2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.72

A Schema-based Method of Summarizing XML Documents

Teng Lv[1], Weimin He[2] and Ping Yan[3+]

[1]Teaching and Research Section of Computer, Artillery Academy, Hefei 230031, P.R.China

[2]Department of Computing and New Media Technologies, University of Wisconsin-Stevens Point, Stevens Point, WI 54481, USA

[3]School of Science, Anhui Agricultural University, Hefei 230036, P.R.China

Abstract. XML has become one of the de facto standards of data exchange and representation in many applications. An XML document is usually too complex and large to understand and use for a human being. A summarized XML document of the original document is useful in such cases. Three standards are given to evaluate the final summarized XML document: document size, information content, and information importance. A method of summarizing an XML document based both on the document itself and the schema is given, which applies schema to summarize XML documents because there are many important semantic and structural information implied by the schema. In our method, redundant data are first removed by abnormal functional dependencies and schema structure. Then tags and values of the XML document are summarized based on the document itself and schema. Our mehtod is a semi-automatic approach which can help users to summarize an XML document in the sense that some parameters must be specified by the users.

Keywords: XML, document summarization, schema, key, functional dependency

1. Introduction

XML (eXtensible Markup Language)[1] has become one of the de facto standards of data representation over World Wide Web and elsewhere. More and more data are stored in XML format. To understand these XML documents with complex structure and abundant data, a human being must spend much time to read such documents. In some cases, it is impracticable, even not impossible, for a human being to read the whole XML document when the document is very large and complex. So it is necessary to present a human being with a summarized form of the original complex and large XML document. Such a summarized XML document is also useful in other applications: querying XML documents, comparing two XML documents, displaying or storing XML documents in a mobile or embedded device which has limited CPU processing ability, display screen and storage spaces, etc. Although such a summarized XML document is very useful, it is difficult to generate a good summarized XML document. Although a human being has good ability of summarizing and analyzing, a computer is not good at doing such things. So the challenge of summarizing XML documents is how to generate such a summarized XML document by computers.

A summarized XML document should grasp the core information of the original document so that a human being can have a basic understanding of the original document. Of course, such a summarized XML document should have less size than the original document considering the storage space and complexity. A good summarized XML document can be evaluated by the following 3 standards:

(1) Document size. The first goal of summarizing XML document is to obtain an XML document with an acceptable size comparing with the original one according to specific applications.

(2) Information content. A perfect summarized XML document should contain the entire information content of the original one, i.e. it is equivalent to the original one in the aspect of information content. But in

+ Corresponding author.
E-mail address: want2fly2002@163.com.

reality, it is impossible for a summarized XML document with less size to contain the entire information content of the original document which has no redundant information.

(3) Information importance. As a summarized XML document can not contain the entire information content of the original one in most cases, it is necessary and practicable to contain the most important information of the original one.

Related work. Text summarization [2, 3] focused on free-flowing texts in text datasets, which is not always applicable to XML summarization as the structure information and semantic information are often important to XML. XML schema summarization[4] is one related topic which summarizes XML schemas rather than XML documents. XML structure summarization [5] is another related topic which summarizes XML structures rather than XML documents. Compression technique [6, 7] is another related topic to reduce the document size without considering the readability to human beings. Other works focused on constructing XML summarization for XML efficient query estimation: StatiX [8] explores schema transformation and schema validation to obtain statistics for query selectivity estimation in XML documents. TREESKETCH synopses [9] can produce fast, accurate approximate answers for XML documents. Bloom histogram [10] is a method for XML path selectivity estimation in a dynamic environment. XSEED [11] is a method to estimate cardinality of XPath queries. Ref. [12] proposed a method to summarize XML data streams other than XML documents. A semi-automatic method to summarize XML collections is proposed in Ref.[13], which applies a template to specify the user requirement and matching rules to extract the summarized XML collections.

To the best of our knowledge, the most related work is Ref.[14] which proposed a method of XML document summarization based on document itself alone. As we know that a schema defines the structure and semantic of an XML document which implies many important information of an XML document. From the former observation, we propose a method of summarizing an XML document based on both document itself and the schema. Such method can make the summarized XML document have a good balance of document size, information content and information importance. More specifically, our approach have the advantages over Ref.[14] in the following aspects: (1) Some data redundancies introduce by abnormal functional dependencies and nested structures are first removed in the summarizing process which can make the XML document concise and meaningful. (2) It can reserve key information of the XML document. (3) It can deal with the difficult situation when a tag occurs many times but with little importance in an XML document.

Contributions. In this paper, we give a method of summarizing an XML document based on the document itself and the schema of the document. We apply schemas to summarize XML documents because there are many important semantic and structural information implied by the schemas (including functional dependencies, keys, and structure information). Our method can help users to summarize an XML document in the sense that some parameters must be specified by the users according to the different requirement such as intended summarized document size, intended information content, intended information importance, and specific XML documents.

Organization. The rest of the paper is organized as following: Section 2 illustrate the first two steps to summarize an XML document by removing data redundancies according to functional dependencies and structure. Sections 3 and 4 are the third and fourth steps to summarize an XML document by summarizing tags and values in the XML document. We conclude the paper and give the future work in Section 5.

