An Efficient Search Method For XML document

Xie Qian, and Dong-Sub Cho
Dept. of Computer Science and Engineering, Ewha Womans University
e-mail : kazu@ewhain.net

Abstract

Because of the rapid development of internet, there are more and more documents stored by the XML-based format. When there is a great deal of XML documents, how to get the valuable Information is an important subject. This paper proposes an effective XML document search method to search text contents and structures of XML documents. We build the keyword matrix of text contexts and structure matrixes of structures in XML documents to improve the efficiency of query time. When there is a great deal of XML documents, the search method we propose can improve much efficiency of query time.

1. Introduction

XML (Extensible Markup Language) is a specification of W3C (Bray, 1998). It is developed to complement HTML for data exchange on the Web. In recent years, XML has been more and more used in large information systems, such as digital libraries or information centers. In most of these systems, search engine is a major module.

XML focus on the schema of the document. So there are significant features and advantages in the methods of data exchange or display. Figure 1 is an example of the XML document named CDshop.xml.

```
<? xml version="1.0" encoding="UTF-8"?>
<CDshop>
     <CD category="SINGLE"
          <title lang="jp">ONE DROP</title>
          <publisher>Jone Records</publisher>
          <singer>KAT-TUN</singer>
          <price>7,400</price>
    </CD>
    <CD category ="ALBUM"
          <title lang="jp">Real Face</title>
          <publisher>Je Records</publisher>
          <singer>KAT-TUN</singer>
          <price>13,300</price>
   </CD>
  </CDshop>
```

(Figure 1). Example of CDshop.xml.

With more and more popular used, the amount of relative hidden information is growing, so the storage and query processing of the XML file as an important issue, in which how to query the result quickly and efficiently based on the user need from a large number of XML documents become more and more important.

Based on the above, the purpose of this paper is in face of a growing number of XML documents, hoping to make a query method, for text content and structure (element, tag, attribute, etc.) be part of the query for processing, and changed during the past most of the query language queries, XML documents need to do the whole tree search, causing a waste of time cost for the users, improvement in time efficiency is more able to meet the query requirements. And the method of conducting this study, a large number of XML file query, the more efficiency can play in the time effect.

2. XML Query Language

XML query method is divided into two areas: Structured Query Language and Keyword Search. Structured query language expression through the path expression of the query language, such as XPath [1], XQuery [2], etc. Such data query language query the entire XML document need to conduct a comprehensive search tree, so need to spend more time costs. And as the characteristics of the path representation, it cannot query the XML document only based on the text content; keyword search, such as XSEarch [3], XRANK [4], XSeek [5], can query the XML document only based on the text content, and can also query the file based on the structure of the content such as Element, Tag, Attribute ,etc, and compared with structured query language syntax, key-word search does not require prior knowledge of the query syntax, it is easier for user to use. But all of the keyword query and structured query language have the same problem, that is, query XML documents to be targeted at the whole tree structure to do the search, to spend a lot of time costs.

3. Problem Description

With the increasing common use of XML files, that resulted in the increase in the amount of data. When users want to get the information which he desired during a large number of XML files, the search methods we mentioned before have some disadvantages.

Most of the keyword query, such as XSEarch, XRANK, XSeek and so on, making inquiries through the XML file needed to conduct an over-all search tree structure, because it will advance to give each node in the tree structure of an ID, to help the use of the query, for example, mentioned above, XRANK give tree node is the Dewey IDs, and the XSEarch will be given a number for each non-leaf node.

However, XML documents through keyword queries, although simultaneously for the elements, attributes and text content of the query, but with the structured query language have the same problem, which is conducting inquiries on the XML file hierarchical tree structure need to search the whole tree and this will cause a waste of time cost.

Therefore, in this paper we propose an effective XML document search method to search text contents and structures (element, tags, attribute etc.) of XML documents. We build the keyword matrix of text contexts and structure matrixes of structures in XML documents to improve the efficiency of query time. When there is a great deal of XML documents, the search method we propose can improve much efficiency of query time.

4. An Effective XML Document Search Method

In this section, we will represent an effective XML document search method. This method provides a user-friendly information search method for web and scientific users to easily access XML data without the need of learning a structured query language or studying possibly complex and evolving data schemas.

Before explaining the method we will meant to sketch some pre-processing of the text content and structure which in the XML document. First, we need to create the keyword matrix and structure matrix which the search method is needed. A briefly introduction with an example will be illustrated in the following.

There is a fragment of a CDshop XML document (in Figure 1) we talked before; all of our example represent in this paper will use this XML document.

