Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 169 (2005) 365–381
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

A subjective measure of web search quality

M.M. Sufyan Beg *

Department of Electrical Engineering, Indian Institute of Technology, Delhi, New Delhi 110 016, India

Received 16 September 2003; received in revised form 29 June 2004; accepted 5 July 2004

Abstract

In an Internet search, the user uses a query language to describe the nature of documents, and, in response, a search engine locates the documents that "best match" the description. A number of search engines are available to Internet users today and more are likely to appear in the future. These systems differ from one another in the indexing technique they use to construct the repository of documents and the search algorithm they employ to generate the response. As a result, the results for the same query from different search engines vary considerably. In this work, our aim is to outline a procedure for assessing the quality of search results obtained through several popular search engines. This procedure would enable us to compare the performance of the popular search engines like *AltaVista*, *DirectHit*, *Excite*, *Google*, *HotBot*, *Lycos* and *Yahoo*, etc.

We measure the "satisfaction" a user gets when presented with the search results. 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 therefrom. The implicit ranking thus provided by the user is 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 results pertaining to 7 public search engines and 15 ad hoc queries. Our emphasis is more to demonstrate the procedure of quality measurement than to carry out the actual performance measurement of these search engines.

* Tel.: +91 11 2659 6120; fax: +91 11 2658 1274.
E-mail address: mmsbeg@ee.iitd.ac.in

0020-0255/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2004.07.003

Keywords: World Wide Web; Search engines; Performance evaluation; User feedback; Biased meta-search

1. Introduction

Search engines are among the most popular as well as useful services on the web. But the problem we face is due to the large number of search engines publicly available for the purpose. We need to know how these search engines compare. The comparison could be done on many bases, such as the number of web pages they are indexing (web coverage), the time of their response, their availability and busyness, the correctness of their ranking, etc. Out of these, the first and the last ones seem to be most crucial. In this work, we mainly concentrate on the last one. We can wait for a while provided that we are ensured of good results in the end. Now this goodness of results is very subjective. The users' vote should be counted in this matter. How are the users rating the results of a search engine should be taken into account to evaluate that search engine as a whole. 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. This ranking may then be compared with the original search engine ranking to get a correlation coefficient. An average of this coefficient for a representative set of queries would give a measure of search quality of a search engine. But the problem is to obtain a correct feedback. The problem with the form-based approach is that it 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 therefrom.

This paper is organized as follows. In Section 1.1, we briefly review the related work. In Section 2, we outline our procedure of measuring the search quality. We present our experimental results in Section 3. Some discussion is given in Section 4. Finally, we conclude in Section 5.

1.1. Related work

To the best of our knowledge, no attempt has been made to quantify the user's satisfaction to the search results. However, by other means, efforts have been made to compare the performance of various search engines.

Web Coverage: The underlying principle is to collect a uniform sample of the web pages by carrying out random walks on the web. This uniform sample is then used to measure the size of indices of the various search engines. This index size is an indirect means to estimate the performance of a search engine. Larger the index size, more is the web coverage and so more likely is the emergence of good search results. Some of the efforts in this direction are given in [1–3]. The relative size and overlap of search engines has also been found in [4], but this time, instead of random walks, random queries have been used. These random queries 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 [5,6], the same goal, viz. comparing the search engines, has been achieved using a standard query log like that of NEC Research Institute. In another instance [7], a test data set has been made available for evaluation of web search systems and techniques by freezing a 18.5 million page snapshot of a part of the web. Using this data, it has been concluded that the standard of document ranking produced by public web search engines is by no means state-of-the-art.

Relevance: 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. Moreover, the vector space model is a content-based technique with no consideration to the satisfaction of the user. Our work is driven by the ultimate goal—the satisfaction of the users.

Precision: 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 quantification of "user satisfaction" on the whole, we aspire to get a complete picture of web search evaluation in this work. In fact, it is acknowledged in [9] itself that the major benefit of subjective evaluation of web searching is the accuracy.

