International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Diversifies XML Keyword Search Based on its Different Contexts in the XML Data

Pooja Chudiwal[1], A. C. Lomte[2]

[1]Pune University, JSPM (BSIOTR), Wagholi 411005

[2]JSPM (BSIOTR), Pune University, Wagholi 411005

Abstract: ***Newly, keyword search has concerned a great deal of attention in XML database. It is inflexible to directly improve the XML keyword search Processing. XML Keyword Search by Constructing Effective Structured Queries. An efficient keyword search method for data centric general documents. It has become a everywhere method for users to access text data in the face of information detonation. In which give an overview of the state-of-the-art techniques for sustaining keyword search on structured and semi-structured data[2], including query result definition, result generation and top-k query processing, query cleaning, performance optimization, and search quality evaluation. The uncertainty of keyword query, makes it difficult to effectively answer keyword queries. To address this problem, propose an approach that diversifies XML keyword search based on XML data. In this firstly define the new problem of studying top-k keyword search over XML data, which is to recover k SLCA results[9] with the k highest probabilities of existence and keyword search candidates of the query by a simple selection model then design an effective XML keyword search diversification reproduction for the qualification of query candidate and then propose two efficient algorithms are compute top-k experienced query candidates as the diversified search intentions. Two selection criteria are targeted. At last, the valuation on real data sets demonstrates the effectiveness of diversification model and the good organization of algorithms.***

Keywords: Extensive markup language (XML); Keyword Search Engine; Context-based diversification.

1. Introduction

There is no uncertainty that XML is rapidly attractive data formats. One of the strengths of XML is that it can be used to represent structured facts (records) as well as unstructured facts (i.e., text). For example, XML can be used in a hospital to signify (structured) information about patients(e.g. ,name, address, birth-date) and (unstructured) [2] comments from doctors. To take benefit of this strength, it is important to have tools that can work successfully with both kinds of data; it is important to have XML query languages which select records from the structured part of an XML document and investigate for information in text. To incorporate keyword search into XML query processing is to query several XML documents at the same time. Searching for information is an necessary component of our lives. [10]Web search engines are widely used for searching textual documents, images, and videos. There are also huge collections of structured and semi-structured data both in enterprises, such as relational databases, XML, data extracted from text documents, etc. Result Generation and Top-k Query Processing [1]. Set up representative algorithms for query result generation and efficient top-k query processing. For keyword search on XML data, encoding and indexing schemes have been broken. For keyword search on relational databases, existing approaches are mainly based on candidate network (CN) production, and change on processing and optimization techniques to execute the CNs. Purpose of Keyword searches on the web are for information examination, and essentially have multiple relevant results. Such queries are secret as informational queries, where a user would like to investigate, evaluate, compare, and synthesize multiple relevant results for information detection and decision making, in contrast to navigational queries whose plan is to reach a particular website. With the use of a few query terms in the current search model, the user expresses her information need. In which the small number of terms often specify the aim perfectly. In the absence of clear information representing user aim, the search engine needs to "guess" the results that are most likely to satisfy different intents. In particular, for an unclear query such as eclipse, the search engine could either take the probability ranking principle approach of taking the "best guess" intent and showing the results [5], or it could prefer to present search results that maximize the probability of a user with a random intent result at least one relevant document on the results page.

Usually, a diversification function can be thinking of as pleasing two application specific inputs , for a given query a consequence function that specifies the consequence of document, and a distance function that captures the pair-wise similarity between any pair of documents in the set of relevant results for a given query. In the situation of web search, one can use the search engine's ranking function1 as the relevance function .In search, it is common to introduce miscellany by mixing in different interpretations of a query. Keyword searches are a widely used for querying in document systems and the World Wide Web. Traditional query processing approaches on relational and XML databases are forced by the query constructs imposed by the languages such as structure query language and XQuery.

With the titanic amount of new information, keyword search is vital for users to access text datasets. These datasets include textual documents, XML documents, and relational

tables. Simply typing in keywords as queries, Users use keyword search to retrieve documents Compared with keyword search methods in information retrieval (IR) that

wish to find a list of relevant documents, keyword search approaches in structured and semi structured data (denoted as DB and IR) [2] [3] focus on specific information contents, e.g., fragments fixed at the smallest lowest common ancestor (SLCA) nodes of a given keyword query in XML. And adopt the well-accepted SLCA semantics as a result metric of keyword query over XML data[9].

