.2000$$

Hence,

$$S^{\otimes}(Q, C_2)= s^{\otimes}\left(\hat{\mathbf{Q}}^{\mathbf{1}},\hat{\mathbf{C}}_{\mathbf{2}}^{\mathbf{1}}\right)=0.2000$$

and

$$s^{\otimes}\left(\hat{\boldsymbol{Q}}^{\mathbf{1}},\hat{\boldsymbol{C}}_{\mathbf{3}}^{\mathbf{1}}\right)= \frac{\mathbf{1}}{\left|\mathbf{T}_{\mathbf{C3}}^{\mathbf{1}}-\mathbf{T}_{\mathbf{Q}}^{\mathbf{1}}\right|+\left|\mathbf{T}_{\mathbf{Q}}^{\mathbf{1}}-\mathbf{T}_{\mathbf{C3}}^{\mathbf{1}}\right|+\mathbf{1}}=\frac{\mathbf{1}}{\mathbf{2+1+1}}=0.2500$$

Hence,

$$S^{\otimes}(Q, C_3)= s^{\otimes}\left(\hat{\boldsymbol{Q}}^{\mathbf{1}},\hat{\boldsymbol{C}}_{\mathbf{3}}^{\mathbf{1}}\right)= 0.2500$$

Now, sorting the documents in decreasing order of their Simplified similarity measure value, we get same ranking $C_1 \succ C_3 \succ C_2$ where '$\succ$' indicates 'is more relevant to query than'.

This may be noted that even without any constraints, $S^{\otimes}$ may give same relative ranking as given by $S^{\oplus}$ in some cases but it might fail in some other cases.
For example, if we compute the simplified Similarity measures $S^{\otimes}$ for expressions given in example 3, we have

$$s^{\otimes}\left(\hat{\boldsymbol{Q}}^{\mathbf{1}},\hat{\boldsymbol{C}}_{\mathbf{1}}^{\mathbf{1}}\right)= \frac{\mathbf{1}}{\left|\boldsymbol{T}_{\boldsymbol{C1}}^{\mathbf{1}}-\boldsymbol{T}_{\boldsymbol{Q}}^{\mathbf{1}}\right|+\left|\boldsymbol{T}_{\boldsymbol{Q}}^{\mathbf{1}}-\boldsymbol{T}_{\boldsymbol{C1}}^{\mathbf{1}}\right|+\mathbf{1}}= \frac{\mathbf{1}}{\mathbf{3+1+1}}=0.2000$$

$$s^{\otimes}\left(\hat{\boldsymbol{Q}}^{\mathbf{2}},\hat{\boldsymbol{C}}_{\mathbf{1}}^{\mathbf{1}}\right)= \frac{\mathbf{1}}{\left|\boldsymbol{T}_{\boldsymbol{C1}}^{\mathbf{1}}-\boldsymbol{T}_{\boldsymbol{Q}}^{\mathbf{2}}\right|+\left|\boldsymbol{T}_{\boldsymbol{Q}}^{\mathbf{2}}-\boldsymbol{T}_{\boldsymbol{C1}}^{\mathbf{1}}\right|+\mathbf{1}}= \frac{\mathbf{1}}{\mathbf{3+1+1}}= 0.2000$$

Hence,

$$S^{\otimes}(Q, C_1)= \frac{\mathbf{s}^{\otimes}\left(\hat{\boldsymbol{Q}}^{\mathbf{1}},\hat{\boldsymbol{C}}_{\mathbf{1}}^{\mathbf{1}}\right)+\mathbf{s}^{\otimes}\left(\hat{\boldsymbol{Q}}^{\mathbf{2}},\hat{\boldsymbol{C}}_{\mathbf{1}}^{\mathbf{1}}\right)}{\mathbf{2}}=0.2000$$

and

$$s^{\otimes}\left(\hat{Q}^1,\hat{C}_2^{\,1}\right)=\frac{1}{\left|T_{C2}^1-T_Q^1\right|+\left|T_Q^1-T_{C2}^1\right|+1}=\frac{1}{1+1+1}=0.333$$

$$s^{\otimes}\left(\hat{Q}^1,\hat{C}_2^{\,2}\right)=\frac{1}{\left|T_{C2}^2-T_Q^1\right|+\left|T_Q^1-T_{C2}^2\right|+1}=\frac{1}{1+1+1}=0.333$$

$$s^{\otimes}\left(\hat{Q}^2,\hat{C}_2^{\,1}\right)=0 \text{ (because } T_Q^2 \cap T_{C_2}^1=\phi)$$

$$s^{\otimes}\left(\hat{Q}^2,\hat{C}_2^{\,2}\right)=\frac{1}{\left|T_{C2}^2-T_Q^2\right|+\left|T_Q^2-T_{C2}^2\right|+1}=\frac{1}{1+1+1}=0.333$$

