Acta Technica *62* No. 2B/2017, 183–192

A text information extraction algorithm based on label path clustering[1]

Jing Qiu[2,3], Yuhan Chai[2]

Abstract. XML documents are widely used in computer and Internet technology. Based on XML documents, data exchange can be effective. However, the extraction of text information determines the utilization and efficiency of the data. The traditional text information extraction algorithm has low accuracy and low efficiency. Therefore, a text information extraction algorithm based on label path clustering was proposed in this paper. Then the traditional PBClusreing algorithm was improved. Finally, the verification test was carried out. As can be seen from the result, compared with the traditional algorithms, the improved text information extraction algorithm has significant advantages in noise elimination and efficiency. Therefore, the improved text information extraction algorithm can be applied to the specific practice, which is conducive to promoting communication and utilization of Internet data, and has positive significance for the development of the Internet.

Key words. Clustering, information extraction, frequent path, XML.

1. Introduction

With the continuous development and application of modern computer technology and Internet technology, people rely more and more on the Internet, and the Internet has greatly changed people's way of life [1]. When people use computers and the Internet to communicate with each other, there are more extensible markup language applications (hereinafter referred to as XML), which can exchange data among different data sources [2]. In terms of structure and function, XML can not only represent structured data, but also represent unstructured data of the Web class, which is widely used in database, data exchange, e-commerce, natural language conversion and so on, and has become a standard language for data representation and exchange [3]. With the increasing of XML class data, how to effectively carry

[1]This paper is supported by the National Natural Science Foundation of China (project No. 61300120).

[2]Department of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, Hebei, 050026, China

[3]Corresponding author

out this kind of data information has become more important. XML data is mostly represented in the form of XML, and in the XML document, the two main forms are hierarchical and semi-structured. At present, the data mining methods for this kind of document include classification processing, clustering analysis and structure mining (Yan et al. 2015) [4]. The clustering processing of XML documents is usually done in two aspects: structured clustering processing and content clustering processing [5]. In this paper, a text information extraction algorithm based on label path clustering was proposed to achieve the rapid and accurate extraction of XML class information. And the algorithm was compared with the traditional method. Thus, the accuracy and practical value of the algorithm were understood.

2. State of the art

The development of information technology leads to exponential growth of Internet information, and it is harder and harder for people to acquire the knowledge they want. It is a challenge to find the desired information from heterogeneous, dynamic and distributed Web information library and convert it into the information and knowledge that users need [6]. When using Web retrieval to query relevant information, a list of results is returned to the user, and according to the correlation algorithm, a variety of factors are calculated to carry out sorting [7]. In these sorting algorithms, the frequency, location, number, importance, and number of clicks being queried are considered synthetically [8]. At present, the related clustering analysis of HTML language files applied in Internet has been studied more and more, and the fast and accurate extraction of text information has been realized. However, the research of information extraction algorithms for XML documents is relatively small, which is not conducive to the use and communication of Internet information [9]. At present, TreeFinder and Tag-based are the main methods to extract structural information of XML documents. In this paper, a collection of paths from roots to leaf nodes was used to represent XML documents [10].

As extensible markup languages, both XML and HTML belong to the category of SGML, and there are many similarities between them. They are the common standard languages in the world [11]. XML is a subset based on the SGML language, but different from the SGML language, it avoids the complex features of SGML language to some extent [12]. As for the basic functions of XML, in terms of file data, documents are varied and diversified by description, and its technology is mainly oriented to data applications, in which, the meaning of some data and material can be explained by marking [13]. In processing patterns, XML documents are often abstracted into semantic layer, syntax layer, character layer, structure layer, binary file and so on, which are similar to network protocol TCP/IP [14].

3. Methodology

3.1. Preprocessing of XML documents

The XML document can be represented in the form of a tree. The main idea of the new algorithm proposed in this paper is to use sequential pattern mining to obtain frequent path sets among different XML documents. XML documents with similar structure have similar frequent paths, while, for XML documents of different structure, similar frequent paths will be relatively few. Therefore, the clustering documents, frequent paths can be used as their characteristics. Converting the XML document into a tree structure, then each node in the tree corresponds to an element of the XML document. In this paper, tree structure was represented based on path. In this way, each tree is represented by a collection of paths, and the ordered nodes form the root node to the leaf node together form the paths. In the study, a simplified XPath was used to represent the path. It contains both element information and hierarchical information in the path, so that the XML document can be represented by a simple collection of XPath expressions [15]. Firstly, the XML document was preprocessed, to extract the corresponding path set from the tree structure. And then the duplicate path was deleted. The specific process is shown in Fig. 1.

