XML Query Construction Based On Keyword Search Using IQP

Dipika Joge, Vaishali Sahare

Abstract— **Keyword search alleviates the usability problem at the price of query expressiveness. As keyword search algorithms do not differentiate between the possible informational needs represented by a keyword query, users may not receive adequate results. This paper presents a system that searches a keyword to fulfill the user informational needs for XML query structure. We have used IQP—a novel approach to bridge the gap between usability of keyword search and expressiveness of database queries. IQP enables a user to start with an arbitrary keyword query and incrementally refine it into a structured query through an interactive interface. Applying this methodology to XML data, an easier way is to be developed to construct a XML query based on keyword search. Without learning SQL and XML, web users can access the XML data. Identification of return nodes are enabled using this technique. This project presents the detailed design for XML Query construction based on keyword search using a novel method IQP. An incrementally constructed query allows the data to be represented in detail. This paper also presents the system which can be directly used by third user called novice user having limited or no knowledge of XML.**

Index Terms— **query, expressiveness, structured query.**

I. INTRODUCTION

Keyword is a word which acts as the key to a cipher or code. It is a word used in an information retrieval system to indicate the content of a document. A type of search that looks for matching documents contains one or more words specified by the user.

It all begins with words typed into a search box. Keyword research is one of the most important, valuable, and high return activities in the search marketing field. Ranking for the right keywords can make or break your website. It's not always about getting visitors to your site, but about getting the right kind of visitors**.** The results can be abundant and not so informative according to visitor. The problem with keyword search is that, the expressiveness of query constructed for generating a search is lagging in most of search engine. They use structured data representation, which are difficult to understand to visitors. Ranking the information is easy but ranking the query is an error-prone task, which can be solved by XML.

These problems can be solved by two issues. First, using IQP, a user can benefit from both, a conventional ranking interface and a more controllable query construction interface. The former allows the user to immediately identify the most common interpretation of her query. The latter enables the user to clarify her search intent step by step, which is especially helpful when the intended query interpretation does not receive a good rank. IQP system consists of three components: 1) a framework that formally defines the process of incremental query construction; 2) a probabilistic model to estimate the probabilities of structural query interpretations; 3) an algorithm for generating the optimal query construction plan (QCP), which enables a user to obtain the intended structured query with a minimal number of interactions [1]. When a user issues a keyword query, *IQP* provides the user with a *ranked list of structured queries* (as interpretations of the keyword query) and the corresponding results, which are presented in the query and result windows, respectively.

Second, extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is a textual data format with strong support via Unicode for different human languages. The design goals of XML emphasize simplicity, generality and usability across the Internet. The design of XML focuses on documents and data structures in web services. Due to the lack of expressivity and inherent ambiguity, there are two main challenges in interpreting the semantics when performing keyword search on XML data. First, unlike a structured query where the connection among the data nodes matching the query is specified precisely in the 'where' clause (in XQuery or SQL) and/or as variable bindings (in XQuery), we need to automatically connect the match nodes in a meaningful way. Second unlike a structured query where the return nodes are specified using either a 'return' clause (in XQuery) or 'select' clause (in SQL), we should effectively identify the desired return information. This problem can be solved using IQP system.

II. XML Query Construction

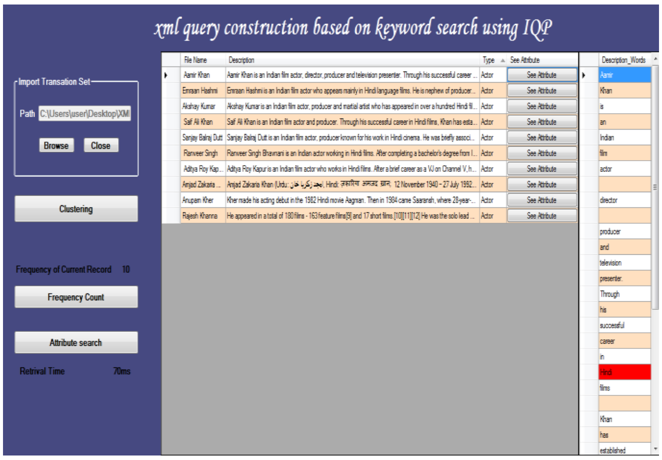
Analyzing XML Data Structure

To decide what information should be returned, we need to understand the roles and relationships of nodes in the data. The information in XML documents can be recognized as a set of real world entities, each of which have attributes and interact with other entities through relationships. This mimics the Entity-Relationship model in relational databases.

