Identifying Meaningful Return Information for Keyword Based Incremental Query Construction Scheme

Mr. Mahendra P Shinde*, Prof Jyoti Mankar
K.K. wagh college of Engineering

Abstract

Keyword search on database have interesting feature to express users exact informational need but problem is keyword queries are not expressive and fail to retrieve exact information from database, while structural queries retrieve exact information from database but structural queries construction manually is a laborious and error prone task, also it needs exact schema knowledge of database. This paper presents novel system which makes use of keyword to map with predefined query templates and produce structural queries It also returns exact informational need of user by differentiating keywords as predicate or a return node and according to that prediction structured queries will generate. propose system consists of

1. Probabilistic model to access the possible information need
2. Obtaining optimal query construction framework
3. Involvement of other relational operator such as projection and apply effective algorithm for meaningful return information

Keywords: Data mining, keyword search, query construction

1. Introduction

Keyword search on database is a interesting tool for users to express informational need lacks in expressiveness and may fail to retrieve exact informational need. Structural queries can describe exact informational need but it require database schema knowledge also manual creation is tedious job and error prone .To take advantage of both user friendly keyword search on database and structural queries to describe what system should return exactly , some recent approaches [3][10][11][12] translate

keyword queries to structural queries and present rank list of structural queries to user so that he can select best match of his keywords on interpretation but these approach having limitation that only higher rank result get priority but all the time highest rank result is not user intended result. For example if anyone search "'Mumbai"' on database then keyword Mumbai having multiple occurrence in database. But rank list will display only highest rank result which may not be user intended.

If user's intention is to find out lower ranked result then it might take long time to search entire records. For example if we want to search new player or new team or untouched information then it may increases search time which decreases efficiency of system. Only work [3] has taken care of incremental query construction. In this approach if user is not satisfied with ranked result then facility of incremental query construction has been provided. Which subsumed users intended queries and refine it to exact query construction. Which is based on a priory principle on calculating probability of users intended query Bayesians method. This approach calculate frequency of occurrence of keyword in database full text index and assign probability to keyword as well as templates which are predefined and stored in schema table . With this known parameters probability of occurrence of keyword can be calculate. But problem with this approach is that it consider only subset of relational operator such as selection, join and returns query "select* from" For example consider queries IQp [3] has taken care of incremental query construction.In this approach if user is not satisfied with ranked result then facility of incremental query construction has been provided. Which subsumed users intended queries and refine it to exact query construction. But problem with this approach is that it consider only subset of relational operator such as selection, join and returns query "select * from " For example Consider queries

Q1= (mumbaiIN) ;
Q2= (sachin, captain) ;
Q3= (sachin, role);
Q4= (team, mumbaiIN,captain) ;
Q5= (MumbaiIN,player)

which on applying data as shown in fig 1 user intention as For "Q1" user is interested in particular team from IPL named mumbaiIN,for "'Q2"' user is interested in information about player whose name is sachin and who is captain of team for "'Q3"' particular piece of information role of sachin for "'Q4 "'captain of particular team in IPL named mumbaiIN for "'Q5"' player in MumbaiIN. IQp will return all "'select * from"' queries which on joining sub queries intended result will get generated this approach fails to give exact informational need as it will return node with all relevant node and users have to explore all information.

If we look at "'Q5"' it will return sub tree with root node team and user have to find intended information in particular sub tree But question is how to identify the information to be displayed at the beginning and subsequently at each expansion step. Proposed system providing solution to above problem by inferring where clause as projection operator and return only user intended query. Our research includes other operator too such as project, which restrict the search by inferring where clause in constructed query. Involvement of where clause in query construction will eliminate unnecessary increase in interpretation space.

