QUERY RECOMMENDATIONS AND ITS EVALUATION IN WEB INFORMATION RETRIEVAL

R. Umagandhi\(^1\) and A.V. Senthil Kumar\(^2\)
\(\text{\textsuperscript{1}}\)Department of Computer Technology, Kongunadu Arts and Science College, India
E-mail: umakongunadu@gmail.com
\(\text{\textsuperscript{2}}\)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 satisfy 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 \((X \rightarrow Y) = \frac{\text{Support}(XUY)}{\text{Support}(X)}\)

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 \( U \); for the input query \( Q \). 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

\[
\begin{bmatrix}
A & B & C & D & E \\
A & 0 & 0 & 0 & 1 \\
B & 0 & 0 & 1 & 1 & 0 \\
C & 0 & 0 & 0 & 0 \\
D & 0 & 0 & 0 & 0 \\
E & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

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.

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

From Fig. 2,

\[\text{Hub}(Q) = \text{Number of out links from } Q\]

\[= \text{Authority (A)} + \text{Authority (B)} + \text{Authority (C)}\]

\[= 3\]

**Hub**

**Authority**

\[A \quad B \quad C \]

**Fig. 2. Multiple Authorities**

\[A \quad B \quad C \]

**Fig. 3. Multiple Hubs**
From Fig. 3:

Authority (Q) = Number of in links to Q
= Hub (A) + Hub (B) + Hub (C)
= 3

\[ t-	ext{measure} = \frac{\text{Cluster number}(u_i) \sum_{j=1}^{n} j}{n} \] (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 fair agreement, and agreement below 0.67 is seen as poor agreement. A kappa value between 0.67 and 0.8 is taken as good agreement, a kappa value above 0.8 is taken as very good agreement, and agreement below 0.4 is seen as poor agreement. A kappa value. As a rule of thumb, a kappa value above 0.8 is seen as poor agreement, a kappa value above 0.8 is seen as very good agreement, and agreement below 0.4 is seen as poor agreement.

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. 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.

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

\[ DCG_N = \sum_{i=1}^{N} \frac{G_i}{\log_2(i+1)} \] (6)

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

\[ NDCG_N = \frac{DCG_N}{IDCG_N} \] (7)

---

**Evaluation of Relevancy Score in Recommended Queries**

---

**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) \( \leq \) 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) \( \geq \)1 and support (query) \( \geq \) 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

<table>
<thead>
<tr>
<th>Anon ID</th>
<th>Query</th>
<th>Query Time</th>
<th>Item Rank</th>
<th>ClickURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>227</td>
<td>psychiatric disorders</td>
<td>2006-03-02 17:30:36</td>
<td>1</td>
<td><a href="http://www.merck.com">http://www.merck.com</a></td>
</tr>
<tr>
<td>227</td>
<td>Cyclothymia</td>
<td>2006-03-02 17:34:08</td>
<td>1</td>
<td><a href="http://www.psycom.net">http://www.psycom.net</a></td>
</tr>
<tr>
<td>647</td>
<td>rabbit hole the broad way play</td>
<td>2006-03-01 22:15:33</td>
<td>2</td>
<td><a href="http://www.10hbc.com">http://www.10hbc.com</a></td>
</tr>
<tr>
<td>1038</td>
<td>tow truck</td>
<td>2006-03-01 23:17:31</td>
<td>No Click</td>
<td>NoRank</td>
</tr>
</tbody>
</table>

Table 2. ModifyPrefixSpan - Frequent Queries

<table>
<thead>
<tr>
<th>ModifyPrefixSpan</th>
<th>ModifyPrefixSpan with Hub and Authority</th>
<th>ModifyPrefixSpan with t-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent Queries with ID</td>
<td>Support</td>
<td>Frequent Queries</td>
</tr>
<tr>
<td><a href="http://www.pokemon.com">www.pokemon.com</a> 18</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td><a href="http://www.gamewinners.com">www.gamewinners.com</a> 17</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>lotto 21</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>mickey dolenz 11</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>cliff notes 9</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>mapquest com 22</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>american spirit tobacco 33</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.pokemon.com">www.pokemon.com</a> 18</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.gamewinners.com">www.gamewinners.com</a> 17</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Precision and Recall for the Frequent Queries

<table>
<thead>
<tr>
<th>ModifyPrefixSpan</th>
<th>ModifyPrefixSpan with Hub and Authority</th>
<th>ModifyPrefixSpan with t-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent Query</td>
<td>Recall</td>
<td>Precision</td>
</tr>
<tr>
<td>18</td>
<td>0.333</td>
<td>1.000</td>
</tr>
<tr>
<td>17</td>
<td>0.667</td>
<td>1.000</td>
</tr>
<tr>
<td>21</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>
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, Table 3 depicts the NDCG for the frequent URLs identified using the algorithms PrefixSpanBasic and MHitsPrefixSpan [25]. The Fig.5 shows that the URLs 21 and 87 got the highest NDCG value.

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).

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.

