A search quality evaluation based on objective-subjective method

Fugui Wang,Yajun Du,Qinhua Dong
School of Mathematics & Computer Science
Xihua University, Chengdu, Sichuan, 610039, China
Xhu218@163.com, efg218@139sc.com

Abstract

Commercial search engines, especially meta-search engines was designed to retrieve the information by submitting users' queries to multiple conventional search engines and integrating their partial searching results generated by each search engine. How to find the best search engine for user queries is know as the selection problem of search engines. Recently, the selection problem has become a research hotspot in meta-search engines. Therefore, it is the key technology to evaluate the quality of the search result of every search engine. In this paper, the objective-subjective measure method was proposed. At first, we focus on objective evaluation of search quality. From global viewpoint, we calculated the average correlation coefficient of one search result list with other search result lists, the correlation coefficient described the relation among the original lists of search results, and it is objective measure of the search result of search engine；Second, we focus on the subjective evaluation of search quality. We measure the user satisfaction to each search result of search engine by comparing the implicit ranking provided by the user feedback with the original ranking given by the search engine. At last, the objective-subjective measure method is the weighted sum of the average correlation coefficient and the user satisfaction. What more, we discussed the algorithm of our proposed method.

Keywords

Correlation Coefficient; Search Engines; Quality Evaluation; User Feedback

1. Introduction

Search engines are the most popular useful services on the web. Meta-search engines become more and more important. After submitting the user query to the meta-search engine, the user query divided into several sub queries, the meta-search integrated their partial searching results which generated by each conventional search engine into the final search result for user query. In process of dividing the user query, how to choose a best conventional search engine？ We need to know what the different among these search engines. The different among the conventional search engine is in many factors, such as the number of web pages they are indexing (web coverage), the time of their response, their availability, the correctness ranking, etc., which are adopted to evaluate these search engines. Of these factors, the first and the last ones seem to be most crucial. In this work, we mainly concentrate on the last one.

Statistic Model [1-4] proposed two indexes， NoDoc and AvgSim to evaluate the quality of search result. NoDoc denoted the high relation documents' count which search engine retrieved from the index databases for user query; Avgsim stands for high relational documents' average similarity ratio.

User Model [5-7] is different from traditional model. The user model focused on more factors about user feedback. It analyzes the search result quality by monitor user's action, such as the time that the user spends on this document, whether user copies, prints, emails, bookmarks the document or not, and so on.

As we know, the few researches focus on discussion combining the correlation of the search result sets and qualify the users' satisfaction. However, by other means, efforts have done to compare the performance of various search engines.

In this paper, we proposed the objective-subjective approach for search quality evaluation. At first, we focus on objective evaluation of search quality. From global viewpoint, we calculated the average correlation coefficient of one search result list with other search result list; the correlation coefficient described the relation among the original lists of search results. It is objective measure of the search result of search engine. Second, we focus on the subjective evaluation of search quality. We measure the user satisfaction by comparing the implicit ranking provided by the user feedback with the original ranking given by the search engine. At last, the objective-subjective measure method proposed that is the weighted sum of the average correlation coefficient and the user satisfaction. This paper is organized as follows. In Section 2, we outline our procedure of measuring the search quality, in section 3, Experiments, and results , finally, we conclude in Section 4.

2. Search quality evaluation

In this section, we discuss the objective-subjective approach and its algorithm.

Definition 1 Consider that a universe D is all web pages of a search result for a user query. Let D= [d_1, d_2… d_n], without loss of generality, we may consider that the position of web page d_i is i in D. Let $T \subseteq D$, an ordered list l with respect to T is $d_1 \succ d_2 \succ ... \succ d_{|T|}$, where "$\succ$" is some ordering relation on T. For example, a search result set l, is given as [$d3$, $d4$, $d2$, $d1$, $d5$], has the ordering relation $d_3 \succ d_4 \succ d_2 \succ d_1 \succ d_5$. The universe D= [$d_1$, d_2, d_3… d_4, d_5] may be taken as {1, 2, 3, 4, 5} with, say $d_1 \equiv 1, d_2 \equiv 2, d_3 \equiv 3, d_4 \equiv 4, d_5 \equiv 5$. With such an assumption, we have l = [3, 4, 2, 1, 5].