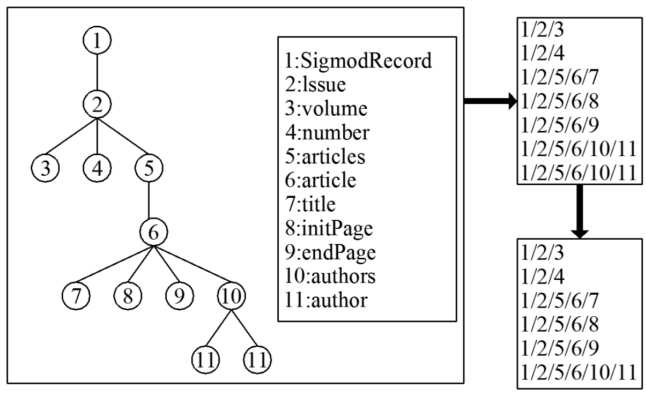

Fig. 1. The path set for the XML document

When using this representation, in the path, in order to indicate a change in the hierarchy of the different elements, a "/" will be added behind the element. Therefore, the Xpath paths obtained after preprocessing include both path element information and hierarchical information. Since the XML path is a relatively simple sequence, each element is a XML node and can't be a collection containing multiple items. Then, in the XML path, if a path is not included in all other paths, then the path is the maximum path. Path support is involved, the computational method of which is the ratio of the number of occurrences of a path to the number of documents in a path set. If in the XML document the path A is contained in the path B, then the path A can be called a subsequence of path B. It is assumed that a XML path

set is D, which can be determined by the document number and serial numbers to which each path belongs, respectively $doc - id, seq - id$.

The number of occurrences used to compute path support refers to the number of XML documents that s is contained in D. In this paper, only the frequent paths among different documents were searched, while the XML paths that were repeated in the same document were not counted repeatedly. If the given support threshold is $\min\sup p$, the support number of the sequence database for path$s$ in XML is no less than $\min\sup p$, and s is known as the path model. The document set of XML is converted to the corresponding path set. According to the given minimum support threshold $\min\sup p$, the sequential pattern mining algorithm is used to find the path patterns of all XML. After removing all the paths contained by other paths, the maximum path pattern can be obtained.

3.2. Frequent path mining process and document clustering

The two XML document trees are input to sample the mining process of the XML frequent path pattern, as shown in Fig. 2. The minimum support is set to 100 %. Preprocessing is performed in the manner described earlier to obtain a XML path set.

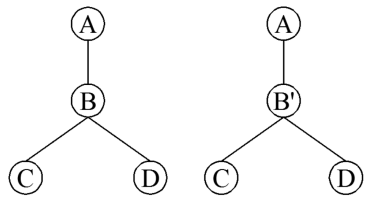

Fig. 2. Two XML document counts

The specific steps of XML frequent path mining are:

(1) All frequent nodes satisfying minimum support, referring to the frequent paths of length 1, should be found out.

(2) According to the frequent nodes found, a Hash mapping table based on frequent node sets is constructed. By using this mapping table, the path of the XML document machine can be converted and represented. Each frequent node name is replaced by a corresponding integer, and the result of the conversion is shown in Fig. 1. After the processing and transformation, the integer sequence can be used to represent the path, thus reducing the identification space of the XML document path sequence set, and improving the subsequent mining speed.

(3) AprioriAll algorithm is used to carry out iteration, and thus a collection of frequent sequences of lengths from 1 to maximum length can be obtained.

(4) The path contained in other patterns in the frequent sequence is eliminated, so as to obtain the maximal frequent sequence.

The path transformation representation is described in Table 1.

The XML frequent path is used as clustering feature. The number of its maximum path patterns is affected by the minimum support of the path mining phase.