Instance 1, think a query q = {database, query} over the DBLP data set. In which 21,260 publications containing the keyword "database", and 9,896 publications containing the keyword "query", which contribute 2,040 results that contain the two known keywords simultaneously. If suppose directly read the keyword search results, it would be instant overriding and not user friendly. It takes 54.22 s for just computing all the SLCA results of q by using X-Rank. Even if the system processing time is acceptable by accelerating the keyword query estimate with well-organized algorithms the uncertain and frequent search intentions in the large set of retrieved results will make users irritated. To address the problem, gain different search semantics from the contexts of the XML data, which can be used to explore different searches intentions of the original query.

Table 1: Zenith 10 chosen aspect Terms of q

Keyword	Aspects
Folder	relational; protein; disseminated; image; explore; large; reproduction; system
Query	words; extension; log; resourceful; transformation; evaluation

The contexts can be modeled by extracting most important aspect terms of the query keywords from the XML data, as shown in Table 1. And then can compute the keyword search results for each search plan. Table 2 shows part of statistic information of the answers related to the keyword query q, which classifies each ambiguous keyword query into different search intentions. 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 improvement based on the analysis of result set and the query logs.

These works are difficult to be applied in real application due to the following three limitations:

- Generation and evaluation of many structured XML queries can be done;
- The structured queries to be evaluated will find matching results due to the structural constraints is not assured;
- The process of constructing structured queries depends on the metadata information in XML data

Table 2: Part of Statistic Information for q

	Words	Growth	Optimization	Estimate
Result	71	5	68	13
	Log	Capable	Circulated	Semantic
Result	12	17	50	14
	Words	Growth	Optimization	Estimate
Result	40	0	20	8
	Log	Capable	Circulated	Semantic
Result	2	11	5	7

When the user enters the keywords "XML, VLDB", user expects to get the sub tree circled by the double dotted lines as the output, with the two authors associated to the answer as corresponding nodes related to the keyword query. DBLP is a distinctive data-centric XML document. Many of the existing studies use the sub trees, which have all the keywords as the results of the given keyword query.

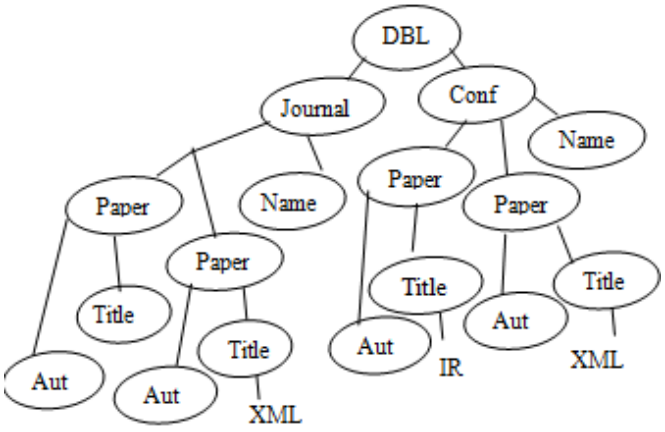

Figure 1: Example XML document

2. Literature Survey

Literature survey is the process of presenting the summary of the journal articles, study resources and conference papers. So this section helps to study the related topics summarized.

2.1 Model Definition

2.1.1 Selection Model

Assume there is an XML tree T with its sample result set R(T). Consider Prob(x, T) to be the probability of term x appearing in R(T), i.e., Prob(x, T) = |R(x,T)| / |R(T)| where |R(x,T)| is number of results which contains x. Let Prob(x,y,T) be probability of terms x and y occurring in R(T), i.e., Prob(x, y, T) = |R(x, y, T)| / | R(T)|. If terms x and y are autonomous then know that x does not give any information about y and y does not give any information about x, so their mutual information is zero. So when working, use the popularly-accepted mutual information model as follows:

MI(x,y,T) = Prob(x, y, T) * log Prob(x, y, T) / Prob(x;,T) * Prob(y, T)

It is necessary to find a set of feature terms for each term in XML data where the feature terms can be chosen in any way, e.g., top-m marks terms or the feature terms where a given threshold based on domain applications is lower than mutual values. e.g.,"to retrieve information by expanding the query" available in Encyclopedia of Database Systems in 2009. To change the generated query to search specific publications of query expansion over relational database replace the term "systems" with "relational", as no work is reported to the problem over relational database in DBLP data set the returned results are empty.