2. Removing XML Data Redundancies by Schema

In real XML documents, there are many data redundancies as the causal design of XML schemas and documents. We focus on two kinds of XML data redundancies: data redundancies caused by functional dependencies and data redundancies caused by structure.

2.1 Removing XML data redundancies by functional dependencies

The first kind of XML data redundancies is caused by abnormal XML functional dependencies proposed in our previous work[15]. We do not give the formal definitions of functional dependencies and normal forms here considering the space. Detailed descriptions can be referred to Ref.[15]. In this paper, we focus

on the third normal form of XML document considering the simplicity and applicability in real applications. Higher normal forms may be too complex to understand and apply in real XML documents.

2.2 Removing XML data redundancies by structure

The second kind of XML data redundancies is caused by the nested structure of XML documents. Considering the following XML document D4 which contains a nesting tag Orders:

```
<CustomerOrders>
  <Orders>
      <Orders>
        <Order>
       <OrderID>10643</OrderID>
       <CustomerID>9232</CustomerID>
       <OrderDate>2010-03-25</OrderDate>
      </Order>
    </Orders>
    <Order>
      <OrderID>10692</OrderID>
      <CustomerID>9349</CustomerID>
      <OrderDate>2009-10-03</OrderDate>
    </Order>
    <CompanyName>Alfreds Futterkiste</CompanyName>
  </Orders>
</CustomerOrders>
```

To reduce such redundancies, the tags in sub-tree rooted on the nested tag Orders (i.e. the second Orders tag here) are moved up as sub-tags of the nesting tag A (i.e. the first Orders tag). Of course, the nested tag Orders is removed in the moving-up process. The above XML document D4 can be transformed to the following XML document D5 by above method.

```
<CustomerOrders>
  <Orders>
      <Order>
      <OrderID>10643</OrderID>
      <CustomerID>9232</CustomerID>
      <OrderDate>2010-03-25</OrderDate>
    </Order>
    <Order>
      <OrderID>10692</OrderID>
      <CustomerID>9349</CustomerID>
      <OrderDate>2009-10-03</OrderDate>
    </Order>
    <CompanyName>Alfreds Futterkiste</CompanyName>
  </Orders>
</CustomerOrders>
```

3. Summarizing Tags in An XML Document

After the previous summarizing procedures, an XML document has no data redundancies caused by abnormal functional dependencies and nested structures proposed in Sections 2. In this section, we will focus on the problem of determining the important tags in the original XML document to be included in the summarized XML document.

3.1 Keys of XML documents

As keys[16] are important for querying and understanding an XML document (an example is given in Section 5.5), the keys have priority over other tags to be contained in the summarized XML document. As how to deal with other tags that are not a key or a part of a key, i.e., whether or not they are contained in the summarized XML document, we will propose the method in Section 3.2. For example, considering the following XML document D6:

```
<book>
  <title>database</title>
  <author>Peter Lee</author>
  <price>5USD</price>
  … …
</book>
```

Suppose {title, author} is a key (a book is uniquely determined by the combination of its title and author), the summarized XML document must contain the key information of a book: title and author. This is also a common sense that the book title and its author is the priority information when we browse a book.

3.2 Other tags

For a tag which is not a key or a component part of a key, we can determine the tag importance by its occurrence in the XML document. In general, a tag A is more important than another tag B if A's occurrence is higher than that of B. But in some special cases, it is not always true. Consider the following XML document D7:

```
<book>
  <title>database </title>
  <author>Peter Lee</author>
  <comment>This book introduces the basic concepts of …</comment>
  <comment>Normal forms are discussed …</comment>
  <comment>Query optimization is …</comment>
  <comment>The author publish …</comment>
</book>
```

4. Summarizing the Values of Tags in XML Documents

After the previous summarizing procedures in Sections 2-3, the summarized XML document contains no redundancies proposed in Section 2 and only the important tags with corresponding values of the original document are preserved in the summarized XML document. In this section, we will focus on summarizing the values of the tags in the XML document.

4.1 Summarizing the Values of Tags in XML documents

If the tags with same tag name have multiple values and all the tags have a same parent tag in the XML document, we just include the first tag with its value in the summarized document and a mark is left to indicate that there is more information about the same tag. If the reader is interested in that information, he or she can unfold the mark to browse the information when necessary. Of course, this will not increase the size of summarization, but only increase the process time when unfold the mark to browse the hide information. The intuitive motivation of this treatment is that the information of the first tag and its value is more important than subsequent tags (with same tag names as the first tag) and their values under the same parent tag in general. For example, the first author is more interested than the co-authors of a book for a reader in general. If it is not the case in extreme situations, the reader can browse the interested information when necessary. For example, a book has four authors in an XML document D8 as shown in following:

```
<Book>
  <Title>database</Title>
  <Author>author1</author>
  <Author>author2</author>
  <Author>author3</author>
  <Author>author4</author>
</Book>
```