Hence,

$$S^{\otimes}(Q, C_2)=\frac{s^{\oplus}\left(\hat{Q}^1,\hat{C}_2^{\,1}\right)+s^{\oplus}\left(\hat{Q}^1,\hat{C}_2^{\,2}\right)+s^{\oplus}\left(\hat{Q}^2,\hat{C}_2^{\,1}\right)+s^{\oplus}\left(\hat{Q}^2,\hat{C}_2^{\,2}\right)}{4}=0.250$$

and

$$s^{\otimes}\left(\hat{Q}^1,\hat{C}_3^{\,1}\right)=\frac{1}{\left|T_{C3}^1-T_Q^1\right|+\left|T_Q^1-T_{C3}^1\right|+1}=\frac{1}{2+1+1}=0.250$$

$$s^{\otimes}\left(\hat{Q}^2,\hat{C}_3^{\,1}\right)=\frac{1}{\left|T_{C3}^1-T_Q^2\right|+\left|T_Q^2-T_{C3}^1\right|+1}=\frac{1}{1+0+1}=0.500$$

Hence,

$$S^{\otimes}(Q, C_3)=\frac{s^{\oplus}\left(\hat{Q}^1,\hat{C}_3^{\,1}\right)+s^{\oplus}\left(\hat{Q}^2,\hat{C}_3^{\,1}\right)}{2}=0.375$$

Now, sorting the documents in decreasing order of their Simplified similarity measure value, we get same ranking $C_3 \succ C_2 \succ C_1$ where '$\succ$' indicates 'is more relevant to query than'.

We obtain the ranking $\Re_C$ of the documents by sorting them in the decreasing order of their Boolean similarity measures with the query.

4. RANK AGGREGATION USING MODIFIED SHIMURA TECHNIQUE

Rank aggregation is the problem of generating a "consensus" ranking for a given set of rankings. We begin with the Shimura technique of fuzzy ordering [16], as it is well suited for non-transitive rankings, as in our case.

4.1Shimura technique of fuzzy ordering

For variables x_i and x_j defined on universe X, a relativity function $f(x_i|x_j)$ is taken to be the membership of preferring x_i over x_j. This function is given as

$$f(x_i|x_j) = \frac{f_{xj}(x_i)}{\max(f_{xj}(x_i), f_{xi}(x_j))}$$

where, $f_{xj}(x_i)$ is the membership function of x_i with respect to x_j and $f_{xi}(x_j)$ is the membership function of x_j with respect to x_i .For X = [x_1 , x_2 ,….,x_n], $f_{xi}(x_i)$ =1. $C_i = \min_{j=1}^{n} f(x_i|x_j)$ is the membership ranking value for the i^{th} variable. Now if a descending sort on C_i (i=1 to n) is caried out, the sequence of i's thus obtained would constitute the aggregated rank. For the lists l_1, l_2,.. l_N from the N participating evaluation techniques, we can have

$$f_{xj}(x_i) = \frac{\left|k \in [1, N] \right| \wedge (l_k(x_i) < l_k(x_j))\right|}{N}$$

In our case N=3.