2. Search quality

Let us begin with some useful definitions.

Definition 1. Given a universe U and $T \subseteq U$, an *ordered list* (or simply, a *list*) l with respect to U is given as $l = [d_1, d_2, \ldots, d_{ITI}]$, with each $d_i \in T$, and $d_1 \succ d_2 \succ \ldots \succ d_{|T|}$, where "$\succ$" is some ordering relation on T. Also, for

$i \in U \wedge i \in l$, let $l(i)$ denote the position or rank of i, 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 $U = \{1, 2, \ldots, |U|\}$.

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

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

Definition 3 (Partial List). A list l containing elements, which are a strict subset of U, is called a partial list. We have a strict inequality $|l| < |U|$.

Definition 4 (Spearman Rank Order Correlation Coefficient). Let the full lists $[a_1, a_2, \ldots, a_n]$ and $[b_1, b_2, \ldots, b_n]$ be the two rankings for some query Q. Spearman rank-order correlation coefficient (r_s) between these two rankings is defined as follows.

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} [l(a_i) - l(b_i)]^2}{n(n^2 - 1)}$$

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 $r_s = -1$.

2.1. User feedback vector

The underlying principle of our approach [10] of performance 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 kth 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 shall 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 shall 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 shall 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 shall 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 and pastes a portion of the document, indicate the level of importance that document holds for the specified query.

2.2. The search quality measure (SQM)

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

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

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 s 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 σ_j will yield a sequence Σ. Let the full list ρ be the sequence in which the documents were initially short-listed. Without loss of generality, it could be assumed that $\rho = (1, 2, 3, \ldots, N)$, where N is the total number of documents listed in the result. We compare the sequences Σ and ρ, and find *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 quantitative measure of the search quality (SQM). The above procedure is illustrated in Fig. 1.

For realizing our test bed, we load our evaluation software on a machine. The software executes the following steps.

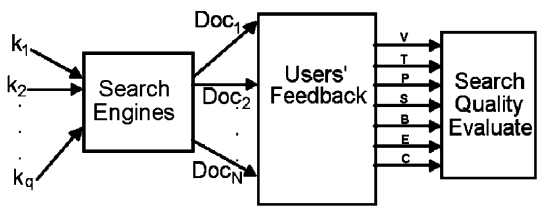

Fig. 1. Search quality evaluation.

1. Take the list of search engines to be compared.
2. Take the set of queries from the users.
3. Distribute a query to all the search engines in the given list.
4. Get the results from all the search engines and display them before the respective user one by one.
5. To note:
   (a) which document is picked up at what sequence,
   (b) time spent in reading the picked up documents,
   (c) if the picked up document was printed,
   (d) if the picked up document was saved,
   (e) if the picked up document was bookmarked,
   (f) if the picked up document was e-mailed,
   (g) what portion of the document has been copied-and-pasted elsewhere.
6. Compute the weighted sum σ_j for each document j selected by the user using Eq. (1).
7. Rank the document in a decreasing order of their weight σ_j.
8. Complete the sequence Σ by filling up the remaining spaces by a reverse count.
9. Compare Σ with the initial listing ρ given by the search engine to evaluate r_s.
10. Repeat steps 5–9 for the results of all the search engines.
11. Repeat steps 3–10 for the given set of queries.
12. Take the average of r_s for each search engine over all the queries and output this list.

2.3. Some finer points in SQM

We must note here that it is a very common practice that the user views only those documents whose snippets displayed before him by the search engine he finds to be worth viewing. This would give only the ranking of the documents viewed by the user. This is to say that the list Σ would almost always be a partial list. In such a case, it is assumed that the rest of the documents are implicitly declared irrelevant by the user and so they are sequenced in a reverse order to complete the sequence, so as to penalize the search engine for displaying irrelevant information. For instance, the user looks at the documents in such a way that sorting documents based on σ_j gives the sequence as 5, 2 and 7. Then for a total of ten documents, the sequence Σ would be obtained as 5, 2, 7, 10, 9, 8, 6, 4, 3 and 1.

