PERSONALIZED USER AND QUERY DEPENDENT RANKING FOR WEB DATABASES

Mrs. M. Priya  
Associate Professor, Dept of Computer Science & Engineering, 
Bharathiyar College of Engineering & Technology, Karaikal.

P. Mangai  
PG Scholar, Dept of Computer Science & Engineering, 
Bharathiyar College of Engineering & Technology, Karaikal.

ABSTRACT

With the blooming of the Web, Querying the Web databases to get the needed results has become a routine task. These emergences of the Web have given a new idea to the concept of ranking the results of a Query given by the User. Former techniques used for ranking include analyzing frequencies of database values and Query logs or establishing User Profiles.

Yet the User and Query independent approach followed there has some limitations. To overcome the limitations of User and Query independent approach, a ranking model is proposed based on User and Query dependent approach. The assumptions behind the proposed model is i) Similar Users prefer comparable ranking preferences for the results of same Queries and ii) Same User prefers comparable ranking preferences for the results of different Queries.

Keywords: Automated Ranking, Query Similarity, User Similarity, Web Databases.

1. INTRODUCTION

The growth of the internet and Web leads to the development of a large number of Web Databases for a variety of applications. Automated ranking of the results of a Query is a popular aspect of the Query model in Information Retrieval. In contrast, Database systems support only a Boolean Query Retrieval model.

It often leads to the scenarios that when the Query is too selective, the answer may be empty and when the Query is not too selective, too many tuples may be in the answer. Such a case would be inappropriate for the User to choose the exact answer. To handle such situations, Web Databases
sorts the Query result on the values of a single attribute. However the ordering based on multiple attribute values would be closer to the Web User’s expectation.

Consider the following two scenarios.

**EG 1:** Two Users, a Software Firm Executive (U1) and a Manager of a Data entry Firm (U2), seeks answer to the same Query (Q1): “Working Area = Computer AND Location = Delhi”. In response to this Query, consider nineteen thousand tuples are generated. By assumption, U1 would search for candidates with Programming skills in particular language, say JAVA and hence would prefer candidates with “Condition = Programmer AND language = JAVA” to be ranked higher than the others. In contrast, U2 would search for Data entry operators with maximum speed, hence for U2 candidates with “Condition = Data entry operator AND Qualification = Plus Two “should be ranked higher than the rest.

**EG 2:** The same User (U2) moves to do some transcription work and issues a different Query (Q2).”Working area = Medical filed AND Location = Mumbai”. Now consider he prefer candidates with “Condition = Data entry operator and Qualification = Degree” to be ranked higher than the others.

EG 1 depicts, towards the results of the same Query, different Web Users may have contrasting ranking preferences. EG 2 depicts, the same User may prefer different ranking order for the results of different Queries.

Automated ranking of Web Database provides a single ranking order for the given Query across all the Users. In contrast ranking using the User profile techniques, do not differentiate the Users, hence provide a single ranking order for any Query given by the same User. This paper introduces an application model based on User and Query dependent approach for ranking the results of Web Database Queries. The assumption behind the proposed model is i) Similar Users prefer comparable ranking preferences over the results of same Query and ii) Same User prefers comparable ranking preferences over the results of different Queries. The model has two functionalities i) Query Similarity and ii) User Similarity. The former analyzes the similarities between the Queries on their attribute values, whereas the latter analyzes the similarity among the Users issuing similar Queries.

2. RELATED WORK

In Document Retrieval systems, ranking functions are often based on the frequency of occurrence of Query values in documents. Information Retrieval system contains documents that often satisfy a given keyword based Query. However when the returned tuples are too many, it is cumbersome for the User to choose the needed results. Approaches to overcome this problem range from Query reformulation techniques in which the User is prompted to refine the Query to make it more selective, to automatic ranking of the Query results by their degree of relevance to the Query and returning only the top K subset. It is evident that automated ranking can have compelling applications in the Database context.

However in the Database context, especially in the case of categorical data, term frequency is irrelevant as tuples either contain or do not contain a Query value. Hence ranking functions need to also consider values of unspecified attributes. Hence Probabilistic ranking of Database Query results proposed the ranking function based on two factors: a global score which captures the global importance of unspecified attribute values, and a conditional score which captures the strengths of dependencies between specified and unspecified attribute values. Yet the quality ranking cannot be achieved. For a given Query it provides the same ordering of tuples across all Users. Employing User Personalization does not consider that the same User may have varied ranking preferences for different Queries.
The ranking process in Relational Database involves manual specification of the ranking preferences thereby insisting the Users to possess knowledge about Query languages and ranking functions. Collaborative filtering is a method of making automatic predictions about the interests of a User by collection preferences from many Users. Here ranking is done based on the Similarity between the Users preferences for a particular item. A major drawback of this ranking method is that they do not consider the same User may prefer varied ranking for different Queries.

The proposed framework provides an automated Query and User dependent ranking by considering the two facts that i) Similar Users prefer different ranking preferences for a given Query and ii) Same User prefers different ranking preferences for different Queries.

3. RANKING ARCHITECTURE

The Ranking problem can be defined as “For a Query Q given by the User, determine a ranking order by assigning a score to every tuple”. The Ranking model (shown in Fig 1) consists of the following components: i) Query Similarity ii) User Similarity and iii) Ranking Process.

When a User issues a Query, the Query Similarity function determines the set of Queries that are more similar to the Query given by the User.

