XML Retrieval with Results Clustering on Android

Pengfei Liu,Yanhua Chen,Wenjie Xie, Qiaoyi Hu
Department of Mathematics
South China Agricultural University
Guangzhou, China
pfliu@scau.edu.cn, chen_yanhua123@126.com, wenjiexie@stu.scau.edu.cn,huqiaoyi@scau.edu.cn

***Abstract*—XML receives widely interests in data exchanging and information management on both traditional desktop computing platforms and rising mobile computing platforms. However, traditional XML retrieval does not work on mobile devices due to the mobile platforms' limitations and diversities.**

Considering that XML retrieval on mobile devices will become increasingly popular, in this article, we have paid attention to the design and implementation of XML retrieval and results clustering model on the android platform, building on jaxen and dom4j, the XML parser and retrieval engine; furthermore, the K-means clustering algorithm.

As an example of usage, we have tested the prototype on some data sets to the mobile scenario and illustrated the feasibility of the proposed approach. The model demonstrated in this article is available on the mobile XML Retrieval project website: http://code.google.com/p/mobilexmlretrieval/.

XML; retrieval; clustering; Android

I. INTRODUCTION

XML (Extensible Markup Language) is designed to store and process data, structure and semantic information, and have become the standard language for data transmission and exchanging on the web due to its flexibility and self-description [1].

With the mobile computing development and a large number of documents from the web or other origins, more and more information on the mobile platform is being organized in XML format. There is an increasing need to be able to automatically process those structurally rich documents for information retrieval applications on the mobile platform and it is somewhat challenging.

Information retrieval is the foundation for the XML search engine. Jaxen is an open-source XPath library written in Java [2]; while dom4j is an open-source library that works with XML, XPath and XSLT on the Java platform [3].

XML clustering is to group similar documents to facilitate searching due to similar documents can be searched and processed within a specific category. The clustering of XML documents is effective for document management and the storage of XML documents [4].

Evidently, XML retrieval and mining plays an important role in computing research areas and is mostly done on computers in the past, but few relative researches on mobile sets are reported. Supposing that one day every mobile device can easily connect to other devices, including data generating devices and can exchange data freely via Bluetooth or WIFI, and then we can process data just by mobile devices realtime. The idea of mining XML documents anywhere and anytime is very amazing but not crazy.

Clustering XML data is more complicated than common text data as XML allows inserting structural and conceptual aspects into document content. An XML document includes tags and data; while tags describing names of elements contain concepts as text data. Besides that, structure tags also show the relationship between elements.

Now, research work dedicated to XML document clustering mainly has three types, including structure-based method, content-based method and combination of both.

Recently, several clustering techniques that consider the structure and the contents of XML documents are studied. References [5-8] had applied different clustering techniques to XML documents represented in different models.

Most of the popular XML retrieval models are designed to work on computers. There are no corresponding models on mobile computing environments and not much works about data mining or XML mining devoted on mobile environment. Liu *et al* have designed and implemented a mobile data mining tool on the android platform [9]. Gillick *et al* have presented the demonstration of cluster visualization technique for mobile devices [10].

Considering the advantages of mobile XML data mining, which need not send local data to other servers; the model just runs the data on local and this is an interesting idea. This paper has proposed a model about XML document retrieval and clustering on the Android platform. The experiments have shown that the model works well and efficiently.

The rest of the paper is organized as follows. In section II, we have proposed a method to apply classical K-means clustering algorithm to XML documents. In section III, we have evaluated the effectiveness of our model with several examples. Finally, we have made a conclusion in section IV.

II. METHOD

In this part, we have offered an implementation of the core retrieval functions underlying jaxen and dom4j. TF-IDF, term frequency-inverse document frequency, is often used as a weighting factor in information retrieval and text mining [11]. It is used in this project as a numerical statistic that reflects how important a word is to a document in a collection or corpus .

Published by Atlantis Press, Paris, France.

