ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2015.0139

QUERY RECOMMENDATIONS AND ITS EVALUATION IN WEB INFORMATION RETRIEVAL

R. Umagandhi[1] and A.V. Senthil Kumar[2]
[1]*Department of Computer Technology, Kongunadu Arts and Science College, India*
E-mail: umakongunadu@gmail.com
[2]*Department of Computer Applications, Hindustan College of Arts and Science, India*
E-mail: avsenthilkumar@yahoo.com

Abstract

Search Engine retrieves significant and essential information from the web based on query terms given by the user. Due to the lack of background knowledge about the information required, shorter length queries posed by the user, the ambiguity of query keywords and dynamic growth of the web, irrelevant and redundant results are also retrieved by the search engine. Query recommendations is an important technique which analyze the real search intent of the user and suggests the alternative queries to be used by the user in future to satisfies their information need. The proposed method recommends and ranks the alternative queries and evaluates the ranking order of the recommendations with the help of the ranking measures Normalized Discounted Cumulative Gain (NDCG) and Coefficient of Variance (CV). These measures identify the relationship between the ranking techniques. The proposed strategies are experimentally evaluated using a real time search engine query log.

Keywords:

Queries, PrefixSpan, NDCG, CV, Kappa Measure

1. INTRODUCTION

Web is the largest and voluminous data source in the world. The plentiful unstructured or semi-structured information on the web leads to a great challenge for the users, who hunt for prompt information. The scenario grows pathetic and distressing to provide personalised service to the individual users from billions of web pages. The unpredictable amount of web information available becomes a menace of experiencing ambiguity in the web search. To prevent the web users from getting overwhelmed by the quantity of information available in the web, search engines are used.

Searching the web information using search engines is a habitual activity of web users. At the end of the nineties the size of the web to be around 200 million static pages [1]. The number of indexable documents in the web exceeds 11.5 billion [2]. According to the survey done by Netcraft, Internet Services Company, England there is 739,032,236 sites in September 2013 and 22.2M more than the month August 2013. Fig.1 shows that the growth in number of web sites from 1995 to 2013. Every year, millions of web sites are newly added in the information world. Hence a proper tool is needed to search the information on the web.

Search Engine retrieves significant and essential information from the web, based on the query term given by the user. The retrieved result may not be relevant all the time. At times irrelevant and redundant results are also retrieved by the search engine because of the short and ambiguous query keywords [3].

A study on private Alta Vista Query Log has shown that more than 85% of queries contain less than three terms and the average length of the queries are 2.35 with a standard deviation of 1.74 [4]. For the second AltaVista log, instead, the average query length is slightly above 2.55. It is to be understood that the shorter length queries do not provide any meaningful, relevant and needed information to the users. For example, consider a user who submits the query term 'apple' in the search process, but he review the result only for 'apple iPod' and not for the 'apple fruit'. Here the user's interest is on apple iPod only. Due to the inappropriate keyword, the retrieved results consist of both the fruit and system. The proposed query recommendation system provides suggestions on the iPod when the same query 'apple' is triggered by the same user next time. Here the recommendation is given by considering the user's past navigations.

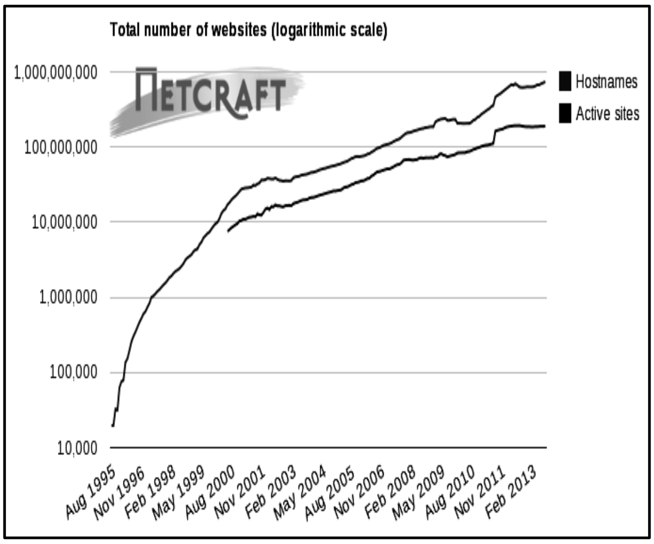

Fig.1. Statistics on number of web sites from 1995-2013

In this situation, Query recommendation is an important technique which provides suggestions to the search user to frame relevant and meaningful queries in future to retrieve the relevant results. The recommended queries are ranked. The un-ranked list is evaluated using the measures precision, recall and f-measure, but the ranked lists is evaluated using mean average precision, R-Precision, Breakeven Point, NDCG, kappa statistics and etc. The major contributions of the work are summarized as follows:

- Frequently accessed queries are identified using Modified PrefixSpan Approach, it consists of 3 processes. First process generates the frequently accessed queries. In the next process, hub and authority weights are calculated. Final process assigns t-measure to the frequent queries.

