Diversification Analysis of Keyword Query from XML Data Based on Contexts of the Query Keywords in Data

[1]CHAVALI BRUGHU ABILASH YADAV, [2]N.NAVEENKUMAR

[1]M. Tech Student, Department of CS,Nalanda Institute Of Engineering and Technology, KantepudiVillage, SattenapalliMandal.GunturDist, Andhra Pradesh, India.

[2]Associate Professor, Department of CSE,Nalanda Institute Of Engineering and Technology, KantepudiVillage, SattenapalliMandal.GunturDist, Andhra Pradesh, India.

ABSTRACT –The problem of diversifying keyword search is initially considered as IR community. Most of them 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. The ambiguity of keyword query makes it difficult to effectively answer keyword queries, especially for short and vague keyword queries. To address this challenging problem, in this paper we propose an approach that automatically diversifies XML keyword search based on its different contexts in the XML data. Given a short and vague keyword query and XML data to be searched, we first derive keyword search candidates of the query by a simple feature selection model. And then, we design an effective XML keyword search diversification model to measure the quality of each candidate. After that, two efficient algorithms are proposed 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 while they have to cover maximal number of distinct results. At last, a comprehensive evaluation on real and synthetic data sets demonstrates the effectiveness of our proposed diversification model and the efficiency of our algorithms.

Keywords*: XML Keyword Search, Context-Based Diversification*

I.INTRODUCTION

Xml has been successfully used in many applications, such as that in scientific and business domains, as the standard format for storing, publishing and exchanging data. Compared witch structured query languages, such as X path and X query, keyword search is also gained popularity on XML data as it relieves users from understanding the complex query languages and the structure of the underlying data, attention due to the results are not the entire documents anymore but nested fragments.Typically, an XML document can be modelled as a node-labelled tree T. For a given keyword query Q, several semantics have been proposed to define meaningful results, for which the basic semantics is Lowest Common Ancestor. Based on LCA, the most widely adopted query semantics are Exclusive LCA (ELCA) and smallest LCA (SLCA) . SLCA defines a subset of LCA nodes, of which no of LCA is an ancestor of any other LCA, as a comparison ELCA tries to capture more meaningful results; it may take some LCAs that are not SLCAs as meaningful results. X Search, a semantic search engine for XML, is presented. X Search has a simple query language, suitable for a naive user. It returns semantically related document fragments that satisfy the user's query. Query answers are ranked using extended information-retrieval techniques and are generated in an order similar to the ranking. Advanced indexing techniques were developed to facilitate efficient implementation of X Search. The performance of the different techniques as well as the recall and the precision were measured experimentally. These experiments indicate that X Search is efficient, scalable and ranks quality results highly. We consider the problem of efficiently producing ranked results for keyword search queries over hyperlinked XML documents. Evaluating keyword search queries over hierarchical XML documents, as opposed to (conceptually) flat HTML documents, introduces many new challenges. First, XML keyword search queries do not always return entire documents, but can return deeply nested XML elements that contain the desired keywords. Second, the nested structure of XML implies that the notion of ranking is no longer at the granularity of a document, but at the granularity of an XML element. Finally, the notion of keyword proximity is more complex in the hierarchical XML data model. In this paper, we present the XRANK system that is designed to handle these novel features of XML keyword search. Our experimental results show that XRANK offers both space and performance benefits when compared with existing approaches. An interesting feature of XRANK is that it naturally generalizes a hyperlink based HTML search engine such as Google. XRANK can thus be used to query a mix of HTML and XML documents. Keyword search in XML documents based on the notion of lowest common ancestors (LCAs) and modifications of it has recently gained research interest . In this paper we propose an efficient algorithm called Indexed Stack to find answers to keyword queries based on X Rank's semantics to LCA . The complexity of the Indexed Stack algorithm is O(kd|S1| log |S|) where k is the number of keywords in the query, d is the depth of the tree and |S1| (|S|) is the occurrence of the least (most) frequent keyword in the query. In comparison, the best worst case complexity of the core algorithms in is O (k d |S|). We analytically and experimentally evaluate the Indexed Stack algorithm and the two core algorithms in . The resultsshow that the Indexed Stack algorithm outperforms in terms of both CPU and I/O costs other algorithms by orders of magnitude when the query contains at least one low frequency keyword along with high frequency keywords. This is important in practice since the frequencies of keywords typically vary significantly. Keyword search is integrated in many applications on account of the convenience to convey users' query intention. Recently, answering keyword queries on XML data has drawn the attention of web and database communities, because the success of this research will relieve users from learning complex XML query languages, such as X Path/X Query, and/or knowing the underlying schema of the queried XML data. As a result, information in XML data can be discovered much easier. To model the result of answering keyword queries on XML data, many LCA (lowest common ancestor) based notions have been proposed. In this paper, we focus on ELCA (Exclusive LCA) semantics, which is first proposed by Guo et al. and afterwards named by X u and Papakonstantinou. We propose an algorithm named Hash Count to find ELCAs efficiently. Our analysis shows the complexity of Hash Count Algorithm is O (kd|S1|), where k is the number of keywords, d is the depth of the queried XML document and |S1| is the frequency of the rarest keyword. This complexity is the best result known so far. We also evaluate the algorithm on a real DBLP dataset, and compare it with the state-of-the-art algorithms. The experimental results demonstrate the advantage of Hash Count Algorithm in practice. Keyword search is a proven, user-friendly way to query HTML documents in the World Wide Web. We propose keyword search in XML documents, modeled as labeled trees, and describe corresponding efficient algorithms. The proposed keyword search returns the set of smallest trees containing all keywords, where a tree is designated as "smallest" if it contains no tree that also contains all keywords. Our core contribution, the Indexed Lookup Eager algorithm, exploits key properties of smallest trees in order to outperform prior algorithms by orders of magnitude when the query contains keywords with significantly different frequencies. The Scan Eager variant is tuned for the case where the keywords have similar frequencies. We analytically and bexperimentally evaluate two variants of the Eager algorithm, along with the Stack algorithm. Finally, we extend the Indexed Lookup Eager algorithm to answer Lowest Common Ancestor (LCA) queries