As seen in Section 3, this harsh step of penalization brings down the search quality measures of the search engines drastically. In an effort to moderate this step, we took to an average ranking method. Let there be a partial list l_j and a full list l, with the number of elements in them being $|l_j|$and $|U|$, respectively.

In order to evaluate the *Spearman rank order correlation coefficient* between l_j and l, we first complete l_j into a full list as follows.

$$l_j(i) = \begin{cases} \text{unchanged}, & \text{if } i \leqslant |l_j| \\ x | x \in U \vee x \notin l_j, & \text{otherwise} \end{cases}$$

Next we modify the positions in the full list l as follows.

$$l(i) = \begin{cases} \text{unchanged}, & \text{if } i \leqslant |l_j| \\ \frac{\sum_{k=(|l_j|+1)}^{(|U|)} l(k)}{(|U|-|l_j|)}, & \text{otherwise} \end{cases}$$

Now, we can find the Spearman rank order correlation coefficient (r_s) between the lists l and l_j as explained in Section 2.2.

Example 2. For $|U| = 5$, let the full list be $l = \{5,4,1,3,2\}$ and the partial list l_j with $|l_j| = 3$ be $l_j = \{2,1,4\}$. We shall first complete l_j into a full list as $l_j = \{2,1,4,3,5\}$ and also modify l as $l = \{5,4,1,2.5,2.5\}$.

An extreme case would be when no matching document is found by a search engine for a given query. In such a case, the engine must be penalized by assuming the sequence Σ to be just the reverse of ρ, and hence the Spearman rank-order correlation coefficient (r_s) would be taken as -1 for that case. Similarly, if a search engine lists a document location, but that location when accessed shows non-existence of that document there, either due to "404 error" or the document might have been moved elsewhere, the engine is penalized for this out-dated information by taking its time fraction ($T_j/T_j^{\max}$) as well as the word fraction (C_j/C_j^{total}) to be zero.

2.4. Applications of SQM

One obvious application of SQM is to know the performance of public web search engines. In this section, we wish to point to another novel application of SQM, the *Biased Meta Search*. A conventional meta-search engine is the one that doesn't have a database of its own, rather it takes the search results from other public search engines, collate those results and present the combined result before the user. This is how we get the combined advantage of different search techniques being employed by the participating search engines. This collation of results is achieved through what is called as *Rank Aggregation* [11–16]. One of the classical methods of Rank Aggregation, for instance, is the Borda's method.

Definition 5 (Borda's Method (BM) of Rank Aggregation). Given k lists $l_1, l_2, \ldots, l_k$, for each candidate c_j in list l_i, we assign a score $S_i(c_j) = |c_p : l_i(c_p) > l_i(c_j)|$. The candidates are then sorted in a decreasing order of the total Borda score $S(c_j) = \sum_{i=1}^{k} S_i(c_j)$.

Example 3. Given lists $l_1 = [c, d, b, a, e]$ and $l_2 = [b, d, e, c, a]$.

$S_1(a) = |e| = 1$, as $l_1(e) = 5 > l_1(a) = 4$.
Similarly,
$S_1(b) = |a, e| = 2$, as $l_1(e) = 5 > l_1(b) = 3$ and $l_1(a) = 4 > l_1(b) = 3$.
Proceeding this way, we get

$$
\begin{aligned}
&S_1(c) = |a, b, d, e| = 4,\\
&S_1(d) = |a, b, e| = 3,\\
&S_1(e) = || = 0,\\
&S_2(a) = || = 0,\\
&S_2(b) = |a, c, d, e| = 4,\\
&S_2(c) = |a| = 1,\\
&S_2(d) = |a, c, e| = 3,\\
&S_2(e) = |a, c| = 2,\\
&S(a) = S_1(a) + S_2(a) = 1 + 0 = 1,\\
&S(b) = S_1(b) + S_2(b) = 2 + 4 = 6,\\
&S(c) = S_1(c) + S_2(c) = 4 + 1 = 5,\\
&S(d) = S_1(d) + S_2(d) = 3 + 3 = 6,\\
&S(e) = S_1(e) + S_2(e) = 0 + 2 = 2.
\end{aligned}
$$

Now, sorting the elements based on their total scores, we get the combined ranking as $b \approx d \succ c \succ e \succ a$. The '$\approx$' symbol indicates a tie.