Definition 2[5] (Spearman Rank Order Correlation Coefficient).Let the lists [a_{11},a_{12},…a_{1n}]and [a_{21},a_{21},…a_{2n}] be the two ranking. Spearman rank-order correlation coefficient (rs) between these two rankings is defined as follows.

$$rs = 1 - \frac{6\sum_{i=1}^{n}[l(a_{1i}) - l(a_{2i})]^2}{n(n^2-1)} \quad (1)$$

The function $l(x)$ means the position of the element x. The Spearman rank-order correlation coefficient (rs) is a measure of closeness relationship of two rankings. The coefficient rs between -1 and 1 when the two rankings are identical, rs = 1, and when one of the rankings is the inverse of the other then rs = -1.

2.1 Objective search quality evaluation

2.1.1 Basic theory of objective evaluation

When user submit a query ,Let SE_1,SE_2 …SE_n be the search result set of search engine 1 to n .LSE_1,LSE_2 ...LSE_n are ordered list based on search rank policy R_1,R_2,…,R_n .which response to SE_1 SE_2 ,…,SE_n . LSE_{ij} is ordered list base on $SE_i \cap SE_j$ according to LSE_j 's order . For example LSE_i = [$1,2,3,4,5,6,7$] LSE_j =[3,2,4,1,7],we get LSE_{ij} = [$3,2,4,1,7$]. Another ordered list LSE_{ji} base on $SE_j \cap SE_i$ according to LSE_i order. We get LSE_{ji}=[$1,2,3,4,7$] and then

$$r_s(LSE_{ij}, LSE_{ji}) = 1 - \frac{6\sum_{i=1}^{n}[LSE_{ij}(i) - LSE_{ji}(i)]^2}{n(n^2-1)} \quad (2)$$

We get rs(LSE_{ij},LSE_{ji}) = 0.3 it is means the similar between LSE_{ij} and LSE_{ji} is 0.3 .and the value is more large ,the two list are more similarly. When the value is -1, it means that the two order list are reverse. When the value is 1, it is means that the two order are all the same.

For get an objective evaluation of search engine. Now we should analysis the affection by other search engine. The search engine sets = {SE_1,SE_2,…SE_m}(m stands for search engine's count),so

$$obj(SE_i) = \sum_{j=1}^{n} w_j * r_s(LSE_{ij}, LSE_{ji}) \quad (3)$$

w_j means the important for SE_j $\sum_{j=1}^{n} w_{se} = 1$

The evaluation of one search result depends not only by this query experience ,but also by the weight of others search engine. The weight is higher, and the affect is bigger.

2.1.2 Algorithm implement

We distribute a query list as [q_1,q_2,…q_n] to search list as [SE_1,SE_2,…SE_m] the algorithm is as fellow:

It assume all the webpage in the results, $page_1 \equiv 1$, $page_2 \equiv 2$,…,$page_n \equiv n$. As Table1,we only analysis the top 20 of each search result. We can get the correlation coefficient of each search engine result. cc_{ij} denote the value of rs value multiply by the weight of SE_j

Table 1. The Correlation coefficient of seven-search engine

	SE_1	SE_2	SE_3	SE_4	SE_5	SE_6	SE_7
SE_1	---	cc_{12}	cc_{13}	cc_{14}	cc_{15}	cc_{16}	cc_{17}
SE_2	cc_{21}	---	cc_{23}	cc_{24}	cc_{25}	cc_{26}	cc_{27}
SE_3	cc_{31}	cc_{32}	---	cc_{34}	cc_{35}	cc_{36}	cc_{37}
SE_4	cc_{41}	cc_{42}	cc_{43}	---	cc_{45}	cc_{46}	cc_{47}
SE_5	cc_{51}	cc_{52}	cc_{53}	cc_{54}	---	cc_{56}	cc_{57}
SE_6	cc_{61}	cc_{62}	cc_{63}	cc_{64}	cc_{65}	---	cc_{67}
SE_7	cc_{71}	cc_{72}	cc_{73}	cc_{74}	cc_{75}	cc_{76}	---