II.RELATED WORK

In this by considering the keyword and its relevant context in XML data , searching should be done using automatically diversification process of XML keyword search is the major area of concern .

In this for structured and semi-structured data, various state-of-the-art techniques are discussed for keyword search. In this query optimization , ranking phases , top k important query processing is discussed. Different data models such as XML , graph-structured data is discussed. Application of these concepts is also discussed in which keyword based search is having prime importance. In this paper some problems like Diverse Data Models, Query Forms: Complexity versus Expressive Power , Search Quality Improvement , Evaluation are also discussed .

XRANK system is discussed in this paper. Ranked search technique over XML data is considered here. In this paper space saving and performance gaining techniques such as index structure and query evaluation are also focused. XRANK can help in searching for HTML as well as XML documents. Disadvantage: For instance, authors have currently taken a document-centric view, where they assume that query results are strictly hierarchical. Index maintenance is major problem for effective search and which is bottleneck area .

In this SLCA-based keyword search approach is discussed. Queries called the Multiway - SLCA approach (MS) is helpful to promote the keyword search beyond and old methods like AND / OR. After LCA analysis improved algorithms are put to solve search problems based on keywords .

In this Indexed Lookup Eager and Scan Eager, algorithms are discussed. XML search based on keyword according to SLCA semantics is prime topic of discussion and for this these algorithm are used. Instant search result is the beauty of theses algorithm. XKSearch architecture implementation is discussed in it. The XKSearch system inputs a list of keywords and returns the set of Smallest Lowest Common Ancestor nodes .

Query and information relevance is calculated so that unnecessary checks are avoided and effective search is achieved. Hence effective text retrieval and summarization is achieved. The Maximal Marginal Relevance (MMR) achieves the stopping of redundancy. This approach provides very much relevant data in terms of search result to the end user by effectively minimizing the redundancy.

In this paper Risk of dissatisfaction of user is major area of concern. To minimize it systematic approach to diversifying results is discussed in it. For this several techniques such as NDCG, MRR, and MAP are discussed in detail in it. A Greedy Algorithm for Diversification used in it. Among the search result user should find most relevant data is the aim of diversification. Also another aim of this paper is to minimize the rank of best fitted result .

This paper also uses greedy approach. Different datasets are considered in this to get approach tested thoroughly and relevant document in terms of search result is expected as search result .

In this using test collection based on TREC question answering track this paper discussed the framework which achieves novelty and diversity. In this approach document is linked with the relevant information in it. Chunk of information is in this way get attached with document and which is helpful in at time of search. This piece of information is having content as well as document properties. The major drawback of this approach is that unusual features of document may cause judging error. Some raw data related with the document may delay the search result .