But the problem with the existing Rank Aggregation methods is that during the process of coming up with the consensus ranking, the results of all the participating engines are combined with equal weightage. However, we wish to incorporate the scenario in which the voters may not be given equal weightage for their votes. The vote of a more educated voter should be weighted higher than the one coming from a less educated one. To address this issue, our idea of *Biased Rank Aggregation* is to use the SQM to weigh up the results of the corresponding search engines. The weighted Borda's method would then be as follows.

Definition 6 (Weighted Borda's Method (WBM)). For each element c_j in the list l_i coming from the ith search engine having SQM q_i, we may assign the

score as $S_i(c_j) = q_i \cdot |c_p: l_i(c_p) > l_i(c_j)|$. The candidates may then be sorted as usual, in a decreasing order of the total Borda score $S(c_j) = \sum_{i=1}^{k} S_i(c_j)$.

With this weighting scheme, the better performing search engines would be given due weightage to their rankings before the rank aggregation is applied. As reported in [17], the user satisfaction, on a scale of -1 to 1, is 0.2 for the Borda's Method and 0.8 for the weighted Borda's method. This way, we may drift from the concept of "simple democracy" to that of "educated democracy".

3. 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, as also to simplify our experiments, that we have obtained all our results with the weights in Eq. (1) being $w_V = 1$, $w_T = 1$, $w_P = 1$, $w_S = 0$, $w_B = 0$, $w_E = 0$ and $w_C = 0$. For example, the observation corresponding to the query *document categorization query generation* is given in Table 1. This table shows that from the results of *AltaVista*, the second document was picked up first by the user, but a zero time was spent on it, most probably because the documents must not be at this location anymore, and so no printout could be taken. The second pick up was made by the user on what was listed as the first document by *AltaVista*, the document was read by the user for just 0.11% of the time required to read it completely, and no printout was taken once again. This gives an importance weight (σ_j) of 1.00 and 0.5011 to the second and the first documents, respectively. So the implicit ranking given by the user is document 2> document 1, where ">" indicates "more relevant than". Since the user does not seem to be interested in the rest of the ten documents listed by *AltaVista*, it is assumed that the user finds all of them irrelevant. *AltaVista* must be penalized for listing the unwanted documents and thus putting the user into trouble of glancing through all of them before reaching to a decision. This is done by putting the rest of the documents in a reverse order, so as to complete the sequence Σ. Thus, Σ=(2, 1, 10, 9, 8, 7, 6, 5, 4, 3). This would be compared with $\rho = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ to give the Spearman rank order correlation coefficient (r_s) = -0.030303 for *AltaVista*. This way the value of r_s would be found for rest of the search engines as shown in Table 1 for the query *document categorization query generation*.

We experimented with 15 queries in all. These queries and their results are summarized in Table 2. We have given the values of the *Spearman rank order correlation coefficient* (r_s) between the listing of each of the seven search engines

Table 1
Results for the query "document categorization and query generation"

