ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

XML Keyword Search for Proposed Diversification Model

P.Sai Kiran
M.Tech(SE) Student
Department of IT
Geethanjali College of Engineering & Technology.

K.Srinivas
Professor
Department of CSE
Geethanjali College of Engineering & Technology.

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.

INTRODUCTION

Keyword search on structured and semi-structured data has attracted much research interest recently, as it enables common users to retrieve information from such structured data sources without the need to learn sophisticated query languages and database In general, the more keywords a given keyword query contains, the easier the search semantics of the keyword query can be identified. However, when the given keyword query only contains a small number of vague keywords, it will become a very challenging problem to derive the search semantics of the query due to the high ambiguity of this type of keyword queries. Although sometimes user involvement is helpful to identify search semantics of keyword queries, it is not always applicable to rely on users because the keyword queries may also come from system application.

In this application case, web or database search engine may need to automatically compute the search semantics of short and frequent keyword queries only based on the data to be searched. The derived search semantics will be maintained and updated in an off-line way. Once a keyword query is issued by the real users, its corresponding search semantics can be directly used to make an instant response. In this paper, we mainly pay attention to the problem of effectively deriving the search semantics of keyword queries with the consideration of data only, which does not receive much closer attention in the previous works.

EXAMPLE 1

Consider a simple keyword query q={database, query} over the DBLP dataset. There are 21,260 publications containing the keyword “database”, and 9,896 publications containing the keyword “query”, which contributes 2040 results that contain the two given keywords together. If we directly explore and understand the keyword search results, it would be time consuming and not user-friendly due to the huge number of results. It needs to take 54.22 seconds for just computing all the SLCA results of q by using XRank Even if the system processing time is acceptable by accelerating the keyword query evaluation with efficient algorithms the unclear and repeated search intentions in the large set of retrieved results will make users frustrating. To address the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

problem, we derive different search semantics of the original query from the different contexts of the XML data to be searched, which can be used to represent the different search intentions of the original query. In this work, the contexts can be modeled by extracting some relevant feature terms of the query keywords from the XML data, as shown in Table 1. And then, we compute the keyword search results for each search intention. 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.

By exploring the different feature terms of the query keywords, we have two benefits: the first is to diversify the keyword search results automatically by the different search intentions, which can return more distinct and diversified results to users; and the second is to improve the efficiency of keyword search because the contexts of diversified keyword queries can be used to reduce the size of relevant keyword node lists. Therefore, we are motivated to study the problem of keyword search diversification based on the contexts of query keywords in XML data to be searched, which is denoted as intent-based diversification. Although the intent-based diversification has been discussed in information retrieval (IR), e.g., models user intents at the topical level of the taxonomy and obtains the possible query intents by mining query logs, they are not always applicable because on the one hand, it is not easy to get the useful taxonomy and query logs; on the other hand, diversified results are modelled at different level, i.e., documents in IR vs. fragments in XML.

To the best of our knowledge, is the most relevant work that first maps each keyword to a set of attribute-keyword pairs, and then constructs a set of structured queries. It assumes that each structured query represents a query interpretation. However, the assumption is too strict to be applied for XML data because contextual information may not be necessarily structured, i.e., it may appear in the form of either attribute labels or texts.

Table 1: Top 10 selected feature terms of q

keyword	features
database	*systems; relational; protein; distributed; oriented; image; sequence; search; model; large.*
query	*language; expansion; optimization; evaluation; complexity; log; efficient; distributed; semantic; translation.*

Table 2: Part of statistic information for q

	database systems query +				
	language	*expansion*	*optimization*	*evaluation*	*complexity*
#results	71	5	68	13	1
	log	*efficient*	*distributed*	*semantic*	*translation*
#results	12	17	50	14	8
	relational database query +				
	language	*expansion*	*optimization*	*evaluation*	*complexity*
#results	40	0	20	8	0
	log	*efficient*	*distributed*	*semantic*	*translation*
#results	2	11	5	7	5
...	...				

The problem of diversifying keyword search is firstly proposed and studied in IR community Most of the techniques perform diversification as a post-processing or re-ranking step of document retrieval based on the analysis of result set and/or the historic query logs. In IR, keyword search diversification is designed at the topic or document level. For structured databases or semi structured databases, it is necessary to be redesigned at the tuple or fragment level. To address the main difference, the authors in propose to navigate SQL results through categorization, which takes into account user preferences. It consists of two steps: the first step analyzes query history of all users in the system offline and generates a set of clusters over the data, each corresponding to one type of user preferences; for an issued query, the second step presents to the user a navigational tree over clusters generated in the first step. By doing this, the user can browse, rank, or categorize the results in selected clusters. The authors in introduce a pre-indexing approach for efficient diversification of query results on relational databases based on the pre specified diversity orderings among the attributes over relations. The authors in first work out a small number of tuples by choosing one representative from each of clusters

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

and return them in the first page, which helps users learn what is available in the whole result set and directs them to find what they need. The authors in differentiate the keyword search results by comparing their feature sets where their feature types are limited to the labels of XML elements in the keyword search results. All of these methods can be classified as post-process search result analysis. They will encounter two challenging problems: the first one is effectiveness because the comparison of results will become difficult when the content of a result is not too much informative; the second is efficiency because they have to compute all the results, analyse and compare them one by one.

