An Improved Algorithm of Similarity Based on Clustering in XML

Puqing Wang

School of Information, Nanchang Institute of science & Technology, Nanchang, 330108, China.

luochenpaper@163.com

Keywords: XML; clustering; similarity; diversification; weight.

Abstract. There are two query methods of the structure query and keyword query on XML query. The keyword query is a more common way of XML query, which is often not accurate to the ambiguous of the keyword. In ordered to change it, the keyword cluster analysis is introduced to this paper, and the similarity of the diversified weight clustering to improve the accuracy of clustering, which can improve the accuracy of the keyword query on XML query.

1. Introduction

The XML language provides a representational form of semi-structure for data management, which makes more clearly logical relationship and more explicit expression on XML query of semi-structured and no-structured documents, and can ensure clear goal on XML query. However, the query result will be affected by keyword. The ambiguous keyword will lead to wrong or incomplete result. In views of the problem, Li proposes method of clustering query keyword in the paper [1]. Using the method, the ambiguous and incomplete keyword can be repaired and improved, which will query accurate and complete information, and improve recall and precision of data query.

2. The Research Status

There were two the traditional similarity algorithms in XML document. The first similarity algorithm was based on structure. The solution method of the similarity was a process that compares the contained elements of document fragments to edge of connected nodes in the algorithm, which paid attention to structure. The advantage of this method was simple and clear, but it neglected the content information, which led to inaccurate clustering. The other similarity algorithm is based on content and structure. The solution method of its similarity was to translate document fragments into space vector, which paid attention to content but neglected structure relationship.

The similarity algorithm based on structure calculated document fragments in accordance with logic structure feature of document, the contained elements of document fragments and edge of connected nodes [2]. There were some common similarity algorithms based on structure [3], such as node-edge method, path matching method, edit-distance method and Fourier transform method, etc.

XML document information is composed of data structure and data content [4]. The similarity computing based on structure only focused structure and neglected content, which led to inaccurate result. In ordered to compute accurately, some researchers proposed some methods of similarity computing, such as Set/Bag model method, inner product method, cosine method and Euclidean Distance method, etc [5].

1) Set/Bag Model method. In compute document similarity, it took on element of document as the feature vector, and match with feature vector. The percentage of the same feature vector presented as similarity degree. The same vectors more, then the similarity degree are higher, conversely lower. For document A and document B, the similarity degree of the feature vector is as shown in Figure 1.

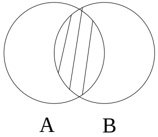


Figure.1 Degree of similarity

The left of Figure 1 is document A, and the right is document B, this shaded part represents the similar part of two documents. The similarity degree is proportional to the size of the vector, which is often shown as Dice and Jaccard coefficient. The Dice coefficient is defined as formula (1):

$$sim(D_i, D_j) = \frac{2\sum_{k=1}^{m} d_{ik} d_{jk}}{\sum_{k=1}^{m} d_{ik}^2 + \sum_{k=1}^{m} d_{jk}^2} \tag{1}$$

Where $\sum_{k=1}^{m} d_{ik} d_{jk}$ refers to $\left|D_i \cap D_j\right|$, which refers to sum of number of the same vectors in two documents. $\sum_{k=1}^{m} d_{ik}^2$ refers to $\left|D_i\right|$,which refers to sum of number of parametric vectors in document D_i, and $\sum_{k=1}^{m} d_{ik}^2$ refers to $\left|D_j\right|$,which refers to sum of number of parametric vectors in document D_j.

2) The cosine method. The documents D_i and D_j are translated into vector, and the cosine formula is as shown in the formula (2):

$$\cos\theta = \frac{\left|D_i \cap D_j\right|}{\sqrt{\left|D_i\right|\left|D_j\right|}} \tag{2}$$

Therefore, the cosine of two documents represents similarity of two documents, that is, if two documents are same, then the angle between the vectors is 0^o and the cosine value is 1. And if two documents are totally different, then the angle between the vectors is 90^o and the cosine value is 1. So the range of values which is represented by the cosines is [0, 1].