Search engine	Document preference		Fraction of time (T)	Printed? (P)	Importance of document (σ_j)	User's sequence (Σ)	r_s (ρ, Σ)
	Sequence (V)	Document no.					
AltaVista	1	2	0.0	0	1.00	2, 1, 10, 9, 8, 7, 6, 5, 4, 3	−0.030303
	2	1	0.0011	0	0.5011		
DirectHit	1	10	0.00091	0	1.00091	10, 9, 8, 7, 6, 5, 4, 3, 2, 1	−1.000000
Excite	1	7	0.012	0	1.012	7, 10, 9, 8, 6, 5, 4, 3, 2, 1	−0.927273
Google	1	1	0.092	0	1.092	5, 2, 1, 3, 10, 9, 8, 7, 6, 4	+0.381818
	2	2	0.88	0	1.38		
	3	3	0.0	0	0.25		
	4	5	0.94	1	2.065		
HotBot	1	1	0.092	0	1.092	6, 1, 10, 9, 8, 7, 5, 4, 3, 2	−0.393939
	2	6	0.88	0	1.38		
Lycos	1	1	0.0	0	1.000	2, 7, 1, 10, 9, 8, 6, 5, 4, 3	−0.030303
	2	2	0.88	0	1.38		
	3	7	0.92	0	1.17		
Yahoo	1	1	0.092	0	1.092	2, 1, 5, 9, 3, 10, 8, 7, 6, 4	+0.406061
	2	2	0.88	0	1.38		
	3	3	0.0	0	0.25		
	4	5	0.92	0	1.045		
	5	9	0.47	0	0.5325		

and the ranking provided implicitly by the user by virtue of his feedback vector, in response to each of those 15 queries. The results of Table 2 are presented in Fig. 2. We have taken these 15 queries for the sake of demonstrating our procedure. We may have taken many more. In fact, the actual measure would require a continuous evaluation by taking a running average over an appropriate window size of the successive queries being posed by the users.

From Fig. 2, we see that *Yahoo* is the best, followed by *Google*, *Lycos*, *HotBot*, *AltaVista*, *Excite* and *DirectHit*, in that order. We also observe that most of the publicly available search engines taken in this study are much below the users' expectations. That is the reason why all but one of them is getting a negative value of Spearman rank order correlation coefficient averaged over all the queries.

Moreover, we were also a bit harsher in penalizing these search engines for giving irrelevant results. As explained in Section 2.3, the documents that are not at all touched by the user are sequenced in a reverse order to complete the sequence Σ, so as to penalize the search engine for displaying irrelevant information. This harsh step has brought down the search quality measures of these search engines. In an effort to moderate this step, we took to the average ranking method. For instance, if the user looks at the documents in such a way that sorting documents based on σ_j gives the sequence as 4 and 2. Then for a total of ten documents, the sequence Σ would be obtained as 4, 2, 1, 3, 5, 6, 7, 8, 9, 10. But the original search engine ranking ρ with which it is to be compared to get r_s is taken as 1, 2, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5. The logic is that the documents 4 and 2 are anyway at the 1st and 2nd position, respectively, but the rest of the documents are jointly at the next position, which in turn, happens to be the average of the remaining ranks, i.e. $(3 + 4 + 5 + 6 + 7 + 8 + 9 + 10)/8 = 6.5$. It may be noted that this way, it becomes immaterial whether we take the remaining documents to complete the sequence Σ in the same order or the reverse order, or any order for that matter. With this average ranking method, we got the performances improved homogeneously, as shown in Fig. 3.

Comparing the Figs. 2 and 3, we see that *Yahoo* still stands out as the best of the lot, followed by *Google*, *Lycos*, *HotBot*, *AltaVista* and *Excite*, in that order. *DirectHit*, which appeared worst in Fig. 2, has however, improved its performance over *AltaVista* and *Excite*, in Fig. 3. This is because *DirectHit* was giving more irrelevant results, and so was penalized due to the harsh measures taken for Fig. 2. This eased out substantially in Fig. 3. Let us, however, reiterate that these rankings of the search engines by us are just a pointer to what to expect, and should not be taken as the last word as yet. Our aim in this paper is just to bring out a procedure for ranking search engines. We have just taken a few ad hoc queries, 15 to be precise. For a more confident ranking of the search engines, we need to have a more comprehensive set of test-queries.

