International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

# Literature Review on Automated Database Searching Methods

**Prashant S. Chavan[1], Prof. Dr. B. D. Phulpagar[2]**

[1]Postgraduate Research Scholar, PESMCOE, Pune, India
[2]Department of Computer engineering, PESMCOE, Pune, India

**Abstract:** ***Now a days, Database searching from the database is time critical task. Modern database contain enormous data and also it may be heterogeneous. From these types of databases, getting user desired instances is tedious task. To get the satisfactory result multiple refining of the attributes is necessary. Use of traditional predefined query interfaces in this type of databases does not give satisfactory results. To meet the need of aforementioned, many methods and approaches have been proposed. Present paper reviews these research methods.***

**Keywords:** query results, database

## 1. Introduction

A database is only as useful as its query interface allows it to be. If a user is unable to convey to the database what he or she wants from it, even the richest data store provides little or no value. Static Query forms or predefined query forms are used by the DBA to retrieve the information from the database. But current used databases contain number of attributes and relations. So, retrieving information with static query forms is difficult. Also it is not possible to design static query form with too many attributes to handle. Many database management tools provide mechanisms to design predefined query forms. The process is complex because user must manually edit to design predefined query forms. If a user is unaware of the database schema then handling attributes in the process of designing predefined query forms becomes too complex to handle. To propose a solution for the problem many researchers have attempted various approaches. Every approach has some pros and cons. This paper takes into account the allied research in database searching methodologies.

## 2. Related Work

**Automated Ranking of Database Query Results [1]:**

Ranking of query results and returning it accordingly is very popular paradigm in database searching and information retrieval. In the paper titled „automated ranking of database query result" Sanjay Agarwal et al. (Microsoft Research) analysed various approaches of ranking of the query results.
Automated ranking of the results of a query is a popular aspect of the query model in Information Retrieval (IR). Mostly all the information retrieval systems work on Structured Query Languages. SQL returns all the candidates selected by the query. But in many cases results are so many that user may face "many answer" problem.

Adding the filters like WHERE and LIKE clauses in SQL we can achieve refined results. But, if the query is too selective it may pose "empty answer" problem [9][10]. Thus it is needed that rather than cutting the results out ranking of the results may solve the problem. But when query is not too selective many tuples may be in the answer and user will not be satisfied with the result. In this scenario, we can rank instances according to the user need. This type of approach very vital in the searching of product catalogue or similar type of systems.

To implement this type of system priorities to the query results should be given. Here many questions may arise:

- How to assign priorities?
- What actors to be considered for ranking?
- How to execute it efficiently over large databases?
- How to get away problem of many answer and empty answers?

In the paper authors used TF-IDF idea that is based on frequency of the occurrence of the attribute values in the database. But database contains mixed information like numerical and categorical. They have extended TF-IDF concept to include numerical values also and created IDF Similarity, a database positioning capacity that stretches out TF-IDF ideas to databases containing a heterogeneous blend of all out and additionally numeric information. While IDF Similarity functions admirably for some database positioning applications, some of the time its adequacy is very constrained. In specific examples the importance of information qualities for positioning may be because of different elements notwithstanding their frequencies [11]. This has been noted in the IR area also, where here and there one needs to go past TF-IDF weightings to infer precise positioning capacities. This makes one wonder: what else could be the premise of non specific positioning in databases? In the said paper they demonstrated that gathering the workload on the database can be entirely helpful for positioning. As it were, this may be seen as a poor man's decision of pertinence criticism and collective separating where a client's last decision of applicable tuples is not recorded. Regardless of its primitive nature, such workload data can assist focus the recurrence with which database qualities and qualities are referenced. At the point when utilized as a part of conjunction with IDF, workload data helps positioning quality. They create QF Similarity, a positioning capacity that influences such workload data. A significant part of the examination in this paper concentrates

