Research Paper

Volume : 5 | Issue : 8 | Special Issue August-2016 • ISSN No 2277 - 8179

Divcontext: a paradigm for Outsourcing Styled Contents over Image based XML Source

Engineering

KEYWORDS: Index Terms— XML, keyword search, context based diversification, outsourcing

Anjana B G — PG Scholar, Computer Science and Engineering Vedavyasa Institute of Technology Karad parampa, Malappuram

Ginnu George — Assistant Professor, Computer Science and Engineering Vedavyasa Institute of Technology Karad parampa, Malappuram

ABSTRACT

The keyword query is a common technique for ordinary users to search vast amount of data. It helps the users for information retrieval without any knowledge about sophisticated query languages and data structure. But there would be a problem to effectively answer the keyword queries due to the ambiguity of keyword query. This problem is solved by applying diversification. XML source is a best method for storing large amount of information. Also, the styles have much importance here. So, the outsourcing of styled contents over image based XML source is emphasized here. The main objective of this paper is to a review study on keyword based search on XML data and the problems in keyword based search on XML data. The outsourcing of styled contents is done over the image based XML source. It also includes the solutions for the challenging problems

I. INTRODUCTION

As the technology develops, the amount of data or information required become very large. So, the storage as well as the retrieval of desired information become an important aspect in this world. Searching for information is an indispensable component of our lives. Web search engines are widely used for searching textual documents, images, and videos. There are also vast collections of structured and semi-structured data both on the Web and in enterprises, such as relational databases, XML, data extracted from text documents, workflows, etc. Traditionally, to access these resources, users have to learn structured query languages, such as SQL and XQuery; they also need to access data schemas of each individual application domain, which are most likely complex, fast-evolving, or even unavailable in Web applications.

There exists so many techniques to store structured and semistructured data. Also ,databases are used in order to store huge amount of data. Among all of these, XML is quite useful and most acceptable method to store data. XML possesses many benefits when compared to other data storing mechanisms. There exists many types of databases to store data as well as XML.

The traditional methods require to use query languages to retrieve relevant answers from XML data. These methods are good but very hard to nonexpert users. These query languages are difficult to comprehend for nondatabase users., Keyword query is easy and today's recognizable to most Internet users as it just requires the input of keywords for querying XML data.

The reading of keyword search result requires much more time and it is not user friendly due to large number of results. Though the keyword query evaluation can be accelerated, the unclear and repeated search intentions in the large set of retrieved results will become a problem for the users. To solve this problem, we are applying diversification to make the search process more easier and efficient. The diversification is based on the context. Also, the styles are applied to the text data in order to consider the users' search intentions. The search is done on the image based XML source.

This chapter has given an overview about the storage and retrieval of data, image based XML source, outsourcing of styled contents.Further chapters discuss the related work, existing system with it's limitations, proposed system and conclusion.

II.RELATED WORK

In [1], Y. Chen, W. Wang, Z. Liu, and X. Lin conducted a study on keyword search on structured and semi structured data. Accessing databases using the keywords is a more useful technique for searching. It doesn't require the steep knowledge about the structured query languages and data schemas when accessing structured data. Also, it allows users to easily access heterogeneous databases. They proved that in traditional database applications, the query results are fully specified by structured queries. But in the case of keyword search, is to define query results which automatically gather relevant information that is generally fragmented and scattered across multiple places.

Sanjay Agrawal, Surajit Chaudhuri and Gautam Das proposed a new system for keyword based search over relational databases in [2]. It is named as DBXplorer, an efficient and scalable keyword search utility for relational databases. It has been deployed on real databases from the intranet within Microsoft. The system also allows one to search multiple databases simultaneously.

E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl have describes about a new system in [3] named DivQ. It describes about the diversification for keyword search over structured databases. Diversification is an important process used in searching of data. It aims to minimize the users' dissatisfaction by balancing novelty and relevance of search results. Keyword queries over structured data are notoriously ambiguous offering an interesting target for diversification. DivQ translates a keyword query to a set of structured queries, also known as query interpretations. To minimize the risk of user's dissatisfaction in this environment, diversification is required to provide a better overview of the probable query interpretations, rather than a ranking based only on relevance.