Table 2
Summary of results for 15 queries

S. no.	Query	Spearman rank order correlation coefficient (r_s) for						
		AltaVista	DirectHit	Excite	Google	HotBot	Lycos	Yahoo
1	"Measuring search quality"	−0.842424	−0.745455	−0.418182	−0.018182	−0.951515	−0.272727	0.32121
2	"Mining access patterns from web logs"	0.006061	−0.272727	−0.442424	−0.042424	0.054545	0.212121	0.321212
3	"Pattern discovery from web transactions"	−0.575758	−0.454545	−0.454545	0.321212	0.406061	−0.29697	0.321212
4	"Distributed association rule mining"	−0.878788	−0.563636	−0.563636	−0.018182	−0.115152	0.272727	−0.018182
5	"Document categorization and query generation"	−0.030303	−1	−0.927273	0.381818	−0.393939	−0.030303	0.406061
6	"Term vector database"	−0.878788	−0.927273	−1	−0.115152	−0.454545	−0.030303	−0.018182
7	"Client-directory-server model"	−1	−1	−1	0.284848	−0.575758	−0.381818	−0.454545
8	"Similarity measure for resource discovery"	−0.030303	−0.2	−0.345455	0.066667	−0.018182	−0.006061	0.066667
9	"Hypertextual web search"	−0.115152	−1	−1	0.054545	−0.563636	−0.381818	0.127273
10	"IP routing in satellite networks"	−1	−0.454545	−1	−0.527273	−0.030303	0.054545	−0.369697
11	"Focussed web crawling"	−0.2	−1	−0.927273	0.054545	−0.018182	−1	0.10303
12	"Concept based relevance feedback for information retrieval"	−0.115152	−0.454545	−0.151515	−0.2	−0.115152	−0.272727	−0.2
13	"Parallel sorting neural network"	−0.115152	−0.030303	−0.2	−0.006061	−1	0.187879	−0.006061
14	"Spearman rank order correlation coefficient"	0.175758	−0.430303	0.018182	−0.272727	−0.018182	0.575758	0.212121
15	"Web search query benchmark"	−0.2	−0.660606	−0.224242	−0.454545	−0.018182	−0.454545	−0.454545
	Average	−0.386667	−0.612929	−0.575757	−0.032727	−0.250101	−0.121616	0.023838

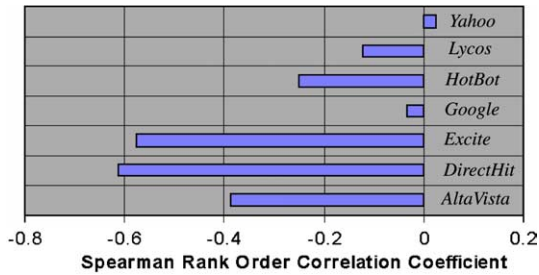

Fig. 2. Performance of search engines based on three components of user feedback vector; (V, T, P). (For colour see online version.)

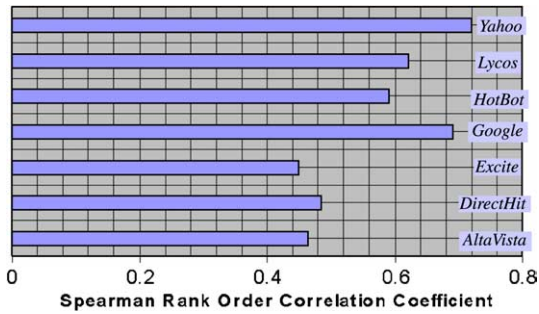

Fig. 3. Performance of search engines based on three components of user feedback vector; V, T, P (sequence of untouched documents averaged). (For colour see online version.)

4. Discussion

We see that most of the publicly available search engines taken in this study are much below the users' expectations. That is the reason why all but one of them is getting a negative value of *Spearman Rank Order Correlation Coefficient* averaged over all the queries. This clearly shows that most of the times, most of them are either not able to fetch the desired documents, or if at all they are doing that, they are not able to rank them as per their relevance. What is important is not just the fetching of all the relevant results, but also they should be ranked properly. All the search engines fall much behind this target.