4.2 Modified Shimura technique

It is observed that classical Shimura technique gives worse performance in comparison to other rank aggregation techniques. We feel that the poor performance coming from the Shimura technique, is primarily due to the employment of "min" function in finding $C_i = \min_{j=1}^{n} f(x_i|x_j)$. The "min" function results in many ties, when a descending order sort is applied on C_i. There is no method suggested by Shimura to resolve these ties. So when resolved arbitrarily, these ties result in deterioration of the aggregated result. We, therefore replace this "min" function by an OWA operator [17]. The OWA operators, in fact, provide a parameterized family of aggregation operators, which include many of the wel-known operators such as the maximum, the minimum, the k-order statistics, the median and the arithmetic mean.

we will be using the relative fuzzy linguistic quantifier "at least half" with the pair (a = 0.0, b = 0.5) for the purpose of finding the vector C_i as folows.

$$C_i = \sum_j w_j . z_j ,$$

where z_j is the j^{th} largest element in the i^{th} row of the matrix $f(x_i|x_j)$. Now, as with the Shimura technique, if a descending sort on C_i (i=1 to m) is caried out, the sequence of i's thus obtained would constitute the aggregated rank.

We will be using the modified Shimura technique for aggregating the three rankings $\Re_A$, $\Re_B$ and $\Re_C$ obtained from the three different evaluation procedures as described in preceding sections. Let us say the aggregated ranking as $\Re_{Comp}$

Let the full list $\Re_{SE}$ be the sequence in which the documents were initially short-listed. Without loss of generality, it could be assumed that $\Re_{SE}$ = (1,2,3,...,N_R), where N_R is the total number of documents listed in the result. We compare the sequences and, and find Modified Spearman Rank Order Correlation Coefficient (r_s'). We repeat this procedure for a representative set of queries and take the average of r_s'. The resulting average value of r_s' is the required measure of the search quality (SQM). The overall procedure is illustrated in Figure 1.

It may be noted that it is very common practice that a user views only those documents whose snippet displayed before him by the search engine he finds to be worth viewing. *Modified Spearman Rank Order Correlation Coefficient* is a better choice than *Spearman Rank Order Correlation Coefficient* to measure the closeness of the two rankings. Since, it is capable of working on a full list and a partial list and the sequence Σ of documents viewed by a user is almost always a partial list, which in turn is used in getting the rankings $\Re_A$, $\Re_B$ and $\Re_C$ and

hence aggregated ranking $\Re_{Comp}$ is also a partial list. Use of *Modified Spearman rank-order correlation coefficient (r_s')* saves computational efforts both in conversion of partial list to full list and also in the computation of r_s' for truncated lists.

