International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 06 | June-2016 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of XML Context Diversified Search

Simon Farande[1], Dr.D.S.Bhosale[2]
[1,2]Ashokrao Mane Group of Institutions, Vathar,Maharashtra,India

---

Abstract - *An When user searches a keyword he types short and obscure keywords and therefore the ambiguity of keyword queries makes it tough to effectively answer keyword queries. This paper focuses on wide-ranging XML keyword search supported on its completely different contexts within the XML information . If the user enters short and obscure queries it's searched in XML information, and derive keyword search candidates of the queries by a straight forward choice model. Then design a good XML keyword search diversification model by implementing two algorithms.*

***Key Words*: Search, XML, Baseline, Anchor based pruning , indexing ,diversification**

1.INTRODUCTION

Keyword search on structured and semi-structured information has attracted a lots of analysis interest recently, because it permits users to retrieve with no requirements to learn query languages and database structure . Compared with keyword search ways in data Retrieval (IR) that favor to realize an inventory of relevant documents, keyword search approaches in structured and semi-structured information (denoted as DB&IR) concentrate a lot of on specific data contents, e.g., fragments nonmoving at the smallest lowest common ancestor (SLCA) nodes of a given keyword question in XML. Given a keyword question, a node v is considered associate degree SLCA if(1) the subtree nonmoving at the node v contains all the keywords, and(2) there doesn't exist a descendant node v′ of v specified the subtree nonmoving at v′ contains all the keywords.

In different words, if a node is an SLCA, then its ancestors are undoubtedly excluded from being SLCAs, by that the lowest data content with SLCA linguistics may be accustomed represent the precise leads to XML keyword search. during this system the well accepted SLCA linguistics as a result metric of keyword question over XML information is to be adopted. In general, the a lot of keywords a user's question contains, the better the user's search intention with regards to the question may be known. However, once the given keyword question solely contains a tiny low variety of obscure keywords, it might become a really difficult drawback to derive the user's search intention owing to the high ambiguity of this sort of keyword queries. though generally user involvement is useful to spot search intentions of keyword queries, a user's interactive method is also time intense once the scale of relevant result set is massive. to handle this, develop a technique of providing various keyword question suggestions to users supported the context of the given keywords within the information to be searched. By doing this, users could opt for their most well-liked queries or modify their original queries supported by various question suggestions.

When the given keyword question solely contains alittle variety of obscure keywords, it'd become a really difficult drawback to derive the user's search intention as a result of the high ambiguous queries. though typically user involvement is useful to spot search intentions of keyword queries, a user's interactive method could also be time overwhelming once the scale of relevant result set is giant. To address this, Jianxin Li, Chengfei Liu *Member,* and Jeffrey Xu Yu [1] proposed a technique of providing various keyword question suggestions to users supported the context of the given keywords within the knowledge to be searched. By doing this, users might select their most popular queries or modify their original queries based on the came back various question suggestions.

The most easy and effective querying methodology for non-structured document collections is that the well-known keyword search. one amongst its key blessings is simplicity, since users solely have to be compelled to specify the keywords they're fascinated by. However, XML document collections have each content and structure, and should be queried by content, structure or each. **Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Eduardo Vicente-López**[2], keep the straight forward keyword search question interface, though they exploit XML structure throughout the question process, in order that the retrieved results will be any quite document elements.

Data Model and classification of nodes can be done as proposed by **Youqiang Guo , Guixiu Tao, Yuqing Liang, Lei Wang, Honghao Zhu[3].**

XML documents, the subtrees connected by the content nodes that directly contain some keywords aren't simply retrieved. consequently, it's abundant more durable to retrieve the integrated subtrees (like the documents and reticular tuples) joined by the content nodes and complementary nodes that don't contain any keyword however contain some relevant and purposeful knowledge, since it's rather tough to work out that nodes are complementary nodes. **LI Guoliang ,FENG**

Jianhua And ZHOU Lizhu[4] mentioned how to retrieve those compact, integral subtrees, referred to as self-integral trees (SI-Trees), that contain complementary nodes that capture the main target of keyword queries, besides the content nodes. additional significantly, every self-integral tree represents associate degree integrated aiming to answer a keyword question..

Jianxin Li, Chengfei Liu, Rui Zhou and Bo Ning[5] suggest Given a keyword question alphabetic character ANd an XML schema tree T, a collection of structuredalphabetic characterueries alphabetic character could also be created and evaluated over the info supply conformist to T for respondent q. the solution to the XML keyword question alphabetic character could also be a giant variety of relevant XML fragments. In distinction, {the ANswer|the solution} to the top-k keyword question is an ordered set of fragments, wherever the ordering reflects however closely every fragment matches the given keywordquestion. Therefore, solely the highest k results with the very best connection w.r.t. alphabetic character have to be compelled to be came to users.