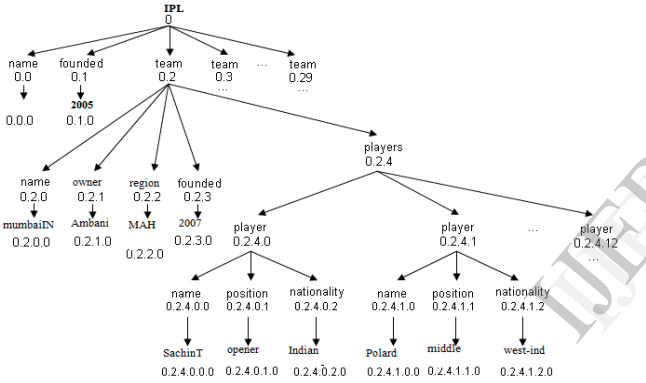
Rest of paper will organize as :

Section 2 "summarizes related work and study of existing technology" such as "'IQp"', "'DBexplorer"', "'Bank: efficiency and drawback .

Section 3 presents the 'conceptual framework' for propose system

Section 4 gives result and efficiency of proposed system

Section 5 provides a conclusion.



2. Related Work

In recent years, keyword search on the database becomes increasingly important. The technology developed from

simple retrieval and ranking of tuples[3],[4],[10],[11] , [12],to ranking and selection of structural queries. As most of these approaches aim to rank and display query as per prior probability also these approaches forming queries which are considering attributes values only while proposed system aims to interpret users intention while entering queries and for that we are considering attributes values, entity , and connection node.

In SUITS [4] keyword is mapping with database administer provided templates and rank structural queries as standardized expected result. To store Ranked list of keyword they use information retrieval techniques such as inverted table index. After that predefined query templates mapped with keywords as per ranking and user habit to search particular information. For example if almost all user search sachin as player from Mumbai Indians team then system will automatically attach query template of player to sachin. Or this can be done manually by database administer; both functionality has been provided with this approach. But this approach is lagging in giving users intended query construction and map keyword as per ranking which may not give required result for low ranked result. Proposed system designed to bridge gap between ranking centric approach and possibility of user's intention while providing keyword.

In IQp[3]system special attention given on incremental construction of query. If any user fails to describe informational need then system provides interactive

query construction option which on proper selection refine into intended result. But system having drawback as it consider only subset of operators such as selection and join and forming query as "select * from "involvement of projection considered in proposed system. For example on entering keyword "sachin, captain" IQp will return select * from DB where name="sachin" + select * from DB where position="captain" " which is not user intended result while proposed system will return "select role from DB where name="sachin" which is most relevant this will help to reduce unnecessary query construction [8] beautifully explain how to indentify return node. This approach explains it for XML keyword search. In this approach system have to find out smallest least common ancestor (SLCA) of matched keyword and according to that system will identify return node, join node or simply attribute. This approach is mainly for XML keyword search and not paying attention on relational databases. We are going to use similar technique but difference in identifying leaf node and we are going to use it for relational database.

In bank[12] keywords query map with nodes in dataset and structure query will generate and hyperlink will provide to user for selection. This approach is very good enough to tackle users informational need and provide required result set.This approach fail to give exact user intended result but our propose work inferring where clause and return node which is not considered in previous any approach. Another approach User intended query formulation has been consider in faceted search organize search results into groups of meaningful facets, by applying some clustering algorithms; proposed system similar to facet search but it also includes query construction option which improves search quality and by providing control over user to express informational need which on incrementally refine produces intended result .

3. Programmers Design

Lots of study and research on identifying selection condition from keywords. This can be done by connecting and grouping keyword matches in meaningful way for example xRank connect these keywords matches by least common ancestor node that contents minimum one occurrence of all keyword. On excluding the occurrence of keywords in their successor that already contents all keywords it also introduces concept of interconnection. For this two matches are interconnected and therefore should be in same group if this is not happening and no two nodes which is having same tag on the path between two nodes excluding themselves Xseek map according to their meaningful LCA

3.1. Experimental setup

For experimental setup we required sample dataset which will model as rooted tree like structure and assign a special Dewey Id as shown in fig 1. Sample dataset of ipl which consists of information of various team, player league owner and so on.This can be done by using closer table otherwise the nested set model known as the modified pre-order tree traversal algorithm Nested Set is a good solution for storing rooted tree like data that provides fast access for reading data . However, updating nested set trees is more costly. Therefore this solution is best suited for rooted tree like structure that are much more frequently read than written to. For system We installed the data sets on a MySQL database server , Intel core e duo 64 bit 2 GHz processor along with 4 GB RAM. The inverted data index was constructed using Lucien function . Our system will be implement using .NET.

