IJDCST @Jan-2015, Issue- V-3, I-2, SW-01
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Dynamic Query Forms an Integrated Mining Framework

Chakka Lakshmi Tulasi[1], D.HariKrishna[2], Dr. K. Rama Krishnaiah[3]

[1] Dept. of CSE, Nova College of Engineerng & Technology,Vijayawada,AP,India.

[2]Assistant Professor, Nova College of Engineerng & Technology,Vijayawada,AP,India.

[3] Professor & Principal, NVR College of Engineering and Technology, Tenali,AP,India.

ABSTRACT

Modern applications are based on huge very huge data. Working with such a vast data requires more efforts in forming queries for analyzing the data, this being the major issue in the existing systems implemented based on static predefined queries in the present day. Even though there are fixed number of queries being in a finite loop, identifying the best query among the listed queries will again be a big challenge to the users as it evidences large number of queries. Apart from these, one more major problem is of refining the query for the accurate results based on the user inputs. The current proposal of approach drives a solution to these problems prioritizing more user friendly format by allowing user to perform all the operations on the data. The system representing Dynamic Query Form will allow the user to interact with the data source with a dynamic interface by giving them a freedom to form the customized queries for analyzing the data. The next key approach is of ranking method based on the user preferences which will make the frequent queries available on the user profile. The resultant approach leads in more reliable and efficient way of handling large and complex database schemas with great quality by prioritizing user satisfaction.

Key words: DQF- *Dynamic Query Form,* QBE: *Query-By-Example,* DDM- *Data Driven Model,* Key word search, Dynamic Faceted Search.

2. INTRODUCTION

Query structure is a standout amongst the most broadly utilized client interfaces for questioning databases. Customary inquiry structures are composed and predefined by developers or DBA in different data administration frameworks. With the quick improvement of web data and logical databases, present day databases get to be huge and complex. Web crawlers have turned into the fundamental section point to web substance and a vast piece of the \visible" Web comprises in what is displayed by them as top recovered results. In this manner, it would be attractive if the initial couple of results [1].

We catch client inclination utilizing both chronicled inquiries and run-time input, for example, navigate. Test results demonstrate that the element approach frequently prompts higher achievement rate and less complex inquiry structures contrasted and a static methodology. The positioning of structure segments likewise makes it less demanding for clients to alter question structures.

3 REVIEWS ON LITERATURE

3.1 Similarity measures for categorical data.

By utilizing Gianluca Departing Every day news about questionable themes are distributed on the web. In addition, individuals talk about their thoughts and assessments in online journals and social sites. As the measure of Web substance is quickly developing [4], web search tools have turned into a vital instrument for clients to discover data. For the same reason, the quantity of pages which are pertinent to a question is becoming, constraining clients to trust the web crawler in what it introduces them. In this way, it may happen that rankings focused around fame of pages (e.g., Page Rank), on topical significance, and even point assorted qualities are inclined towards a certain notion.

3.2 Probabilistic information retrieval approach for ranking of database query results.

Our work brings forth several intriguing open problems [6]. For example, many relational databases contain text columns in addition to numeric and categorical columns. It would be interesting to see whether correlations between text and non-text data can be leveraged in a meaningful way for ranking. Secondly, rather than just query strings present in the workload, can more comprehensive user interactions be leveraged in ranking algorithms [6], tracking the actual tuples that the users selected in response to query results? Finally, a comprehensive quality benchmarks for database ranking need to be established.

3.3 Combining keyword search and forms for ad hoc querying of databases.

We examine the methodology of utilizing essential word inquiry to lead clients to structures for impromptu questioning of databases. We consider various issues that emerge in the usage for this methodology [8]: outlining and producing structures in an efficient manner, taking care of watchword inquiries that are a mix of information terms and diagram terms, separating out structures that would deliver no results concerning a client's question, and positioning and showing structures in a manner that help clients discover helpful structures all the more rapidly [8].

Fig 1: Custom search query, filtering image search.

4. EXISTING SYSTEM

A considerable measure of examination works focus on database interfaces which support customers to question the social database without SQL. QBE (Query-By-Example) and Query Form are two most extensively used database addressing interfaces [10]. At present, request structures have been utilized as a part of most genuine business or exploratory information schemas. Current studies and works basically focus on the best way to make the inquiry structures.

