International Journal of Research
Available at
https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Keyword Search Over Xml Data Using Context Based Diversification

Musthyala.Swapna & Mrs. CH. Srivalli,
[1]M.Tech Student, [2]Assistant Professor,
[1,2]Aurora's Scientific, Technological and Research Academy, Bandlaguda, Hyderabad-500005.
mswapna94@gmail.com & pravar.sri@gmail.com

***Abstract**:*

While keyword query empowers ordinary users to search vast amount of data, 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

Data Mining, Search Engine Optimization, XML Dataset, Baseline Algorithm, Candidate Keyword ,XML Keyword search, feature selection, diversification process.

1. 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[1],[2],[3],[4],[5],[6],[7], due to the results are not the entire documents anymore but nested fragments.

Typically, an XML document can be modeled as a node-labeled tree T. For a given keyword query Q, several semantics [1],[2], [5], 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) [2], [4] and smallest LCA (SLCA) [5],[7]. 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 [10, 14, 22]. 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 [10]. 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 [10] is O (k d |S|). We analytically and experimentally evaluate the Indexed Stack algorithm and the two core algorithms in [10]. The results show 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 experimentally evaluate two variants of the Eager algorithm, along with the Stack algorithm [13]. Finally, we extend the Indexed Lookup Eager algorithm to answer Lowest Common Ancestor (LCA) queries

2. System Analysis

A. Existing System

The problem of diversifying keyword search is firstly studied in 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.

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.

Disadvantages 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 these 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.

B. 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 improved algorithms based on the observed properties of diversified keyword search results.

Advantages of Proposed System:

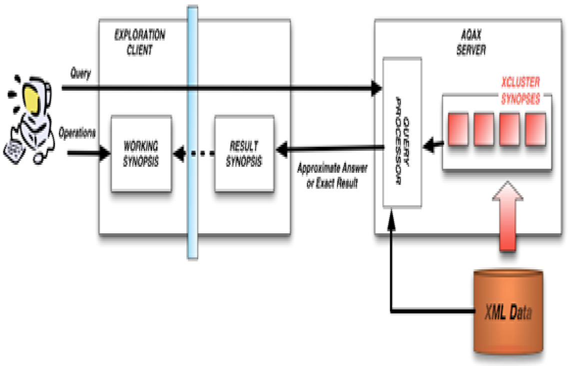
- Reduce the computational cost.
- Efficiently compute the new SLCA results
- We get that our proposed diversification algorithms can return qualified search intentions and results to users in a short time.

3. Design and Implementation

Modules:

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

Modules Description:



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. The table names and attributes are prefix, name, sex, dob, addr, city, zip, mailid, ph, date, Age, problem, Height,Weight,BP_Before,BP_After.

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.

4. Conclusion and Future Scope

In this paper, we first 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 DBLP data set based on the DCG measure and the possibility of diversified query suggestions. Meanwhile, we also demonstrated the efficiency of our proposed algorithms by running substantial number of queries over both DBLP and XMark data sets.

5. 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. B€uttcher, and I. MacKinnon, “Novelty and diversity 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.

[15] H. Peng, F. Long, and C. H. Q. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[16] W. Shen, Q. Shang, S. Shen, Y. Fan, and X. Zen, “A high-throughput VLSI architecture for deblocking filter in HEVC,” in Proc. IEEE Int. Symp. Circuits Syst., May 2013, pp. 673–676.

[17] J. Zhu, D. Zhou, G. He, and S. Goto, “A combined SAO and de-blocking filter architecture for HEVC video decoder,” in Proc. IEEE Int. Conf. Image Process., Sep. 2013, pp. 1967–1971.