For quick computing, we have used the vector space model (VSM) as the algebraic model for representing text documents; It is used for representing text documents as vectors of identifiers. VSM is widely used in information filtering, information retrieval, indexing and relevancy rankings [12]. We have used cosine similarity as measure of similarity between two vectors of an inner product space [13]. The general clustering algorithm of k-means [14] to locate a document in its proper cluster; it tries to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean.

In this project, we adopt the combined similarity of structure and content method as the solution for clustering XML documents considering previous researches.

We have specified a similarity to each existing cluster. For each entering document, if the similarity is bigger than the threshold, then it is placed in the corresponding cluster; else a new cluster is created, and the document is placed in it. Below is the flow chart of the algorithm. A strictly prove of the algorithm's convergence is not given, but in the practical tests in next section, it can achieve a fast convergence.

Input: a number of leaf nodes got by retrieval operations.

Output: Clustering results of several groups.

Step 1: If the input number of leaf nodes is less than the threshold, just set each leaf node as a cluster and return; else go to Step 2;

Step 2: Select the first batch of leaf nodes as the cluster centers.

Step 3: Calculate the similarity between each leaf node and each cluster center by distance combination of tag and content, the leaf nodes belongs to the most similar cluster.

Step 4: Recalculate the cluster centers including tag centers and content centers, and determine whether the cluster centers change; if changed, then repeat step 3 and step 4, otherwise turn to step 5.

Step 5: Returns clustering results.

TABLE I. CODE OF BUILDING TAG VSM

```
public double[] buildTagTF_IDF(int[] wrod_frequency,
                    int[] contentString,
            int numString, int sizeString, int a) {
                            double[] tf_idf = new
              double[wrod_frequency.length];
                  for (int i = 0; i < wrod_frequency.length;
                            i++) {
                                    double tf = (double)
               wrod_frequency[i] / sizeString;
                                            double idf =
  Math.log10((double) numString / contentString[i]);
                                tf_idf[i] = tf * idf / (a + i);
                                        }

                            return tf_idf;
                          }
```

TABLE II. CODE OF CLUSTERING

```
public XMLTreeElement[][] cluster(double[][] tag,
                double[][] content, int k) {
                    int numTagKeyWord = tag[0].length;
                        int numContentKeyWord =
                  content[0].length;
                        double[][] tagCenter = new
              double[k][numTagKeyWord];
                         double[][] contentCenter = new
            double[k][numContentKeyWord];
                        double[][] newTagCenter = new
              double[k][numTagKeyWord];
                       double[][] newContentCenter = new
            double[k][numContentKeyWord];
                                int[][] clusters;
                             for (int i = 0; i < k; i++) {
                  contentCenter[i] = content[i].clone();
                                tagCenter[i] = tag[i].clone();
                                        }
                                    while (true) {
          clusters = group(tag, content, tagCenter, contentCenter);
            newTagCenter = newClusterCenter(tag, clusters);
             newContentCenter = newClusterCenter(content,
                                clusters);
                    if (!equal(tagCenter, newTagCenter)
                  && !equal(contentCenter, newContentCenter))
                                        {
                        tagCenter = newTagCenter.clone();
                      contentCenter = newContentCenter.clone();
           newTagCenter = new double[k][numTagKeyWord];
                        newContentCenter = new
                  double[k][numContentKeyWord];
                                            } else
                                                break;
                                        }
                             XMLTreeElement[][] result = new
                  XMLTreeElement[clusters.length][];
                            for (int i = 0; i < result.length; i++){
           result[i] = new XMLTreeElement[clusters[i].length];
                      for (int j = 0; j < clusters[i].length; j++){
                            result[i][j] = elements[clusters[i][j]];
                                                }

                                        }
                                    return result;
                                }
```

To demonstrate the VSM,TF-IDF and clustering application in XML searching and clustering coding, a VSM and a TF-IDF builder based on the XML's tag and content, also the clustering method are displayed to fulfill the models demanded by the architecture; they are listed in TABLE I, II,III respectively. Of course, there are still many key codes

of the project that not listed; nevertheless, we plan to release this work as an open source project in the future.