Using past query and its analysis provides proper direction for diversification. Past query reformulation provides exact query related behavior of user. Client data request, his ranked structure and query is observed and analysed at client side for proper diversified result. Large query logs are analyzed in this paper from search engine .

In this single swap and multi swap algorithms are used in this paper. On structured data differentiation of search results is carried out. Degree of difference is quantified so that it represents the accuracy of search result. Features from the search result are traced and this result is prominently considered in calculation .

In this by considering query result and its redundancy, new scheme named re-ranking query interpretations is discussed to diversify the search result. For sub-topics and relevance new proposed technique such as propose α-n DCG-W and WS-recall is promoted in it. Algorithm named as Diversification algorithm is used in it. For database query search query similar measure and greedy algorithm is used to obtain diversified query interpretation and its relevance .

III. EXISTING SYSTEMS

The problem of diversifying keyword search is firstly studied in IR community. Most of them perform diversification as a post-processing or reranking 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. 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 is limited to metadata in XML and it is also a method of post-process search result analysis.

A. Disadvantage of existing system:

When the given keyword query only contains a small number of vague keywords, it would become a very challenging problem to derive the user’s search intention due to the high ambiguity of this type of keyword queries.

Although sometimes user involvement is helpful to identify search intentions of keyword queries, a user’s interactive process may be time-consuming when the size of relevant result set is large.

It is not always easy to get this useful taxonomy and query logs. In addition, the diversified results in IR are often modeled at document levels.

A large number of structured XML queries may be generated and evaluated.

There is no guarantee that the structured queries to be evaluated can find matched results due to the structural constraints.

The process of constructing structured queries has to rely on the metadata information in XML data.

IV. PROPOSED SYSTEM

To address the existing issues, we will develop a method of providing diverse keyword query suggestions to users based on the context of the given keywords in the data to be searched. By doing this, users may choose their preferred queries or modify their original queries based on the returned diverse query suggestions. □ To address the existing limitations and challenges, 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 (also denoted as specific search intentions). 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 improvedalgorithms based on the observed properties of diversified keyword search results.

A. Advantages of proposed system.

Reduce the computational cost.

Efficiently compute the new SLCA results

We get that our proposed diversification algorithms can return qualified search intentions and results to users in a short time.

V.DESIGN AND IMPLEMENTATION

Modules:

- Pre-processing
- Query Initialization
- Rewriter
- DOM Tree Construction

Pre-Processing

Data Preparation and filtering steps can take considerable amount of processing time. Includes cleaning, normalization, transformation, feature extraction and selection etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis.

Query Initialization

In this module, user has to give the query for the further propose and to obtain the optimized query. Here we consider the static tables and data's.

Rewriter

In this module, have to rewrite the user given query into the representation format based on the selection, project and joint. Based on this rewrites query only have to prepare the execution plans. The selection is represented by sigma then the projection is represented by pi then the joint is represented by ><.

DOM Tree Construction:

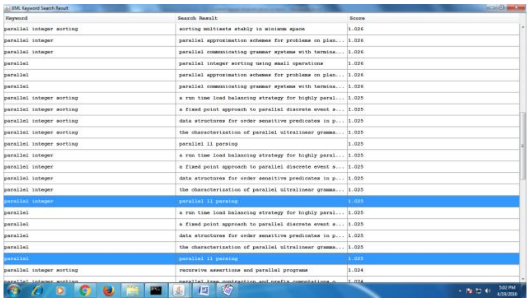
Get the Input Query Result Page from the User. Given a query result page, the DOM Tree Construction module first constructs a DOM tree for the page rooted in the <HTML> tag. Each node represents a tag in the HTML page and its children are tags enclosed inside it. Each internal node n of the tag tree has a tag string tsn, which includes the tags of n and all tags of n's descendants, and a tag path tpn, which includes the tags from the root to n. **Data Region Extraction**

The Data Region Extraction module identifies all possible data regions, which usually contain dynamically generated data, top down starting from the root node. We first assume that some child sub trees of the same parent node form similar data records, which assemble a data region. Many query result pages some additional item that explains the data records, such as a recommendation or comment, often separates similar data records. Hence, we propose a new method to handle non-contiguous data regions so that it can be applied to more web databases. The data region Extraction algorithm discovers data regions in a top-down manner. Starting from the root of the query result page DOM tree, the data region identification algorithm is applied to a node n and recursively to its children ni, i =1 . . .m. Compute the similarity simij of each pair of nodes ni and nj, i , j = 1 . . .m and i # j, using the node similarity calculation method. The data region identification algorithm is recursively applied to the children of ni only if it does not have any similar siblings. Segment the data region into data records using the record segmentation algorithm.