In [4], L.Guo, F.Shavo, C.Botev and J.Shanmugasundaram describes about another system, XRANK,a system for keyword search over XML documents. It is the first system that takes into account the hierarchical and hyperlinked structure of XML documents, and a two-dimensional notion of keyword proximity, when computing the ranking for XML keyword search queries. It is designed to naturally generalize a HTML search engine such as Google. It can query over a mix of HTML and XML documents.Also, XRANK offers both space and performance benefits when compared with existing approaches.

R. Agrawal conducted a research on diversifying search results in [5].The results are diversified in the presence of ambiguous queries. A systematic approach for diversifying the search results is formed here.Also, a natural greedy algorithm is proposed.

DivDB is another system for diversifying query results proposed by Marcos R. Vieira, Humberto L.Razente and Maria C.N. Barioni in [6].

The main idea of this system is also diversification

DivDB is a technique to diversify the search results. The DivDB system provides several diversification algorithms in a common framework, where their speed and quality of the result vary depending on the query parameters. It uses SQL like language.

A. Angel and N. Koudas conducted a study on efficient diversity aware search in [7]. The content based result diversification is focused here. An efficient threshold algorithm, named ,DIVGen is used here. It utilizes novel data access primitives. It achieves significant performance improvements via novel data access primitives. DIVGen is being able to drastically reduce the number of documents it examines.

In [8], a study was conducted on top-k keyword search over probabilistic xml data by J. Li, C. Liu, R. Zhou, and W. Wang. This is the first work that studies keyword search over probabilistic XML data. A strategy have been proposed by them to compute the SLCA probabilities without generating possible words. Based on the strategy, proposed two efficient algorithms: PrStack and EagerTopK.

In [9], C. O. Sakar and O. Kursun have conducted a research on hybrid method for feature selection based on mutual information and canonical correlation analysis in 2010. They gives idea about Kernel Canonical Correlation Analysis (KCCA), a nonlinear correlation measure for determining statistical dependencies between two sets of random variables. MI is more useful than KCCA in the sense that it gives an entropy-based score that quantifies the strength of the dependence.

In 2010, R. L. T. Santos, C. Macdonald, and I. Ounis proposed a new framework XQuAD (eXplicit Query Aspect Diversification) in [10]. It is a novel probabilistic framework for search result diversification. It explicitly models the aspects underlying an initial query, in the form of sub queries.This approach achieves an efficient diversification performance. It is done by directly estimating the relevance of the retrieved documents to multiple sub queries, instead of comparing documents to one another .

III. PROBLEM STATEMENT AND HYPOTHESES

XML (eXtensible Markup Language) has received a great deal of attention as the likely successor to HTML for expressing much of the content of the Web. However, XML also has the potential to benefit databases and data sharing by providing a common format in which to express data structure and content. By describing the broad challenges presented by data access and data interoperability, we highlight both the potential contributions of XML technologies to databases and data sharing, and the problems that remain to be solved. In some areas, XML promises to provide significant and revolutionary improvements, such as by increasing the availability of database outputs across diverse types of systems, and by extending data management to include semi-structured data. While some of the benefits of XML are already becoming apparent, others will require years of development of new database technologies and associated standards.

BENEFITS OF XML:

- Simplicity: Information coded in XML is easy to read and understand,plus it can be processed easily by computers.
- Openness: XML is a W3C standard,endorsed by software industry market leaders.
- Extensibility: There is no fixed set of tags.New tags can be created as they are needed.
- Self description: XML documents can be stored without schemas because they contain meta data; any XML tag can possess an unlimited number of attributes such as author or version.
- It is a better way to store the data by using XML. The retrieval of information is by context based diversification search. Diversifying the search results will give more novel and similar results.

Consider an XML data T and its relevance-based term-pair dictionary W. The distinct term-pairs are selected based on their MI.MI is used as a criterion for feature selection and feature transformation in machine learning.

It is used to characterize both the relevance and redundancy of variables, such as the minimum redundancy feature selection. 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.Each combination of the feature terms and the original query keywords may represent one of diversified contexts

First, measure the correlation of each pair of terms using MI. During XML data tree traversal ,we extract the meaningful text information from the entity nodes in XML data.Then produce a set of term pairs by scanning the extracted text.

MI score for each term-pair is calculated.MI is the criterion for feature selection and feature transformation in machine learning.

The MI score is calculated as:

$$MI(x,y,T) = Prob(x,y,T) * log \frac{Prob(x,y,T)}{Prob(x,T) * Prob(y,T)}$$

After calculating the MI scores, the top k term pairs are found.Then correlated graph is drawn using the top k term pairs.