Association between the frequent queries are generated and it is used for recommendations.

- NDCG and CV measure is calculated for the frequent queries retrieved from the query log.
- The ranking order of recommended queries is compared using CV measure.
- The users assigned the relevancy score for the recommended queries. The relevancy score is evaluated using kappa statistics. Finally the users are clustered based on the relevant value assigned to the recommended queries.

2. RELATED WORKS

2.1 QUERY RECOMMENDATIONS

Information Retrieval (IR) is a method for delivering relevant information to the people who need it. Query recommendation is an essential technique for the search users to suggest set of queries used in future for relevant and required information retrieval. The goal of Recommender Systems (RECSYS) is suggesting items based on users profile and items content, in order to direct users to the items that best meet their preferences and profile. Different techniques suggested for the query recommendation process is center-piece subgraph [5], Query Flow Graph [6] and Term Query (TQ)-Graph [7]. The queries are selected and suggested from those appearing frequently in query sessions [8] to use clustering to devise similar queries on the basis of cluster membership. Clustering approach is used in query recommendations by using click-through data information to devise query similarity [9] [10] [11]. [12] Proposed a model for generating queries to be suggested based the concept of query rewriting. A query is rewritten into a new one either by means of query or phrase substitutions [13] or using tools [14].

2.2 EVALUATION OF RECOMMENDATIONS

Ranked Support Vector Machine (RSVM) algorithm is used to rank the recommended queries [15]. The evaluation on the performance of a ranking model is carried out by comparison between the ranking lists output by the model and the ranking lists given as the ground truth. Several evaluation measures are widely used in IR. These include NDCG, DCG (Discounted Cumulative Gain), MAP (Mean Average Precision), and Kendall's Tau [16]. NDCG [17] is widely used evaluation metric for learning-to-rank (LTR) systems. It is designed for ranking tasks with more than one relevance levels. There are many open source tools are available for computing the NDCG score for the ranked list [18].

Nevertheless, it is interesting to consider and measure how much agreement between judges on relevance judgments. In the social sciences, a common measure for agreement between judges is the kappa statistic [19] [20] [21]; it is designed for categorical judgments and corrects a simple agreement rate for the rate of chance agreement.

3. GENERAL TERMS

Item: In this context an item is a query.

Support: An item set X has support s in T if $s\%$ of the transactions in T contains X. support of the query is calculated by number of times query is issued by the same user.

Confidence: Confidence is an interestingness measure of an association rule. The rule $X \rightarrow Y$ holds in T with confidence c if $c\%$ of transactions in T that contain X also contain Y.

Confidence $(X \rightarrow Y)$ = Support (XUY) / Support(X)

Frequent Item: An item I is frequent if its support is higher than the user specified minimum support threshold.

Association Rules from Query log file: Associations between the queries and the clicked URLs from the query log file are represented as a rule. Consider the following traversal path of the user U_1 for the input query Q_1. The user clicks the document B and E from the document A. The referring URL for the documents C and D is B. The adjacency matrix representation for the traversal path is

$$\begin{array}{c|c} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The association rules generated for the above traversal path is

$A \rightarrow B$, $A \rightarrow E$, $B \rightarrow C$, $B \rightarrow D$, $AB \rightarrow C$, $AB \rightarrow D$.

Hub: The hub identifies the URLs clicked for the query Q. In Fig.2, the URLs A, B and C are accessed for the query Q.