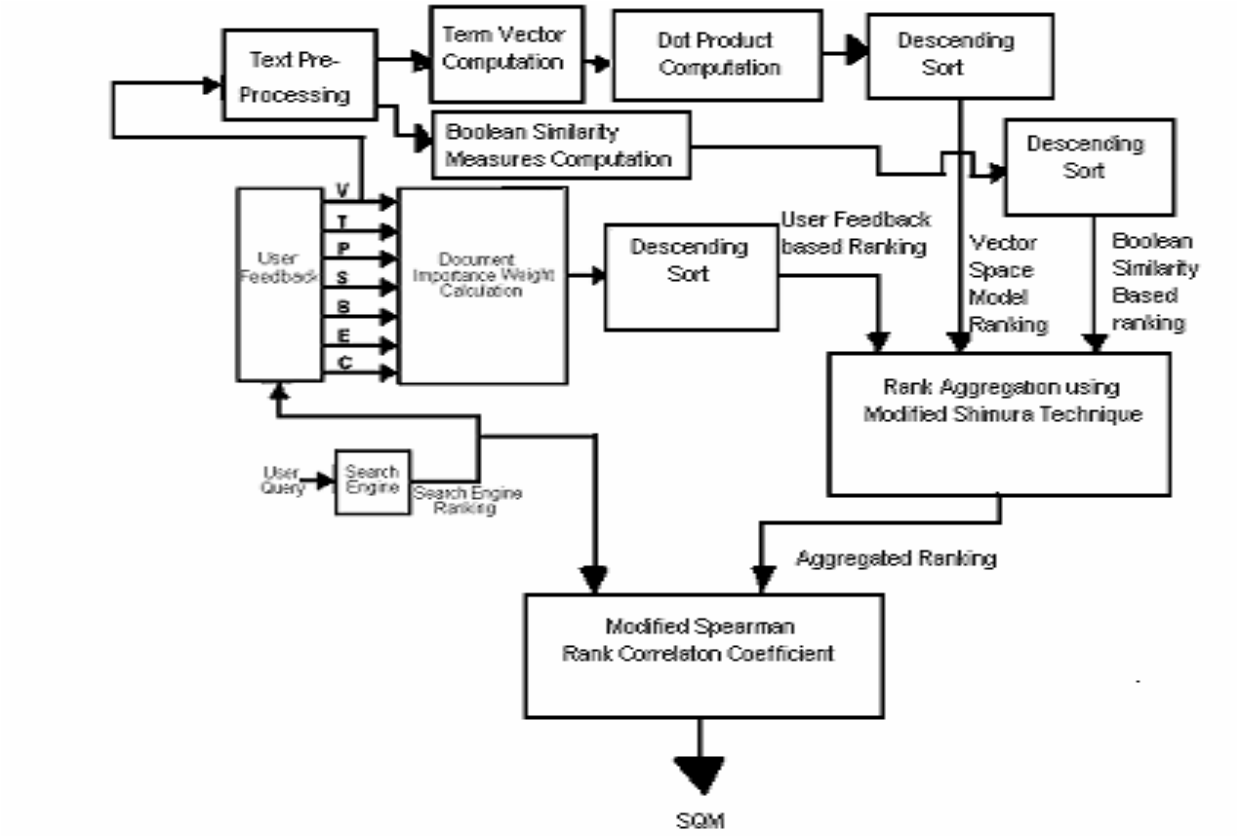

Figure 1: Comprehensive Search Quality Evaluation

4 EXPERIMENTS AND RESULTS

We experimented with a few queries on seven popular search engines, namely, AltaVista, DirectHit, Excite, Google, HotBot, Lycos and Yahoo. It may be noted here that our emphasis is more to demonstrate the procedure of quality measurement than to carry out the actual performance measurement of these search engines. It is for this reason that for subjective measure, we have obtained all our results with the weights in equation (3) being $w_V=1$, $w_T=1$, $w_P=1$, $w_S=1$, $w_B=1$, $w_E=1$ and $w_C=1$. For example, the observation corresponding to the query *similarity measure for resource discovery* is given in Table 1.

Table 1 shows that from the results of *AltaVista*, the document listed first was the document picked up first by the user, the document was read by the user for 20% of the time required to read it completely. It was not printed but saved and book marked. It was neither e-mailed to anyone nor was any of its portions copied and pasted elsewhere. The user then picked the second document listed by *AltaVista* and spent on it 20% of the time required actually to read it completely. It was not printed but was saved, book marked and e-mailed. None of its portion was copied and pasted. This gives an importance weight (σ_j) of 3.200 and 3.700 to the first and

second documents, respectively. So the implicit ranking given by the user is document 2 $\succ$ document 1, where "$\succ$" indicates "more relevant than". .i.e $\Re_A$ =(2,1) for *AltaVista* for the query. This way the value of $\Re_A$ is found for rest of the search engines as shown in Table 1for the query *similarity measure for resource discovery*.

Table 1: User Feedback model results for the Query:
similarity measure for resource discovery

Search Engine	User Feedback (*V,T,P,S,B,E,C*)	Document Weight (σ_j)	Ranking of documents based on user feedback ($\Re_A$)
AltaVista	(1,0.2,0,1,1,0,0.0)	3.200	2
	(2,0.2,0,1,1,1,0.0)	3.700	1
DirectHit	(1.0.2,0,1,1,0,0.0)	3.200	5
	(5,0.2,0,1,1,1,0.0)	3.700	1
Excite	(6,0.2,0,1,1,0,0.0)	3.200	4
	(4,0.2,0,1,1,1,0.0)	3.700	6
	(9,0.2,0,1,1,0,0.0)	2.490	9
Google	(1,0.4,0,0,1,0,0.0)	2.400	1
	(3,0.3,0,0,1,0,0.0)	1.800	3
	(5,0.3,1,0,0,0,0.0)	1.550	5
Hotbot	(2,0.2,0,1,1,0,0.0)	3.200	2
Lycos	(2,0.2,0,1,1,0,0.0)	3.200	2
	(3,0.5,1,0,1,0,0.0)	3.000	3
	(7,0.5,1,0,0,0,0.0)	1.750	7
Yahoo	(2,0.2,0,1,1,0,0.0)	3.200	2
	(4,0.2,0,0,1,0,0.0)	1.700	4
	(9.0.3,0,0,0,0,0.0)	0.550	9