Mainly two algorithms are used here; baseline algorithm and anchor based pruning 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 F

1: $M_{m\times n}$ =getFeatureTerms(q, G);

2: while (q_{new} = GenerateNewQuery ($M_{m\times n}$)) != null do

3: f =null and prob s k= 1;

4:l_{ixjy} = getNodeList(s_{ixjy} , T) for s_{ixjy} €q_{new} ^ $1 \le i_x \le m$ ^ $1 \le j_y \le n$;

5: Retrieve the pre computed node lists of the keyword-feature term pairs in q_new from each xml file.

6: f = ComputeSLCA({l_{ixjyj});

7: prob q new = prob_s_k *|f|;

8:if F is empty then

9:score(q_{new})= prob q new;

10:else
11:for all Result candidates r_x €f do

12:for all Result candidates r_y € F do

13:if $r_{x==}$ r_y or r_x is an ancestor of r_y then

14:f.remove(r_{xj};

15: else if r_x is a descendant of r_y then

Volume : 5 | Issue : 8 | Special Issue August-2016 • ISSN No 2277 - 8179

16: F.remove(r_y);

17: score(q-new)=prob(q-new)*

18: if Q < k then

19: put q_{new} : score(q_{new}) into Q;

20: put q_{new} : f into F;

21: else if score(q_{new}) > score({q'_{new} €Q}) then

22: replace q'_{new} : score(q'_{new}) with q_{new} : score(q_{new});

23: F:remove(q'_{new});

24:return Q and result set F;

The baseline algorithm returns the diversified search results.

Anchor Based Pruning 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 F
1: $M_{m\times n}$ =getFeatureTerms(q, G);

2: while (q_{new} = GenerateNewQuery ($M_{m\times n}$)) != null do

3: Lines 3-5 in Algorithm 1;

4: if F is not empty then ;

5: for all vanchor € F do

6: get lixjy-pre, lixjy-des, and lixjy-next by calling for Partition(lixjy , vanchor);

7: if lixjy –pre ≠null then

8: f'= ComputeSLCA({lixjy –pre}, vanchor);

9: if lixjy- des ≠null then

10: f" = ComputeSLCA({lixjy- des}, vanchor);

11: f+=f' + f";

12: if f" ≠ null then

13: F.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: f = ComputeSLCA({lixjy});

19: score(q-new)=prob(q-new)*

20: Lines 18-23 in Algorithm 1;

21: return Q and result set F;

It avoids unqualified SLCA results.The interrelationships between the intermediate SLCA candidates are analyzed here.

Given an anchor node va and a new query candidate qnew = {s1; s2; . . . ; sn}, its keyword node lists L = {ls1; ls2 ; . . ; lsn} can be divided into four areas to be anchored by va, i.e., the keyword nodes that are the ancestors of va, denoted as Lva-anc; the keyword nodes that are the previous siblings of va, denoted as Lva-pre; the keyword nodes that are the descendants of va, denoted as Lva-des; and the keyword nodes that are the next siblings of va, denoted as Lva-next. We have that Lva-anc does not generate any new result; each of the other three areas may generate new and distinct SLCA results individually; no new and distinct SLCA results can be generated across the areas.

But, here the search can be performed only in case of plain text data in XML. Context based diversified search can't be used in case of styled text contents. Also, the image files can't do so. This is the major limitation of this paper and this needs to be tackled.

IV. SYSTEM OVERVIEW
The existing methodology focuses on searching for the plain text data over the XML source. The main aim of this approach is to searching according to the context and diversifying the search results. The user enters a keyword and a number of feature term pairs are generated. Then, top k term pairs are displayed and a correlated graph is generated. After that the search process is executed according to the two algorithms, baseline algorithm and anchor based pruning algorithm. But, this methodology only considers the plane text data. The image files are not considered here. Also styled contents are not used.

So, in order to avoid the drawback of the existing methodology, a new method is proposed. It considers both the styled text data as well as the image data. The outsourcing of the styled contents is done over image based xml source.

Adding styles to the text data has more benefits than the normal text data. The styled contents are much more attractive. The readers can understand and analyze the data more faster. The styles can be bold,italic,underlined...etc.This can be done using the web tool, tiny mce.