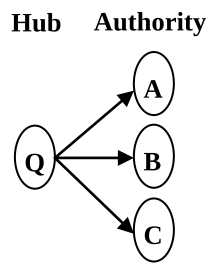

Fig.2. Multiple Authorities

Authority: The authority identifies the URLs pointed for the query Q. In Fig.3, A, B and C are the URLs which have resources for the query Q. For example,

From Fig.2,

Hub(Q) = Number of out links from Q

= Authority (A) +Authority (B) +Authority(C)

= 3

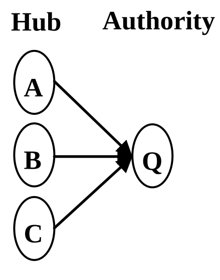

Fig.3. Multiple Hubs

From Fig.3,

Authority (Q) = Number of in links to Q

= Hub (A) +Hub (B) + Hub(C)

= 3

t-measure: A new measurement called t-measure assigns the weights for the time period where the query occurs. If URL u_1 is accessed in a discrepant couple of time periods t_1 and t_2 (t_1 occurs earlier than t_2), then the t-measure of u_1 at t_1 is lesser than the t-measure of u_1 at t_2.

$$t-measure(u_i) = Cluster\ number(u_i) \Big/ \sum_{j=1}^{n} j \tag{1}$$

where, n = number of clusters [25].

Kappa Statistic: The kappa value will be 1 if two judges always agree, 0 if they agree only at the rate given by chance, and negative if they are worse than random. If there are more than two judges, it is normal to calculate an average pair wise kappa value. As a rule of thumb, a kappa value above 0.8 is taken as good agreement, a kappa value between 0.67 and 0.8 is taken as fair agreement, and agreement below 0.67 is seen as data providing a dubious basis for an evaluation. Kappa value is calculated using Eq.(1).

$$kappa = \frac{p(A) - p(E)}{1 - p(E)} \tag{2}$$

where, $p(A)$ is the proportion of the times the users agreed the recommended queries, and $p(E)$ is the proportion of the times they would be expected to agree by chance.

Precision and Recall: The two most frequent and basic measures for unranked retrieval sets in IR effectiveness are precision and recall. The measures precision and recall is used to evaluate the retrieval process [27].

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} = P\left(\frac{relevant}{retrieved}\right) \tag{3}$$

$$Recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = P\left(\frac{retrieved}{relevant}\right) \tag{4}$$

CV: A better way to measure the dispersion is square the differences between each data and the mean before averaging them. Standard deviation shows how much variation is there from the mean. A low value indicates that the data points tend to be very close; whereas a higher value indicates that the data spread over a large range of values. The Coefficient of variance (CV) is calculated using

$$\mathrm{CV} = \frac{\text{Standard Deviation}}{\text{Mean}} * 100 \tag{5}$$

NDCG: It is a ranking measure widely used in web applications

$$DCG_N = \sum_{i=1}^{N} \left(G_i / \log_2(i+1)\right) \tag{6}$$

where, G_i represents the relevance gain assigned to the label of the document at position i.

$$NDCG_N = Z_n \sum_{i=1}^{n} \left(\left(2^{ri} - 1\right) / \log(i+1)\right)$$

$$NDCG_N = DCG_N / IDCG_N \tag{7}$$

4. QUERY RECOMMENDATIONS

4.1 ARCHITECTURE FOR RECOMMENDATIONS

The Fig.4 shows the overall process of the proposed technique. Set of queries are recommended to the web users by analysing the past behaviour of the user. Query log is a precise and imperative repository in web usage mining which contains the input query and its navigations in the search process. The log is analysed and frequently accessed queries are identified by using the algorithm ModifyPrefixSpan. The authority weight and t-measure is assigned to frequently accessed queries. The queries with higher weight are provided as the recommendations to the user. In the same way frequently accessed URLs are also identified and may be used in the recommendations [23]. NDCG measure is calculated for the frequent queries identified from the previous phase, which is the best technique to weight the URL or query. Next, the recommended queries are re-ranked using the preference, t-measure and preference with t-measure. The ranking order is evaluated by using the metric CV. The users are instructed to assign the relevancy score for the recommendations. The relevancy score is evaluated and the users are clustered based on the relevancy score.

4.2 FREQUENT QUERY GENERATION

In order to give the suggestions to frame the future queries, the search histories are analysed from the query log. The search histories are organized under the attributes

- AnonID - An anonymous user identifier
- Query -The query issued by the user
- QueryTime - The date and time on which the query was triggered by the user
- ItemRank - The rank of the clicked item in the search result
- ClickURL - If the user clicked on the search result, the domain portion of the URL in the web snippets is listed.

