International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 4, April 2016

"Context-Based Diversification for Keyword Queries over XML Data"

[1] Snehal Ingole [2] DR. S.S.Prabhune
[1][2] Department of Computer Science and Engineering, SSGMCE Shegaon , India [1]
snehal.dilip3112@gmail.com, [2] ssprabhune@gmail.com

***Abstract*: 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.**

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.

I. 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 structure [1]. 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 chal-lenging problem to derive the search semantics of the query due to the high ambiguity of this type of keyword queries. Although Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage an d that copies bear this notice and the full citation on the first page. To cop y otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Sometimes user involvement is helpful to identify search semantics of keyword queries, it is not always applicable to rely on users be-cause 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.

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.

II. PROBLEM DEFINITION

Given a keyword query q and an XML data denoted by T , we consider a set of possible search

intentions Q that are generated by bounding each query keyword to a context using its relevant feature terms in T . Here, search intentions are also represented in the format of keyword query. Naturally, we need present to the users the top k qualified queries in terms of high relevance and maximal diversification.

2.1 Feature Selection Model

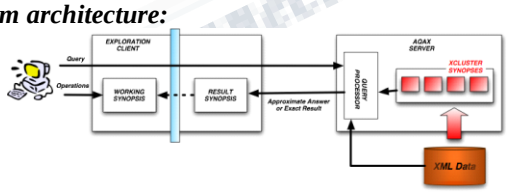
Consider an XML data T and a set of term-pairs W that can appear in T. The composition method of W depends on the application context and will not affect our subsequent discussion. As an example, it can simply be the full or a subset of the terms comprising the text in T , the contents of a dictionary, or a well-specified set of term-pairs relevant to some applications.

In this work, the distinct term-pairs are selected based on their mutual information as Mutual information has been used as a criterion for feature selection and feature transformations in machine learning. It can be used to characterize both the vance and redundancy of variables, such as the minimum redundancy feature selection. Assume we have an XML tree T and its sample result set R(T). Let P rob(x, T) be the probability

III. EXTRACTING FEATURE TERMS

Although we can pre-compute and manipulate the co-related terms up to any size, the use of two-term co-occurrences presents the most reasonable alternative in most applications In addition, two-term co-occurrences can be computed and stored efficiently a s de-scribed in Co-occurrences of higher order can be utilized at the expense of space and, most importantly, time. For the scale of the applications we envision, materializing co-occurrences of length higher than two is probably infeasible. Therefore, in this work, we materialize two-term co-occurrences, which involves the computation of a sorted list.

System architecture:



Modules:

1.Admin
2.User
3. xmlQuery Answering

Admin:

Admin maintains the total information about the hole application.

Admin maintain the data in XML format only.

User:

User search queries and he got the reply in xml format.

Xml Query Answering:

In this project user search the information in semi structure document. He got reply in xml format only.

IV. KEYWORD SEARCH DIVERSIFICATION

Algorithms

In this section, we first introduce the procedure of generating a new query from the matrix of the original keyword query w.r.t. the data to be searched. And then based on the matrix, we propose a baseline algorithm to retrieve the diversified keyword sea rch results. At last, two anchor-based pruning algorithms are designed to improve the efficiency of the keyword search diversification by utilizing the intermediate results.

4.1Generate Search Intentions

Given a keyword query q, we first retrieve the corresponding feature terms for each query keyword and then construct a matrix of search intentions. In the matrix, the feature terms in each column are sorted based on their mutual information scores. Each combination of the feature terms (one term per column) represents a search intention. We iteratively choose the combination with the maximal aggregated mutual information score as the next best search intention until the terminal requirements are reached.