In general, we make the following interfaces on node categories

1. A node represents an entity if it corresponds to a *-node in the DTD.
2. A node denotes an attribute if it does not correspond to a *-node, and only has one child, which is a value.
3. A node is a connection-node if it represents neither an entity nor an attribute. A connection node can have a child that is an entity, an attribute or another connection node.

Dipika Joge, G.H.Raisoni Institute of Engineering & Technology for Women, Nagpur
Vaishali Sahare, G.H.Raisoni Institute of Engineering & Technology for Women, Nagpur



xml query construction based on keyword search using IQP

Import Transation Set

Path C:\Users\user\Desktop\XM

Frequency of Current Record 10

Retrival Time 70ms

File Name	Description	Type	See Attribute
Aamir Khan	Aamir Khan is an Indian film actor, director, producer and television presenter. Through his successful career ...	Actor	See Attribute
Emraan Hashmi	Emraan Hashmi is an Indian film actor who appears mainly in Hindi language films. He is nephew of producer...	Actor	See Attribute
Akshay Kumar	Akshay Kumar is an Indian film actor, producer and matial artist who has appeared in over a hundred Hindi fil...	Actor	See Attribute
Saif Ali Khan	Saif Ali Khan is an Indian film actor and producer. Through his successful career in Hindi films, Khan has esta...	Actor	See Attribute
Sanjay Balraj Dutt	Sanjay Balraj Dutt is an Indian film actor, producer known for his work in Hindi cinema. He was briefly associ...	Actor	See Attribute
Ranveer Singh	Ranveer Singh Bhavnani is an Indian actor working in Hindi films. After completing a bachelor's degree from I...	Actor	See Attribute
Aditya Roy Kap...	Aditya Roy Kapur is an Indian film actor who works in Hindi films. After a brief career as a VJ on Channel V, h...	Actor	See Attribute
Amjad Zakaria ...	Amjad Zakaria Khan (Urdu: امجد زکریا خان, Hindi: ज़कारिया अमजद ख़ान; 12 November 1940 – 27 July 1992...	Actor	See Attribute
Anupam Kher	Kher made his acting debut in the 1982 Hindi movie Aagman. Then in 1984 came Saaransh, where 28-year-...	Actor	See Attribute
Rajesh Khanna	He appeared in a total of 180 films - 163 feature films[9] and 17 short films.[10][11][12] He was the solo lead ...	Actor	See Attribute

Description_Words
Aamir
Khan
is
an
Indian
film
actor
director
producer
and
television
presenter.
Through
his
successful
career
in
Hindi
films
Khan
has
established

Fig. 1. *IQP User Interface*

Analyzing Keyword Match patterns:

Besides studying the structure of XML data and inferring inherent entities and attributes presented in the data, we also analyze the pattern of the keyword matches to infer search predicate and return node specifications. Some keywords indicate 'predicates' that restrict the search, corresponding to the 'where' clause. Some keywords specify 'return nodes' as the desired output type, corresponding to the 'return' clause in XQuery or the *select* clause in SQL.

IQP on XML:

Incremental query construction is used in Xml query construction, instead of VLCA node technology. Query building is done in some steps like first translation of keyword query to a structured query. Then query is interpreted to form query hierarchy i.e. sub-query relationship is generated for more than one result. Followed by query construction plan query structures are being processed.