In order for a diversified search over plain xml source, we can improve the dataset architecture with multimedia contents specifically image files. Instead of plain contents, we can load styled contents so as to view it in an efficient manner. For that, We propose a technique to align data units into different groups so that the data units inside the same group have the same semantic. Instead of using only the DOM tree or other HTML tag tree structures of the SRRs to align the data units (like most current methods do), our approach also considers other important features shared among data units, such as their data types (DT), data contents (DC), presentation styles (PS), and adjacency (AD) information.

STEPS IN PROCESS:

- Node extraction: The nodes are extracted from the XML source. Then Take each keyword from grid.Extract the feature terms from thesaurus. Create a directory named ex-contents. Create subdirectory for each XML files in the ex-contents directory with the xml filename. Save the extracted contents to an html page with the name as keyword.

- Feature term extraction: The feature terms of each node are collected. Load the html files in the chosen xml file.Split each filename with '.'.then we will get filename only .Read the file content. Split split the html content with separator"" and . We will get the inner features of each word.If no features are there,then add as No attributes. Create a dictionary to store. Keyword: feature terms.it will store elements like <key,value> pair. Take that to a session in order to access it in save to dictionary option. Finally the keywords and feature terms will be viewed in grid.

- Save to dictionary: The collected feature terms and nodes are saved into a predefined dictionary.

- Term pair generation: The term pairs of the feature term and the node are generated.
- MI Score calculation: The mutual information score of each term pair is calculated using the proper equations given.
- Generation of top k term pairs: Among the calculated MI scores of each term pairs, top k term pairs are found and listed.
- Correlated graph generation: The correlated graph is drawn using the top k term pairs.
- Finally find the search results by approximate search and filtered search. The search results are obtained.

Here we apply two more algorithms along with the baseline algorithm and anchor based pruning algorithm. They are: Data alignment algorithm and annotation algorithm.

Data alignment algorithm
1.Merge text nodes.
2.Align text nodes.
3.Split (composite) text nodes.
4.Align data units.

Annotation algorithm
1. Submit the keyword query.

2. Load the whole Custom Book Depo containing styled contents and images.

3. Given the 1st priority to category, the search hits on the xml files having the name as the search keyword.

4. If matching, then populates the whole book records with styled contents within the specific category. Otherwise check it with the inner text of title node or description node or image name node or author node or publish_date node or price node.

5. Finally, the contents with style and the corresponding images will be annotated from the custom styled Xml dataset.

V. RESULT ANALYSIS

The proposed methodology is analyzed. When the user gives a keyword query, after some steps the search results are obtained.

There are differences between existing system and proposed systems. But, they both work on XML data. The existing system aims at search for only normal text data. But the proposed system works on styled contents over image based XML source. DivContext uses image files and styles text data. Also the data source can be customized. But in previous works, only predefined and static datasets can be used to perform search. DivContext is more useful and efficient than the already existing search methods. The main differences are tabulated as in Table1.

Table 1: Comparison between Context based diversified search and DivContext

Context based diversified search	DivContext
Only plain text can be searched	Styled text data also can be searched
Image files are not included for search	Image files are included
Not as much efficient	More efficient
Only predefined and static XML dataset can be used	Data source can be customized

A. Results

The fig 1 shows the term pairs with their mutual scores for the particular input. Fig 2 shows the correlated graph of with respect to mutual score and term pairs.