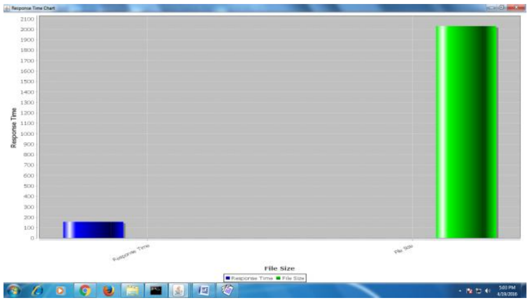
VI .EXPERIMENTAL RESULTS

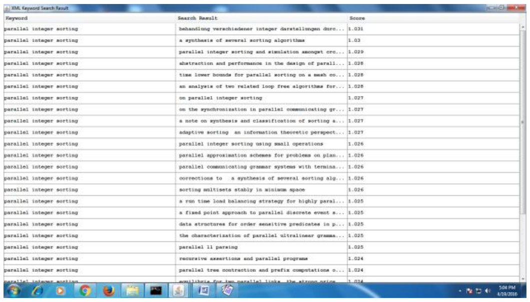
A large variety of structured XML queries could also be generated and evaluated. There's no guarantee that the structured queries to be evaluated will notice matched results because of the structural constraints. The method of constructing structured queries must rely on the metadata information in XML data. to deal with the present problems, we'll develop a way of providing various keyword question suggestions to users supported the context of the given keywords within the data to be searched. By doing this, users could select their most well-liked queries or modify their original queries supported the returned various query suggestions. To deal with the present limitations and challenges, we tend to initiate a proper study of the diversification drawback in XML keyword search, which may directly reckon the distributed results while not retrieving all the relevant candidates. Towards this goal, given a keyword question, we tend to initially derive the co-related feature terms for every query keyword from XML knowledge supported mutual information within the probability theory that has been used as a criterion for feature choice. The choice of our feature terms isn't restricted to the labels of XML components. Every combination of the feature terms and therefore the original query keywords could represent one in all distributed contexts (also denoted as specific search intentions). And then, we tend to valuate every derived search intention by activity its connection to the first keyword query and therefore the novelty of its made results. To with efficiency calculate diversified keyword search, we have a tendency to propose one baseline formula and two improved algorithms supported the ascertained properties of distributed keyword search results. Scale back the machine value; with efficiency compute the new SLCA results that our planned diversification algorithms will return qualified search intentions and results to users in a very short time.

Experimental Results are shown in the screens given below.

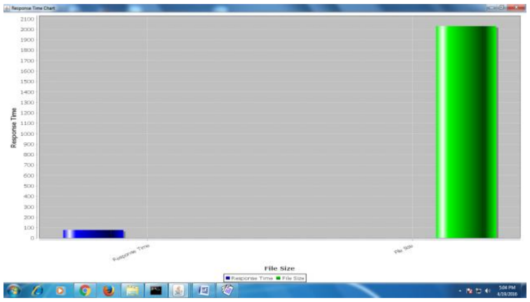


Keyword	Search Result	Score
parallel integer sorting	sorting multisets stably in minimum space	1.026
parallel integer	parallel approximation schemes for problems on plan...	1.026
parallel integer	parallel communicating grammar systems with termina...	1.026
parallel	parallel integer sorting using small operations	1.026
parallel	parallel approximation schemes for problems on plan...	1.026
parallel	parallel communicating grammar systems with termina...	1.026
parallel integer sorting	a run time load balancing strategy for highly paral...	1.025
parallel integer sorting	a fixed point approach to parallel discrete event s...	1.025
parallel integer sorting	data structures for order sensitive predicates in p...	1.025
parallel integer sorting	the characterization of parallel ultralinear gramma...	1.025
parallel integer sorting	parallel ll parsing	1.025
parallel integer	a run time load balancing strategy for highly paral...	1.025
parallel integer	a fixed point approach to parallel discrete event s...	1.025
parallel integer	data structures for order sensitive predicates in p...	1.025
parallel integer	the characterization of parallel ultralinear gramma...	1.025
parallel integer	parallel ll parsing	1.025
parallel	a run time load balancing strategy for highly paral...	1.025
parallel	a fixed point approach to parallel discrete event s...	1.025
parallel	data structures for order sensitive predicates in p...	1.025
parallel	the characterization of parallel ultralinear gramma...	1.025
parallel	parallel ll parsing	1.025
parallel integer sorting	recursive assertions and parallel programs	1.024
parallel integer sorting	parallel tree contraction and prefix computations o...	1.024





