Scientific Journal of Impact Factor(SJIF): 4.14

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Special Issue for ICPCDECT 2016,Volume 3 Issue 1

Keyword query search using context based diversification in xml data with performance analysis

Kirti Agarwal
Department of Computer Engineering,
Pune Vidyarthi Griha's College of Engg & Tech.
Pune,India
Email : agarwalkirtidilip@gmail.com

Mihir Joshi
Department of Computer Engineering,
Pune Vidyarthi Griha's College of Engg & Tech.
Pune,India
Email : mihirjoshi20@gmail.com

Shreenivas Latad
Department of Computer Engineering,
Pune Vidyarthi Griha's College of Engg & Tech.
Pune,India
Email : latad.shreenivas@gmail.com

Abstract-*Keyword query empowers ordinary users to search large data. But the ambiguity of keyword query makes it difficult to effectively answer keyword queries, especially if the query is short and vague. To solve this challenging problem, we propose an approach that automatically diversifies XML keyword search based on its different contexts in the XML data. When a short and vague keyword query is provided and XML data to be searched, we first derive keyword search candidates of the query by a simple feature selection model. Then, we design an effective XML keyword search diversification model to measure the quality of each candidate. After that, two efficient algorithms are used to incrementally compute top-k qualified query candidates as the diversified search intentions. Two selection criteria are targeted: the k selected query candidates are most relevant to the given query and they have to cover maximal number of distinct results. At last, a comparative analysis of the existing system and proposed system demonstrates the effectiveness and efficiency of our algorithms and diversification model.*

Keywords*—XML keyword search, context-based diversification, smallest lowest common ancestor (SLCA), anchor nodes, DBLP dataset, feature selection*

I. INTRODUCTION.

Data mining is the process of extracting useful information from large volumes of data which includes various relational databases, object-oriented databases, data warehouses, transactional databases etc. With the help of data mining tools we can predict behavior and future trends and make knowledge-driven decisions. But due the enormous data, it is difficult for the user to get relevant documents. Given a set of documents in the database and a query given by user, subset of documents relevant to the query is to be retrieved in any Information Retrieval system. In Relational databases, if the user knows the schema of the database he can form suitable query for his needs using Structured Query Language (SQL). Information needed to answer a keyword query is often split across the tables/tuples.

But in case of online databases, user does not have knowledge of schema or query languages. Hence in this case the desired results can be obtained using Keyword queries. In response to a user's query the Search engines usually do keyword matching and return ranked list of all the documents containing the keywords specified in the query. The relevant documents may not be retrieved and/or retrieved instances may not be relevant. Keyword query search provides a lot of advantages to the users. They don't need to learn complex structured query languages to access information and don't have to be aware of the complex data schemas when accessing structured data allowing them to access heterogeneous databases.

In currently used HTML based search engines, HTML is a presentation language and is not able to capture semantics.XML allows for extensible element tags which can capture additional semantics. It is a simultaneously human- and machine-readable format and supports Unicode, allowing almost any information in any written human language to be communicated. Also XML can represent the most general computer science data structures: records, lists and trees

@IJAERD-2016, All rights Reserved

But XML also has some disadvantages. The result of a query could be deeply nested. Returning the deepest node which usually gives more context information is difficult to extract. Ranking mechanisms used for XML based search is different from HTML based search. Notion of proximity is more difficult in XML. No intrinsic data type support in XML. It provides no specific notion of "integer", "string", "boolean", "date", and so on. Expressing overlapping (non-hierarchical) node relationships requires extra effort. Inspite of these disadvantages XML is widely used and standard format.

II. LITERATURE SURVEY.

A. Present Work.

The query processing time is accelerating with efficient algorithms, but search intentions that are unclear and repeated in the large set of retrieved results will make users frustrated. Most of works before perform diversification as a post-processing or re-ranking step of document retrieval based on the analysis of result set and/or the query logs. In IR, keyword search diversification is designed at the topic or document level. For e.g. Agrawal et al. model user intents at the topical level of the taxonomy and Radlinski and Dumais obtain the possible query intents by mining query logs. In addition, the diversified results in IR are usually modeled at document levels.