Processing query structures

A query construction plan (QCP) is binary tree. Each node of tree represents a structured query. The left and right node represents the acceptance and rejection of query construction option i.e. partial interpretation, respectively. Construction of query, matching a keyword with query and ranking the queries are processed in this phase. Without knowing the exact informational need of the user, IQP translates the keyword query into a number of structured queries, which give different interpretations to User's keywords. For example, one possible structured query searches for the information of "Amitabh" and entitled "Singer", the possible interpretations and corresponding results are ranked and presented in the query and result windows. Simultaneously, IQP generates a set of construction options, and presents these options in the query construction window.

Generating search results

Generation of result according to matching keyword algorithm and generating result algorithm is obtained. These algorithms provides grouping of matched keywords according to the nodes. We infer return nodes either explicitly from keywords by analyzing keyword match patterns, or implicitly by considering both keyword matches and relevant entities in the data. The data nodes that match return nodes are output based on their node categories: attributes entities and connection nodes. Besides outputting the matches to return nodes, data nodes that match search predicates are also output such that the user can verify the meaning of the matches.

III. Implemented Algorithm

Framework and Definitions

The Query construction framework states an interpretation of keyword, query interpretation generation, sub-query relationships and query construction plan. To support an efficient query construction, it is important to have an accurate assessment of the probability of whether a XML query interprets a user's keyword correctly. Here we IQP compute these probabilities. There are two types of keyword interpretation.

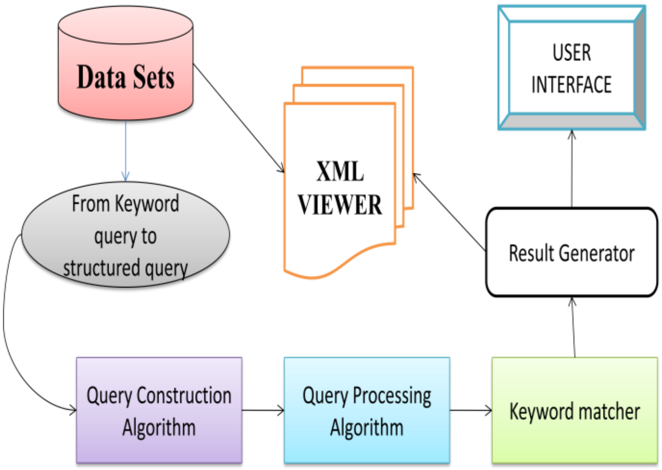

Data Sets
XML VIEWER
USER INTERFACE
From Keyword query to structured query
Result Generator
Query Construction Algorithm
Query Processing Algorithm
Keyword matcher

Fig. 2. *Framework of proposed methodology*

The first type interprets a keyword as a part of a query template, such as table name, an attribute name etc. the second type maps a keyword to a 'contains' predicate, interpreting the keyword as a value of an attribute. The next one has two phases i.e. 1. Query construction algorithm and 2. Query processing algorithms. The query construction algorithm is based on a greedy algorithm which constructs a query. Next the query processing algorithm has two algorithms named Match keyword and grouping algorithm and generating result algorithm. The first algorithm is to match the keyword to the constructed query and group the similar keywords. Next algorithm generates the result of keyword match and constructed XML query.

Query Construction Algorithm

An algorithm to create a plan that imposes as little effort on the user as possible, i.e., a minimum query construction plan is stated here. We present the pseudo code of the greedy algorithm for query construction. IQP generates query interpretations by expanding the query hierarchy in a bottom-up fashion.

Greedy Algorithm:

Instead of fully expanding the query hierarchy, the greedy algorithm stops when the size of the top level of the query hierarchy reaches a certain threshold (denoted by T). Then, it searches for the best query construction option (denoted by best_r) within the current query hierarchy and presents the option to the user.

- If the user accepts the option, the algorithm keeps the part of the top level subsumed by this option and discards the rest.
- If the user rejects an option, the algorithm discards the part of the top level subsumed by this option.
- The algorithm processes till user reaches the final outcome. We are using the greedy algorithm for XML query construction.