The strength of our search quality measure appears to be the fact that the ultimate intelligence of human beings is used for obtaining the "true" ranking (Σ) of the documents, which in turn, could be used as a standard to compare the ranking given by the search engines (ρ) against it. But for this, we are not at all bothering the user, rather inferring on our own the user feedback from his actions on the results. Moreover, our measure is absolute and not relative.

While carrying this exercise out, we also came across a few minor shortcomings in our approach. Let us enumerate them below.

(a) We are relying solely on the user feedback with the good hope that we would get honest response from majority, if not all, of them. If, however, a group of users combine to bring down the reputation of a particular search engine for some reason or the other, not much can be done with our approach.
(b) This approach does not take into account the time of retrieval of search results.

4.1. A word about query generation

Broadly speaking, we can call a net searcher to be either an *expert* or a *novice*. An expert knows exactly the keyword he is looking for and is also interested in exact results. A novice, on the other hand, supplies vague or far terms and gets happy with any reasonable result. The example of an expert could be a researcher carrying out literature survey in a focussed area of his research interest, whereas a novice could be a schoolboy looking for some reasonable material to complete the write up for his home assignment. If we draw a parallel to this, we may classify the resulting queries also into two corresponding categories. *Broad* queries come from a novice and may contain very few terms, e.g. ''election'', ''giraffe'', ''tropical forest'', etc. Narrow queries, on the other hand, are expected from experts and may contain many qualifying terms, e.g. ''focussed web crawling'', ''concept based relevance feedback for information retrieval'', ''parallel sorting neural network'', etc.

For the purpose of evaluating the quality of a search engine as a whole, the opinion of an expert should be given utmost importance. The novice would get satisfied even with the second grade results and this would hardly help in the search engine evaluation. In fact, the novice might even mislead in the evaluation procedure due to his lack of knowledge in that domain. It is for this reason that we are studying the user feedback to the results of narrow queries only. Our aim in this work is to demonstrate a procedure for evaluating the quality of search results. For this purpose, we picked up some 15 queries and tried them on 7 public web search engines. Our system requires a continuous evaluation of these search systems by means of a running average over the successive queries. However, if a one-time estimate of the search quality is to be made, we must have a carefully chosen set of queries. For this, we may resort to a real search query log, such as the one from NEC Research Institute used in [5,6]. Another alternative is the one used in [4]. With the assumption that *Yahoo* hierarchy has a fairly decent collection of web documents, a broad crawl is made on those nearly 300,000 documents. Some 400,000 index words were picked up

from this collection. Now these words may be combined randomly to generate the required queries. However, even then we won't be able to achieve a particular level of statistical confidence in stating our results regarding the quality of search systems.

Another more decent approach could be by using the works reported in [1–3]. These works have been concentrating on obtaining a near uniform sample of the web by carrying out random walks on the web. Once this near uniform sample is obtained, we may use it to our benefit. These sampled pages could be analyzed for the index terms in them using the standard TF–IDF (term frequency–inverse document frequency) procedure. The index terms thus obtained would be a sample set of what to expect on the web. So these terms could then be combined to form queries for search quality evaluation. But since, our aim has been to outline a procedure for search quality evaluation rather than the comparison of the search engines as such, we leave the query generation problem as a future direction of research.

5. Conclusion

We have tried to quantify the search quality of a search system. We have used the notion of "user satisfaction" for this purpose. More satisfied a user is with the search results presented to him in response to his query, higher is the rating of the search system. The "user satisfaction" is being gauged by the sequence in which he picks up the results, the time he spends at those documents and whether or not he prints, saves, bookmarks, e-mails to someone or copies-and-pastes a portion of that document. Proper formulation has been done for the combination of these metrics. Our proposition is tested on 7 public web search engines using some 15 queries. With this limited set of queries, it has been found that *Yahoo* gives the best performance followed by *Google*, *Lycos*, *HotBot*, *AltaVista*, *Excite* and *DirectHit*, in that order. To say it with more confidence, we need to have a better set of queries. Our aim in this paper is just to bring out a procedure for ranking search engines. We also discuss briefly the biased meta-search, an application of the Search Quality Measure of the web search engines. While aggregating the ranks of documents from different search engines, we can take into account the quality of those search engines as well.