Algorithm

1 Baseline Algorithm

input: a queryqwithnkeywords and XML dataT **output:** Top-ksearch intentionsQand overall result setΦ

1: Mm×n = getFeatureTerms(q, T);

2: **while** (qnew= GenerateNewQuery(Mm×n))6=null **do**
3: φ = null and prob_s_k = 1;
4: liXjY = getNodeList(siXjY , T) for siXjY∈ qnew∧ 1 ≤ ix≤ m ∧ 1 ≤jy≤ n;
prob_s_k ∈qNEW(|lIX JY|);
= fI J
∈sI J getNodeSize(IX JY
f

6: φ = ComputeSLCA({liXjY});
7: prob_q_new = prob_s_k * |φ|;
8: **if** Φis empty **then**
9: score(qnew) = prob_q_new;
10: **else**
11: **for all** Result candidatesrx∈φ **do**
12: **for all** Result candidatesry∈Φ **do**
13: **if** rx==ryorrxis an ancestor ofry **then**
14: φ.remove(rx);

15: **else if** rxis a descendant ofry **then**
16: Φ.remove(ry);
17: score
18: **if** |Q|< k **then**
19: put qnew : score(qnew) into Q;
20: put qnew : φ into Φ;
21: **else if** score(qnew)> score({qnew′∈Q}) **then**
22: replace qnew′ : score(qnew′) with qnew : score(qnew);
23: Φ.remove(qnew′);
24: **return** Qand result setΦ;

Anchor-based Pruning Solution

By analyzing the baseline solution, we can find that the main cost of this solution is spent on computing SLCA results and removing unqualified SLCA results from the newly and previously generated result sets. To reduce the computational cost, we are motivated to design an anchor-based pruning solution, which can avoid the unnecessary computational cost of unqualified SLCA results (i.e., duplicates and ancestors). In this subsection, we first analyze the interrelationships between the intermediate SLCA candidates.

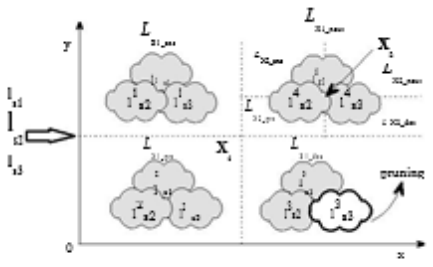


Figure 1: The usability of anchor nodes

2.The Anchor-based Pruning Algorithm

input: a queryqwithnkeywords and XML dataT **output:** Top-kquery intentionsQand result setΦ
1: Mm×n = getFeatureTerms(q, T);

2: **while** qnew= GenerateNewQuery(Mm×n)6=null **do**
3: Line 3-Line 5 in Algorithm 1;
4: **if** Φis not empty **then**
5: **for all** vanchor∈Φ **do**
6: get liXjY _pre, liXjY _des, and liXjY _next by calling for Partition(liXjY , vanchor);

7: **if** ∀liXjY_pre=6null **then**
8: φ′ = ComputeSLCA({liXjY_pre}, vanchor);
9: **if** ∀liXjY_des=6null **then**
10: φ″ = ComputeSLCA({liXjY_des}, vanchor);
11: φ += φ′ + φ″;
12: **if** φ″=6null **then**
13: Φ.remove(vanchor);
14: **if** ∃liXjY_next=null **then**
15: Break the FOR-Loop;
16: liXjY = liXjY _next for 1 ≤ ix ≤ m ∧ 1 ≤jy≤ n;
17: **else**

18: φ = ComputeSLCA({liXjY});
19: score(qnew) = prob_q_new * |φ|* |Φ|+|φ|;
20: Line 18-Line 23 in Algorithm 1;
21: **return** Qand result setΦ;
20: **Line 18-Line 23 in Algorithm 1;**
21: **return Qand result setΦ;**

V. EXPERIMENTS

In this section, we show the extensive experimental results for evaluating the performance of our baseline algorithm (denoted as *b*aseline *e*valuation *BE*) and anchor-based algorithm (denoted as *a*nchor-based *e*valuation *AE*), which were implemented in Java andrun on a 3.0GHz Intel Pentium 4 machine with 2GB RAM running Windows XP. For our anchor-based parallel sharing algorithm (de-noted as *ASPE*), it was implemented using six computers, which can serve as six processors for parallel computation.