Santos used probabilistic framework to diversify document ranking, by which web search result diversification is addressed. They also applied the similar model to discuss search result diversification through sub-queries. **Gollapudi and Sharma** proposed a set of natural axioms that a diversification system is expected to satisfy, by which it can improve user satisfaction with the diversified results.

Hasan developed efficient algorithms to find top-k most diverse set of results for structured queries over semi-structured data as [2]. A structured query can be used to express much clearer search intention of a user. Therefore, diversifying structured query results is less significant than that of keyword search results. In his work, **Panigrahi** focused on the selection of diversed item set, providing no importance to structural relationships of the items to be selected.

Liu et al. is the first work to measure the difference of XML keyword search results by comparing their feature sets. However, the selection of feature set in is limited to metadata in XML and it is also a method of post-process search result analysis. Different from the post-process methods, another type of works addresses the problem of intent-based keyword query diversification through constructing structured query candidates. Their brief idea is to first map each keyword to a set of attributes (metadata), and then construct a large number of structured query candidates by merging the attribute-keyword pairs. They assume that each structured query candidate represents a type of search intention, i.e., a query interpretation. However, these works are not easy to be applied in real application due to the following three limitations:

1. A large number of structured XML queries may be generated and evaluated
2. No guarantee that structured queries to be evaluated can find matched results due to the structural constraints
3. The process of constructing structured queries has to rely on the metadata information in XML data.

The most relevant work to ours is the approach **DivQ** in [6]**,** where **Demidova** first identified the attribute-keyword pairs for an original keyword query and then constructed a large number of structured queries by connecting the attribute-keyword pairs using the data schema (the attributes can be mapped to corresponding labels in the schema).The challenging problem is that two generated structured queries with slightly different structures may still be considered as different types of search intentions, which may hurt the effectiveness of diversification. The different diversification techniques put forth a threshold-based method to have a control on the tradeoff between relevance and diversity features in their diversification metric. But it is a big challenge for users to set the threshold value.

B. Proposed Work.

We derive different search semantics of the original query from different contexts of the xml data, which can be used to explore different search intentions of the original query. And then, we can compute the keyword search results for each search intention. In our work, the contexts can be modeled by extracting some relevant features terms of the query keywords from the xml data. To improve the precision of query diversification in structured databases or semi structured data, we have consider both structure and content of data in diversification model.

@IJAERD-2016, All rights Reserved

Our algorithms can incrementally generate query suggestions and evaluate them. The diversified search results can be returned with the qualified query suggestions without depending on the whole result set of the original keyword query. Our diversification model utilizes the mutually co-occurring feature term sets as contexts to represent different query suggestions and the feature terms are selected based on their mutual correlation and the distinct result sets together. The structures of data are considered by satisfying the exclusive property of SLCA semantics in [3].

To address the above limitations of intent based diversification models, we initiate a formal study of the diversification problem in XML keyword search, which can directly compute the diversified results without retrieving all the relevant candidates. Towards this goal, given a keyword query, we first derive the co-related feature terms for each query keyword from XML data based on mutual information in the probability theory, which has been used as a criterion for feature selection. The selection of our feature terms is not limited to the labels of XML elements. Each combination of the feature terms and the original query keywords may represent one of diversified contexts. And then, we evaluate each derived search intention by measuring its relevance to the original keyword query and the novelty of its produced results. To efficiently compute diversified keyword search, we propose one baseline algorithm and two improved algorithms based on the observed properties of diversified keyword search results as [1].

III. PROBLEM DEFINITION

Given a keyword query q containing one or more keywords and an XML data T,our aim is to find top-k expanded query candidates having high relevance and maximal diversification for q in T.In this case each query candidate represents a context or a search intention.