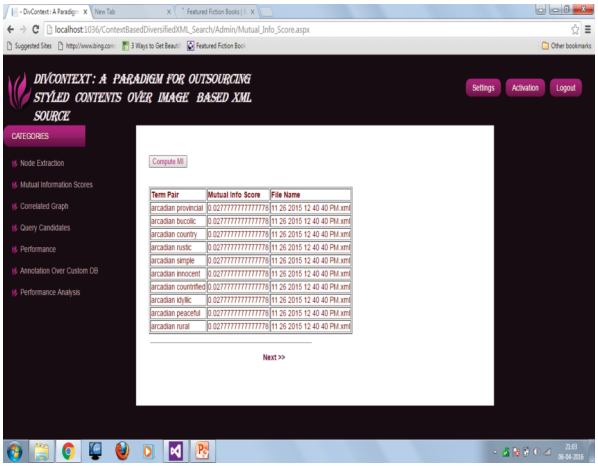


Fig.1. Term pairs and their MI scores

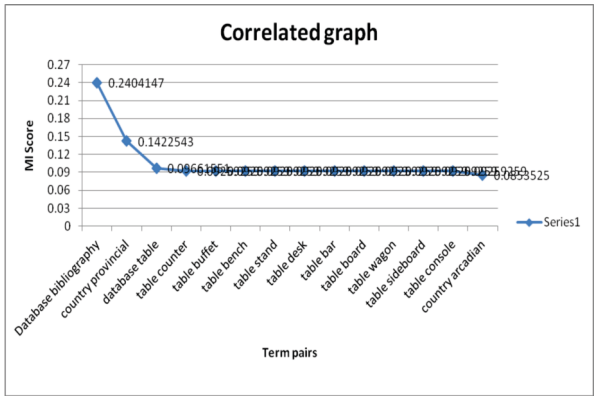


Fig.2.Correlated graph

A. Performance Analysis

The search process is being executed in two different search engines, Google and Bing. Then, the processing speed of each one is calculated and the performance is analyzed. By analyzing this, a graph is drawn.

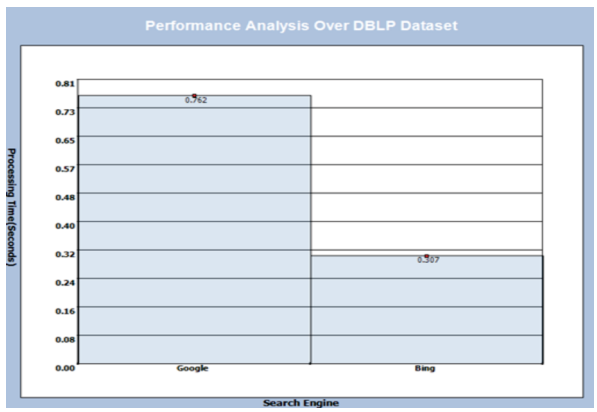


Fig.3. Performance Analysis

Fig 3 shows the analysis of performance of the search engines while run under the proposed system.

DivContext uses four algorithms for outsourcing styled contents over image based XML source.

VI. IMPLEMENTATION

The fig 4 shows the system architecture of our proposed system Divcontext. It is done in book domain. There exists many categories such as adventure, fiction, horror...etc. The contents are loaded as data units. Then stored in web DBs. Then going through the alignment phase and the annotation phase. Finally, the styled

Volume : 5 | Issue : 8 | Special Issue August-2016 • ISSN No 2277 - 8179

content extraction is done. The search results are returned according to the query given by the user.

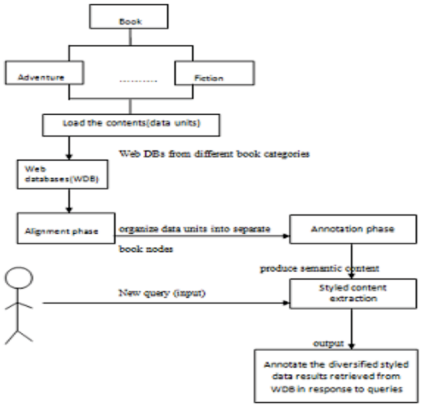


Fig.4. System Architecture

VII. CONCLUSION

As the technology emerges, the need for storing and retrieving the information become a significant thing in our day today life.

A number of techniques used for the information search is discussed here. In order to make the search more efficient, we proposed a new system, Divcontext, is a context based diversification model for keyword queries. The highlight of this system is that it considers the styled contents over the image based XML source. The search can be done via approximate search and filtered search.

The feature terms are extracted first. Then the term pairs are generated. The MI score are calculated. After that, the top k term pairs are listed and the correlated graph is drawn. Them the algorithms are performed. After all these, the user will get the best search results while giving a keyword query.

This is an efficient system than other systems since it also considers the image files along with the styled text data.

The future work will be to append the audio files and to perform the search according to them.

VIII. 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.] Sanjay Agrawal,Surajit Chaudhuri, Gautam Das,"DBXplorer: A System for Keyword-Based Search over Relational Databases",IEEE Conf.,2002,

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

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

[5.] 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.

[6.] Marcos R. Vieira, Humberto L.Razente, Maria C.N. Barioni,"DivDB: A System for Diversifying Query Results"

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

[8.] 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.

[9.] C. O. Sakar and O. Kursun, "A hybrid method for feature selection based on mutual information and canonical correlation analysis," in Proc. 20th Int. Conf. Pattern Recognit., 2010, pp. 4360–4363.

[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.

[11.] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng, , and Clement Yu,"Annotating search results from web databases", 2013

[12.] www.wikipedia.org