The appropriate minimum support can effectively eliminate noise data and avoid excessive features missing, so as to ensure that the classification features of the XML document set can be effectively represented in the smaller dimension feature space.

Table 1. Path transformation representation

Document number	1	1	2	2
Path number	1	2	1	2
Original path	A/B/C	A/B/D	A/B'/C	A/B'/D
Converted path	A//C	A//D	A//C	A//D
After mapping	1,2,2,3,	1,2,2,4	1,2,2,3	1,2,2,4

In the vector intermediate model, the PBClusreing algorithm represents the XML document set. If the number of maximal frequent paths obtained by XMNL frequent paths in the excavation stage is n, then an n dimensional vector is used to represent each XML document. If an XML document contains the maximum frequent path m, then in the corresponding vector $S_m = 1$ or $S_k = 0$. When PBClusreing algorithm calculates the similarity between documents, the traditional Euclidean distance is used to perform calculations, and then the clustering algorithm is adopted to perform clustering.

3.3. Improvement of text information extraction algorithm

There are some problems in the existing information extraction algorithms, and previous studies have also tested the performance of the PBClusreing algorithm. The problems existing in the traditional PBClusreing algorithm are mainly embodied in the following aspects. The accuracy of clustering is not high. Even if the minimum support of very frequent and verifiable frequent paths is selected, the accuracy of the final clustering results is only about 70%, which is also the reason for the improvement of the algorithm. In addition, the sensitivity of PBClusreing algorithm to noisy data is high. In the performance of artificial data and actual high noise data, the actual noise situation has not been fully considered. And the scalability of the traditional PBClusreing algorithm is poor. Due to the use of aggregation hierarchical clustering algorithm and AprioriAll algorithm, the time complexity required is higher. In the face of large amounts of data, it is difficult to effectively deal with the data, and its processing efficiency is low. Therefore, aiming at many problems existing in the algorithm, some targeted improvements and research were made in this paper.

The principle of PBClusreing algorithm is to define the distance between documents directly through Euclidean distance. Although it is very easy to understand in the low latitude space, the distribution of the data will obviously improve the noise level when the high-dimensional data is encountered. And thus, in the conventional sense, Euclidean distance definition is no longer valid, and cosine similarity can be used to document similarity calculation.

Boolean vectors are often used when documents containing frequent paths are

represented by the PBClusreing algorithm. However, the differences among the lengths of the frequent paths are not fully taken into account. There are nowa, b, c documents, and two documents a, b contain the frequent path s_1, and b, c documents contain the frequent path s_2. If the frequent pathss_1 is longer than s_2, it is considered that a is closer to the document b than c. The vector of the XML document can be weighted, and integer vectors are used to represent an XML document, such as $\{s_1, s_2, \cdots, s_n\}$. If there is a k maximum path model in a XML document, then its corresponding length assignment is s_k, or $s_k = 0$. The key codes based on experiments to obtain weighted paths are shown in Fig. 3.

```
public AprioriAlgorithm(double minsupp, string[] files)
        {
            StringBuilder strb = new StringBuilder();
            strb.AppendFormat("{0}:start reading {1} xml docs. . . .",
                System.DateTime. Now. ToString(),
                files.Length;
             strb.AppendLine();
             ininInfo = strb. ToString();
            this. sequenceData = new SequenceDataSet(minsupp, files);
              StringBuilder strb2 = new StringBuilder();
              strb2.AppendFormat("{0} :generate length of 1:{1}",
                  Sysytem.DateTime.Now. ToString(),
           . . . . . .
```

Fig. 3. Key codes for weighted paths

In the use of PBClusreing algorithm, the relatively classic AprioriAll algorithm is applied, the principle of which is anti-monotone. If a sequence fails to pass the test, all sequences that contain it are considered impossible to pass the test. Based on this feature, the search space can be cut to make sequential pattern mining more efficient. But whether it is AprioriAll algorithm or agglomerative hierarchical clustering algorithm, the time complexity is still relatively large, and can't meet the requirements of large document set processing. On this basis, the more efficient PrefixSpan algorithm can be applied. In addition, the K-mean algorithm can be used in the clustering division, and the efficiency of the algorithm is also higher. In this paper, different types of XML document formats can be considered as clustering centers, and the high-dimensional vector space was relatively sparse, so that the K-mean algorithm was also applicable. The K-hierarchy algorithm was used to replace the hierarchical clustering algorithm. Thus, the improved PBClusreing algorithm based on the K-mean algorithm and the PrefixSpan algorithm can be obtained.