To address the above limitations, 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 the XML data based on the mutual information in 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 represents one of diversified contexts that express specific search intentions. And then, we evaluate derived search intentions by considering their relevances to the original keyword query and the novelty of the produced results. To effi- ciently compute diversified keyword search, we propose one baseline algorithm and two efficient algorithms based on the observed properties of diversified keyword search results. The remainder of this paper is organized as follows. In Section 2, we introduce a feature selection model and define the problem of diversifying XML keyword search. We describe the procedure of extracting the relevant feature terms for a keyword query based on the explored feature selection model in Section 3. In Section 4, we first show the procedure of generating search intentions from the derived feature terms and then propose three efficient algorithms, based on the observed properties of XML keyword search results, to identify a set of qualified and diversified keyword queries and compute their corresponding results. In Section 5, we provide extensive experimental results to show the effectiveness of our XML keyword search diversification model and the performance of our proposed algorithms. We describe the related work in Section 6 and conclude in Section 7.

Algorithm

Baseline Algorithm:

input: a query q with n keywords, XML data T and its term correlated graph G
output: Top-k search intentions Q and the whole result set Φ

$M_{m\times n}$ = getFeatureTerms(q, G);
while (q_{new} = GenerateNewQuery ($M_{m\times n}$)) $\neq$ null **do**
  ϕ = null and $prob\_s\_k = 1$;
  $l_{i_x j_y}$ = getNodeList($s_{i_x j_y}$, T) for $s_{i_x j_y} \in q_{new} \wedge 1 \leq i_x \leq m \wedge 1 \leq j_y \leq n$;
  $prob\_s\_k = \prod_{f_{i_x j_y} \in s_{i_x j_y} \in q_{new}} (\frac{|l_{i_x j_y}|}{getNodeSize(f_{i_x j_y}, T)})$;
  ϕ = ComputeSLCA($\{l_{i_x j_y}\}$);
  $prob\_q\_new = prob\_s\_k * |\phi|$;
  if Φ is empty **then**
    $score(q_{new}) = prob\_q\_new$;
  else
    for all Result candidates $r_x \in \phi$ **do**
      for all Result candidates $r_y \in \Phi$ **do**
        if $r_x == r_y$ or r_x is an ancestor of r_y **then**
          $\phi.remove(r_x)$;
        else if r_x is a descendant of r_y **then**
          $\Phi.remove(r_y)$;
    $score(q_{new}) = prob\_q\_new * |\phi| * \frac{|\phi|}{|\phi| + |\Phi|}$;
  if $|Q| < k$ **then**
    put $q_{new} : score(q_{new})$ into Q;
    put $q_{new} : \phi$ into Φ;
  else if $score(q_{new}) > score(\{q'_{new} \in Q\})$ **then**
    replace $q'_{new} : score(q'_{new})$ with $q_{new} : score(q_{new})$;
    $\Phi.remove(q'_{new})$;
return Q and result set Φ;

Anchor-Based Pruning Algorithm:

Motivated by the properties of computing diversified SLCAs, we design the anchor-based pruning algorithm. The basic idea is described as follows. We generate the first new query and compute its corresponding SLCA candidates as a start point. When the next new query is generated, we can use the intermediate results of the previously generated queries to prune the unnecessary nodes according to the above theorems and property. By doing this, we only generate the distinct SLCA candidates every time.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

That is to say, unlike the baseline algorithm, the diversified results can be computed directly without further comparison.