In vector space model, each query and document is represented by a term vector. Once the text pre-processing such as removal of stop-words, stemming is performed on query as well as on documents picked by user, normalized term vectors for the query and the documents are obtained. For example, the normalized term vector for the query *similarity measure for resource discovery* is (0.500,0.500,0.500,0.500). The vector contains only four components because there are only four terms in the query namely "*similarity", "measure", "resource" and "discovery"*. The fifth word in the Query was "*for"* which is a stop-word and hence it is removed in pre-text processing. Each component has the same value (0.500) because all the four term appear in the query same number of times (only once). The vector is normalized to one since the magnitude of vector is 1. Similarly, normalized term vectors for the documents picked up from the results of the query are obtained. The observation corresponding to the query *similarity measure for resource discovery* is given in Table 2. From the results of *AltaVista*, first and second documents were picked up as first and second document respectively. Table 1 shows that the term vector for first document is (0.500,0.500,0.500,0.500) which is same as that of the query. That means the first documents contains all the four terms present in the query and also each term appear in the document same number of times. The dot product of this term vector with that of query is 1.000.The term vector for the second document, on the other hand, is (0.378,0.882,0.252,0.126).

Here, each component has different value since the four different terms appear different number of times in the document. The dot product of this with query term vector is 0.819. So the implicit ranking given by the vector space model is document 1 > document 2, where ">" indicates "more relevant than". We are not computing the term vectors for the rest of the documents listed by *AltaVista* as they were not clicked by user, thereby assuming that none of them contain relevant information. Thus, $\Re_B$ =(1, 2) for *AltaVista* for the query. This way the value of $\Re_B$ would be found for rest of the search engines as shown in Table 2 for the query *similarity measure for resource discovery*.

Table 2: Vector Space Model Results for the Query:
similarity measure for resource discovery

Search Engine	Picked Document	NomalizedTermVector (c1,c2,c3, c4)	Dot product cos(θ)	Ranking of documents based on Vector Space Model ($\Re_B$)
AltaVista	1	(0.500,0.500,0.500,0.500)	1.000	1
	2	(0.378,0.882,0.252,0.126)	0.819	2
DirectHit	1	(0.500,0.500,0.500,0.500)	1.000	1
	5	(0.378,0.882,0.252,0.126)	0.819	5
Excite	6	(0.500,0.500,0.500,0.500)	1.000	6
	4	(0.378,0.882,0.252,0.126)	0.819	9
	9	(0.500,0.500,0.500,0.500)	1.000	4
Google	1	(0.436,0.655,0.436,0.436)	0.982	3
	3	(0.500,0.500,0.500,0.500)	1.000	1
	5	(0.378,0.882,0.252,0.126)	0.819	5
Hotbot	2	(0.500,0.500,0.500,0.500)	1.000	2
Lycos	2	(0.500,0.500,0.500,0.500)	1.000	2
	3	(0.706,0,706,0.314,0.314)	0.738	3
	7	(0.064,0.032,0.993,0.096)	0.592	7
Yahoo	2	(0.500,0.500,0.500,0.500)	1.000	2
	4	(0.436,0.655,0.436,0.436)	0.982	4
	9	(0.400,0.400,0.200,0.800)	0.900	9

In Boolean Similarity based model, each query and document is represented by a Boolean expression. We assume that Boolean expressions of the query Q and documents to be compared (C_1, C_2, ….., C_n), just contain only AND terms i.e their CDNF contain only a single compact descriptor. Once the text pre-processing such as removal of stop-words, stemming is performed on query as well as on documents picked by user, set of descriptors in the compact atomic descriptors of the query and the documents are obtained. For example, set of descriptors in the compact atomic descriptor of the query *similarity measure for resource discovery* is (similarity, measure, resource, discovery). The set contains only four terms because there are only four AND terms in the Boolean expression of the query namely "*similarity*", "*measure*", "*resource*" and "*discovery*". The fifth word in the query was "*for*" which is a stop-word and hence it is removed in pre-text processing. Similarly, sets of descriptors in the compact atomic descriptors of all the documents picked up by user from the results of the query are obtained. Once we have obtained set of descriptors present in the compact atomic descriptors of the query and documents, we can

easily compute the simplified Boolean similarity measure using equation (17). The observation corresponding to the query *similarity measure for resource discovery* is given in Table 3.