2.FEATURE SELECTION AND DIVERSIFICATION

As mentioned in [6] when the document is uploaded to the search engine its information is stored in XML file (T) and the relevance based term pair dictionary (W) is also stored in that file. At the time of search the distinct term-pairs are selected based on their mutual information .Mutual information is to be used as principle or standard for feature selection and feature transformation in machine learning. It can be used to distinguish both the relevance and redundancy of variables, such as the minimum redundancy feature selection. This can be shown in Fig1.

2.1 A. Baseline Solution

As mentioned in [1] baseline solution can be implemented as follows:

1. Given keyword query q with n keywords, we first load its pre-computed relevant feature terms from the term co-related graph G of xml data T. which is used to construct a matrix Mm*n .
2. We generate a new query candidate qnew from the matrix Mm*n by calling the function GenerateNewQuery ().
3. Generation of new query candidates is in the descending order of their mutual information score.
4. Compute the SLCA results of qnew; we need to retrieve the recomputed node lists of the keyword-feature term pairs in qnew from T by getNodeList (sixty, T).
5. Compare the SLCA result of the current query and the previous queries in order to obtain the distinct and diversified SLCA result.
6. We compute the final score of qnew as diversified query candidate with respect to previously generated query candidate in Q.
7. At last we compare the new query and previously generated and replace the unqualified once in Q.we can return the top k generated queries with their SLCA result.

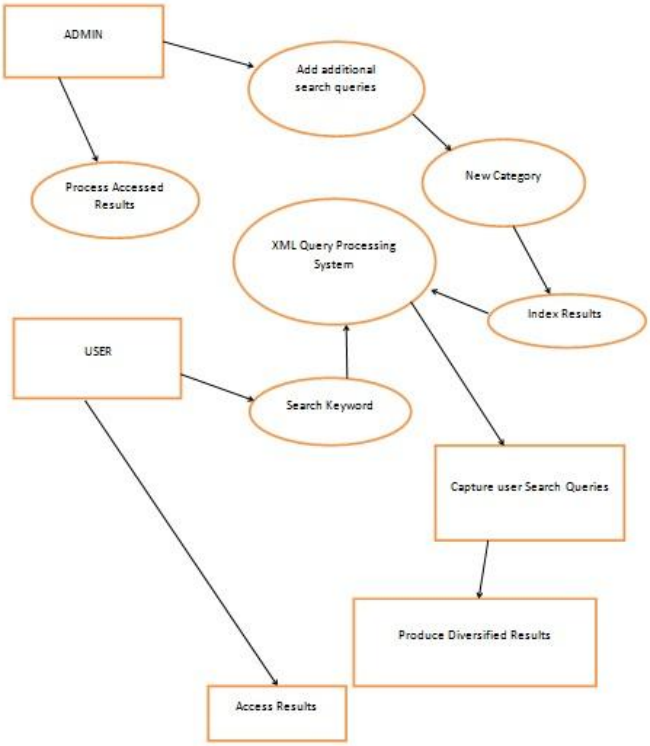


Fig1-*Design of XML Context Diversified Search*

2.2.Anchor-based Pruning Solution

All As mentioned in [1] baseline solution can be implemented as follows:

1. Given keyword query q with n keywords, we first load its pre-computed relevant feature terms from the term co-related graph G of xml data T. which is used to construct a matrix Mm*n .

2. We generate a new query candidate qnew from the matrix Mm*n by calling the function GenerateNewQuery ().

3. Compare new SLCA result with the previously generated SLCA result.

4. For the first generated query, we can compute the SLCA results using any existing XML keyword search method .use stack based method to implement the function computeSLCA(). Result of the 1st query will be taken as anchors to prune the node list of the next query for reducing its evaluation cost.

5. Given an anchor node anchor, for each node list lixjy of a query keyword in the current new query, we may get three effective node lists lixjy_pre, lixjy_des and lixjy_next using R-tree index by calling for the function Partition ().

6. If a node list is empty, e.g., lixjy_pre =null, then we don't need to get the node lists for the Other query keywords in the same area.

7. If all the node lists have at least one nodein the same area, then we compute the SLCA results by the function ComputeSLCA (), If the SLCA results are the descendant of vanchor, then they will be recorded as new distinct results and vanchor will be removed from the temporary result set.