2.1.2 Contributions of this study

Here, proposed terms help in constraining resulting solution by using a set of natural axioms for result diversification that

help in the option of an idea function. This work is analogous to the recent work on axiomatization of ranking and clustering systems. We revise the functions that occur out of the condition of fulfilling a set of simple properties and display an unfeasibility result which states that all the properties cannot be satisfied by using the diversification function f. To conclude we do a groundwork classification of the alternative of an objective and its underlying properties) using the natural events of significance and innovation Finally, we present an appraisal method that computes the actions based on the pages in, public-domain assessment data set implemented for working with a diversification system.

2.1.3 Axioms of diversification

Here, f is satisfies the set of axioms given, where they appear perceptive for the setting of diversification. Also, we show that any correct subset of these axioms is maximal, i.e. all axioms cannot be satisfied by the diversification function. This gives a natural method of selecting between a variety of purpose functions, as one can prefer the necessary properties for any exacting diversification system.

Consistency: The output of the ranking should not be changed by making the output documents more appropriate and more varied and creating other documents less applicable and less varied Now, given any two functions $\alpha : U \rightarrow R+$ and $\beta : U \times U \rightarrow R+$, we change the relevance and weight functions as follows:

$$w(u) = \begin{cases} w(u) + \alpha(u), & u \in S \wedge k \\ w(u) - \alpha(u), & \text{otherwise} \end{cases}$$

$$d(u, v) = \begin{cases} d(u, v) + \beta(u, v), & u, v \in S \wedge k \\ d(u, v) - \beta(u, v), & \text{otherwise} \end{cases}$$

The ranking function f must be such that it is still maximized by S ^ k.

- Richness: It states that given the right choice of relevance and distance function any possible set of output could be achieved.
- Scale invariance: This property states that the given set of selection function should be insensate to the scaling of the input functions.
- Strength of Relevance: It ensures that no function f ignores the relevance function. Now, the following properties should cling to for any x ∈ S.

Acquire a innovative relevance function w 0 (·) from w(·), where w 0 (·) is matching to w(·) except that
w 0 (x) = a0 > w(x).
f(S, w0 (·), d(·, ·), k) = f(S, w(·), d(·, ·), k) + δ0

Adjust the relevance function w(·) to get a new relevance function w 0 (·) which is similar to w(·) except that w 0 (x) = a1 < w(x). Now,
f(S, w0 (·), d(·, ·), k) = f(S, w(·), d(·, ·), k) − δ1

2.1.4 Diversification Function:

I. As described in the introduction, our high-level purpose is to opt for a small group of items that are different with respect to their component features or proportions from a huge quantity of multidimensional items. Here, we take a look at numerous variants of the dilemma all of which symbolize the complex goal of diversification, and eventually join to a meticulous crisis formulation that we spotlight on for the remaining paper. Consider that there are two versions of the diversification problem depending on whether the complete set U is existing in the algorithm prior to it starts selecting the items in S. In this case, the diversification algorithm requests to select news items as they turn up i.e. devoid of having admission to the total input set of items. We entitle this the online version of the problem. Therefore, in the online model, on the onset of an item j, the algorithm must straight away either select or abandon it, subject to the limitation that the total number of selected items cannot surpass B. We presuppose that every item in the online input stream is drawn from some probability distribution on a set of features that is unidentified to the algorithm.

Perhaps the simplest objective function D that one can aspire for while choosing a representative subset from a big set of items is to exploit the sum of fractional coverage's of all the features, i.e. D(c) = X i∈F ci. A diversification function that reflects this perception is D(c) = X i∈F pi.