on the empty answers issue. Taking care of the numerous answers issue postures extra difficulties in light of the fact that a positioning capacity that just relies on upon the conditions in the client inquiry is lacking for this issue. They have augmented positioning capacities with extra question free parts that measure the "significance" of tuples in a worldwide sense. At last, regardless of the possibility that we get the positioning capacities right, for extensive databases, we need to minimize their effect on question handling. Albeit upset records are prominent information structures for proficient recovery in IR, they are deficient for our reasons as we look for uncertain matches including clear cut and numerical qualities. We contemplate uses of some late calculations for Top-K inquiry preparing, which drives us to yet another commitment of this paper; a record based Top-K question handling calculation, ITA that endeavours our positioning capacities. We have manufactured a framework in which our positioning calculations have been actualized on a social DBMS. The framework has two noteworthy segments, a pre-handling segment and an inquiry preparing segment. The pre-preparing part is a positioning capacity extractor that influences information and workload attributes. The question preparing segment is a Top-K calculation that uses the positioning capacity and endeavours the physical database plan. We have performed client investigates our framework to assess its viability. Be that as it may, in spite of our earnest attempts, our client investigations are preparatory. Dissimilar to IR which depends on broad accessible client ponders and benchmarks, no framework is accessible today to evaluate database positioning.

**Similarity Measures for Categorical Data [2]**

Measuring closeness or separation between two elements is a key stride for a few information mining and learning discovery assignments. The thought of likeness for ceaseless information is generally surely known, yet for straight out information, the comparability calculation is not clear. Several information driven comparability measures have been proposed in the writing to register the similitude between two straight out information occurrences however their relative execution has not been assessed. In this paper we ponder the execution of a mixed bag of comparability measures in the setting of a specific information mining assignment: anomaly detection [8]. Results on a mixed bag of information sets demonstrate that while nobody measure overwhelms others for a wide range of problems, a few measures have the capacity to have reliably high execution.

Measuring similitude or separation between two information focuses is a center prerequisite for a few information mining and information disclosure assignments that include distance calculation. Cases incorporate bunching (k-means), separation based exception location, classification (knn, SVM), and a few other information mining assignments. These calculations normally regard the similitude calculation as an orthogonal step and can make utilization of any measure. For constant information sets, the Minkowski Distance is a general strategy used to figure separation between two multivariate focuses. Specifically, the Minkowski Separation of request 1 (Manhattan) and request 2 (Euclidean) are the two most generally utilized separation measures for persistent information. The key perception about the above measures is that they are free of the basic information set to which the two focuses have a place. A few information driven measures, for example, Mahalanobis Distance, have additionally been investigated for ceaseless information. The idea of similitude or separation for straight out information is not as direct concerning nonstop information. The key normal for straight out information is that the different values that an unmitigated property takes are not inalienably requested. Along these lines, it is unrealistic to straightforwardly think about two different unmitigated qualities. The least difficult approach to find closeness between two clear cut ascribes is to allocate a likeness of 1 if the qualities are indistinguishable and a closeness of 0 if the qualities are not indistinguishable. For two multivariate clear cut information focuses, the closeness between them will be straightforwardly relative to the quantity of properties in which they match. This basic measure is otherwise called the cover measure in the writing.

One clear downside of the cover measure is that it doesn't recognize the different values taken by a quality. All matches, and in addition mismatches, are dealt with as equivalent.

**Expressive Query Specification through Form Customization [3]**

A structure based inquiry interface is generally the favored intends to provide an unsophisticated client access to a database [16]. Not just is such an interface simple to utilize, obliging no specialized training, however it moreover it requires next to zero learning of how the information is organized in the database. Be that as it may, form structure is static and can express just a exceptionally restricted arrangement of queries i.e. form is static or fixed. Without space for change, question specification is restricted by the ability and vision of the interface developer at the time the structure was made. On the off chance that an accessible structure can't express a wanted question, then user is stuck.

In this paper, authors proposed a mechanism to let a client change an existing structure to express the wanted question. These modifications are themselves specified through filling forms to make an expression in a fundamental form language defined[15]. The specialized complexity needed to change structures is very little more noteworthy than structure filling. They have built up a form editor that executes this structure manipulation language. They have additionally built up a query generator that modifies the form's original query taking into account a client's progressions.