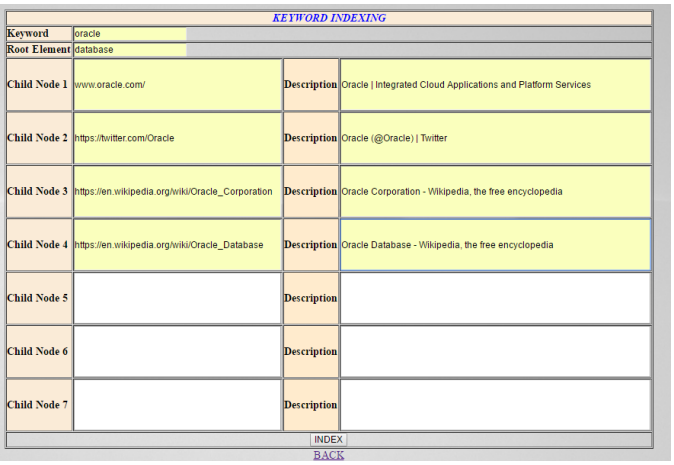
8. After all the necessary queries are computed, the top-k diversified queries and their results will be returned.

3. METHODOLOGY

A real dataset, based on keywords is used for testing the projected XML keyword search diversification model and designed algorithms. Use xml dataset to live the effectiveness of diversification model during this work. for every XML dataset used, choose some terms supported the subsequent criteria: (1) a particular term ought to typically seem in user-typed keyword queries;(2) a particular term ought to highlight totally different linguistics once it co-occurs with feature terms in several contexts.

3.1 Indexing the keywords

Information to be uploaded is given with the keyword to be searched, the category in which it should be displayed and the web links for it. Once it is submitted JaxB Marshalling takes place where the java objects will be converted to XML format. XML document is generated for the above information.Keyword indexing is shown in Fig2.



KEYWORD INDEXING

Keyword	oracle		
Root Element	database		
Child Node 1	www.oracle.com/	Description	Oracle | Integrated Cloud Applications and Platform Services
Child Node 2	https://twitter.com/Oracle	Description	Oracle (@Oracle) | Twitter
Child Node 3	https://en.wikipedia.org/wiki/Oracle_Corporation	Description	Oracle Corporation - Wikipedia, the free encyclopedia
Child Node 4	https://en.wikipedia.org/wiki/Oracle_Database	Description	Oracle Database - Wikipedia, the free encyclopedia
Child Node 5		Description	
Child Node 6		Description	
Child Node 7		Description	

INDEX
BACK

Fig2-Indexing of Keywords

3.2.UnMarshalling the required contents

User will fetch the required information using Keywords. The results will be diversified where the related informations to required data will also be displayed. This is done by JaxB UnMarshalling where the XML document will be converted to java objects and displayed to the users.

3.3. Pruning Results in form of Categories

When the user searches required information with a query, the information required by him is retrieved along with all other related information. So in this module pruning is done to differentiate the diversified data. The diversified data will appear in form of categories.

3.4. Reranking the Categorised results

Once the results are pruned in form of categories, which helps a user to view the most visited links by other users. This is done by keeping track of all active sessions that is occurred and statistically assigning to 1 . The links which have been opened will increase the session count and based upon these sessions that pages will be hierarchically arranged in top down order.

3.5. User Uploads

When user knows some information which is not available in the site, he can request for posting his views towards it. The request will be moved to an Auditor whose role is to check for users request and accept it if it is useful or deny it. Indexing is done in users upload also.

REFERENCES

[1] Jianxin Li, Chengfei Liu *and Jeffrey Xu Yu* "Context-based Diversification for Keyword Queries over XML Data" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING , VOL. 27, NO.3, March 2015,page 660-672

[2]Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Eduardo Vicente-López "Using Personalization to Improve XML Retrieval" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 5, MAY 2014

[3] Youqiang Guo , Guixiu Tao, Yuqing Liang, Lei Wang, Honghao Zhu, " XML Keyword Search Based on Node Classification and Hierarchical Semantics" Communications in Information Science and Management Engineering Jan. 2014, Vol. 4 Iss. 1, PP. 6-12

[4] LI Guoliang ,FENG Jianhua And ZHOU Lizhu," Keyword Searches in Data-Centric XML Documents Using Tree Partitioning" in TSINGHUA SCIENCE AND TECHNOLOGY ISSNl l1007-0214l l02/ 21l lpp7-18 Volume 14, Number 1, February 2009

[5] Jianxin Li, Chengfei Liu, Rui Zhou and Bo Ning ," Processing XML Keyword Search by Constructing Effective Structured Queries" Advances in Data and Web Management Lecture Notes in Computer Science Volume 5446 , 2009, pp 88-9

[6] Simon Farande,Dr.,D.S.Bhosale,"Survey Paper on XML Context Diversified Search" in International Journal of Advance Research in Science and Engineering Vol. No.5,Special Issue No.01,March 2016,IJARSE,ISSN 2319-8354,page 1-4.