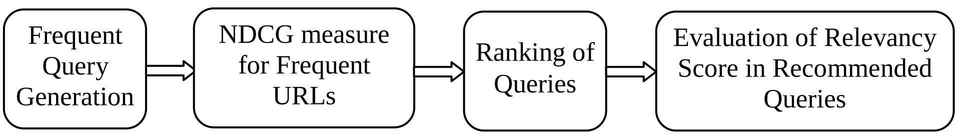

Fig.4. Process of the Proposed Technique

Algorithm ModifyPrefixSpan

Input: Query Cluster, support threshold

Output: Frequently accessed queries and Association rules, authority and t-measure

begin

Process 1:

Step 1: From Query Cluster, generates the query patterns

Step 2: Find the count for each query pattern

Step 3: If support (pattern) < = threshold then delete the pattern

Otherwise generate the association rule for that pattern

Step 4: if support (rule) > threshold then

Queries used in the rule are considered as frequent

Process 2:

Step 1: Identify the frequent queries and their associations from Process 1

Step 2: Calculate the hub and authority weight for each query using Hits Algorithm

Step 3: If authority weight (query) > =1 and support (query) >= threshold then

Generate the association rule and queries used in the rule are considered as frequent queries.

Process 3:

Step 1: Calculate the t-measure for the frequent queries identified from Process 2

end

To evaluate the proposed technique, the real search engine AOL query log data set is considered [22], which is a free open data set and it is downloaded from zola.di.unipi.it/smalltext/datasets.html. 2006 -03 - 01 to 2006 – 05 – 31. The data set contains 1975811 records and 19131507 words in 174 MB, based on our system's memory and its speed we consider a maximum of first 200 pre-processed records. Table.1 depicts the sample query log entries from AOL data set.

Table.1. Sample AOL Log Entries

Anon ID	Query	Query Time	Item Rank	ClickURL
227	psychiatric disorders	2006-03-02 17:30:36	1	http://www.merck.com
227	Cyclothymia	2006-03-02 17:34:08	1	http://www.psycom.net
309	whec tv in rochester ny	2006-05-11 14:54:43	1	http://www.10nbc.com
366	Intravenous	2006-03-01 17:16:19	3	http://en.wikipedia.org
647	rabbit hole the broad way play	2006-03-01 22:15:33	2	http://www.entertainment-link.com
1038	tow truck	2006-03-01 23:17:31	No Click	NoRank

Table.2. ModifyPrefixSpan - Frequent Queries

ModifyPrefixSpan		ModifyPrefixSpan with Hub and Authority		ModifyPrefixSpan with t-measure	
Frequent Queries with ID	**Support**	**Frequent Queries**	**Authority weight**	**Frequent Queries**	**Authority with t-measure**
www.pokemon.com 18	5	18	5.0	18	5.0519
www.gamewinners.com 17	4	17	3.9646	17	3.9936
lotto 21	9	21	7.8849	21	8.029
mickey dolenz 11	2				
cliff notes 9	2				
mapquest com 22	2				
american spirit tobacco 33	2				
www.pokemon.com 18 ww.gamewinners.com 17	2				

Table.4. Precision and Recall for the Frequent Queries

ModifyPrefixSpan			ModifyPrefixSpan with Hub and Authority			ModifyPrefixSpan with t-measure		
Frequent Query	**Recall**	**Precision**	**Frequent Query**	**Recall**	**Precision**	**Frequent Query**	**Recall**	**Precision**
18	0.333	1.000	18	0.333	1.000	18	0.333	1.000
17	0.667	1.000	17	0.667	1.000	17	0.667	1.000
21	1.000	1.000	21	1.000	1.000	21	1.000	1.000

The log entries are pre-processed [25], unique queries and URLs clicked for the unique queries are identified. The unique queries are assigned with an identifier. For illustration,

Identifier	Query
1	psychiatric disorders
2	cyclothymia
3	grooming in harrisburg pa
4	subsidized housing in harrisburg pa
5	whec tv in rochester ny
6	pen pals for kids
7	intravenous
8	rabbit hole the broadway play
9	cliff notes
10	on line casino

The generation of association among all the unique queries are very tedious and ineffective process. Hence, the frequently accessed queries are obtained by considering the prefix patterns generation procedure. The queries which satisfy the minimum support 2 are considered as frequently accessed queries. Next, calculate the hub and authority weight for the frequently accessed unique queries which are identified from Process 1 of the ModifyPrefixSpan [24] [26] algorithm. The queries which satisfy the minimum authority of 1 are considered for recommendations. Next, t-measure is calculated for the frequently accessed items along with hub and authority weight.