Definition 1 (Structured query). A structured query Q in is form of relational algebra which can be represented
as tables, operators or predicates while

Definition 2 (Keyword query). A keyword query K represents just a sack of keywords. We differentiate three types of information in data

Definition 3. Entity in the real world
Entity in real world means leaf node which represent exact information about typed keyword Attributes of Entity Attributes represents nodes as child of the element node which are associated, and not distinguishing them from element nodes
Definition 4 .Connection node
Connection node joins the remaining structural queries which are separate from attributes and entity we are going to classify user entered keyword into two types One that specify search predicate One that identify return information we are going to classify user entered keyword into two types
One that specify search predicate
One that identify return information

3.2. how to identify predicate and return node: :

1) Differentiate data type from data values. When schema info is available we can directly obtain type information another way is to use nodes name for indicating data types
2) Predicate consists of type and value while return node specifies values only and not specifying data types If input keyword matches a node name or type and there is no keyword k2 matching node value v such that U is ancestor of V then ki specifies a return node if any keyword doesn't indicate return node then it will be treated as predicate consider a queries in example ; in Q2 captain is considered as predicate Sachin in Q2 and Q3 considered as predicate team in Q4 considered as predicate but role in Q3 is considered as return node as it matches with two nodes .similarly player in Q4 also treated as return node.

3.3. Steps in Query construction process

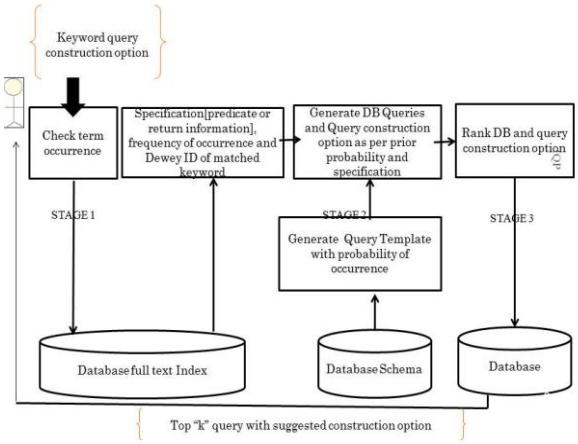
1) On issuing keywords Checking term of occurrence of these keywords in database full text index, return
Dewey id of node, classify keyword as return node or predicate.
2) After that as per probability of occurrence of predefined query template mapping as a selection or
Projection
3) Rank generated structured query and present it to user for further acceptance or rejection as intended query
4) Construct and refine top k queries. And present it to user

3.4 Data Flow

Data flow in proposed approach is as shown in the figure 5 when any user enter query it will first search in database full text index similar to after that information store in database full text index in rooted tree like format will return Dewey id along with return type to the system this also returns frequency of occurrence of typed keyword so that probability of occurrence will calculate. This will pass to the schema table where predefined query templates has been stored and this
query templates mapped with keyword to generate partial structure query. these query along with return type of keyword will return exact structural query. For example In Q2, since both "'Sachin"' and "'captain"' are identified as predicates, no explicit return nodes are

specified in the keywords. We first identify the player node (0.2.4.0) as the SLCA node. It is known to be master entity, and the only relevant entity in given group of keyword. Therefore player is treated as the implicit return node for Q2. then generated structural query points to relational database which will result intended result .

.