Customized Query Form: Existing database customers and devices attempt extraordinary endeavors to help engineers plan and create the inquiry structures, for example, Easy Query, Cold Fusion, SAP, Microsoft Access and so on..,

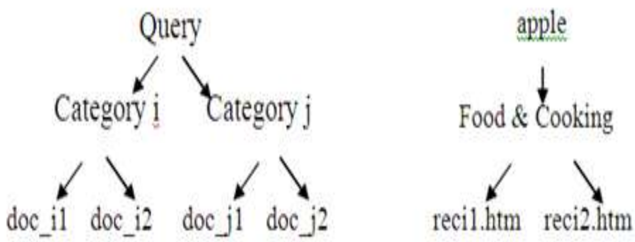

Figure 2: Example for search results.

Automatic Static Query Form: As of late, proposed programmed methodologies to produce the database question structures without client interest. Exhibited information driven strategy. It first discovers a set of information traits, which are undoubtedly questioned focused around the database mapping and information occasions.

Auto completion for Database Queries: In novel client interfaces have been produced to help the client to sort the database inquiries focused around the inquiry workload, the information circulation and the database schemas.

Query Refinement: Question refinement is a typical pragmatic method utilized by most data recovery frameworks.

Dynamic Faceted Search: Dynamic faceted pursuit is a kind of web indexes where applicable realities are introduced for the clients as per their route ways.

PROBLEM STATEMENT

Advanced experimental databases and web databases keep up huge and heterogeneous information. These genuine databases contain over hundreds or even a large number of relations and properties. Customary predefined inquiry structures are not ready to fulfill different specially appointed questions from clients on those databases. With the quick improvement of web data and exploratory databases, current databases get to be substantial and complex. It is hard to plan a set of static question structures to fulfill different specially appointed database questions [10] on those complex databases. Query to let clients make redid inquiries on databases. In the event that a client is not acquainted with the database outline ahead of time, those hundreds or a great many information traits would befuddle the defined client collaborations in web databases. So the better framework was presupposed amid this necessity effectively [10].

4 PROPOSED SYSTEM

We give techniques to conclude a set of applicable classifications for every client question focused around the recovery history of the client. The set of classes can be concluded utilizing the client's profile just, or utilizing the general profile just or utilizing both profiles.

We make the accompanying examinations:

(a) The exactness of joining the client profile and the general profile versus that of utilizing the client profile just.

(b) The exactness of joining the client profile and the general profile versus that of utilizing the general profile just.

The issue is to customize web look. We present a methodology in this area. Initially, we propose a tree model to speak to a client's hunt history and portray how a client's inquiry history can be gathered without his/her steer contribution.We utilize lattices to speak to client look histories and client profiles. Figure 5 demonstrates a sample of the grid representations of a quest history and a profile for a specific client, who is intrigued by the classes "COOKING" and "SOCCER". Which is built from the client inquiries and the applicable archives? (In the accompanying discourse, we utilize "archives" to mean both questions and applicable reports in the grids and). Is a report class framework, A client profile is spoken to by a class term framework. In this illustration, D1, D2, … are archives; lowercase words, for example, "football", "fruit",… are terms; uppercase words, for example, "SOCCER", "COOKING", … are categories.

Doc\Term	apple	recipe	pudding	football	soccer	fifa
D1	1	0	0	0	0	0
D2	0.58	0.58	0.58	0	0	0
D3	0	0	0	1	0	0
D4	0	0	0	0.58	0.58	0.58

(a) Document-Term matrix DT

Doc\Category	COOKING	SOCCER
D1	1	0
D2	1	0
D3	0	1
D4	0	1

(b) Document-Category matrix DC

Cate\Term	apple	recipe	pudding	football	soccer	fifa
COOKING	1	0.37	0.37	0	0	0
SOCCER	0	0	0	1	0.37	0.37

(c) Category-Term matrix M represents a user profile

Figure 3: Matrix Representations of the processing states.

Network is developed from the questions (the root hubs in the tree model) and their applicable archives (the leaf hubs in the tree model) in the client's inquiry records. is the quantity of archives in a client's pursuit history and is the quantity of different terms happening in these reports. Each one question or report is a line vector of weighted terms in

We propose a Dynamic Query Form framework: DQF, a question interface which is fit for progressively creating inquiry structures for clients. It begins with an essential question structure which contains not very many essential properties of the database. Not quite the same as customary record recovery, clients in database recovery are frequently eager to perform numerous rounds of activities (i.e., refining question conditions) before recognizing the last applicants. User capturing is the main task in Dynamic Query Form for providing efficiency of the query formation. So the better system was required for capturing those results efficiently.