3.1. Feature Selection Model

3.1.1. Mutual Information Model

Mutual Information score has been used as a criterion for feature selection. It characterizes the relevancy of variables and used to select minimum redundant features. Suppose we have an XML tree T and its result set R(T). Assume Prob(x,y,T) be the probability of term x and y co-occurring in R(T) i.e.,$Prob(x,y,T) = \frac{|R(x,y,T)|}{|R(T)|}$.

$$MI(x,y,T)=Prob(x,y,T)*log\frac{Prob\,(x,y,T)}{Prob\,(x,T)*Prob\,(y,T)}, \quad (1)$$

Where Prob(x,T) be the probability of term x appearing in R(T).Similarly Prob(y,T) be the probability of term y appearing in R(T).

3.1.2. Diversification Model

We consider the probability of new queries as the relevance. But for new and diversified results we not only need to consider relevance i.e. degree of interpretation of context to the user generated query but also novelty i.e. degree of dissimilarity between the new generated query being considered and previously generated query set Q.

$$score(q_{new})=Prob(q_{new}|q,T)*DIF(q_{new},Q,T)\,, \quad (2)$$

where $Prob(q_{new}|q,T)$ is the probability of search intention q_{new} when q is the original query over data T and $DIF(q_{new},Q,T)$ represents the percentage of dissimilarity between the results produced by q_{new} and the queries in Q.

IV. PROPOSED SYSTEM.

Our approach towards feature terms extraction integrates ideas of both Sarkas et al. and Bansal et al. Towards this approach, we first extract the meaningful text information from the entity nodes in XML data during the XML data tree traversal. And then we produce a set of term pairs by scanning the extracted text. For all the term pairs we will then calculate the correlation value using the mutual information model. The term pairs extracted will be stored along with their correlation value in the correlated graph as in [1]. The term pairs with their correlation value lower than a threshold can be filtered out to reduce the size of the correlation graph. Based on the graph built we can derive top-m distinct terms as its features for each keyword in the query.

4.1. Diversification Algorithms

To efficiently compute diversified results for keyword query search we propose baseline algorithm [1] and the improved anchor based algorithm[1] based on the observed properties of diversified keyword search results.

@IJAERD-2016, All rights Reserved

4.1.1. Baseline Algorithm

For a given keyword query, we first retrieve relevant feature terms with high mutual scores from the correlated graph and the arrange them in descending order of mutual information score. And then we compute SLCA as search results for each query candidate and then finally we measure its diversification score. The top-k diversified query candidates with high diversification score can be returned.

As multiple query candidates are evaluated in which results should be distinct from each other , therefore , we need to remove the duplicated or ancestor SLCA results as [3].

Algorithm 1. Baseline Algorithm

Input: Query q with n keywords and XML data T and pre-computed correlated graph G

Output: Top-k search intentions Q and the whole result set Φ

1: M_{mxn}=getFeatureTerms(q,G);
2: while(q_{new}=GenerateNewQuery(M_{mxn})) ≠ null do
3: ϕ=null and prob_s_k=1;
4: $l_{i_xj_y}$=getNodeList($s_{i_xj_y}$,T) for $s_{i_xj_y} \in q_{new} \wedge l \le i_x \le m \wedge l \le j_y \le n$;
5: prob_s_k=$\prod_{f_{i_xj_y} \in s_{i_xj_y} \in q_{new}} ((|l_{i_xj_y}|))/getNodeSize(f_{i_xj_y}, T)$);
6: ϕ=ComputeSLCA({$l_{i_xj_y}$ });
7: prob_q_new=prob_s_k*|ϕ|;
8: if Φ is empty then
9: score(q_{new})=prob_q_new;
10: else
11: for all Result Candidates $r_x \in \phi$ do
12: for all Result Candidates $r_y \in \Phi$ do
13: if r_x == r_y or r_x is an ancestor of r_y then
14: ϕ.remove(r_x);
15: else if r_x is a descendant of r_y then
16: Φ.remove(r_y);
17: score(q_{new})=prob_q_new*|ϕ|*$\frac{|\phi|}{|\phi|+|\Phi|}$;
18: if |Q|<k then
19: put q_{new} :score(q_{new}) into Q;
20: put q_{new}:ϕ into Φ;
21: else if score(q_{new})>score({$q'_{new} \in Q$})then
22: replace q'_{new}:score(q'_{new}) with q_{new}:score(q_{new});
23: Φ.remove(q'_{new});
24: return Q and result set Φ;