The detailed procedure is shown in Algorithm 2. Similar to the baseline algorithm, we need to construct the matrix of feature terms, retrieve their corresponding node lists where the node lists can be maintained using R-tree index. And then, we can calculate the likelihood of generating the observed query q when the issued query is qnew. Different from the baseline algorithm, we utilize the intermediate SLCA results of previously generated queries as the anchors to efficiently compute the new SLCA results for the following queries. For the first generated query, we can compute the SLCA results using any existing XML keyword search method as the baseline algorithm does, shown in line 18. Here, we use stack-based method to implement the function ComputeSLCA().

input: a query q with n keywords, XML data T and its term correlated graph G
output: Top-k query intentions Q and the whole result set Φ

1: $M_{m\times n} = \text{getFeatureTerms}(q, G)$;
2: **while** $q_{new} = \text{GenerateNewQuery}(M_{m\times n}) \neq \text{null}$ **do**
3: Lines 3-5 in Algorithm 1;
4: **if** Φ is not empty **then**
5: **for all** $v_{anchor} \in \Phi$ **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** $\forall l_{i_x j_y\_pre} \neq \text{null}$ **then**
8: $\phi' = \text{ComputeSLCA}(\{l_{i_x j_y\_pre}\}, v_{anchor})$;
9: **if** $\forall l_{i_x j_y\_des} \neq \text{null}$ **then**
10: $\phi'' = \text{ComputeSLCA}(\{l_{i_x j_y\_des}\}, v_{anchor})$;
11: $\phi += \phi' + \phi''$;
12: **if** $\phi'' \neq \text{null}$ **then**
13: $\Phi.remove(v_{anchor})$;
14: **if** $\exists l_{i_x j_y\_next} = \text{null}$ **then**
15: Break the FOR-Loop;
16: $l_{i_x j_y} = l_{i_x j_y\_next}$ for $1 \le i_x \le m \wedge 1 \le j_y \le n$;
17: **else**
18: $\phi = \text{ComputeSLCA}(\{l_{i_x j_y}\})$;
19: $score(q_{new}) = prob\_q\_new * |\phi| * \frac{|\phi|}{|\Phi|+|\phi|}$;
20: Lines 18-23 in Algorithm 1;
21: **return** Q and result set Φ;

EXISTING SYSTEM:

- 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.

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.

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

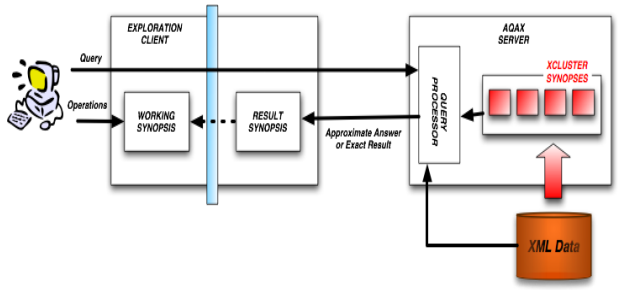
directly compute the diversified results without retrieving all the relevant candidates.

- 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.

SYSTEM ARCHITECTURE:



IMPLEMENTATION MODULES:

- Pre-processing
- Query Initialization
- Rewriter
- DOM Tree Construction
- Data Region Extraction

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

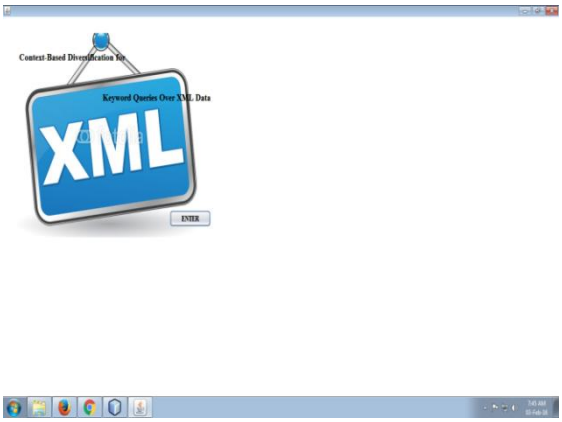
The Data Region Extraction module identifies all possible data regions, which usually contain dynamically generated data, top down starting from the root node. We first assume that some child sub trees of the same parent node form similar data

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

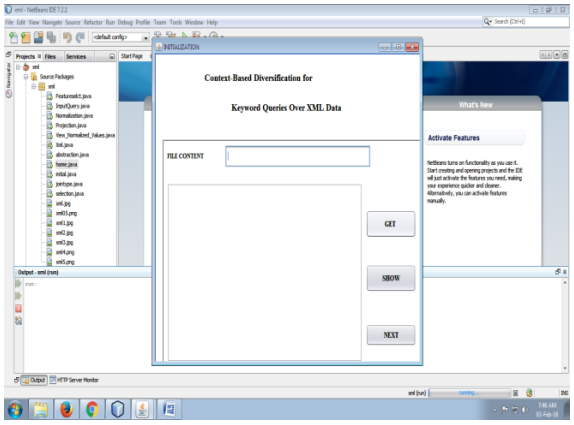
records, which assemble a data region. Many query result pages some additional item that explains the data records, such as a recommendation or comment, often separates similar data records. Hence, we propose a new method to handle non-contiguous data regions so that it can be applied to more web databases. 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 n_i, i =1 . . .m. Compute the similarity sim_{ij} of each pair of nodes n_i and n_j, 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 n_i only if it does not have any similar siblings. Segment the data region into data records using the record segmentation algorithm.