4. Result analysis and discussion

4.1. Similarity calculation of documents

In this study, a comparative analysis of different clustering algorithms was carried out. In the four classes of use, each class contained 150 documents. When the minimum support degree of the frequent path was 5 %, Euclidean distance and cosine similarity coefficients were compared and analyzed. And the clustering hierarchy and

K-mean clustering algorithm were adopted. The specific results are shown in Fig. 4.

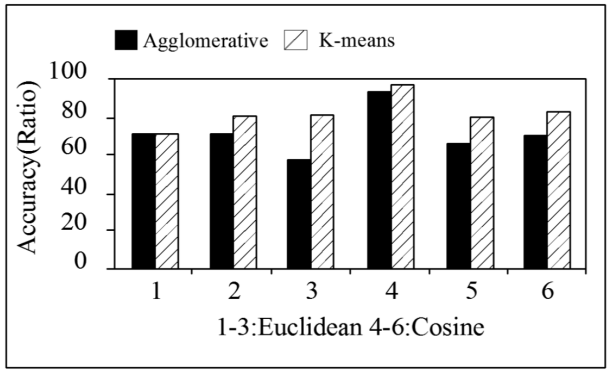

Fig. 4. Contrastive analysis of two methods of measurement

It can be seen from Fig. 3 that the clustering accuracy obtained by cosine similarity coefficients is better than that of Euclidean distance under the same algorithm. And in the case of using the same metric method, the K-mean clustering algorithm has better effect than the hierarchical clustering algorithm. It can be seen that, when calculating the document similarity of the path schema, the cosine similarity coefficient can achieve better results compared with the traditional Euclidean distance method.

The path weighted analysis was further carried out. In 11 classes of documents, each type of document contained 150 files. When the minimum support of the frequent path was 8 %, the contrast experiments were conducted to analyze whether the 10 groups joined the path length weight. From the results of the study, it can be seen that when the path length is added, the accuracy of clustering is improved, and the range of improvement is about 5–10 %. Therefore, the path model can effectively reduce noise and eliminate the influence of short path noise.

4.2. Algorithm comparison

In the case of frequent path with minimum support of 5 %, PrefixSpan algorithm and AprioriAll algorithm were compared in terms of the time efficiency. The size of the document, ranging from 200 to 1400, was divided into 7 levels. The specific results are shown in Table 2.

Table 2. Comparison and analysis of time efficiency between PrefixSpan and AprioriAll algorithms

Time (s)	200	400	600	800	100	1200	1400
PrefixSpan	0.29	0.85	1.21	1.51	1.76	1.82	2.03
AprioriAll	13	40	57	58	53	111	125

As can be seen from Table 2, the PrefixSpan algorithm is more efficient and has a significant advantage compared with the AprioriAll algorithm. Four types of

documents were further selected, 150 in each category. When the support was 5 % and 15 %, respectively, 117 and 49 frequent paths were obtained. Then the K-mean clustering and aggregation hierarchy were compared and analyzed in terms of time and accuracy. The specific results are shown in Table 3. As can be seen from Table 3, the clustering accuracy obtained by the two algorithms is comparable in the 117 and 49 dimensions. However, further analysis shows that the K-mean algorithm has a more significant advantage than the aggregation level algorithm.

Table 3. Comparative analysis of time efficiency between aggregation level and K-mean

Time (s)	Aggregation level	K-mean
49 dimensional	112 (100 %)	0.017 (98 %)
117 dimensional	173 (96 %)	0.03 (94 %)

5. Conclusion

XML documents are used more frequently in data usage. However, there are many problems such as low efficiency and low accuracy in the traditional text information extraction algorithm. Therefore, a more efficient and reliable algorithm for text information extraction is urgently needed. Based on the traditional PB-Clusreing algorithm, the K- mean algorithm and the PrefixSpan algorithm were introduced in this paper, and the large data text set can be processed. In order to verify the practicability of the algorithm, the computation of document similarity and the performance of other algorithms were compared and analyzed. It can be seen that the cosine similarity coefficient has better effect, and the path model can reduce the influence of noise effectively. In addition, the PrefixSpan algorithm will have higher time efficiency than the traditional AprioriAll algorithm. As for the accuracy of clustering, the K- mean algorithm also has a more prominent advantage. In summary, compared with the original algorithm, the improved text information extraction algorithm has better application effects, especially for high noise and large number of processing. And the accuracy is also improved by 10–20 %, and the time efficiency is also significantly improved.