4.1.2. Anchor-Based Pruning Algorithm

By observing the properties of the baseline algorithm, we recognize that computing power is much spent on computing SLCA results and filtering unqualified SLCA results from the result sets. To efficiently compute result set, we design an improved anchor-based pruning algorithm. It avoids computational cost of unnecessary SLCA results.

Towards this approach in [1], given a set of already processed query candidates Q and a new query candidate q_{new}, the generated SLCA results Φ of Q can be taken as anchors for efficiently computing SLCA results of $q_{new}$by partitioning keyword nodes of q_{new}.

Algorithm 2. Anchor-Based Pruning Algorithm

Input: A query q with n keywords , XML data T and the correlated graph G

Output: Top-k query candidates Q and result set Φ

1: M_{mxn}=getFeatureTerms(q,G);

@IJAERD-2016, All rights Reserved

2: while q_{new}=GenerateNewQuery(M_{mxn}) ≠ null do
3: Lines 3-5 in Algorithm 1;
4: if Φ is not empty then
5: for all v_{anchor}∈Φ do
6: get $l_{i_x j_y\_pre}$,$l_{i_x j_y\_des}$,and $l_{i_x j_y\_next}$ by calling for Partition($l_{i_x j_y}$,v_{anchor});
7: if∀$l_{i_x j_y\_pre}$≠ null then
8: ϕ'=ComputeSLCA({$l_{i_x j_y\_pre}$},v_{anchor});
9: if ∀ $l_{i_x j_y\_des}$ ≠ null then
10: ϕ"=ComputeSLCA({$l_{i_x j_y\_des}$},v_{anchor});
11: ϕ+=ϕ'+ϕ";
12: if ϕ" ≠ null then
13: Φ.remove(v_{anchor});
14: if ∃$l_{i_x j_y\_next}$=null then
15: Break the FOR-Loop;
16: $l_{i_x j_y}$=$l_{i_x j_y\_next}$ for $l \le i_x \le m \wedge l \le j_y \le n$;
17: else
18: ϕ=ComputeSLCA({$l_{i_x j_y}$ });
19: score(q_{new})=prob_q_new*|ϕ|*$\frac{|\phi|}{|\phi|'+|\Phi|}$;
20: Lines 18-23 in Algorithm 1;
21: return Q and result set Φ;

The results of the first query will be taken as anchors to prune the node lists of the next query for reducing its evaluation costs in line 5-16.

V. PERFORMANCE ANALYSIS.

As of today there already exists a range of benchmarks for XML database systems and also a range of performance analysis has been performed so far. XBench (Yao et al., 2004) is a family of XML benchmarks. It generates various types of XML documents (data-oriented and varying in their size) and simulates different applications by inserting and reading data using XQuery. The Michigan Benchmark (Runapongsa et al., 2006) runs 45 different queries (loading, inserting, deleting and updating) on one large XML document with a recursive structure. It mostly measures the performance of the implementation but is not suited to measure across different systems involving relational databases and SQL queries.