3) The Euclidean Distance method. The Euclidean Distance method is also one of the most common methods, which uses the distance between two vectors to represent the similarity of two vectors. The computing formula is as shown in formula (3):

$$L_p(Di, Dj) = \left[\sum\left|d_{ik} - d_{jk}\right| p\right]^{\frac{1}{p}} \tag{3}$$

Where d_{ik} refers to weight of K item in document D_i, and d_{jk} refers to weight of K item in document D_j.

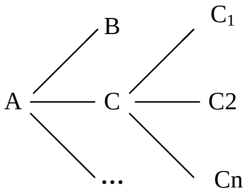
Though the distance method can acquire good result in simple data clustering, it is unfit to cluster for complex and especial data because of high timecomplexity, so it can not be used extensively in the practical application.

3. The Improved Similarity Algorithm

3.1 The 3-way of XML Document Similarity.

In the similarity algorithm based on content and structure only paid attention to contained content of the document fragments while ignored XML document content and its mark, and the document fragments are divided into isolated fragments, so it is easy to lead to failures. In views of this problem, this paper improves the document fragments.

In similarity computing, only considering information content of the entire descendant nodes of keyword, and translating them into space vectors. For example, only query the node C of document tree. All nodes are translated into vector space (c_1, c_2, c_3), where $c_n = (c_{nk1}, c_{nk2}, ..., c_{nkm})$ represents the nth vector of m keyword. The corresponding vector of nodes is shown in figure 2.



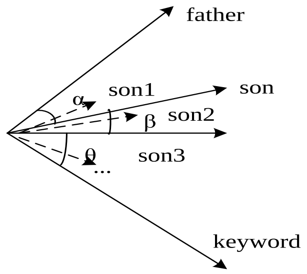


Figure.2 The corresponding vector of nodes

Figure.3 The 3-way of similarity

The mark vector of query node and mark vector of father node are composited, formed vector space (father, keyword, son), where the corresponding information of father nodes can be queried by query node. According to composition of three vectors, the similarity can be computed accurately. The composited vectors are shown in figure 3.

And thus, the similarity can be computed only with keyword content, besides, the similarity between the father node and its need to be composited.

$$sim = \cos\alpha \, sim(father) + \cos\beta sim(son) + \cos\theta \, sim(keyword) \tag{4}$$

Where α, β and θ represent referenced coefficients, their values is determined by vector importance.

3.2 The Weight Analysis of XML Document Similarity.

When comparing document similarity, the same numbers of two documents are generally considered. In real word application, the nonzero items play a crucial role in similarity computing when querying document A. But nonzero items of document B affect its accuracy of similarity computing. For this problem, this paper proposes directed similarity computing, that is, similarity computing is to analyze documents first, compare documents second.

The example of Set/Bag model shows the similarity computing. Let document B to be query object, document A and document C to be comparing documents. If document A and document B have same elements more than the document B and document C, then document A and document B should be clustered. But in practice elements cardinality of the document C are much less than elements cardinality of the document A, The Dice coefficient is formula (5).

$$sim(D_i, D_j) = \frac{2\sum_{k=1}^{m} d_{ik} d_{jk}}{\sum_{k=1}^{m} d_{ik}^2 + \sum_{k=1}^{m} d_{jk}^2} \tag{5}$$

And thus, the similarity of document A and document B is shown in formula (6):

$$sim(A,B) = \frac{2|A \cap B|}{|A| + |B|} = \frac{2m}{n_A + n_B} \tag{6}$$

Where m represents the number of same item in the document A and document B, n_A respresents the mumber of nonzero in the document A, n_B respresents the mumber of nonzero in the document B. In the same way, the similarity of document A and document B is shown in formula (7).