ALGORITHMS TO LEARN PROFILES

Taking in a client profile framework from the client's inquiry history (lattices and) and mapping client questions to classes can be seen as a particular multi-class content classification assignment. We portray four calculations to take in a client profile:

Relevance(q)
Input:
1) the query fusion graph, $\mathcal{QFG}$
2) the jump vector, g
3) the damping factor, d
4) the total number of random walks, $numRWs$
5) the size of neighborhood, $maxHops$
6) the given query, q

Output: the fusion relevance vector for q, $\mathbf{rel}_q^F$

Initialize $\mathbf{rel}_q^F = \mathbf{0}$
$numWalks = 0$; $numVisits = 0$
while $numWalks < numRWs$
  $numHops = 0$; $v = q$
  while $v \neq NULL \wedge numHops < maxHops$
    $numHops{+}{+}$
    $\mathbf{rel}_q^F(v){+}{+}$; $numVisits{+}{+}$
    $v = SelectNextNodeToVisit(v)$
  $numWalks{+}{+}$
For each v, normalize $\mathbf{rel}_q^F(v) = \mathbf{rel}_q^F(v)/numVisits$

Algorithm for calculating the query relevance by simulating random walks over the query fusion graph.

SelectNextNodeToVisit(v)
Input:
1) the query fusion graph, $\mathcal{QFG}$
2) the jump vector, g
3) the damping factor, d
4) the current node, v

Output: the next node to visit, q_i

if $random() < d$
  $V = \{q_i \mid (v, q_i) \in \mathcal{E}_{\mathcal{QF}}\}$
  pick a node $q_i \in V$ with probability $w_f(v, q_i)$
else
  $V = \{q_i \mid g(q_i) > 0\}$
  pick a node $q_i \in V$ with probability $g(q_i)$
return q_i

Algorithm for selecting the next node to visit.

Figure 4: Representation of query fusion process with contributions.

Similarities using prior approaches can be inefficient for large amount of search records. Our efficient in both computation and storage, and is adaptive. The following formula is used:

IJDCST @Jan, Issue- V-3, I-2, SW-01
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Algorithm Scope: Query Construction

Algorithm Name: Ranking Algorithm
Input Arguments: Query Id, Date of Ranking, Rank, Usage Frequency
Output Arguments: List of Queries based on the criteria's
Algorithm:
Do
{
While (Query Id==Null)
{
If (Rank ==Maximum && Usage Frequency==Very High)
{
Add the query to the wish list;
}
Query Id--;
}
Usage Frequency--;
Rank--;
}
While (Rank ==least && Usage Frequency==low);

$$M(i,j)^t = \frac{N_i^{t-1}}{N_i^t} M(i,j)^{t-1} + \frac{1}{N_i^t}\sum_k DT(k,j) * DC(k,i)$$

Combining methods and compare them with the above two baseline cases. Let c^u and c^g be the category vectors for the user profile and the general profile respectively. The following computation is done for every category.

(a) Use only the user profile: $Sim(q,c) = Sim(q,c^u)$.

(b) Use only the general profile: $Sim(q,c) = Sim(q,c^g)$.

(c) Combining Method 1:

$$Sim(q,c) = (Sim(q,c^u) + Sim(q,c^g))/2 .$$

(d) Combining Method 2:

$$Sim(q,c) = 1-(1-Sim(q,c^u)) * (1-Sim(q,c^g)) .$$

(e) Combining Method 3:

$$Sim(q,c) = \max(Sim(q,c^u), Sim(q,c^g)) .$$

The classifications are positioned in slipping request of the consolidated likenesses, i.e. Sim(q, c), and the main 3 classifications are decided to reflect the client's pursuit aim. The reason that it is sufficient to utilize the main 3 classes just is that, for a given question, most clients are intrigued by stand out or two classifications in the two-level class chain of importance. A user queries a search engine tries to construct user profile based on his Ip address from its user search history repositories. If the user already exists, the search engine checks from its user search history repositories up to a certain threshold whether the user already queried the same query previously.

Figure 5: Algorithm for calculating the query construction by simulating random walks over the Ranking method.

If the user did, then search engine further retrieves click points from user search history repositories and reformulates query by generating click graphs. Click graphs contain useful information on user behavior when searching online. This step is called query fusion graph. Uses random walk propagation over the query fusion graph instead of time-based and keyword similarity based approaches. This entire process is called organizing user search histories into query groups.

PERFORMANCE ANALYSIS

Trial Setup: We think about the conduct and execution of our calculations on parceling a client's inquiry history into one or more gatherings of related questions. Case in point, for the succession of inquiries "Caribbean voyage"; "bank of America"; "convenient"; "budgetary explanation", we would expect two yield parts: to start with, {"caribbean journey", "expedia"} relating to travel-related questions, and, second, {"bank of America", "monetary statement"} relating to cash related questions.