Query Processing Algorithm

Match keywords and grouping match nodes algorithm

- For a set of input keywords, we start with the procedure find match and retrieves the list of data nodes KWmatch that match a keyword.
- The node lists are obtained by accessing the name index and value index using NAMEID and VALUEID operations.
- For each match, we record whether it is a name or a value. Then the groupMatch procedure group then keyword matches KWmatch based on their ancestor.
- Then the groupMatch procedure groups the keyword matches KWmatch based on their ancestor node (if exists).

Generating result algorithm

- After keyword matches are grouped according to their nodes, if no explicit return node are specified in a group, implicit return nodes will be inferred.
- Then our search engine generates search results by outputting data nodes that match search predicates and return nodes.
- The genResult procedure navigates the paths from the master entity to each match in a group, identifies and outputs the matches to predicates and explicit or implicit return nodes.

IV. Performance Analysis

➢ Query type 1

Select attribute : single keyword

Result : All statements present in dataset are shown containing the entered keyword highlighted.

Xml result : The result is shown in separate xml window with keyword type and description.

Example:

Keyword : GANGA

Result : (i) The Ganga was ranked as the fifth most polluted river of the world in 2007.

(ii) Ganga is an Indian name mostly given to girl child in India.

➢ Query type 2

Select attribute: multiple keywords

Result: Information present in dataset containing these multiple keywords together is shown in result.

Xml result: The result is shown in separate xml window with keyword type and description, arranged with the differentiated datasets.

Example:

Multiple keyword: Bhartiya Janta Party

Result: (i) Modi, a leader of the Bharatiya Janata Party.

(ii) Rajnath Singh is an indian political leader of Bhartiya Janata Party.

➢ Frequency of current record is measured while clustering the data.

➢ Frequency count is another performance measure of this system. The record of all keywords searched by random users is shown with attribute count with respect to its date and time.

➢ Retrieval time for each keyword in query type 1 and query type 2 is measured and shown in milliseconds.

V. Graphical Analysis

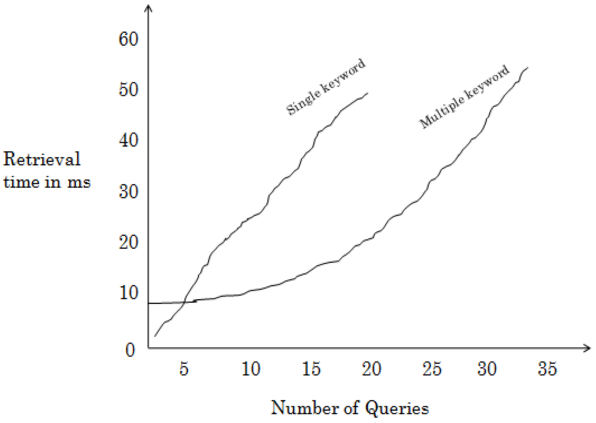


Fig. 3. Processing Time with incremental queries

VI. Conclusion

We present an XML keyword search engine that solves the problem of inferring nodes. We analyze XML data structure as well as keyword match patterns. The pattern matching also results in top query ranking and minimization of retrieval time for keyword search for various query types. We presented the conceptual query construction framework for incremental query construction and probabilistic model for accessing user required information. The performance analysis shows the high rate of search of top-ranked queries, which verified our motivation and approaches.

Acknowledgment

The authors would like to thank GHRIETW Nagpur for supporting the implementation of the user study. This work has been partially supported by the Prof. Anil Jaiswal and Prof. Vaishali Sahare.

References

[1] Elena Demidova, Xuan Zhou, and Wolfgang Nejdl, “A Probabilistic Scheme for Keyword-Based Incremental Query Construction”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012.

[2] Ziyang Liu, Yi Chen, “Identifying Meaningful Return Information for XML Keyword Search”, ACM, *SIGMOD* June 2007.