Table 3: Boolean Similarity Model Results for the Query:
similarity measure for resource discovery

Search Engine	Picked Document	$\left\|T_{C_k}^j - T_Q^i\right\|$	$\left\|T_Q^i - T_{C_k}^j\right\|$	$S^{\otimes}$ (Q, C_k)	Ranking of documents based on Boolean Similarity Measures ($\Re_C$)
Altavista	1	56	0	0.017857	1
	2	94	0	0.010638	2
DirectHit	1	56	0	0.017857	1
	5	94	0	0.010638	5
Excite	6	56	0	0.017857	6
	4	94	0	0.010638	9
	9	56	0	0.017857	4
Google	1	171	0	0.005848	3
	3	56	0	0.017857	5
	5	94	0	0.010638	1
Hotbot	2	56	0	0.017857	2
Lycos	2	56	0	0.017857	2
	3	450	0	0.002222	3
	7	496	0	0.002016	7
Yahoo	2	56	0	0.017857	2
	4	171	0	0.005848	4
	9	448	0	0.002232	9

From the results of *AltaVista*, first and second documents were picked up as first and second document respectively. Table 2 shows that the values of $\left|T_{C_k}^j - T_Q^i\right|$ and $\left|T_Q^i - T_{C_k}^j\right|$ to be used in equation (16) for the first document are 56 and 0 respectively. The Boolean similarity measure for this document, with that of query is 0.017857. The values of $\left|T_{C_k}^j - T_Q^i\right|$ and $\left|T_Q^i - T_{C_k}^j\right|$ for the second document, on the other hand, are 94 and 0 respectively.

The Boolean similarity measure for this with query is 0.010638. So the implicit ranking given by the Boolean similarity based model is document 1 $\succ$ document 2, where "$\succ$" indicates "more relevant than". We are not computing the Boolean similarity measures for the rest of the documents listed by *AltaVista,* as user did not click them, thereby assuming that none of them contain relevant information. Thus, $\Re_C$ = (1,2) for the *AltaVista* for the query. This way the value of $\Re_C$ would be found for rest of the search engines as shown in Table 3 for the query *similarity measure for resource discovery.*

All these three rankings $\Re_A$, $\Re_B$ and $\Re_C$ are then aggregated using *Modified Shimura Technique*. The Aggregated Ranking $\Re_{Comp}$ thus obtained is then compared with original ranking

$\Re_{SE}$ to get the *Modified Spearman Rank Order Correlation Coefficient*. Thus, *AltaVista* gets $\Re_{Comp}$ =(1,2) for the query. This would be compared with $\Re_{SE}$ =(1,2) to give the *Modified Spearman Rank Order Correlation Coefficient* (r_s' =1.000) for *AltaVista*. This way the value of r_s' would be found for rest of the search engines as shown in Table 4 for the query *similarity measure for resource discovery*.

Table 4: aggregated ranking ($\Re_{Comp}$ obtained using modified shimura technique) and modified spearman rank order correlation coefficient (r_s') for the Query: ***similarity measure for resource discovery***

Search Engine	Picked Document	Aggregated Ranking ($\Re_{Comp}$)	Correlation Coefficient (r_s')
Altavista	1	1	1.000000
	2	2	
DirectHit	1	1	0.812500
	5	5	
Excite	6	6	0.687500
		9	
	4	4	
	9		
Google	1	1	0.930556
		3	
	3	5	
	5		
Hotbot	2	2	0.666667
Lycos	2	2	0.875000
		3	
	3	7	
	7		
Yahoo	2	2	0.829167
	4	4	
		9	
	9		

Table 5:List of Test Queries

1	*"measuring search quality"*
2	*"mining access patterns from web logs"*
3	*"pattern discovery from web transactions*
4	*"distributed associations rule mining"*
5	*"document categorization query generation"*
6	*"term vector database"*
7	*"client -directory-server-model"*
8	*"similarity measure for resource discovery"*
9	*"hypertextual web search"*
10	*"IP routing in satellite networks"*
11	*"focussed web crawling"*