The first 200 pre-processed records are considered for the evaluation of the proposed technique. Out of 200, 113 unique queries are recognised, the ModifyPrefixSpan algorithm identify 8 queries are frequently accessed and 3 queries satisfied the authority weight threshold. The generation of support, authority weight and t-measure are explained in [24] [25] [26] and it is given in Table.2. First process in the proposed algorithm ModifyPrefixSpan, identifies 36 association rules which satisfy the minimum confidence of 20. Process 2 and 3 generates 3 rules. The rules and their confidence values are given in Table.3. The association rules,

18 =>>21
17 =>>18
17 =>>21

generates the recommendations for the queries 18, 17 and 21. For the query 18, the query 21 is recommended and for the query 17 the queries 18 and 21 are recommended. In Information Retrieval, the measures precision and recall is used to evaluate the retrieval process [27]. These are first defined for the simple case where the search engine retrieves set of recommended queries. Precision and Recall for the frequent and relevant queries 18, 17 and 21 are given in Table.4. The Eq.(3) and Eq.(4) given in section 3 are used to calculate the precision and recall measures. For example,

Precision (Query 18) = 3 / 3 = 1.00

Recall (Query 18) =1 / 3 = 0.33

Precision (Query 17) = 3 / 3 = 1.00

Recall (Query 17) =2 / 3 = 0

Table.3. Association Rules and their Confidence

ModifyPrefixSpan with Hub and Authority Rule & Confidence	ModifyPrefixSpan with t-measure Rule & Confidence
18 =>>21 & 25.769	18 =>>21 & 25.774
17 =>>18 & 22.611	17 =>>18 & 22.619
17 =>>21 & 29.888	17 =>>21 & 29.907

4.3 NDCG MEASURE FOR FREQUENT URLS

NDCG [17] is a widely used evaluation metric used in Ranking Algorithms. NDCG has two advantages compared to many other measures. First, NDCG allows each retrieved document has graded relevance while most traditional ranking measures only allow binary relevance. That is, each document is viewed as either relevant or not relevant by previous ranking measures; while there can be degrees of relevancy for documents in NDCG. Second, NDCG involves a discount function over the rank while many other measures uniformly weight all positions. This feature is particularly important for search engines as users care top ranked documents much more than others.

The Table.5 lists the frequent URLs identified using the algorithm PrefixspanBasic proposed in [25] and its NDCG value which is calculated by using Eq.(7).

Table.5. Frequent URL and its NDCG

n	Query#	relevance gain	DCG	IDCG	NDCG
1	7	1	1	1	1
2	51	1	2	2	1
3	18	0.6	2.379	2.631	0.904
4	17	0.8	2.779	3.031	0.917
5	21	1	3.21	3.289	0.976
6	87	0.5	3.403	3.482	0.977

The technique identifies 6 queries numbered 17, 51, 18, 17, 21 and 87 are frequently accessed. Table.6 depicts the NDCG for the queries identified using the algorithms PrefixSpanBasic and MHitsPrefixspan [25]. The Fig.5 shows that the URLs 21 and 87 got the highest NDCG value.

Table.6. Frequent URL and its NDCG of Process 1 and 2

n	URL#	PrefixSpanBasic NDCG	MHitsPrefixspan NDCG
1	7	1	1
2	51	1	1
3	18	0.904	0.904
4	17	0.917	0.917
5	21	0.9	0.976
6	87	0.944	0.977

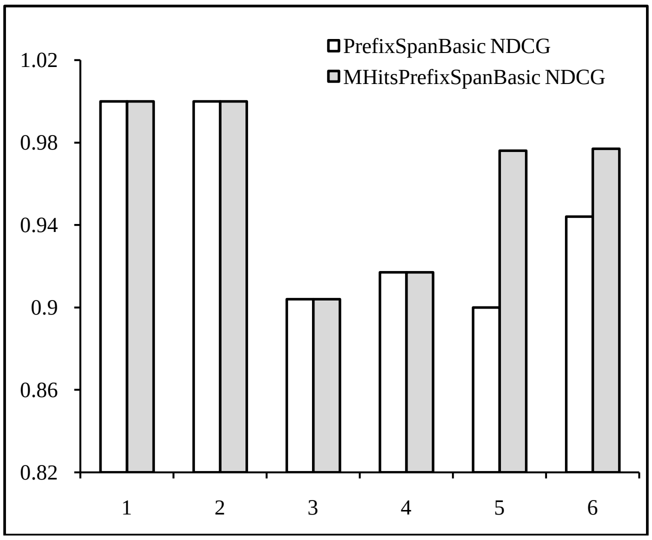

Fig.5. NDCG comparison of Process 1 and 2 of ModifyPrefixSpan

Coefficient of variance for both the NDCG lists are evaluated and it is given below;

CV(PrefixSpanBasicNDCG) = 4.438

CV(MHitsPrefixSpanNDCG) = 3.956

Coefficient of variance for MHitsPrefixSpan is lesser than the PrefixSpanBasic, Hence the values generated using MHitsPrefixspan is consistent other than PrefixSpanBasic.