4. Result And Discussion

To measure search quality we will use precision and recall

Precision = Rel ∩ Ret /Ret

Recall = Rel ∩Ret / Rel

Where Rel is the set of relevant nodes (i.e. desired search results),

Ret is the set of nodes returned by a system every keyword search is expressed as an English sentence and analyzing this data will be extracted from original database and will represent ground truth Recall. we will examine generated output by system ie Ret and on counting nodes in RET appear in relevant node. with the best of our knowledge identifying meaningful return information for keyword based query construction is concise study to identify and infer return node along with predicate information which will give exact users intended result, with incremental query construction option

5. Conclusion

Proposed approach is to enhance searching method describe in . After identifying intention behind entering keyword for search, system will classify keywords into predicate and return node which will be input to the existing system. As predicate will infer where clause while return node will infer select clause .This approach will remove drawback of by inferring where clause in query construction process , which gives exact user intended result by analyzing query and get user intent. use of other operator such as group by ,union ,intersection will be added in future development

6. References

[1] H. Bast, A. Chitea, F. Suchanek, and I. Weber ESTER: Efficient Search on Text, Entities, and Relations, Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2007.
[2] S. Agrawal, S. Chaudhuri, and G. Das, DBXplorer: A System for Keyword-Based Search over Relational Databases , Proc. Int'l Conf. Data Eng. (ICDE), 2002
[3] E. Demidova, X. Zhou, and W. Nejdl, Member, IEEE Computer Society, IQP: Incremental Query Construction, a Probabilistic Approach , Proc. 26th IEEE Int'l Conf. Data Eng. (ICDE)Mar 2012.
[4] E. Demidova, X. Zhou, G. Zenz, and W. Nejdl, SUITS: Faceted User Interface for Constructing Structured Queries from Keywords, , Proc. 14th Int'l Conf. Database Systems for Advanced Applications (DASFAA), 2009.
[5] IXQuery 1.0: An XML Query Language, June 2001. http://www.w3.org/XML/Query. ,
[6] S. Amer-Yahia, C. Botev, J. Dorre, and J. Shanmugasundaram, XQuery Full-Text extensions explained , In IBM Systems Journal,pages 335.352, 2006.
[7] G. Koutrika, A. Simitsis, and Y.E. Ioannidis Explaining Structured Queries in Natural Language , Proc. Int'l Conf. DataEng. (ICDE), 2010
[8] Z. Liu and Y. Chen Identifying Meaningful Return Information for XML Keyword Search , Proc. ACM SIGMOD Int'l Conf.Management of Data (SIGMOD), 2007
[9] Y. Luo, X. Lin, W. Wang, and X. Zhou, SPARK: Top-k Keyword Query in Relational Databases, Proc. ACM SIGMOD Int'l Conf.Management of Data (SIGMOD), 2007
[10] S. Tata and G.M. Lohman, SQAK: Doing More with Keywords , Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD),2008.
[11] H. He, H. Wang, J. Yang, and P.S. Yu, BLINKS: Ranked Keyword Searches on Graphs , . ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), 2007.
[12] Gaurav Bhalotia ,Arvind Hulgeri,Charuta Nakhe,Soumen Chakrabarti S. Sudarshan, BANKS :Browsing and Keyword Searching in Relational Database ,
[13] J. Clark and S. DeRose. XML Path Language (XPath) 1.0 November 1999. http://www.w3.org/TR/xpath
[14] D. Florescu, D. Kossmann, and I. Manolescu, BIntegrating Keyword Search into XML Query Processing
[15] Gaurav Bhalotia ,Arvind Hulgeri,Charuta Nakhe,Soumen Chakrabarti S. Sudarshan, BANKS :Browsing and Keyword Searching in Relational database , . Computer Networks (Amsterdam, Netherlands: 1999)
[16] V. Hristidis, N. Koudas, Y. Papakonstantinou, and D. Srivastav Keyword Proximity Search in XML Trees , IEEE Transactions on Knowledge and Data Engineering, 18(4), 2006

[17] P. Wu, Y. Sismanis, and B. Reinwald, Towards Keyword-Driven Analytical Processing, , Proc. ACM SIGMOD IntŠl Conf. Management of Data (SIGMOD), 2007.
[18] M.M. Zloof BQuery-by-Example: A Data Base Language , IBM Systems J., vol. 16, no. 4, pp. 324-343,1