References

[1] M.R. Henzinger, A. Heydon, M. Mitzenmacher, M. Najork, Measuring index quality using random walks on the web, Computer Networks 31 (1999) 1291–1303.

[2] M.R. Henzinger, A. Heydon, M. Mitzenmacher, M. Najork, On near uniform URL sampling, in: Proc. 9th International World Wide Web Conference (WWW9), May 2000, pp. 295–308.

[3] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, D. Weitz, Approximating aggregate queries about web pages via random walks, in: Proc. 26th Very Large Data Bases (VLDB) Conference, Cairo, Egypt, September 10–14, 2000, pp. 535–544.
[4] K. Bharat, A. Broder, A technique for measuring the relative size and overlap of public web search engines, in: Proc. 7th International World Wide Web Conference (WWW7), April 1998, pp. 379–388.
[5] S. Lawrence, C.L. Giles, Searching the World Wide Web, Science 5360 (280) (1998) 98–100.
[6] S. Lawrence, C.L. Giles, Accessibility of information on the web, Nature 400 (1999) 107–109.
[7] D. Hawking, N. Craswell, P. Thistlewaite, D. Harman, Results and challenges in web search evaluation, in: Proc. 8th International World Wide Web Conference (WWW8), Toronto, Canada, May 1999, pp. 1321–1330.
[8] L. Li, Y. Shang, A new method for automatic performance comparison of search engines, World Wide Web Internet and Web Information Systems 3 (2000) 241–247.
[9] Y. Shang, L. Li, Precision evaluation of search engines, World Wide Web: Internet and Web Information Systems 5 (2002) 159–173.
[10] M.M.S. Beg, C.P. Ravikumar, Measuring the quality of web search results, in: Proc. 6th International Conference on Computer Science and Informatics—A Track at the 6th Joint Conference on Information Sciences (JCIS 2002), Durham, NC, USA, March 8–13, 2002, pp. 324–328.
[11] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proc. 10th World Wide Web Conference (WWW10), Hong Kong, May 1–5, 2001, pp. 613–622.
[12] M.M.S. Beg, N. Ahmad, Genetic algorithm based rank aggregation for the web, in: Proc. 6th International Conference on Computer Science and Informatics—A Track at the 6th Joint Conference on Information Sciences (JCIS 2002), Durham, NC, USA, March 8–13, 2002, pp. 329–333.
[13] N. Ahmad, M.M.S. Beg, Fuzzy logic based rank aggregation methods for the world wide web, in: Proc. International Conference on Artificial Intelligence in Engineering and Technology (ICAIET 2002), Malaysia, June 17–18, 2002, pp. 363–368.
[14] N. Ahmad, M.M.S. Beg, Improved methods for rank aggregation on the world wide web, in: Proc. International Conference on Knowledge Based Computer Systems (KBCS 2002), Mumbai, India, December 19–21, 2002, pp. 193–202.
[15] M.M.S. Beg, N. Ahmad, Soft computing techniques for rank aggregation on the World Wide Web, World Wide Web—An International Journal 6 (1) (2003) 5–22.
[16] M.M.S Beg, N. Ahmad, Fuzzy logic and rank aggregation for the world wide web, in: V. Loia, M. Nikravesh, L.A. Zadeh (Eds.), Fuzzy Logic and the Internet, Springer-Verlag, 2004, pp. 27–46.
[17] M.M. Sufyan Beg. Improved meta-search for the world wide web, in: Proc. Eighth National Conference on Communication (NCC 2002), IIT Bombay, January 25–27, 2002, pp. 135–139.