4.4 RANKING OF QUERIES

The queries are ranked based on the user's preferences on day wise, query wise with t-measure. For example, consider the user and his activities around 5 days. The queries Q_i, $1 \le i \le 6$ are triggered by the user on Day j, $1 \le j \le 5$.

Day 1 - $Q1$, $Q3$, $Q4$

Day 2 - $Q1$, $Q4$, $Q5$

Day 3 - $Q1$, $Q2$, $Q3$, $Q6$

Day 4 - $Q3$, $Q4$, $Q5$

Day 5 - $Q1$, $Q2$, $Q6$

The queries $Q1$, $Q3$ and $Q4$ are issued on Day1. Table.7 depicts the support, confidence, preference and t-measure for the above day wise activities. The weight t-measure is assigned to day wise clusters. Since $Q1$ occurs on Day1, 2, 3 and 5, t-measure of $Q1$ is

1/15+2/15+3/15+5/15= 0.733

Table.7. Preference and t-measure

Query	Support	Confidence (%)	Preference	t-measure
$Q1$	4	80	0.525	0.733
$Q2$	2	40	0.263	0.533
$Q3$	3	60	0.394	0.533
$Q4$	3	60	0.394	0.466
$Q5$	2	40	0.263	0.4
$Q6$	2	40	0.263	0.533

The preference and the combined measure preference with t-measure is calculated using Eq.(8) and Eq.(9) respectively.

$$Preference\ (u,q) = \alpha * Day\_Preference\ (u,q) + \beta * Query\_Preference(u,q) \quad (8)$$

$$Preference\ \ with t-measure = \alpha * preference(u,q) + \beta * t-measure(u,q) \quad (9)$$

The Table.8 shows the changes in the ranking order according to the α value. For all the cases, irrespective of α and β the favourite query of the user is $Q1$. The queries $Q2$ and $Q6$ have equal weight and the query $Q5$ is less accessible. Table.9 shows the changes in the ranking order of 6 queries by using the ranking techniques preference, t-measure and preference with t-measure. Average ranking is assigned to the queries when they have the same measure. For example, the queries $Q3$ and $Q4$ have the same preference 0.394; hence the rank 2.5 is assigned for $Q3$ and $Q4$ instead of 2 and 3 respectively.

Table.8. Ranking of queries

α	β	Ranking of queries
0.1	0.9	Q1, Q3, (Q2, Q6), Q4, Q5
0.3	0.7	Q1, Q3, (Q2, Q6), Q4,Q5
0.5	0.5	Q1, Q3, Q4, (Q2, Q6), Q5
0.7	0.3	Q1, Q3, Q4, (Q2, Q6), Q5
0.9	0.1	Q1, Q3, Q4, (Q2, Q6), Q5

Table.9. Ranking order

Original	Preference	t-measure	Preference + t-measure (when α=0.5)
1	1	1	1
2	5	3	4.5
3	2.5	3	2
4	2.5	5	3
5	5	6	6
6	5	3	4.5

Table.10. Relevancy Score

Query :Cricket	R1	R2	R3	R4	R5	R6	R7	CV
User 1	0	2	2	2	1	--	--	57.143
User 2	1	2	2	1	1	1	0	55.902
User 3	0	2	1	1	--	--	--	70.711
User 4	0	2	1	--	--	--	--	81.650
User 5	2	2	2	1	1	--	--	30.619
User 6	2	2	1	1	0	0	--	81.650

User 7	0	2	2	1	1	--	--	62.361
User 8	0	2	2	2	--	--	--	57.735
User 9	1	2	1	1	1	--	--	33.333
User 10	1	2	2	1	--	--	--	33.333
CV	111.575	0.000	30.619	34.015	44.721	100.000	DIV0	

CV is calculated using

$$\text{CV} = \frac{\text{Standard Deviation}}{\text{Mean}} * 100$$

CV for the different ranking order is

CV (Preference) = 45.175

CV (t-measure) = 45.922

CV (Preference + t-measure when $\alpha = 0.5$) = 48.093

When preference only considered for ranking, the ranking order is consistent, it treats the items are same. When preference along with t-measure is considered for ranking, the ranking order is varied, it ranks the items are in different orders.