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, Table 3 depicts the NDCG for the frequent URLs identified using the algorithms PrefixSpanBasic and MHitsPrefixSpan [25]. The Fig.5 shows that the URLs 21 and 87 got the highest NDCG value.
 Preference clusters. Since above day wise activities. The weight t depicts the support, confidence, preference and t-

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 $Q_i$, $1 \leq i \leq 6$ are triggered by the user on Day $j$, $1 \leq j \leq 5$.

Day 1 - $Q_1$, $Q_3$, $Q_4$
Day 2 - $Q_1$, $Q_4$, $Q_5$
Day 3 - $Q_1$, $Q_2$, $Q_3$, $Q_6$
Day 4 - $Q_3$, $Q_4$, $Q_5$
Day 5 - $Q_1$, $Q_2$, $Q_6$

The queries $Q_1$, $Q_3$ and $Q_4$ are issued on Day 1. 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 $Q_1$ occurs on Day 1, 2, 3 and 5, t-measure of $Q_1$ is

\[
1/15 + 2/15 + 3/15 + 5/15 = 0.733
\]

### Table 7. Preference and t-measure

<table>
<thead>
<tr>
<th>Query</th>
<th>Support</th>
<th>Confidence (%)</th>
<th>Preference</th>
<th>t-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1$</td>
<td>4</td>
<td>80</td>
<td>0.525</td>
<td>0.733</td>
</tr>
<tr>
<td>$Q_2$</td>
<td>2</td>
<td>40</td>
<td>0.263</td>
<td>0.533</td>
</tr>
<tr>
<td>$Q_3$</td>
<td>3</td>
<td>60</td>
<td>0.394</td>
<td>0.533</td>
</tr>
<tr>
<td>$Q_4$</td>
<td>3</td>
<td>60</td>
<td>0.394</td>
<td>0.466</td>
</tr>
<tr>
<td>$Q_5$</td>
<td>2</td>
<td>40</td>
<td>0.263</td>
<td>0.4</td>
</tr>
<tr>
<td>$Q_6$</td>
<td>2</td>
<td>40</td>
<td>0.263</td>
<td>0.533</td>
</tr>
</tbody>
</table>

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

\[
Preference (u, q) = \alpha \cdot Day \_Preference (u, q) + \beta \cdot Query \_Preference (u, q)
\]

\[
Preference with - measure = \alpha \cdot preference (u, q) + \beta \cdot t-measure (u, q)
\]

The Table 8 shows the changes in the ranking order according to the $\alpha$ value. For all the cases, irrespective of $\alpha$ and $\beta$ the favourite query of the user is $Q_1$. The queries $Q_2$ and $Q_6$ have equal weight and the query $Q_5$ 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 $Q_3$ and $Q_4$ have the same preference 0.394; hence the rank 2.5 is assigned for $Q_3$ and $Q_4$ instead of 2 and 3 respectively.

### Table 8. Ranking of queries

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>Ranking of queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>$Q_1$, $Q_3$, ($Q_2$, $Q_6$), $Q_4$, $Q_5$</td>
</tr>
<tr>
<td>0.3</td>
<td>0.7</td>
<td>$Q_1$, $Q_3$, ($Q_2$, $Q_6$), $Q_4$, $Q_5$</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>$Q_1$, $Q_3$, $Q_4$, ($Q_2$, $Q_6$), $Q_5$</td>
</tr>
<tr>
<td>0.7</td>
<td>0.3</td>
<td>$Q_1$, $Q_3$, $Q_4$, ($Q_2$, $Q_6$), $Q_5$</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>$Q_1$, $Q_3$, $Q_4$, ($Q_2$, $Q_6$), $Q_5$</td>
</tr>
</tbody>
</table>

### Table 9. Ranking order

<table>
<thead>
<tr>
<th>Original</th>
<th>Preference</th>
<th>t-measure</th>
<th>Preference + t-measure (when $\alpha=0.5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4.5</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2.5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6.5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4.5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4.5</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 10. Relevancy Score

<table>
<thead>
<tr>
<th>Query : Cricket</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>57.143</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>--</td>
<td>55.902</td>
</tr>
<tr>
<td>User 3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>70.711</td>
</tr>
<tr>
<td>User 4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>81.650</td>
</tr>
<tr>
<td>User 5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>30.619</td>
</tr>
<tr>
<td>User 6</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>81.650</td>
<td></td>
</tr>
</tbody>
</table>

996


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.

\[
\text{Similarity}(\text{User}1, \text{User}2) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{n} \text{Count}(i, j)}{\sum_{i=0}^{n} \sum_{j=0}^{n} \text{Count}(i, j)}
\]

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.

\[
\text{Similarity} (\text{User}1, \text{User}2) = (0+1+2) / (0+1+0+0+1+0+0+1+2) = 3 / 5 = 0.6
\]

Table.12. Relationship between User1 and 2

<table>
<thead>
<tr>
<th>User 1</th>
<th>Score</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Σ</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

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


[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.