Input: search engine list (seList), query list(querylist), the weight for each search engine
Output: the objective evaluation for each search engine in search list
Procedure:

```
int Query_no = querylist.Length;//query term number.
int Search_no = seList.Length;//search engine number.
int Each_result_no = 20;// anlysis the top 20 of each search engine result only.
string[] Query = new string[query_no]; //initialize query term;
string[] SearchEngine = new string[search_no]; //initialize search engine;
string[,,] SearchResult = new string[search_no,query_no,each_result_no];//search
                    result for all the search engine.
string [,] Rspearman_Matrix = new string[Search_no,Search_no] ;//correlational
                     coefficient matrix for all the search engine
 string []SearchWeight = new string[Search_no];//init the weight for each search engine
 string []objSearchEvalute = new string[Search_no];//the objective evalution
 /* Submit the query list to the search list and get the search result       */
  for(int i = 1 ;i < Search_no;i++)//search engine list
    for(int j = 1 ;j < Query_no;j++)//query list
    {
        //Query[j] submit to SearchEngine[i];
        for(int k = 1;k<=Each_result_no;k++)
        {
           initialize SearchResult[i][j][k]
        }
    }
 /* get the correlational coefficient matrix         */
  for(int i = 1;i <Query_no;i++)
     for(int j = 1;j<Search_no;j++)
        for(int k = 1;k<Search_no ;k++)
        {
           if (j == k )continue;
           /* get the Rspearman matrix by Rspearman function  */
           Rspearman_Matrix[j,k]
                 =SearchWeight[k]* rs(SearchResult[i][j],SearchResult[i][k]);
        }
 /* get the objective evalution for each search engine in search engine list */
 for(int i = 1 ;i<=Search_no;i++) //search enginelist
 {
     for(int j = 1;j<Search_no;j++) //search enginelist
     {
        objSearchEvalute[i] += Rspearman_Matrix[i][j];
      }
     objSearchEvalute[i] /= Search_no;
 }
```

2.2 Subject search quality evaluation

2.2.1. Basic theory of subjective evaluation

The underlying principle of subjective search quality 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 a document by a vector (*V*, *T*, *P*, *S*, *B*, *E*, *C*), which consists of the following [5]. *V* means the order of the user visits the document *i*. *T* means a user spends examining the document *i* . *P* means the user prints the document or not; The *B* means whether the user book-marked the document *i* or not; *E* means the user e-mailed the document *i* to someone or not; *C* means the number of words the user copied and pasted elsewhere.

The motivation of collecting users' feedback is well known as a well-educated user is likely to select the more appropriate documents early. Similarly, the time that a user spends examining a document, and whether he prints, saves, bookmarks, e-mails it to someone else or copies and pastes a portion of the document or not, indicate the level of importance that document holds for the specified query. When feedback recovery is complete, we propose to compute the following weighted sum σ_j for each document j selected by the user.

$$\sigma_j = \left(w_v \frac{1}{2^{(v_j-1)}} + w_T \frac{t_j}{t_j^{max}} + w_p p_j + w_s s_j + w_b b_j + w_E e_j + w_c \frac{c_j}{c_j^{total}} \right)$$

Where T_{jmax} represents the maximum time expected a user is to spend in examining the document *j*, and C_{jtotal} 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, we want to give to each of the seven components of the feedback vector. The sum σ_j represents the importance of document *j*. Now, sorting the documents on the descending values of σ_j will yield a sequence *R*. Let the full list ρ be the list in which the documents were initially short-listed. Without loss of generality, it could be assumed that ρ = {1, 2,3, . . .,N}, where *N* is the total number of documents listed in the result. We compare the sequences *R* and ρ, and find Spearman rank order correlation coefficient (rs).