4.1 XML text contents pro-processing—keyword matrix

The query method that we describing in this section first have to do the processing of the text contents in the XML document—to build a keyword matrix.

Table1 shows the keyword matrix of the CDshop.xml document that we mentioned before. The keyword matrix is consists of a set of keywords and index number (X1, X2, etc.)

<Table1> Keyword matrix

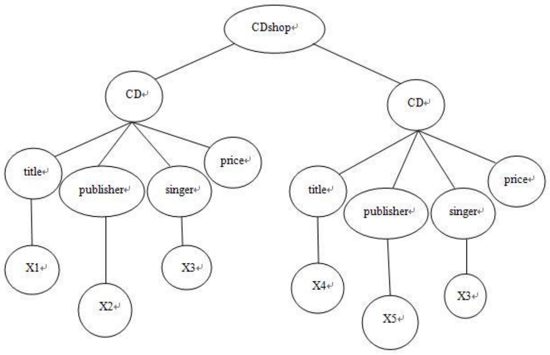
	X1	X2	X3	X4	X5
ONE	1	0	0	0	0
DROP	1	0	0	0	0
Jone	0	1	0	0	0
Records	0	1	0	0	1
KAT-TUN	0	0	1	0	0
Real	0	0	0	1	0
Face	0	0	0	1	0
Je	0	0	0	0	1

Besides, during the process of build the keyword matrix, the index table can also be omitted as shown in table 2:

<Table2> Index Table

Index No.	Content NO.	Text Content
X1	Content 1	ONE DROP
X2	Content 2	Jone Records
X3	Content 3 Content 6	KAT-TUN
X4	Content 4	Real Face
X5	Content 5	Je Records

After we got the index table, the CDshop XML file's tree structure can be shown as in Figure2:



(Figure 2). the new tree structure of the CDshop.xml.

Matrix of XML documents with keywords, which can change the past mode of an XML document query method, not like the structured query language only based on the structure of the content, also not like the keyword query have to do the search through the whole tree structure. And our method is that through the keyword matrix, the keyword which user desired can be corresponding to the index number, and then we can use the index number to do the other processes, such as get the Content Table that we can know the keywords user inquired was in which XML document.

Table 3 is the content table that including numbers of XML documents, also the CDshop.xml was inside it.

<Table3> Number of documents' text contents table

Source document	Content No.	Text Content
CDshop	Content 1	ONE DROP
CDshop	Content 2	Jone Records
CDshop	Content 3	KAT-TUN
CDshop	Content 4	Real Face
CDshop	Content 5	Je Records
CDshop	Content 6	KAT-TUN
...	...	...

With the query method we presenting here, the advantage is: the contents of an XML file will produce a keyword matrix; more copies of the xml document will also generate one keyword matrix, therefore, no matter how many XML documents we have, when we do the text contents query, search in one keyword matrix is enough which can greatly improve the query time cost.

4.2 XML structure pro-processing—structure matrix

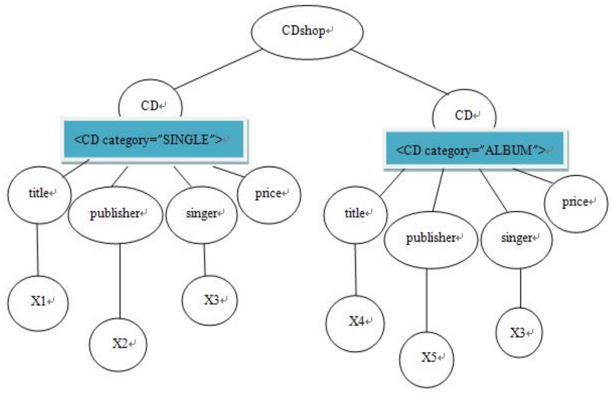
After the text content keyword matrix was produced, we should extract useful information of the structure part of an XML document, which is based on the user query log. Then we can get the words that frequently used during the structural elements query.

As we mentioned before, the CDshop.xml, if users define a set of keywords like this: {Keyword: ONE, Range: SINGLE}, means that want to find a record store in which there is a piece of CD named about "ONE" and the classification is "single". In this case, the keyword is not only in the text content, but also probably in the structure of the attribute or element that only the text content keyword matrix cannot meet the user's requirement. Consequently, we propose a method to solve the problem by establishing the structure matrix that the words users frequently used. Here, we called "frequently used", which determined by the characteristics of an XML document that the tags and attributes are self-describing. Hence, the XML document will be different with different description of elements, tags and attribute, even for the same thing.