$$sim(B,C) = \frac{2|B \cap C|}{|B| + |C|} = \frac{2k}{n_B + n_C} \tag{7}$$

Where k represents the number of same item in the document Band document C, n_C respresents the mumber of nonzero in document C, n_B respresents mumber of nonzero in the document B. If you campare both the formulas, you can find $sim(A,b)$ is greater than $sim(B,c)$, but this result is in contrast to result of the graph. In the same way, the number of nonzero have greatly influenced in result. To avoid the effect of it, removing the denominator base of similarity, the improved formula is shown in the formula (8).

$$sim(D_i, D_j) = \frac{\sum_{k=1}^{m} d_{ik} d_{jk}}{\sum_{k=1}^{m} {d_{ik}}^2} = \frac{|D_i \cap D_j|}{|D_i|} = \frac{m}{n_1} \tag{8}$$

Where D_i represents the queried document, D_j represents the clustering document. Plugging the example into the formula, the similarity of document A and document B is shown in the formula (9).

$$sim(A, B) = \frac{|A \cap B|}{|B|} = \frac{m}{n_B} \tag{9}$$

The similarity of document A and document B is shown in the formula (9).

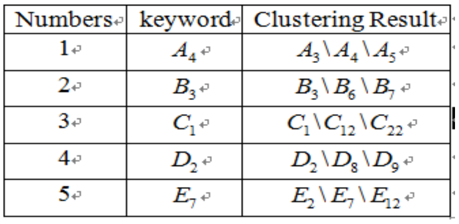
$$sim(B, C) = \frac{|B \cap C|}{|B|} = \frac{k}{n_B} \tag{10}$$

The denominator base of the two formulas are same, and m>k, so $sim(A, B) > sim(B, C)$,when document B is clustered, the document A is prior to document B.

4. The Experiment and Assessment

In order to prove that the improved algorithm is better than the traditional query algorithm, we implement query experiment with A_4, B_3, C_1, D_2 ,and E_7 (keywords). The clustering result is shown in the table 1. The as seen in Fig.5, the accuracy ratio of query with improved similarity algorithm is more than traditional query ways.

Table 1 The clustering result of different keywords



Numbers	keyword	Clustering Result
1	A_4	$A_3 \backslash A_4 \backslash A_5$
2	B_3	$B_3 \backslash B_6 \backslash B_7$
3	C_1	$C_1 \backslash C_{12} \backslash C_{22}$
4	D_2	$D_2 \backslash D_8 \backslash D_9$
5	E_7	$E_2 \backslash E_7 \backslash E_{12}$

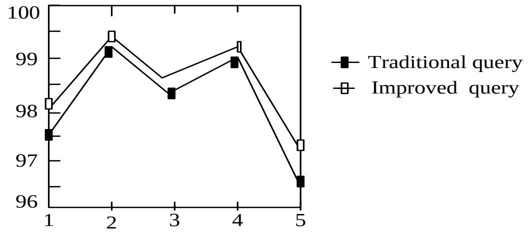


Figure.5 Experimental result

5. Summary

This paper has analyzed the advantage of extending clustering of queried keyword in XML document query, and explained similarity in accordance with document clustering. Firstly, summarizes some existing algorithms of similarity, and then proposes some improved ways in view of existence question. But this paper does not stipulate standardly similarity of clustering and proportion of $\cos\alpha$, $\cos\beta$ and $\cos\theta$, and more research is needed.

References

[1]. LI W. XML Document Clustering Research Based on Weighted Cosine Similarity .Journal of Jilin University (Information Science Edition), Vol. 1 (2010) No.1:68-76.

[2]. FLAVO RIZZOLO,ALEJANDRO A VAISMAN.Temporal XML:Modeling,Indexing, and Query Processing.The VLDB Journal, Vol. 2008(17):1179-1212.

[3]. GONG A, LIU H S. Component Clustering Analysis Based on XML Documents Similarity. Computer Engineering and Design, Vol. 30(2009) NO 10:507-510.

[4]. ZHANG B Q, BAI S. Research on XML Data Similarity. Computer Engineering, Vol. 31 (2010) NO 6:25-27.

[5]. LIU C. Research on XML Documents Clustering. Dalian: Dalian University of Technology, 2010: 125-128.