12	*"concept based relevance feedback for information retrieval"*
13	*"parallel sorting neural network"*
14	*"spearman rank order correlation coefficient"*
15	*"web search query benchmark"*

Table 6: modified spearman rank order correlation coefficient (r_s') obtained using aggregated ranking ($\Re_{Comp}$) for the Queries given in Table 5

Query	Altavista	DirectHit	Excite	Google	Hotbot	Lycos	Yahoo
1	0.800000	0.835417	0.840067	0.981250	0.977778	0.750000	0.877551
2	0.741667	0.875000	0.887500	0.900000	0.850505	0.888889	1.000000
3	0.729167	1.000000	0.777778	0.862500	0.765152	0.866667	0.947917
4	0.791667	0.757576	0.706349	0.833333	0.937500	0.400000	0.771429
5	0.645833	0.222222	0.875000	0.937500	0.833333	0.666667	0.791667
6	1.000000	1.000000	0.797619	1.000000	0.876190	1.000000	0.795833
7	0.930556	1.000000	0.800000	0.933333	0.793651	0.250000	0.906250
8	1.000000	0.812500	0.687500	0.930556	0.666667	0.875000	0.829167
9	0.685714	0.788360	0.848958	0.888889	0.854167	0.845714	0.807500
10	0.550505	1.000000	0.200000	1.000000	0.181818	0.671717	0.790476
11	0.888889	0.905000	0.733333	0.930556	0.882540	0.977778	0.913131
12	0.859375	0.861111	0.947917	0.916667	0.285714	1.000000	0.930556
13	1.000000	0.515873	0.222222	1.000000	0.181818	0.181818	0.666667
14	1.000000	0.914286	0.944444	0.977778	0.937500	1.000000	0.762500
15	0.181818	0.181818	0.181818	0.500000	0.285714	0.181818	0.181818
Avg	**0.787013**	**0.777944**	**0.696700**	**0.906157**	**0.687337**	**0.703738**	**0.798164**

We experimented with 15 queries in all. These queries are listed in Table 5 and their *modified spearman rank order correlation coefficient (r_s')* thus obtained using aggregated ranking ($\Re_{Comp}$) is given in Table 6. The results of Table 6 are pictorially represented in Figure 2. From Table 6 and Figure 2, we observe that Google gives the best performance, followed by Yahoo, AltaVista, DirectHit, Lycos, Excite, and Hotbot, in that order.