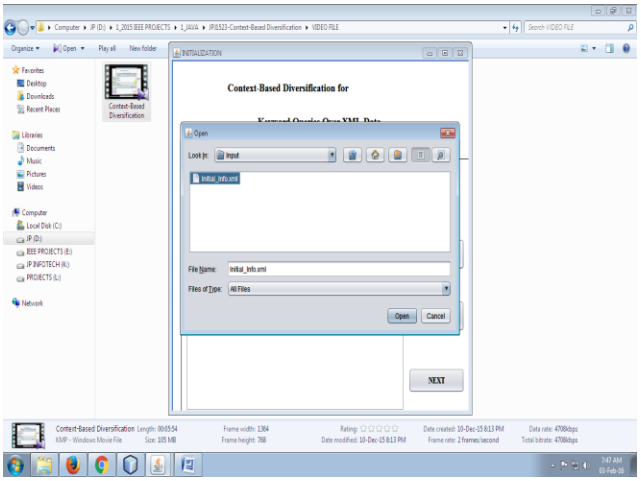
SCREEN SHOTS

Home:

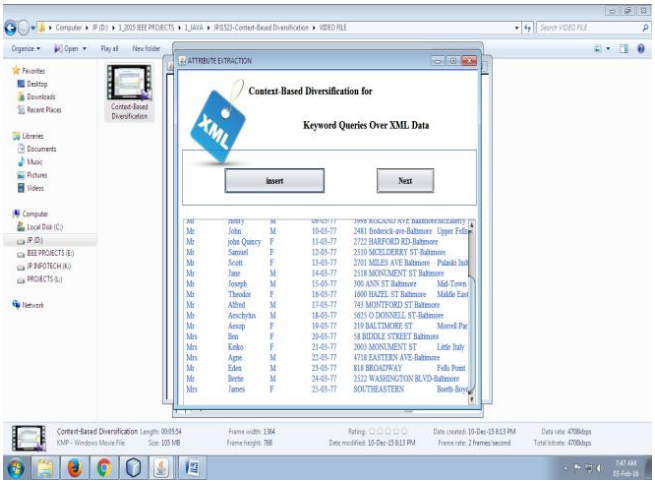


Initalization:

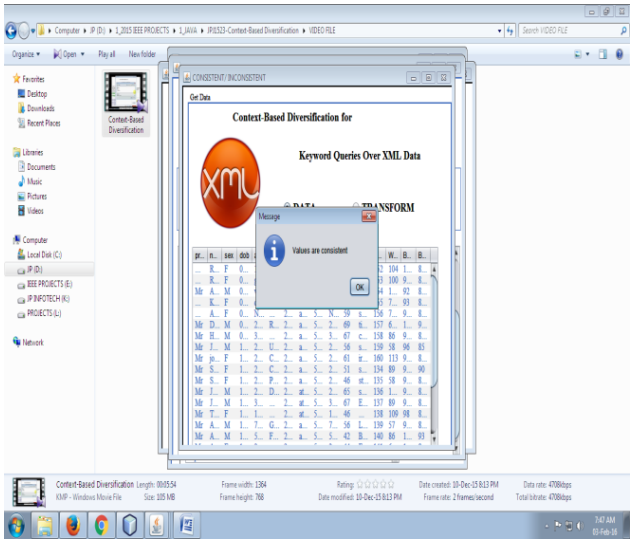




Attribute Extraction:

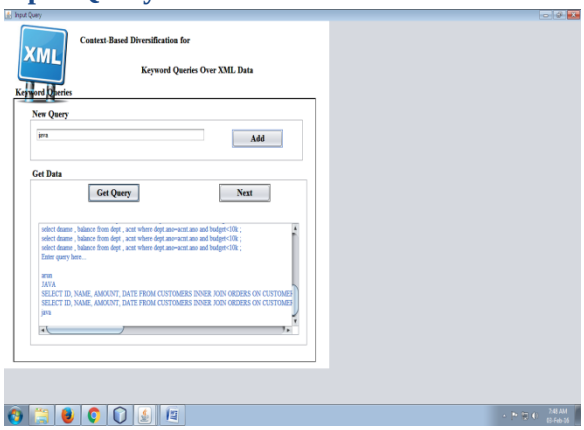


Consistent/Inconsistent:



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Input Query:



CONCLUSION

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 nDCG 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.

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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[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.