[3] Elena Demidova, Xuan Zhou*, Wolfgang Nejdl, “IQP: Incremental Query Construction, a Probabilistic Approach”, IEEE, ICDE Conference 2010.

[4] Youjin Chang, Minkoo Kim, Vijay V. Raghavan, “Construction of query concepts based on feature clustering of documents”, Springer Science+Business Media, LLC 2006.

[5] HolgerBast, AlexandruChitea, Fabian Suchanek, Ingmar Weber, “ESTER: Efficient Search on Text, Entities, and Relations”, ACM,SIGIR'07, Amsterdam, The Netherlands, July 2007.

[6] Yi LuoXuemin Lin Wei Wang, Xiaofang Zhou, “SPARK: Top-k Keyword Query in Relational Databases”, ACM, SIGMOD June-2007.

[7] Daniele Braga, Alessandro Campi, Stefano Ceri, “*XQBE* (*XQ*uery *ByE*xample): A Visual Interface to the Standard XML Query Language”, ACM Transactions on Database Systems, Vol. 30, No. 2, June 2005.

[8] Venkatesan T. Chakaravarthy, Himanshu Gupta, Prasan Roy, MukeshMohania,“Efficiently Linking Text Documents with Relevant Structured Information”, ACM, VLDB Sepetmember-2006.

[9] Arnab Nandi, H. V. Jagadish, “Assisted Querying using Instant-response Interface”, ACM, SIGMOD June-2007.

[10] Ziyang Liu, Jeffrey Walker, Ya Chen, “XSeek: A Semantic Search Engine Using Keyword”, ACM, VLDB Septmber-2007.

[11] Iryna Oleze, Prof. Dr. techn. Wolfgang Nejdl, Prof. Dr. rer. nat. Udo Lipeck, “Integration of YAGO Ontology In The IQP Query Construction System To Support Efficient Query Construction Over a Large-scale Relational Database”, ICDB, 2008.

[12] Fang Liu, Clement Yu, Weiyi Meng, Abdur Chowdhury, “Effective Keyword Search in Relational Databases”, ACM, SIGMOD 2006, June 27-29, 2006.

[13] Hongwei Li, Dongxiao Liu, Yuanshun Dai, Tom H. Luan, Xuemin (Sherman) Shen, IEEE, “Enabling Efficient Multi-keyword Ranked Search over Encrypted Mobile Cloud Data through Blind Storage”, IEEE Transactions on Emerging Topics in Computing, 2003.

[14] Chris K. Anderson, Ming Cheng, “Paid Search: Modeling Rank Dependent Behavior”, 47th Hawaii International Conference on System Science, 2014.

[15] Jianxin Li, Chengfei Liu, Rui Zhou, Jeffrey Xu Yu, IEEE, “Quasi-SLCA Based Keyword Query Processing over Probabilistic XML Data”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 4, APRIL 2014.

[16] Sunita Mohanta, M. Saravanan, “Keyword-Based Incremental Query Construction Using IQP Mechanism**”,** International Conference on Advanced Engineering & Technology, 10th March Chennai, ISBN: 978-93-82702-20-7.

[17] Enrico Minack, Wolf Siberski, Gideon Zenz, Xuan Zhou, “SUITS4R|DF: Incremental Query Constructionfor the Semantic Web”, *International Semantic Web Conference,volume 401 of CEUR Workshop Proceedings, CEUR-WS.org*, 2008.

[18] Sanjay Agrawal, Surajit Chaudhari, Gautam das, “DBXplorer: A System for Keyword-Based Search over Relational Databases”, Proceedings of the 18th International Conference on Data Engineering (ICDE.02), 2002.

[19] Deepika Joge, Minal Kamble, P.S.Chhaware, “REVIEW: Probabilistic Scheme For IQP And XML Query Construction By Keyword Search”, MANTHAN 2014, 21st and 22nd February 2014.