Therefore, we build the structure matrix through the content that user frequently used. The same as an example of CDshop.xml, we assume that the category named "SINGLE" 、"ALBUM" is often be queried that we can build a structure matrix based on "SINGLE" and "ALBUM".

Structure matrix is created by ordinary users need to find the structure content in which hierarchy of the XML document tree structure, then the index number node under this hierarchy, that is, the corresponding node of the contents of this structure.

The figure 3 shows that the tree structure with the structure content "SINGLE"、 "ALBUM":



(Figure3). the new tree structure with content of the CD shop.xml.

From the above figure we can see, under the structure content "SINGLE", the index number is: X1, X2, X3, the same as structure content "ALBUM", the index number is: X3, X4, X5. Then the structure matrix can be described as the following tables:

<Table4> "SINGLE" Structure Matrix

	X1	X2	X3	X4	X5
<SINGLE>	1	1	1	0	0

<Table5> "ALBUM" Structure Matrix

	X1	X2	X3	X4	X5
<ALBUM>	0	0	1	1	1

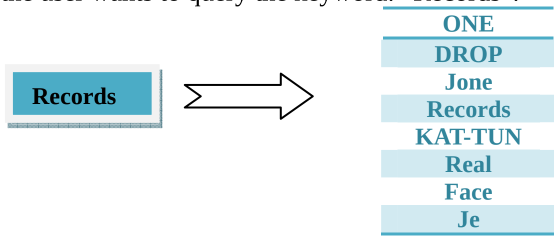
4.3 An effective XML document search method—text content

After building the keyword matrix and the structure matrix, we can go to the query process, first, we will explain the query of the text content, the methodology consist of the following steps:

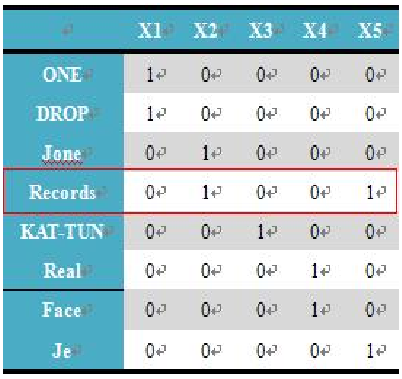
1. comparing the keyword exists in the keyword matrix or not
2. If the keyword we desired was in the keyword matrix then extract the Column matrix which the keyword in.
3. Check the column matrix by the value of each column, if a column value is 1, and then return this column's corresponding index number. Until the check is finished.

Additionally, the step1, When making comparison of the keyword, Once the keyword users' inquired was found, we don't need to search the rest of the collection. Hence, the methodology we represented can save a lot searching time; that is because the same word set of keywords will not be repeated.

In the following, we describe an example of the methodology, and the user wants to query the keyword: "Records".

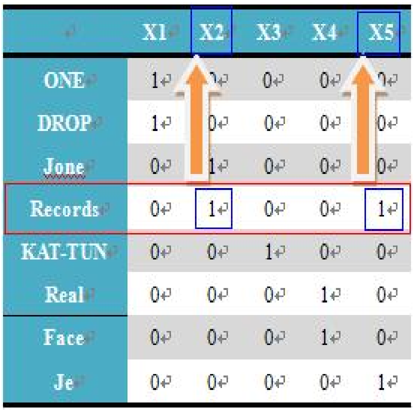


Step1: comparing the keyword exists in the keyword m atrix or not



	X1	X2	X3	X4	X5
ONE	1	0	0	0	0
DROP	1	0	0	0	0
Jone	0	1	0	0	0
Records	0	1	0	0	1
KAT-TUN	0	0	1	0	0
Real	0	0	0	1	0
Face	0	0	0	1	0
Je	0	0	0	0	1

Step: 2 get the "Records" column matrix



	X1	X2	X3	X4	X5
ONE	1	0	0	0	0
DROP	1	0	0	0	0
Jone	0	1	0	0	0
Records	0	1	0	0	1
KAT-TUN	0	0	1	0	0
Real	0	0	0	1	0
Face	0	0	0	1	0
Je	0	0	0	0	1

Step3: check the column matrix's value, get the Index num ber: X2, X5

In addition, If we need to query the compound word, can simply split compound words into two keyword querying in the matrix is possible, consequently with this method will not be the impact of compound words.