TABLE III. CODE OF BUILDING TAG IDF

```
public double[][] buildTagVsm(String[] tag, String[]
                keyWords) {
                    double[][] vsm = new
        double[tag.length][keyWords.length];
                    int[][] tagWordF = new
        int[tag.length][keyWords.length];
                    int sizeWord = 0;
                for (int i = 0; i < tag.length; i++) {
                        Pattern p =
        Pattern.compile("[/!.?<>\\,\" ]");
                    String[] strs = p.split(tag[i]);
                      sizeWord = strs.length;
                       tagWordF[i] =
    wordFrequency(strs, keyWords, includeTagKeyWords);
                                }
                for (int i = 0; i < tag.length; i++) {
                                    vsm[i] =
    buildTagTF_IDF(tagWordF[i], includeTagKeyWords,
                    tag.length, sizeWord, 1);
                                }
                            return vsm;
                             }
```

In the next subsection, we will introduce model's tests with details.

III. EXPERIMENT

To fully test mobileXMLRetrieval, the experimental mobile device of HTC Dream mobile phone (also known as T-Mobile G1) is used. The processor of G1 is ARM based MSM7201A with 528 MHz speed and a 192MB RAM memory; while development tools are Android SDK and development language of Java.

In order to validate the approach proposed, we had evaluated the retrieval and clustering performance of the prototype model by running time. We have measured the effectiveness of models by running time together with a test collection built from the XML Data Repository (http://www.cs.washington.edu/research/xmldatasets/).Test results are shown in table IV and table V.

TABLE IV. RESULTS OF SINGLE DOCUMENT SEARCHING.

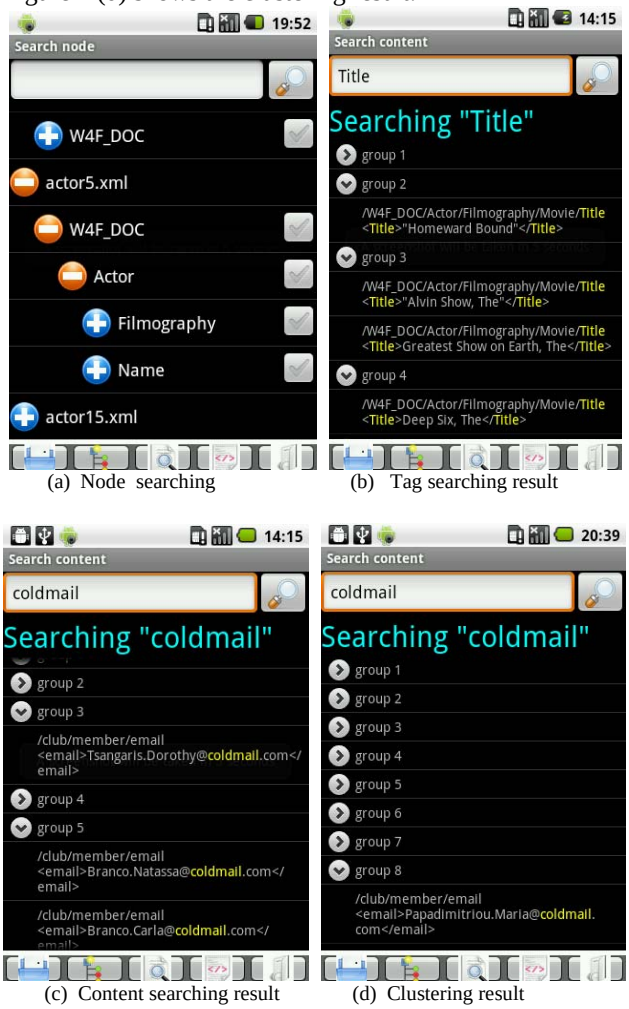
Document name	Size (KB)	Time(s)
club4	10	0.382
club5	3	0.06
club6	7	0.291
club7	8	0.267
club8	9	0.397

TABLE V. THE RESULTS OF MULTIPLE DOCUMENTS SEARCHING.

Amount of documents	Size (KB)	Time(s)
5	37	1.754
4	28	1.472
3	20	0.821
2	17	0.764

The above experiments and their results have shown that the model works well and efficiently.