2.1.5 Keyword Search Diversification Model:

In our model, we not only consider the prospect of new generated queries, i.e., relevance, we also take into account their original and dissimilar results[4]. To represent the relevance and novelty of keyword search jointly two criteria should be fulfilled

1) The generated query qnew has the maximal probability to understand the contexts of unique query q with regards to the data to be searched.
2) The generated query qnew has a maximal distinction from the earlier generated query set Q. Thus, the aggregated scoring function:

Score(qnew) = Prob(qnew | q, T) * DIF(qnew, Q, T);
where Prob(qnew | q, T) represents the probability that qnew is the exploration purpose when the original query q is issued in excess of the data T, DIF(qnew,Q,T) represents the percentage of outcome that are created by qnew, but not by any previously generated query in Q.

Calculating the Probabilistic Relevance of an Intended Query Suggestion w.r.t. the Original Query Based on the Bayes Theorem, we have
Prob(qnew | q, T) = Prob(q | qnew, T) * Prob(qnew | T) Prob(q | T)

2.1.6 Keyword Search Diversification Model Algorithm:

The proposed system uses a baseline algorithm to recover the diversified keyword search outcome. Then to progress the competence of the keyword search diversification by using the intermediary results[8], two anchor-based pruning algorithms are calculated. Specified a keyword query, the perceptive of the baseline algorithm is to first recover the applicable feature terms with elevated common scores from the term connected graph of the XML data T after that produce list of query candidates that are sorted in the downward order of total shared scores; and finally calculate the SLCAs as keyword search results[9] for every query candidate in addition to measure its diversification score.

2.2 Existing System

Recently much research interest has been given to KEYWORD search on structured and semi-structured data, as using it users can recover information lacking the need to study complicated query languages and database structure. Separate that XML data into compressed linked vital sub-trees, to imprison the structural information in the XML document. Due to which, keyword search using XML data can be extra creative. A range of data models will be discussed, with relational data, data streams, workflows, XML data and graph-structured data. We also confer applications that are based upon keyword search, such as query generation, analytical processing and keyword based database selection. Lastly we categorize the problems and opportunities of future research to progress in the field.

Here I extend the XML-QL query language using keyword based search capabilities. At first we illustrate our XML data model and continue by relating the syntax and the semantics of the existing language. Data Model is a significant question whether an XML query is calculated on a set of XML elements, on a set of single XML document or on a set of XML documents. Regarding this delicate point, we make the subsequent statement we query sets of XML documents. Call a set of documents an XML data set. XML elements in a data set can be divided based on their types: an XML element which has the form <tag_name>...</tag_name> is of type tag_name. Thus, an XML data set can have numerous elements of type document.

3. Proposed Architecture

The keyword search approaches in structured and semi structured data (referred as DB and IR) focus more on detailed information contents. Keyword searches in text documents ought to locate the documents that have all the keywords. The outcome should have associated significant information.

3.1 System Architecture

Figure: The Architecture of Context Based Diversification for keyword query over XML data

4. Algorithm

4.1 Baseline Solution

Algorithm 1: Baseline Algorithm
Input: query q with n keywords, XML data T and the term connected graph G
Output: Top-k search intentions Q and the entire result set F
1: Mm*n = getFeatureTerms(q, G);
2:while(qnew=GenerateNewQuery(Mm*n)) =\= null do
3: f = null and prob_s_k = 1;
4:prob s k =Q Fixjy2sixjy2qnew(jlixjy j getNodeSize(fixjy ,T));
5: f = ComputeSLCA({lixjyg);
6: prob q new = prob s k * jfj;
7: if F is empty then
8: score(qnew) = prob q new;
9: else
10: for every Result candidates rx 2 f do
11: f:remove(rx);
12: else if rx is a child of ry then
13: F:remove(ry);
14: score(qnew) = prob q new * jfj* jfj jfj) jFj ;
15: if jQj < k then
16: put qnew : score(qnew) into Q;
17: put qnew : f into F;
18: else if score(qnew) > score(fq0new 2 Qg) then
19: replace q0new : scoreðq0newÞ with qnew : score(qnew);
20: F:remove(q0new);
21: return Q and result set F;

4.2 Max-sum diversification

A normal bi-criteria purpose is to exploit the sum of the relevance and dissimilarity of the chosen set. This aim can be programmed in terms of our formulation in terms of the function f(S), which is defined as follows:
$f(S) = (k - 1)X\ u \in S\ w(u) + 2\lambda\ X\ u,v \in S\ d(u, v)$
where $|S| = k$, and $\lambda > 0$ is a parameter depicting the trade-off among relevance and similarity.