We just contain the first author information in the summarized XML document as following XML document D9, where the mark ⓜ indicates that there is more information about authors of the book:

```
<Book>
  <Title>Database</Title>
    <Author>author1</author>ⓜ
  </Book>
```

4.2 Treating the length of tag values

For different tags, tag values may vary greatly in length. Some tag values just contain several characters ("short values") and others may contain thousands of characters ("long values"). It is obviously that a summarized XML document should not contain such long values without any change because it occupies too much space comparing to its provided simple information. To deal with the long values, it is sufficient that a summarized XML document contains some fixed length of the characters of the whole long values with a mark left to indicate that there is more information about the same tag. Considering the following XML document D10:

```
<Book>
    <author>Peter Lee<authors/>
    <comment>
        I found it to be a very useful text-book. The concepts are easy to understand and the authors
        provide plenty of examples for better understanding. The book is detailed, which means that if you
        want to go into detail you can, e.g. on what normal form is and how to use it, of the difference
        between relational and…
    </comment>
</Book>
```

The length of author value is 9 characters ("Peter Lee"), but the length of comment value is thousands of characters. We can just choose a fixed length, for example 41 characters, to be include in the summarized XML document with the mark ⓜ indicates that there is more information about comment of the book. The final summarized XML document D11 is following:

```
<Book>
    <author>Peter Lee<authors/>
    <comment> I found it to be a very useful text-book.ⓜ</comment>
</Book>
```

5. Conclusions

This paper proposed a semi-automatic method to help users summarize XML documents, which means that some guidelines must be specified by uses according to different requirements such as intended summarized document size, intended information content, intended information importance, and specific XML documents. The contribution of the method is that the summarized XML document can get a good balance of document size, information content, and information importance of the original document. Another contribution is that the summarized XML document can help a human being to understand the original large and complex document well and present a clear and core information of the original one to the user.

We do not give a whole running example to demonstrate the full aspects of the proposed approach considering the clarity and simplicity of the difficult problem. So we use different examples to demonstrate each aspect of the proposed approach without loss of generality.

6. Acknowledgment

This work is supported by the Faculty Professional Development Grant from the University of Wisconsin-Stevens Point, USA, 2010.

7. References

[1] W3C, Extensible Markup Language, http://www.w3.org/XML/

[2] Hahn, U., Mani, I. The challenges of automatic summarization. Computer, 33(11): 29-36(2000)

[3] Amini, M. R., Tombros, A., Usunier, N., Lalmas, M. Learning-based summarisation of XML documents. Information Retrieval, 10(3): 233-255(2007)

[4] Yu, C. and Jagadish, H. V. Schema summarization. In: Proc. of the 32nd international Conference on Very Large Data Bases, New York: ACM Press, pp.319-330(2006)

[5] Dalamagas, T., Cheng, T., Winkel, K., and Sellis, T. A methodology for clustering XML documents by structure. Information Systems, 31(3): 187-228(2006)

[6] League, C. and Eng, K. Type-based compression of XML data. In: Proc. of Data Compression Conference, Washington DC: IEEE Computer Society Press, pp.273-282(2007)

[7] Maneth, S., Mihaylov, N., and Sakr, S. XML tree structure compression. In: Proc. of the 3rd International Workshop on XML Data Management Tools and Techniques, Washington DC: IEEE Computer Society Press, pp.243-247 (2008)

[8] Freire, J., Haritsa, J. R, Ramanath, M., Roy, P, and Simeon, J. StatiX: Making XML count. In: Proc. of ACM SIGMOD, ACM Press (2002)

[9] Polyzotis, N., Garofalakis, M., and Ioannidis, Y. Approximate XML query answers. In: Proc. of SIGMOD, ACM Press(2004)

[10] Wang, W., Jiang, H., Lu, H., and Yu, J. X. Bloom histogram: Path selectivity estimation for XML data with updates. In: Proc. of the 30th international Conference on Very Large Data Bases, pp.240-251(2004)

[11] Zhang, N., Ozsu, M. T, Aboulnaga, A., and Ilyas, I. F. Xseed: Accurate and fast cardinality estimation for XPath queries. In: Proc. of IEEE 22nd International Conference on Data Engineering, IEEE CS Press, pp.61-72(2006)

[12] Mayorga, V. and Polyzotis, N. Sketch-based summarization of ordered XML streams. In: Proc. of IEEE 25th International Conference on Data Engineering, IEEE CS Press, pp.541-552(2009)

[13] Fischer, G. and Campista, I. A Template-Based Approach to Summarize XML Collections. In: Proc. of LWA, October 10-12, 2005, Saarbrcken, Germany (2005)

[14] Ramanath, M. and Kumar, K. S. A rank-rewrite method for summarizing XML documents. In: Proc. of IEEE 24th International Conference on Data Engineering Workshop, Washington DC: IEEE Computer Society Press, pp.540-547 (2008)

[15] Lv, T., Gu, N., Yan, P. Normal forms for XML documents. Information and Software Technology, 46(12): 839-846(2004)

[16] Buneman, P., Davidson. S. B., Fan, W., Hara, C. S., and Tan, W. C. Keys for XML. Computer Networks, 39(5): 473-487(2002)