Utilizing Search Logs: our query gathering calculation depends vigorously on the utilization of hunt logs in two routes: initially, to build the question combination chart utilized as a part of figuring inquiry significance, and, second, to grow the set of inquiries considered when processing inquiry importance. We begin our exploratory assessment, by researching how we can make the most out of the pursuit logs.

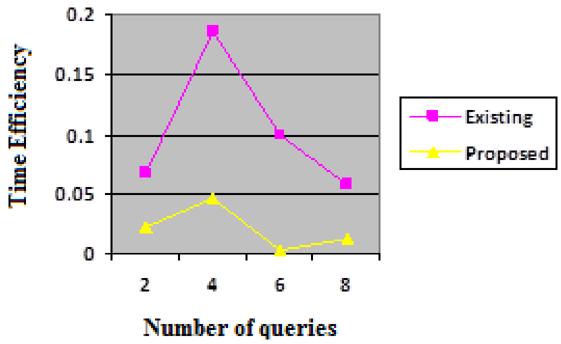

Fig 6: Varying query results in both existing and proposed approaches.

CONCLUSION:

In this paper we propose a dynamic query form generation approach which helps users dynamically generate query forms. The key idea is to use a probabilistic model to rank form components based on user preferences. We capture user preference using both historical queries and run-time feedback such as click- through. Experimental results show that the dynamic approach often leads to higher success rate and simpler query forms compared with a static approach. The ranking of form components also makes it easier for users to customize query forms. As future work, we will study how our approach can be extended to non relational data. As for the future work, we plan to develop multiple methods to capture the user's interest for the queries besides the click feedback. For instance, we can add a text-box for users to input some keywords queries. The relevance score between the keywords and the query form can be incorporated into the ranking of form components at each step.

REFFERENCES: [1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of VLDB*, pages 81–92, Berlin, Germany, September 2003.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of WSDM*, pages 5–14, Barcelona, Spain, February 2009.

[3] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In *CIDR*, 2003.

[4] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of SIAM International Conference on Data Mining (SDM 2008)*, pages 243–254, Atlanta, Georgia, USA, April 2008.

[5] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis. Query recommendations for interactive database exploration. In *Proceedings of SSDBM*, pages 3–18, New Orleans, LA, USA, June 2009.

[6] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Trans. Database Syst. (TODS)*, 31(3):1134– 1168, 2006.

[7] K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh. Usher: Improving data quality with

ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

dynamic forms. In *Proceedings of ICDE conference*, pages 321–332, Long Beach, California, USA, March 2010.

[8] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton. Combining keyword search and forms for ad hoc querying of databases. In *Proceedings of ACM SIGMOD Conference*, pages 349–360, Providence, Rhode Island, USA, June 2009.

[9] S. Cohen-Boulakia, O. Biton, S. Davidson, and C. Froidevaux. Bioguidesrs: querying multiple sources with a user-centric perspective. *Bioinformatics*, 23(10):1301–1303, 2007.

[10] Dynamic Query Forms for Database Queries Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen

URLS

[1] Cold Fusion. http://www.adobe.com/products/coldfusion/.

[2] DBPedia. http://DBPedia.org.

[3] EasyQuery. http://devtools.korzh.com/eq/dotnet/.

[4] Freebase. http://www.freebase.com.

[5] Framework for clustering. http://ieeexplore.ieee.org/

[6] probabilistic information retrieval. http://scholarworks.umass.edu/

About Authors

Ch. Lakshmi Tulasi pursuing M.tech in Nova College of Engineering & Technology, My Interest in Data Mining.

Mr. Hari Krishna.Deevi is a qualified persion Holding M.Sc(CSE) & M.Tech Degree in CSE from Acharya Nagarjuna university, He is an Outstanding Administrator & Coordinator. He is working as an Assistant Professor in NOVA College of Engineering Technology .He guided students in doing IBM projects at NOVA ENGINEERING College. Who has Published 10 research Papers in various international Journals and workshops with his incredible work to gain the knowledge for feature errands.

Dr. K. Rama Krishnaiah is a highly qualified person, an efficient and eminent academician. He is an outstanding administrator; a prolific researcher published 33 research papers in various International Journals and a forward looking educationist. He worked in prestigious K L University for 11.5 years and he contributed his service for NBA accreditation in May 2004, Aug 2007 with ‘record rating’, ISO 9001:2000 in 2004, Autonomous status in 2006, NAAC accreditation of UGC in 2008 and University status in 2009. Later on he worked as Principal at Nova College of Engineering and Technology, Vijayawada for a period of 3.5Yrs. He took charge as the Principal, NVR College of Engineering and Technology, Tenali in May 2014.