**Dynamic Generation of Query-Dependent Faceted Interfaces for Wikipedia [4]**

This paper proposes Facetedpedia, a faceted recovery framework for data revelation and investigation in Wikipedia. Given the set of Wikipedia articles coming about because of a pivotal word inquiry, Facetedpedia creates a faceted interface for exploring the outcome articles. Compared with other faceted recovery frameworks, Facetedpedia is completely programmed and dynamic in both aspect era and chain of importance construction, and the aspects depend on the rich semantic data from Wikipedia. The substance of our methodology is to expand upon the cooperative vocabulary in Wikipedia, all the more specifically the intensive inward

structures (hyperlinks) and folksonomy (class system). Given the sheer size and intricacy of this corpus, the space of conceivable decisions of faceted interfaces is restrictively extensive. We propose measurements for positioning individual feature orders by client's navigational expense, and measurements for positioning interfaces (each with k features) by both their normal pairwise likenesses and normal navigational expenses. We in this manner create faceted interface revelation algorithms that advance the positioning measurements.


**Figure 1:** Facetedpedia Interface

**Assisted Querying using Instant-Response Interfaces [5]**

In this paper Arnab Nandi and H. V. Jagadish attempt to design instant-response interface. The issue of looking for data in huge databases has dependably been an overwhelming errand. In current database systems, the client needs to defeat a huge number of difficulties. To delineate these issues, we take the sample of a client looking for the representative record of an American "Emily Davies" in a venture database. The first major challenge is that of composition unpredictability: expansive associations might have worker records in differed mapping, run of the mill to each department. Second, the client may not be mindful of the accurate estimations of the choice predicates, and may give just a fractional or incorrectly spelled characteristic quality (just like the case with our illustration, where the right spelling is "Davis"). Third, we would like the client to issue questions that are important in terms of result size a question posting all representatives in the organization would not be useful to the client, and could be extravagant for the framework to register. The instant-response interface is comparative in collaboration to different administrations, for example, auto-completion in word processors and mobile phones, and keyword query suggestion in search engines. The client is then anticipated that would reformulate the question by changing the terms entered.

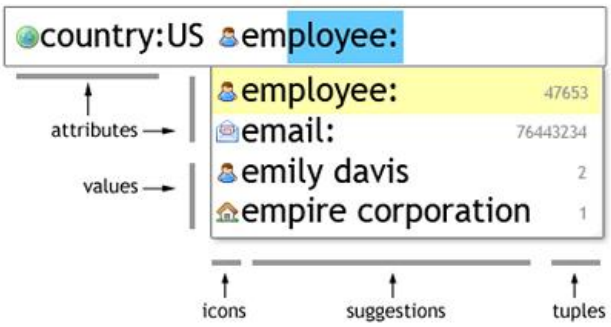


**Figure 2:** The instant-response Interface

In the case of this example, the user first types in “US”, to specify that he is considering only the United States offices of the organization. Since “US” came up as a valid suggestion, the user selected it from the drop down box, which expands the query to country: US. This is beneficial for the database, since it can now do an exact attribute match instead of looking through multiple attributes. Then, the user types in “em”; the suggestion mechanism returns a list of suggestions, with two key values, and two data values. The user notices that there are two employee records which have “Emily Davis” in them, and selects the respective suggestion. Since the query so far will still return a result set of cardinality two, the user decides to refine the query even further, typing in “department:”, which provides the suggestions “Sales”, and “HR”, one for each result tuple. By selecting “HR”, the user is able to locate the single employee record of interest, and directly navigate to it. It should be noted that the complete interaction occurs over just a few seconds, as all suggestions are provided in real time as the user is typing out the query.