5.1 Dataset and Queries

We use a real dataset, DBLP and a synthetic XML bench-mark dataset XMark for testing the proposed XML keyword search diversification model and our designed algorithms. The size of DBLP dataset is 971MB and the size of generated XMark dataset is 697MB. Compared with DBLP dataset, the synthetic XMark.

VI. RELATED WORK

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.

VII. CONCLUSIONS

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 con texts were 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 key-word search results. Finally, we demonstrated the efficiency y of our proposed algorithms by running substantial number of queries over both DBLP and XMark datasets. Meanwhile, we also verified the effectiveness of our diversification model by analyzing the returned search intentions for the given keyword queries over DBLP dataset. From the experimental results, we get that our proposed diversification algorithms can return qualified search intentions an d results to users in a short time.

REFERENCES

[1] Y. Chen, W. Wang, Z. Liu, and X. Lin, "Keyword search on structured and semi-structured data," in *SIGMODConference*, 2009, pp. 1005–1010.

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

[3] C. Sun, C. Y. Chan, and A. K. Goenka, "Multiway slca-based keyword search in xml data," in *WWW*, 2007, pp. 1043–1052.

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

[5] R. Agrawal, S. Gollapudi, A. Halverson, and

S. Ieong, "Diversifying search results," in *WSDM*, 2009, pp. 5–14.

[6] F. Radlinski and S. T. Dumais, "Improving personalized w eb
search using result diversification," in *SIGIR*, 2006, pp. 691–692.

[7] E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl, " *DivQ*: diversification for keyword search over structured

databases," in *SIGIR*, 2010, pp. 331–338.

[8] J. G. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *SIGIR*, 1998, pp. 335–336.

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

[10] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova,

A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *SIGIR*, 2008, pp. 659–666.

[11] A. Angel and N. Koudas, "Efficient diversity-aware sear ch," in *SIGMOD Conference*, 2011, pp. 781–792.

[12] Z. Chen and T. Li, "Addressing diverse user preferences in sql-query-result navigation," in *SIGMOD Conference*, 2007, pp. 641–652.

[13] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and

S. Amer-Yahia, "Efficient computation of diverse query results," in *ICDE*, 2008, pp. 228–236.

[14] B. L. 0002 and H. V. Jagadish, "Using trees to depict a forest," *PVLDB*, vol. 2, no. 1, pp. 133–144, 2009.

[15] Z. Liu, P. Sun, and Y. Chen, "Structured search result differentiation," *PVLDB*, vol. 2, no. 1, pp. 313–324, 2009.

[16] 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. PatternAnal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[17] C. O. Sakar and O. Kursun, "A hybrid method for feature selection based on mutual information and canonical correlation analysis," in *ICPR*, 2010, pp. 4360–4363.

[18] N. Sarkas, N. Bansal, G. Das, and N. Koudas, "Measure-driven keyword-query expansion," *PVLDB*, vol. 2, no. 1, pp. 121–132, 2009.

[19] N. Bansal, F. Chiang, N. Koudas, and F. W. Tompa, "Seekin g stable clusters in the blogosphere," in *VLDB*, 2007, pp. 806–817.

[20] "http://dblp.uni-trier.de/xml/."

[21] "http://monetdb.cwi.nl/xml/."

[22] M. J. Welch, J. Cho, and C. Olston, "Search result divers ity for informational queries," in *WWW*, 2011, pp. 237–246.

[23] R. H. van Leuken, L. G. Pueyo, X. Olivares, and R. van Zwol, "Visual diversification of image search results," in *WWW*, 2009, pp. 341–350.

[24] Z. Liu, S. Natarajan, and Y. Chen, "Query expansion base d on clustered results," *PVLDB*, vol. 4, no. 6, pp. 350–361, 2011.

[25] S. Gollapudi and A. Sharma, "An axiomatic approach for result diversification," in *WWW*, 2009, pp. 381–390.

[26] J. Wang and J. Zhu, "Portfolio theory of information retrieval," in *SIGIR*, 2009, pp. 115–122.