Algorithm 2: Algorithm for MAXSUMDISPERSION

Input : Universe U, k
Output: Set S (|S| = k) that maximizes f(S)
Initialize the set S = ∅
for i ← 1 to b k 2 c do
Find (u, v) = argmaxx,y ∈U d(x, y)
Set S = S ∪ {u, v}
Remove all edges from E that are events to u or v
end
If k is odd, add an random document to S

4.3 Max-min diversification

The second bi-criteria purpose we have, maximizes the minimum relevance and distinction of the selected set. This intention can be encoded in terms of the function f(S), which is defined as follows:
$f(S) = \min u \in S\ w(u) + \lambda \min u,v \in S\ d(u, v)$

Algorithm 3: Algorithm for MAXMINDISPERSION

Input: Universe U, k
Output: Set S (|S| = k) which maximizes f(S)
Initialize set S = ∅;
Find (u, v) = argmaxx,y∈U d(x, y) and
set S = {u, v}; For any x ∈ U \ S,
define d(x, S) = minu∈S d(x, u);
while |S| < k do
Find x ∈ U \ S such that x = argmaxx∈U\S d(x, S);
Set S = S ∪ {x};

5. Evaluation

For the evaluation we use the structure to distinguish differences among diversification objectives. Gears are switched to inspect the further method of distinguishing

among diverse objectives, specifically through their experimental performance. Here, we describe the choice of the objective function and its underlying axioms using two distinguished measures relevance and novelty. We exhibit the convenience of the diversification framework by conducting two sets of experiments. We will use this idea of treatment of a topic to use the Wikipedia data set for evaluating the efficiency of the diversification algorithm. The idea behind the evaluation of novelty for a list is to compute the number of categories represented in the list. With the rising recognition of supporting keyword search on structured data, there is an growing need to supply an evaluation framework to review and lead the system design. An axiomatic framework has been made for evaluating keyword search strategies on XML data. Assistance from the community are exceedingly demanded for increasing inclusive frameworks for evaluating the recovery and ranking strategies of keyword search on a variety of structured data models. We will discuss assessment framework for keyword search engines. That is based on empirical estimation using benchmark data for XML keyword search.

6. Conclusions

Informational queries are persistent in web search, where a user likes to examine assess evaluate and blend multiple relevant results for information detection and decision making. First presented approach to search diversified results of keyword query from XML data based on the contexts of the query keywords in the data. Here , how to design tools that involuntarily distinguish structured search results and show how an obtainable XML query language can be comprehensive in order to maintain keyword search. Furthermore, described how such an extended XML query language can be implemented. The most important data structure required for keyword search is the inverted file. This work presents an approach to characterizing diversification systems using an empirical analysis.

References

[1] 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.
[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] M. Hasan, A. Mueen, V. J. Tsotras, and E. J. Keogh, “Diversifying query results on semi-structured data,” in Proc. 21st ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 2099–2103.
[4] D. Panigrahi, A. D. Sarma, G. Aggarwal, and A. Tomkins, “Online selection of diverse results,” in Proc. 5th ACM Int. Conf. Web Search Data Mining, 2012, pp. 263–272.
[5] R. L. T. Santos, J. Peng, C. Macdonald, and I. Ounis, “Explicit search result diversification through sub-queries,” in Proc. 32nd Eur. Conf. Adv. Inf. Retrieval, 2010, pp. 87–99.
[6] A. Angel and N. Koudas, “Efficient diversity-aware search,” in Proc. SIGMOD Conf., 2011, pp. 781–792.
[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] Z. Liu, P. Sun, and Y. Chen, “Structured search result differentiation,” J. Proc. VLDB Endowment, vol. 2, no. 1, pp. 313–324, 2009.
[9] 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.
[10] R. L. T. Santos, C. Macdonald, and I. Ounis, “Exploiting query reformulations for web search result diversification,” in Proc. 16th Int. Conf. World Wide Web, 2010, pp. 881–890.