4.5 RELEVANCY SCORE FOR RECOMMENDED QUERIES

The proposed recommendations are evaluated by using an evaluation form. The users are asked to search in one query category. On the evaluation form, the users are asked to give the relevancy score for the recommended queries. For each recommended query, the user had to label it with a relevancy score {0, 1, 2} where 0: irrelevant, 1: partially relevant, and 2: relevant. Table.10 shows the relevancy score for the query 'cricket' and coefficient of variance for every user against their relevancy score. The number of recommended queries is varied and depends on the intent of the user. From Table.10, {R1, R2….R7} indicates the recommended queries. Here R1 is always the favourite query of the user.

When the recommended queries R1 to R7 is considered, Coefficient of variance for the second query R2 has the minimum value 0 because the query R2 contains the relevance score 2 for all the users. The recommended query R1 contains the maximum value 111.6 because the score assigned by the users are different. While the users User 1 to User 10 is considered, the User 5 assigns maximum number of same relevancy score for the recommended queries. Hence the CV for User 5 has the minimum value 30.619. User 4 and User 6 assign different relevancy scores for the queries, the CV for User 4 and User 5 is 81.65. Table.11 lists the Kappa statistic between the User 1 and other users User 2 to User 10.

Table.11. Kappa statistics value

Users	Kappa
User 2	-0.111
User 3	1.000
User 4	1.000
User 5	-0.111
User 6	-0.250
User 7	-0.111
User 8	1.000
User 9	-0.111
User 10	-0.142

The kappa value will be 1 if the two users are always agreeing the recommendations, 0 if they agree only at the rate given by chance, and negative if they are worse than random. From Table.11, the users User 3, User 4 and User 8 are agreed with User 1 on the relevancy score assigned to the recommendations. The other users User 2, User 5, User 6, User 7, User 9 and User 10 do not agree with User 1. In the same way, kappa value for all pairs of users is calculated. Next, the similar users are identified by using Eq.(10) and they are clustered based on the similarity measure.

$$Similarity(User1, User2) = \frac{\sum_{i=0}^{n} Count(i,i)}{\sum_{i=0}^{n}\sum_{j=0}^{n} Count(i,j)} \quad (10)$$

where, n is number of relevancy score.

For example, consider the users 1 and 2. The relevancy score value is 0, 1 and 2. Numbers of occurrences of all possible combinations of the scores are generated. The users 1 and 2 have five recommendations, both assigns the score 2 for the 2 recommendations R2 and R3. Table.12 shows that the relationship between the users 1 and 2 in terms of scores assigned to the recommended queries.

Similarity (User 1, User 2) = (0+1+2) / (0+1+0+0+1+0+0+1+2)

= 3 / 5 = 0.6

Table.12. Relationship between User1 and 2

	User 2				
	Score	0	1	2	Σ
User 1	0	0	1	0	1
	1	0	1	0	1
	2	0	1	2	3
	Σ	0	3	2	5

When the highest similarity 1 is considered as a threshold, the users (1, 8), (2, 10) and (3, 4) are clustered. That is the users 1 and 8 have assigned the same relevancy score.

5. CONCLUSION

The proposed technique recommends and evaluates the queries in Web Information Retrieval. The order of the recommendations is also evaluated. The ranking order is evaluated by using NDCG and kappa statistics value. The measure coefficient of variance is used to find the variations between the ranking orders. The relevancy score assigned by the users to the recommended queries is evaluated using the kappa statistics. Users with similar relevancy score is identified and clustered.

REFERENCES

[1] Bharat Krishna and Andrei Broder, "A technique for measuring the relative size and overlap of public web search

engines", *Journal on Computer Networks and ISDN Systems*, Vol. 30, No. 1-7, pp. 379-388, 1998.

[2] Antonio Gulli and Alessio Signorini, "The indexable web is more than 11.5 billion pages", *Special interest tracks and posters of the 14th International Conference on World Wide Web, ACM*, 2005.

[3] Mark Sanderson, "Ambiguous queries: Test collections need more sense", *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM*, pp. 499-506, 2008.

[4] Silverstein Craig, Monika Henzinger, Hannes Marais and Michael Moricz, "Analysis of a very large altavista query log", *Technical Report 1998-014, Systems Research Center, Compaq Computer Corporation*, 1998.

[5] Hanghang Tong and Christos Faloutsos, "Center-piece subgraphs: problem definition and fast solutions", *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pp. 404-413, 2006.