2.2.2 Algorithm implement

Now, we watch user's action on the search result, as in Table 2 shows.After compute each document's σ_j, we sort the σ_j in descending .We get user's order list R

Compute the spearman between R and the initiations ρ .We distribute a query list as [$q_1,q_2,\ldots q_n$] to search list as [$SE_1,SE_2,\ldots SE_m$] the algorithm is as fellow.

Table 2. User's action on a search result

DOC.	V	T(second)	P	S	B	E	C
doc_1	3	5	1	1	0	1	0.3
doc_2	2	6	0	0	0	0	0
doc_3	4	10	0	0	0	0	0
doc_4	-1	0	0	0	0	0	0
doc_5	-1	0	0	0	0	0	0
…	…	…	…	…	…	…	…
doc_{20}	-1	0	0	0	0	0	0

Input: search engine list, query list

Output: the subjective evaluation for each search engine in search list

Procedure:

```
/* we define the doc as ten part as a struct */
    public struct doc
    {
        public int docID;//doc's id
        public string docUrl;//doc's url
        public int docView;//doc's view order
        public float docTime;//doc's cost time
        public bool docPrint;//doc's print
        public bool docSave;//doc's save
        public bool docBookMark;//doc's bookmark
        public bool docEmail;//doc's email
        public float dorspy;//doc'copy
        public float sum;//doc's weight
    }
  int Query_no = querylist.Length;//query term number.
  int Search_no = seList.Length;//search engine number.
  int each_result_no = 20;
  string[] Query = new string[query_no]; //initialize query term;
```

```
string[] SearchEngine = new string[search_no]; //initialize search engine;
doc []sedoc = new doc[each_result_no];//initialize the doc struct
float []subjectSearchEvalute = new float[Search_no];//the objective evalution
/** Submit the query list to the search list and get the search result**/
 for(int i = 1 ;i < Search_no;i++)//search engine
 {
    subjectSearchEvalute[i] = 0.0;
     for(int j = 1 ;j < Query_no;j++)//query list
     {
        //Query[j] submit to SearchEngine[i];
        for(int k = 1;k<=Each_result_no;k++)
         {
           initialize sedoc[k];
           Comprehend user's feed back;
           update sedoc[k];
           count each sedoc[k].sum
           get User's  Order list(userdoc)  by sort  sedoc[k].sum by desc
           subjectSearchEvalute[i] += Rspearman(userdocList,sedocList)
          }
         subjectSearchEvalute[i] /= Query_no;
   }
```

2.3 A search quality measure combined objective and subjective aspect

Now, we put forward a objective-subjective search quality evaluate method, while combine objective aspect and subjective aspect based on 2.1.1 and 2.2.1 as Table 3 shows.

$S(SE_i) = \mu *$ objective-evaluation + $(1 - \mu)$ subjective-evaluation. It means $S(SE_I) = \mu *$ objSearchEvalute[i] + $(1-\mu)$subjectSearchEvalute[i].Here μ is a learn ratio value.

3. Experiments and results

On the base of above analysis, this artic select the most famous search engine. Such as Google,Yahoo,MSN,Altavista, DirectHit,Netscape . It assume that the weight of all the search engine has the same importance, WSEi = 1($1 \leq i \leq m$,m stands for the search engine's count) To be justice, the query term is Radom on This artic, and select different length of term to experiments. Table 4 is objective evaluation of the result on the test condition of "2008 Olympic Games"

Table 3. The search quality evaluation result

	SE_1	…	SE_{m-1}	SE_m
objValue	objSearchEvalute[1]		objSearchEvalute[m-1]	objSearchEvalute[m]
subValue	subjectSearchEvalute[1]		subjectSearchEvalute[m-1]	subjectSearchEvalute[m]
totalValue	SE[1]		SE[m-1]	SE[m]

Table 4. "2008 Olympic Games"'s objective evaluate result

SEj	Google	Yahoo	MSN	AltaVista	Direct Hit	Netscape
Google	--	0.40	0.80	0.60	0.40	0.70
Yahoo	0.40	--	1.00	0.20	1.00	0.50
MSN	0.80	1.00	--	0.40	0.80	1.00
AltaVista	0.60	0.20	0.40	--	0.60	0.60
Direct Hit	0.4	1.00	0.80	0.60	--	0.75
Netscape	0.70	0.50	1.00	0.60	0.75	--
Avg	0.58	0.62	0.8	0.40	0.71	0.71