During the test procedure, we have taken some screenshots of mobileXMLretrieval. Figure 1 (a) shows the node searching screen; Figure 1 (b) shows searching tag results; Figure 1 (c) shows the searching content result; Figure 1 (d) shows the clustering result.



(a) Node searching (b) Tag searching result

(c) Content searching result (d) Clustering result

Figure 1. Screenshots of examples.

The main finding is that retrieval engine and clustering algorithm work well on the Android, and it is a complementary approach for finding information in XML format.

IV. CONCLUSION

We have presented a mobile XML retrieval and clustering model named mobileXMLretrieval that works well on test data sets; that reveals the possibility of XML clustering on mobile devices.

XML clustering on mobile devices is not very practical now; however, it will be accepted without waiting too long with the rapid developments of mobile hardware in short years. Furthermore, mobileXMLretrieval can be transplanted to other mobile sets with the Java environment. We plan to release this work as an open-source project in the future.

As future developments of the mobileXMLretrieval, we plan to enhance the model by improving the clustering algorithm, also completing other functions such as XML classification.

ACKNOWLEDGMENT

Funding: Supported by Scientific Research Foundation of South China Agricultural University (4900-k11045) and Natural Science Foundation of Guangdong Province, China (No.S2011040004387, No.S2011040001127 and No. S2012010009961), National Natural Science Foundation of China, Tian Yuan Special Foundation（No. 11226186）.

REFERENCES

[1] "XML Media Types, RFC 3023". IETF. 2001-01. pp. 9–11.

[2] jaxen: universal Java XPath engine. http://jaxen.codehaus.org/

[3] Dom4j. http://dom4j.sourceforge.net/

[4] Jianghui Liu, Jason T. L. Wang, Wynne Hsu, *et al*, "XML Clustering by Principal Component Analysis," International Conference on Tools with Artificial Intelligence,IEEE Press, Nov,2004, pp. 658-662.

[5] Jong P. Yoon, Vijay Raghavan, "Bitmap Indexing-based Clustering and Retrieval of XML Documents," ACM SIGIR'01 Workshop on Mathematical/Formal Methods in IR Naive clustering of a large XML document collection. ACM Press, 2001.

[6] Gianni Costa, Giuseppe Manco, Riccardo Ortale and Andrea Tagarelli, "A Tree-Based Approach to Clustering XML Documents by Structure," KNOWLEDGE DISCOVERY IN DATABASES: PKDD 2004 Lecture Notes in Computer Science, Springer-Verlag, Volume 3202, Sept,2004, pp.137-148.

[7] Sangeetha Kutty, Richi Nayak, and Yuefeng Li, "XML documents clustering using a tensor space model," In Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining, Springer-Verlag. Volume Part I ,May,2011:488-499

[8] Theodore Dalamagas, Tao Cheng, Klaas-Jan Winkel,*et al*, "A methodology for clustering XML documents by structure," Information Systems. Volume 31, Issue 3, pp. 187-228,2006.

[9] Pengfei Liu, yanhua Chen, Wu lei tang, *et al*, "Mobile weka as a data mining tool on android," Advances in Electrical Engineering and Automation. Springer-Verlag, 2012, pp. 139:75-80.

[10] Gillick, Brett; AlTaiar, Hasnain; Krishnaswamy, Shonali, *et al*, "Clutter-adaptive visualization for mobile data mining," 10th IEEE International Conference on Data Mining Workshops, IEEE Press, 2010,pp.1381-1384.

[11] Salton G; McGill MJ. Introduction to modern information retrieval. McGraw-Hill.1986

[12] G. Salton , A. Wong , C. S. Yang, "A vector space model for automatic indexing," Communications of the ACM, v.18 n.11, pp.613-620,1975

[13] P.-N. Tan, M. Steinbach & V. Kumar, Introduction to Data Mining, Addison-Wesley. 2005.

[14] MacQueen, J. B. "Some Methods for classification and Analysis of Multivariate Observations," Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press.1967, pp. 281-297.