**Query Processing Over Incomplete Autonomous Databases [6]**

Incompleteness due to missing attribute values (aka “null values”) is very common in autonomous web databases, on which user accesses are usually supported through mediators. Traditional query processing techniques that focus on the strict soundness of answer tuples often ignore tuples with critical missing attributes, even if they wind up being relevant to a user query [7]. Ideally we would like the mediator to retrieve such possible answers and gauge their relevance by accessing their likelihood of being pertinent answers to the query. The autonomous nature of web databases poses several challenges in realizing this objective. Such challenges include the restricted access privileges imposed on the data, the limited support for query patterns, and the bounded pool of database and network resources in the web environment. We introduce a novel query rewriting and optimization framework QPIAD that tackles these challenges. Our tech nique involves reformulating the user query based on mined correlations among the database attributes. The reformulated queries are aimed at retrieving the relevant possible answers in addition to the certain answers. QPIAD is able to gauge the relevance of such queries allowing tradeoffs in reducing the costs of database query processing and answer transmission. To support this framework, we develop methods for mining

attribute correlations (in terms of Approximate Functional Dependencies), value distributions (in the form of Naive Bayes Classifiers), and selectivity estimates. We present empirical studies to demonstrate that our approach is able to effectively retrieve relevant possible answers with high precision, high recall, and manageable cost.

## 3. Conclusion

Though all of the above papers thrusts majority of work on how to efficiently fetch the data. They lack in user participation in the refinement process. Taking into account all the facets in ranking of the results we can also rank attributes of the forms so that user can give feedback through clickthrough. Thus we are proposing adaptive Query Interface (AQI) which generates user interface dynamically. In static query forms getting desired result is one step process but if the database is huge, user end up in getting too many instances of results and hence desired information is unsatisfactory. Proposed approach uses many rounds of actions as inputted by the user to generate query forms dynamically. Since filtration of results is based on user actions. Process can be extended till satisfactory result or satisfactory number of filtered results can be found. Proposed approach is also beneficial to the non expert user. It starts with a basic query form which contains very few primary attributes of the database. The basic query form is then enriched iteratively via the interactions between the user and our system until the user is satisfied with the query results. Thus, Query forms generated again can be refined according to user feedback and dynamically be changed hence name given as adaptive query interface.

## References

[1] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In CIDR, 2003.
[2] 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).
[3] M. Jayapandian and H. V. Jagadish. Expressive query specification through form customization. In Proceedings of International Conference on Extending Database Technology (EDBT),pages 416–427, Nantes, France, March 2008.
[4] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das. Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. In Proceedings of WWW, pages 651–660, Raleigh, North Carolina, USA, April 2010.
[5] A. Nandi and H. V. Jagadish. Assisted querying using instant-response interfaces. In Proceedings of ACM SIGMOD, pages 1156–1158, 2007.
[6] G. Wolf, H. Khatri, B. Chokshi, J. Fan, Y. Chen, and S. Kambhampati. Query processing over incomplete autonomous databases. In Proceedings of VLDB, pages 651–662, 2007.
[7] 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.
[8] XMark: An XML Benchmark Project: http://www.xml-benchmark.org/.
[9] Magesh Jayapandian, Adriane Chapman, et al. Michigan Molecular Interactions (MiMI): Putting the Jigsaw Puzzle Together. Nucleic Acids Research (Database Issue), 35, 2007.
[10] Daniele Braga, Alessandro Campi, and Stefano Ceri. XQBE (XQuery by Example): A Visual Interface to the Standard XML Query Language. ACM TODS, 30(2), 2005.
[11] Shishir Gundavaram. CGI Programming on the World Wide Web. O'Reilly, 1996.
[12] D. J. Helm and B. W. Thompson. An Approach for Totally Dynamic Forms Processing in Web-Based Applications. In ICEIS (2), 2001.
[13] Sergey Melnik. Generic Model Management: Concepts and Algorithms. Chapter 7. PhD thesis, University of Leipzig, 2004.
[14] Anders Tornqvist, Chris Nelson, and Mats Johnsson. XML and Objects-The Future for E-Forms on the Web. In WETICE. IEEE Computer Society, 1999.
[15] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. IEEE Computer (COMPUTER), 40(8):34–40, 2007.
[16] 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