References

[1] J. S. Leu, J. H. Chen, K. H. Li: *Hybrid search scheme for social networks supported by dynamic weighted distributed label clustering.* IEEE Transactions on Computers *64* (2015), No. 9, 2586–2594.

[2] M. Piernik, D. Brzezinski, T. Morzy, A. Lesniewska: *XML clustering: A review of structural approaches.* Knowledge Engineering Review *30* (2015), No. 3, 297–323.

[3] S. He, P. Samara, J. Burgers, L. Schomaker: *A multiple-label guided clustering algorithm for historical document dating and localization.* IEEE Transactions on Image Processing *25* (2016), No. 11, 5252–5265.

[4] W. Yan, F. Suo: *Top-k ranking of preference query results for XML based on simi-*

larity of contextual states. Journal of Computational Information Systems *11* (2015), No. 12, 4439–4446.

[5] M. Piernik, D. Brzezinski, T. Morzy: *Clustering XML documents by patterns.* Knowledge and Information Systems *46* (2016), No. 1, 185–212.

[6] Y. Xia, L. Nie, L. Zhang, Y. Yang, R. Hong, X. Li: *Weakly supervised multilabel clustering and its applications in computer vision.* IEEE Transactions on Cybernetics *46* (2016), No. 12, 3220–3232.

[7] N. Shafieian: *A novel method for transforming XML documents to time series and clustering them based on delaunay triangulation.* Applied Mathematics *6* (2015), No. 6, 1076–1085.

[8] L. Guo, W. Zuo, T. Peng, L. Yue: *Text matching and categorization: Mining implicit semantic knowledge from tree-shape structures.* Mathematical Problems in Engineering *2015* (2015), Article ID No. 723469, 1–9.

[9] K. Li, Y. Liu, Q. Wang, Y. Wu, S. Song, Y. Sun, T. Liu, J. Wang, Y. Li, S. Du: *A spacecraft electrical characteristics multi-label classification method based on off-line FCM clustering and on-line WPSVM.* Plos One *10* (2015), No. 11, e0140395.

[10] M. L. Zhang, L. Wu: *Lift: Multi-label learning with label-specific features.* IEEE Transactions on Pattern Analysis & Machine Intelligence, *37* (2015), No. 1, 107–120.

[11] N. Piroonsup, S. Sinthupinyo: *Semi-supervised cluster-and-label with feature based re-clustering to reduce noise in Thai document images.* Knowledge-Based Systems *90* (2015), 58–69.

[12] M. Hucka, F. T. Bergmann, S. Hoops, S. M. Keating, S. Sahle, J. C. Schaff, L. P. Smith, D. J. Wilkinson: *The systems biology markup language (SBML): Language specification for level 3 version 1 core.* Journal of Integrative Bioinformatics *12* (2015), No. 2, paper 266.

[13] S. Cheng, M. T. Martinez-Ingles, D. P. Gaillot, J. M. Molina-Garcia-Pardo, M. Liénard, P. Degauque: *Performance of a novel automatic identification algorithm for the clustering of radio channel parameters.* IEEE Access: Toward Ubiquitous Real-Time Radio Propagation Modeling *3* (2015), 2252–2259.

[14] C. Bothorel, J. D. Cruz, M. Magnani, B. Micenkova: *Clustering attributed graphs: Models, measures and methods.* Network Science Journal *3* (2015), No. 3, 408–444.

[15] J. Levatić, D. Kocev, S. Džeroski: *The importance of the label hierarchy in hierarchical multi-label classification.* Journal of Intelligent Information Systems *45* (2015), No. 2, 247–271.

Received May 7, 2017