![Fig 1. Ranking Model](image)

User Similarity function identifies the set of Users more similar to the User who issued the Query. These ordered sets of Similar Queries and Similar Users are used to rank the results by using Weighted Page Rank algorithm, where the attribute weight and value weight are calculated as tuple score for every tuple.

3.1 Query Similarity

Query Similarity analyzes the similarity between the Queries by considering the attribute values. For illustration, consider a User U1, issues a Query Q1 for which certain ranking order exists. Now another User U2, issues a Query Q2 which is more similar to the Query Q1, already issued by the User U1. In such a case the same ranking order can be derived for the Query Q2 given by the User U2.

Query similarity can be theoretically explained as: Given two Queries Qx and Qy, each with conjunctive selection conditions of the form “WHERE Y1=y1 AND · · · AND Ys=ys” and
‘WHERE Y1=y1’...AND Ys=ys’ respectively. The similarity between Qx and Qy is represented as the product of the similarities between the values yi and yi’ for every attribute Yi, and is shown as

\[ S_{similarity}(Qx, Qy) = \prod_{i=1}^{n} sim(Yi = yi, Yi = yi') \] (1)

The similarity between any two values v1 and v2 for an attribute is defined as: For two Queries Qv1 and Qv2 with condition “WHERE Ai=v1” and “WHERE Ai=v2”, consider Nv1 and Nv2 are the set of returned results from Database. The similarity between the two values is given as the similarity between Nv1 and Nv2 and is determined as the average of all similarity values between the tuples T.

\[ sim(Nv1, Nv2) = \sum_{i} sim(T_i, Nv2) \]

By substituting this in eqn (1) Query similarity is obtained.

3.2 User Similarity

User Similarity analyzes the similar Users issuing the most similar Queries. For illustration, consider a User U1 issuing a Query Q1 is more similar to User U2 who already have the ranking order. Then for the results of Query Q1 issued by the User U1, both the Users will display similar ranking order.

Instead of determining a single similar User, a set of top K similar Users can be determined to improve the ranking quality. While determining similar Users, if Query independent approach is followed then the same ordering will be displayed for all the Queries by a particular User.

To overcome such a limitation, the proposed framework uses Query dependent approach while determining the similar Users as Same User will display different ranking preferences for different Queries. Though the User similarity concept exists in collaborative and content filtering, the technique for examining the similarity varies. Existing system provides static approach. The assumption followed there is that a User behavior does not change across all items/objects. Whereas in proposed work it is assumed that Same User may exhibit different ranking order for different Queries. For a given Query Q by a User U, only the top most similar Queries to Q are selected by assuming that a User may possess dynamic preferences.

3.3 Ranking Process

The ordered sets of similar Queries and similar Users obtained from Query Similarity function and User Similarity function respectively are used to rank the results of Web Databases. Weighted Page Rank (WPR) algorithm is used for ranking the results.

WPR is an extension of the standard Page Rank algorithm. It takes into account the importance of both the in-links and the out-links of the pages and distributes the rank scores based on the popularity of the pages. Popularity of a Web page is decided by observing its in-links and out-links.

WPR assigns a larger rank values to the more important pages rather than separating the rank value of a page evenly among its outgoing linked pages. Each outgoing link gets a value proportional to its consequence. The importance is assigned in terms of weight values to the incoming and outgoing links.

The popularity from the number of in-links and out-links is recorded as \( Win(v,u) \) and \( Wout(v,u) \) respectively. \( Win(v,u) \) is the weight of link(v,u) calculated based on the number of in-links of page v.

\[ Win(v,u) = \frac{Iu}{\sum Ip} \]
where $I_u$ and $I_p$ represents the number of in-links of page $u$ and page $p$ respectively. $R(v)$ denotes the reference page list of page $v$.

$W_{out}(v,u)$ is the weight of link$(v,u)$ calculated based on the number of out-links of page $u$ and the number of out-links of all reference pages of page $v$.

$$W_{out}(v,u) = \frac{O_u}{\sum_{p \in R(v)} O_p}$$

where $O_u$ and $O_p$ represents the number of out-links of page $u$ and page $p$ respectively. $R(v)$ denotes the reference page list of page $v$. Page Rank is calculated using the following formula.

$$PR(v) = (1-d) + d \sum_{v \in B(u)} PR(v) \times Win(v,u) \times W_{out}(v,u)$$

The relevancy of a page to a given query depends on its group and its location and position in the page list. The resultant pages of a query are classified into four categories based on their relevancy to the given query. They are Very Relevant Pages, pages containing very important information allied to a given query. Relevant Pages, though pages are relevant but do not contain any information about a given query. Weak Relevant Pages, pages may have the query keywords but don’t have the appropriate information and Irrelevant Pages, pages without relevant information and query keywords. WPR is able to identify large a large number of relevant pages to a given query compared to standard PageRank algorithm.

4. CONCLUSION

A Query and User dependent approach for ranking the results of Web Databases has been proposed. Based on the above said approach, a ranking architecture has been presented with three major components, Query Similarity, User Similarity and Ranking Process. Weighted Page Rank algorithm has been applied to rank the results of the Query. WPR is able to identify a larger number of relevant pages to a given Query compared to standard PageRank algorithm. The notion of User Similarity proposed in this work with existing User profiles can be combined to analyze if ranking quality can be improved further.

REFERENCES

23. Suresh Subramanian and Dr. Sivaprakasam, “Genetic Algorithm With A Ranked Based Objective Function and Inverse Index Representation For Web Data Mining’’ International journal of Computer Engineering & Technology (IJCET), Volume 4, Issue 5, 2013, pp. 84 - 90, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.