Keyword	Search Result	Score
parallel integer sorting	behandlung verschiedener integer darstellungen durc...	1.031
parallel integer sorting	a synthesis of several sorting algorithms	1.03
parallel integer sorting	parallel integer sorting and simulation amongst crc...	1.029
parallel integer sorting	abstraction and performance in the design of parall...	1.028
parallel integer sorting	time lower bounds for parallel sorting on a mesh co...	1.028
parallel integer sorting	an analysis of two related loop free algorithms for...	1.028
parallel integer sorting	on parallel integer sorting	1.027
parallel integer sorting	on the synchronization in parallel communicating gr...	1.027
parallel integer sorting	a note on synthesis and classification of sorting a...	1.027
parallel integer sorting	adaptive sorting an information theoretic perspect...	1.027
parallel integer sorting	parallel integer sorting using small operations	1.026
parallel integer sorting	parallel approximation schemes for problems on plan...	1.026
parallel integer sorting	parallel communicating grammar systems with termina...	1.026
parallel integer sorting	corrections to a synthesis of several sorting alg...	1.026
parallel integer sorting	sorting multisets stably in minimum space	1.026
parallel integer sorting	a run time load balancing strategy for highly paral...	1.025
parallel integer sorting	a fixed point approach to parallel discrete event s...	1.025
parallel integer sorting	data structures for order sensitive predicates in p...	1.025
parallel integer sorting	the characterization of parallel ultralinear gramma...	1.025
parallel integer sorting	parallel ll parsing	1.025
parallel integer sorting	recursive assertions and parallel programs	1.024
parallel integer sorting	parallel tree contraction and prefix computations o...	1.024
parallel integer sorting	equilibria for two parallel links the strong price...	1.024



VII.CONCLUSION

In this paper, we first introduce an approach to search the diversified results of a keyword query from the XML data based on contexts of the query keywords in the data. The diversification of a context was calculated by exploring their relevance measure to the original query and the novelty of their results. Moreover, we have designed three efficient algorithms based on the observed properties of the XML keyword search results. Finally, we have verified the effectiveness of our diversification model by the analysis of the returned search intentions for all the given keyword queries over the DBLP data set based on nDCG measure and possibility of the diversified query suggestions.

References

[1] 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.

[2] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "Xrank: Ranked keyword search over xml documents," in Proc. SIGMOD Conf., 2003, pp. 16–27.

[3] C. Sun, C. 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.

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

[5] 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.

[6] J. G. Carbonell and J. Goldstein, "The use of MMR, diversitybased reranking for reordering documents and producing summaries," in Proc. SIGIR, 1998, pp. 335–336.

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

[8] H. Chen and D. R. Karger, "Less is more: Probabilistic models for retrieving fewer relevant documents," in Proc. SIGIR, 2006, pp. 429–436.

[9] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Buttcher, and I. MacKinnon, "Novelty and diver-€ sity in information retrieval evaluation," in Proc. SIGIR, 2008, pp. 659–666.

[10] A. Angel and N. Koudas, "Efficient diversity-aware search," in Proc. SIGMOD Conf., 2011, pp. 781–792.

[11] F. Radlinski and S. T. Dumais, "Improving personalized web search using result diversification," in Proc. SIGIR, 2006, pp. 691– 692.

[12] Z. Liu, P. Sun, and Y. Chen, "Structured search result differentiation," J. Proc. VLDB Endowment, vol. 2, no. 1, pp. 313–324, 2009.

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

[14] 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.

Author Details

Author 1:

Chavalibrughuabilashyadav,M.Tech Scholar, Computer Science, Nalanda Institute Of Engineering and Technology, Approved by AICTE, Affiliated to JNTUK, GunturDist,A.P,India, Pin:522438. Mail Id: brughuabilash@gmail.comPhone:9553303216

Author 2:

N.NaveenKumar, Associate Professor, Computer Science and Engineering, Nalanda Institute Of Engineering and Technology, Approved by AICTE, Affiliated to JNTUK, Guntur Dist,A.P., India, Pin:522438