To evaluate the performance of the different XML data storage approaches we used a new benchmark called HYBE. It explicitly includes hybrid systems in the performance analysis and supports several query alternatives such as XPath, XQuery, SQL/XML, XUpdate, W3C XQuery Update Facility. HYBE consists of two parts: 1) a feature list considering the general functionality of the storage approach or database system and 2) a performance analysis with a fixed set of queries measuring the performance. The considered features were: maximal complexity of XML documents, support for schema information, support for different query languages, support for updating queries, and support for combining XML and relational data.

At the end, by comparing our proposed algorithm results with BaseX provided results we provide performance analysis of diversification of keyword search results.

VI. CONCLUSION.

In this system we presented an approach to search diversified results of keyword query from XML data based on the contexts of the query keywords in the data. The diversification of the contexts was measured by exploring their relevance to the original query and the novelty of their results. Furthermore, we designed three efficient algorithms based on the observed properties of XML keyword search results. Finally, we verified the effectiveness of our diversification model by analyzing the returned search intentions for the given keyword queries over our XML database and the possibility of diversified query suggestions.

We also demonstrated the efficiency of our proposed algorithms by running substantial number of queries over using our algorithms and traditional searching algorithm. From the experimental results, we get that our proposed diversification algorithms can return qualified search intentions and results to users in a short time.

@IJAERD-2016, All rights Reserved

VII. ACKNOWLEDGEMENT.

We take this opportunity to thank all those persons who rendered their full services to our work. It's with lot of happiness we are expressing gratitude to our guide **Prof. B. C. Julme** in Computer & Information Technology Department, for timely and kind help, guidance and providing us with most essential materials required for the completion of this project. We are very thankful to our guide for her indomitable guidance. We also thank **Dr. G.V. Garje**, Head of the Department, Computer & Information Technology for the cooperation extended for the successful completion of the project .

REFERENCES.

[1] Jianxin Li, Chengfei Liu, Member , IEEE, and Jeffrey Xu Yu, "Context-Basesd Diversification for Keyword Queries Over XML Data" in IEEE Transactions on Knowledge and Data Engineering , Vol. 27, No. 3, March 2015.

[2]Y. Chen, W. Wang, Z. Liu, and X. Lin, "Keyword search on structured and semi-structured data," in Proc. SIGMOD Conf., 2009, pp. 1005-1010.

[3]Y. Xu and Y. Papakonstantinou, "Efficient keyword search for smallest SLCA in xml databases" in Proc. SIGMOD Conf., 2005, pp. 537-538.

[4]H. Peng, F. Long and H. Q. Ding , "Feature Selection based on mutual information: Criteria of max-dependency, max-relevance and min-redundancy," Aug.2005Pattern Anal. Mach. Intel. vol. 27, no.8, pp.1226-1238 .

[5] R. Agarwal , S. Gollapudi, A. Halverson and S. leong " Diversifying search results, " in Proc. 2nd ACM Int. Conf. Web Search Data Mining , 2009 pp.5-14.

[6]E. Demidova, P. Fankhauser, X. Zhou and W.Nejdl, "DivQ: Diversification for keyword search over structured databases" in Proc. SIGIR Conf., 2010, pp. 331-338.

[7] J. Li. C. Liu, R. Zhou, and B. Ning, "Processing xml keyword search by constructing effective structured queries," in Advances in Data and Web Management . New York , NY, USA: Springer, 2009, pp. 88-99.

[8]J. Li. C. Liu, R. Zhou, and W. Wang, "Top-k keyword search over probabilistic xml data," in Proc. IEEE 27th Int. Conf. Data Eng., 2011,pp. 673-684.

[9]C. Sun, S. Y. Chan, and A. K. Goenka, " Multiway SLCA-based keyword search in xml Data" in Proc. 16th Int. Conf. World Wide Web, 2007,pp. 1043-1052.

[10]L. Guo, F. Shao, C. Bovet and J. Shanmugasundaram, "XRank:Ranked keyword search over xml documents" in Proc. SIGMOD Conf., 2003, pp. 16-27.

@IJAERD-2016, All rights Reserved

@IJAERD-2016, All rights Reserved