[6] Boldi Paolo, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis and Sebastiano Vigna, "The query-flow graph: model and applications", *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 609-618, 2008.

[7] Bonchi Francesco, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi and Rossano Venturini, "Efficient query recommendations in the long tail via center-piece subgraphs", *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 345-354, 2012.

[8] Bruno M Fonseca, Paulo Braz Golgher, Edleno Silva de Moura and Nivio Ziviani, "Using association rules to discover search engines related queries", *Proceedings of the First Conference on Latin American Web Congress*, pp. 66-71, 2003.

[9] Baeza-Yates Ricardo, Carlos Hurtado and Marcelo Mendoza, "Query recommendation using query logs in search engines", *Current Trends in Database Technology-EDBT 2004 Workshops*, pp. 588-596, 2005.

[10] Cucerzan Silviu and Ryen W. White, "Query suggestion based on user landing pages", *Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 875-876, 2007.

[11] Zhang Zhiyong and Olfa Nasraoui, "Mining search engine query logs for query recommendation", *Proceedings of the 15th International Conference on World Wide Web*, pp. 1039-1040, 2006.

[12] Jones Rosie, Benjamin Rey, Omid Madani and Wiley Greiner, "Generating query substitutions", *Proceedings of the 15th International Conference on World Wide Web*, pp. 387-396, 2006.

[13] N. Azimi and S. Kiani, "Accelerating the Response of Query in Semantic Web", *International Journal on Computer Network and Information Security*, Vol. 8, pp. 26-33, 2014.

[14] V. Jain and M. Singh, "Ontology development and query retrieval using protege tool", *International Journal of Intelligent Systems and Applications*, Vol. 5, No. 9, pp. 67-75, 2013.

[15] Neelam Dunhan and A. K. Sharma, "Rank Optimization and Query Recommendation in Search Engines using Web Log Mining Techniques", *Journal of Computing*, Vol. 2, No. 12, pp. 97-102, 2010.

[16] Hang LI, "A Short Introduction to Learning to Rank", *Special Section on Information-Based Induction Sciences and Machine Learning, ieice trans. inf. & syst.*, Vol. e94-d, No. 10, 2011.

[17] Ravikumar, Pradeep D, Ambuj Tewari and Eunho Yang, "On NDCG consistency of listwise ranking methods", *International Conference on Artificial Intelligence and Statistics*, pp. 618-626, 2006.

[18] Busa-Fekete R, Gyorgy Szarvas, Tamas Elteto and Balazs Kegl. "An apple-to-apple comparison of Learning-to-rank algorithms in terms of Normalized Discounted Cumulative Gain", *20th European Conference on Artificial Intelligence*, 2012.

[19] Liu Tie-Yan, "Learning to rank for information retrieval", *Foundations and Trends in Information Retrieval*, Vol. 3, No. 3, pp. 225-331, 2009.

[20] Donner Allan, Mohamed M. Shoukri, Neil Klar and Emma Bartfay, "Testing the equality of two dependent kappa statistics", *Statistics in Medicine*, Vol. 19, No. 3, pp. 373-387, 2000.

[21] Manning Christopher D, Prabhakar Raghavan and Hinrich Schutze, "*Introduction to information retrieval*", Cambridge University Press, Vol. 1, 2008.

[22] AOL data set from 2006 -03 - 01 to 2006 – 05 - 31 (zola. di. unipi. it /smalltext/datasets.html).

[23] Umagandhi R and Senthilkumar A V, "Search Query Recommendations using Hybrid User Profile with Query Logs", *International Journal of Computer Applications*, Vol. 80, No. 10, pp. 7-18, 2013.

[24] Pei Jian, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal and Mei-Chun Hsu, "Mining sequential patterns by pattern-growth: The prefixspan approach", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 11, pp. 1424-1440, 2003.

[25] Umagandhi R and Senthilkumar A V, "Time Dependent Approach for Query and URL Recommendations Using Search Engine Query Logs", *International Journal of Computer Science*, Vol. 40, No. 3, 2013.

[26] Umagandhi R and Senthilkumar A V, "Time Heuristics Ranking Approach for Recommended Queries Using Search Engine Query Logs", *Kuwait Journal of Science*, Vol. 41, No. 2, pp. 127-149, 2014.

[27] Melamed I. Dan, Ryan Green and Joseph P. Turian, "Precision and recall of machine translation", *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Vol. 2, pp. 61-63, 2003.