Table 5. "2008 Olympic Games"'s subjective evaluate result

Search Engine	Document Preference Sequence(V)	Document no.	Fraction of time(T)	Printed (P)	Importance of doc(σ_j)	User's sequence($\sum$)	$r_s(\sum, \rho)$
Google	1	1	0.092	0	1.092	2,1,5,9,3,10,8,7,6,4	0.271415
	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		
Yahoo	1	1	0.092	0	1.38	5,2,1,3,10,9,8,7,6,4	0.381818
	2	2	0.88	0	0.25		
	3	3	0.0	0	2.065		
	4	5	0.94	1	1.092		
MSN	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		
AltaVista	1	7	0.012	0	1.012	7,10,9,8,6,5,4,3,2,1	-0.927273
Direct Hit	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		
Netscape	1	10	0.00091	0	1.00091	10,9,8,7,6,5,4,3,2,1	-1.00000

From above, $S(SE_I)$ = μ * objSearchEvalute[i] + (1- μ)subjectSearchEvalute[i]., assume μ =0.5 .We get S(Google) = 0.851415,S(Yahoo) = 1.001818 S(MSN) =0.769697 S(AltaVista) = -0.527273 S(Direct Hit) = 0.316061,S(Netscape) =-0.29.The search engine quality is decrease as Yahoo,Google,MSN,DirectHit,Netscape,AltaVista.

4. Conclusion

We have tried to quantify the search quality of a search system. We have used the notion of ''search results' correlation coefficient and user satisfaction'' for this purpose. Analysis the correlation coefficient of the return search results for get the search quality. In addition, on subjective aspect, a user is more satisfied with the search results presented to him in response; 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 done for the combination of these metrics. 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.

5. Acknowledgement

This work is supported by the Education Department Foundation of Sichuan Province (Grant No.2006A086). the Application Foundation of Sichuan Province (Grant No.2006J13-056), the Cultivating Foundation of Science, and Technology of Xihua University (Grant No.R0622611), the cultivating foundation of the science and technology leader of Sichuan province.

6. References

[1] Wanbao Lei,Ping Liang,Huo Xue,An Estimation Model Based on the Correlation of the Search Result Sets, Journal of Information No.2,2007

[2] King—Lup Liu.Clement Yu, W eiyi Meng, Wensheng Wu.Napht~i Rishe. A Statistical Method for Estimating the Usefulness of Text Databases. IEEE TRANSACTIONS ON KNOW LEDGE AND DATA ENGlNEERlNG, 2002; 14(6)

[3] FaginR,KumarR ,SivakumarD. ComparingTopK Lists. SIAM Journa1 oll Discrete Mathematies.2003; 17(1)

[4] Judit Bar—llan. Comparing Rankings of Search Results on the Web. Information, Processing and Management, 2005; (41)

[5] M.M. Sufyan Beg, A subjective measure of web search quality, Information Sciences 169(2005)365-381

[6] A mamda Spink ,A user-centered approach to evaluating human interaction with Web search engines:an exploratory study,Information Processing and Management 38(2002 401-426)

[7] M.M. Sufyan Be,Nesar Ahmad,Web search enhancement by mining user actions,Information Sciences 177(2007)5203-5218

[8] Meng Tao, Hongfei Yan, Xiaoming LI ,An Evaluation Model on Information Coverage of Search Engines,ACTA ELECTRONICA SINICA 0372-2112(2003)08-1168-05

[8] Shicong Feng ,Songwei Shan,Zhigang Zhang,Bihong Gong ,Xiaoming Li,Research of personalization technologies based on web mining,Computer Engineering and Design,1000-7024(2004)01-0004-03

[9] Yajun Du: The study and implement of intelligent action of search engine. [Doctor paper]. Chengdou, Sichuan. Southwest jiaotong University. 2005.