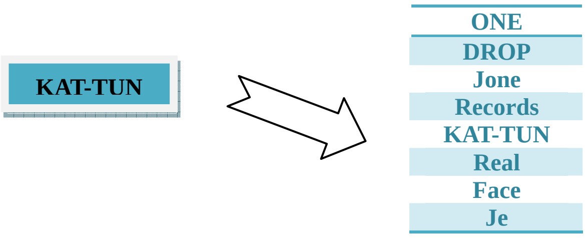
4.4. An effective XML document search method —structure and text content

For users, only for text content query may not be able to meet the needs of their query, in the previous section we describe the query algorithm for the text content, this section we will introduce the algorithm that is able to query both structure and text content.

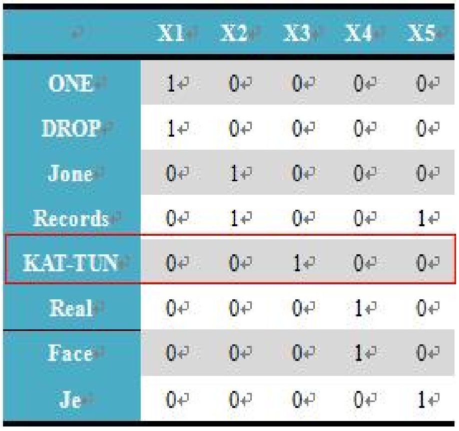
The methodology we mentioned in the last section was querying use the keyword matrix, the methodology in this section will combine the keyword matrix and structure matrix to do the query process. The methodology can be explaining as the following steps:

1. Comparing the keyword exists in the keyword matrix or not.
2. If the keyword we desired was in the keyword matrix then extract the Column matrix which the keyword in.
3. Check the column matrix by the value of each column, if a column value is 1, and then return this row matrix of the structure matrix.
4. Check the row matrix by the value of each row, if a row value is 1, means that the keyword user queried is inside this structure, and then return this row's corresponding index number.
5. Back to step3, until the value of each column matrix have been checked.

The following figure illustrates an example of this methodology we introduced before, and the user wants to query the keyword and structure range like this: {KAT-TUN, single}:

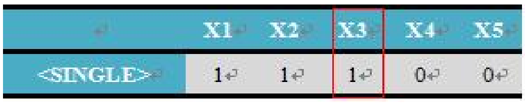


Step1: comparing the keyword "KAT-TUN" exists in the keyword matrix or not



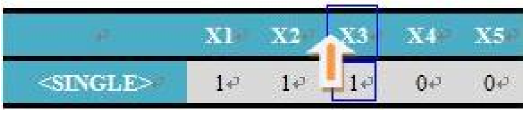
	X1	X2	X3	X4	X5
ONE	1	0	0	0	0
DROP	1	0	0	0	0
Jone	0	1	0	0	0
Records	0	1	0	0	1
KAT-TUN	0	0	1	0	0
Real	0	0	0	1	0
Face	0	0	0	1	0
Je	0	0	0	0	1

Step2: get the "Records" column matrix



	X1	X2	X3	X4	X5
<SINGLE>	1	1	1	0	0

Step3: the third of "KAT-TUN" column matrix's value was 1; return the third row of the SINGLE row matrix



	X1	X2	X3	X4	X5
<SINGLE>	1	1	1	0	0

Step4: Check the value of the column matrix that X3 be the corresponding index number for the {KAT-TUN, single}, also we can know that KAT-TUN was inside the SINGLE range.

5. Conclusion

In this paper we proposed an effective XML document search method to search text contents and structures of XML documents. We showed how to build the keyword matrix of text contexts and structure matrixes of structures in XML documents to improve the efficiency of query time. When there is a great deal of XML documents, the search method we propose can improve much efficiency of query time.

In face of a growing number of XML documents, we have made a query method, for text content and structure (element, tag, attribute, etc.) be part of the query for processing, and changed during the past most of the query language queries, XML documents need to do the whole tree search, causing a waste of time cost for the users, improvements in time efficiency is more able to meet the query requirements. And the method of conducting this study, a large number of XML file query, the more efficiency can play in the time effect.

References

[1] J. Clark and S. DeRose: XML Path Language（XPath）Version 1.0. 1999, available at http://www.w3.org/TR/xpath.
[2] S. Boag, et al: XQuery 1.0: An XML Query Language. 2007, available at http://www.w3.org/TR/xquery.
[3] S. Cohen, et al, "XSEarch：A Semantic Search Engine for XML", Proceedings of the 29th VLDB Conference, pp. 45–56, 2003.
[4] L.Guo, et al, "XRANK：Ranked Keyword Search over XML Documents", Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 16-27, 2003.
[5] Z. Liu and Y. Chen, "Identifying Meaningful Return Information for XML Keyword Search", Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp.329-340, 2007.