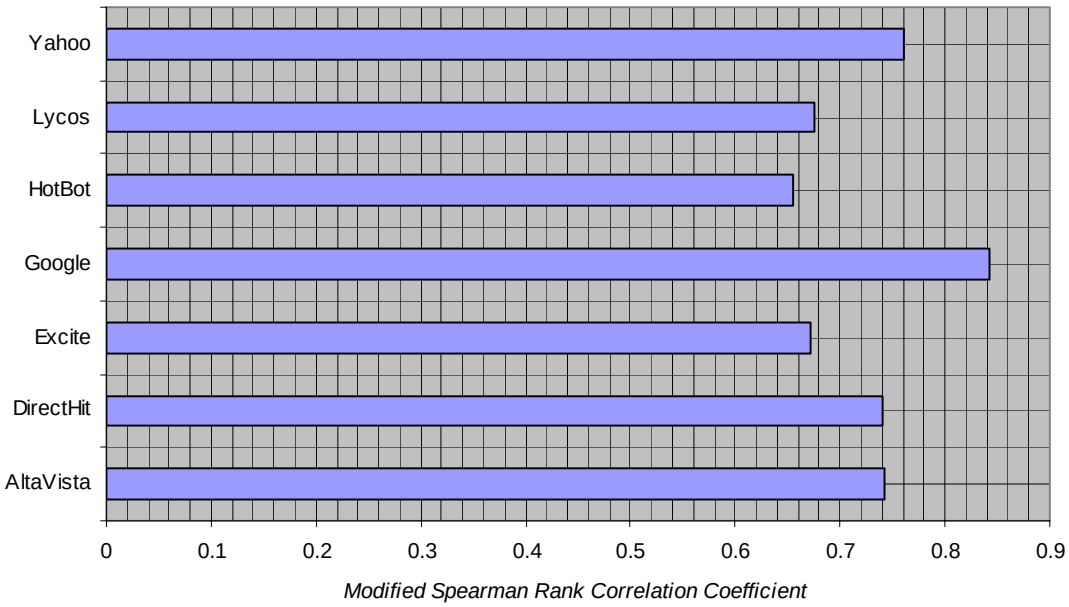

Figure 2: Performance of Search Engines based on Aggregated Ranking Model

5 CONCLUSION

We have tried to combine the user feedback based subjective evaluation with Vector Space Model and Boolean similarity measure based objective evaluation for the public web search engines. For the subjective measure, we have devised a method that observes the actions of the users on the search results presented before him and then infer his preferences there from. For the objective measure, we have used Vector Space Model and Boolean similarity measures. we have proposed and used the simplified version of Li Danzig Boolean similarity measure for computing the similarity between the query and the documents returned by the search engines. We are aggregating the ranking of documents obtained from these three evaluation processes using Modified Shimura Technique. Our results for 15 queries and 7 public web search engines show that Google gives the best performance, followed by Yahoo, AltaVista, DirectHit, Lycos, Excite, and Hotbot, in that order.

6 REFERENCES

[1] Henzinger M. R., Heydon A., Mitzenmacher M. and Najork M., "Measuring Index Quality Using Random Walks on the Web," Computer Networks, 31, 1999, pp. 1291-1303.

[2] Henzinger M. R., Heydon A., Mitzenmacher M. and Najork M., "On Near Uniform URL Sampling," Proc. 9th International World Wide Web Conference (WWW9),May 2000, pp. 295-308.

[3] Bar-Yossef Z., Berg A., Chien S., Fakcharoenphol J. and Weitz D., "Approximating Aggregate Queries about Web Pages via Random Walks," Proc. 26th Very Large Data Bases (VLDB) Conference, Cairo, Egypt, September 10-14, 2000, pp. 535-544.

[4] Bharat K. and Broder A., "A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines," Proc. 7th International World Wide Web Conference (WWW7), April 1998, pp. 379-388.

[5] Lawrence S. and Giles C. L., "Searching the World Wide Web," Science, 5360(280), 1998, pp. 98-100.
[6] Lawrence S. and Giles C. L., "Accessibility of Information on the Web," Nature, 400, 1999, pp. 107-109.
[7] Hawking D., Craswell N., Thistlewaite P. and Harman D., "Results and Challenges in Web Search Evaluation," Proc. 8th International World Wide Web Conference (WWW8), May 1999, Toronto, Canada, pp. 1321-1330.
[8] Li L. and Shang Y., "A New Method for Automatic Performance Comparison of Search Engines," World Wide Web: Internet and Web Information Systems, 3, 2000, pp. 241-247.
[9] Shang Y. and Li L., "Precision Evaluation of Search Engines," World Wide Web: Internet and Web Information Systems, 5, 2002, pp. 159-173.
[10] Weisstein E. W. "Spearman Rank Correlation Coefficient" From MathWorld – A Wolfram Web Resource, © 1999-2004 Wolfram Research, Inc. http://mathworld.wolfram.com/SpearmanRankCorrelationCoefficient.html
[11] Beg M. M. S., "On Measurement and Enhancement of Web Search Quality," Ph.D. thesis submitted to the Department of Electrical Engineering, I. I. T. Delhi, 2002.
[12] Porter M., "An Algorithm for Suffix Stripping," Program: Automated Library and Information Systems, 14(3), 1980, pp. 1980.
[13] Salton G. & McGill M. J., “Introduction to modern information retrieval,” McGraw Hill, 1983.
[14] Li S.H. and Danzig P.B., “Boolean similarity measures for resource discovery,” IEEE Trans. on Knowledge and Data Engineering, volume 9,No. 6, 1997, pp. 863-876
[15] Li S.H. and Danzig P.B., “Boolean similarity measures for resource discovery,” Technical Report, USC-CS-94-579 (Computer Science Department, University of South California, Los Angeles) 1994
[16] M. Shimura, "*Fuzzy Sets Concept in Rank-Ordering Objects*," J. Math. Anal. Appl., vol. 43 pp. 717-733, 1973.
[17] R. R. Yager, "*On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making*," IEEE Trans. Systems, Man and Cybernetics, vol. 18, no. 1,January/February 1988.