Estimation of Structural Similarity of XML Document Based on Frequency and Path

Ren Xueli
School of Computer Science and Engineer
Qujing Normal University
Qujing, China
oliveleave@126.com

Dai Yubiao
School of Computer Science and Engineer
Qujing Normal University
Qujing,China
abiaodai@163.com

***Abstract*—With the continuous development of Internet and rich resources emerging on the Web, information retrieval based on XML has emerged; the similarity of documents is the basis of information retrieval. A method is proposed to compute similarity of XML documents based on path and frequency in the paper. XML document is expressed as a collection of tuple, the paths are extracted and delete the recurring in order to improve efficiency, tag is matched by WordNet; and then path similarity is computed by the fuzzy longest common subsequence and frequency; finally, the structure similarity between documents are calculated. Two experiments are done to show that the method is effective, the experiment 1 test structural similarity of 15 XML documents from 3 DTDs; the similarity computing is applied in the documents classification for real data sets in the experiment 2, and results show the accuracy may arrive at 100%.**

***Keywords*—XML; structural similarity; frequency; sematic; tuple**

I. INTRODUCTION

In recent years, XML has become extremely popular in management and mining of semi-structured/hierarchical text data due to its abilities in representing information in a well-defined, extensible and machine readable format [1-3]. The use of XML covers data representation and storage, database information interchange, data filtering, as well as web services interaction. With the development of web and the wide usage of XML, there is an increasing need to automatically process Web documents for efficient data management, similarity clustering and search applications [4-6]. It has become a hot research topic currently how to realize the automatic retrieval, classification and clustering of XML documents effectively, so it is studied in the paper.

II. SIMILARITY MEASUREMENT OF XML DOCUMENTS

A wide range of algorithms for computing similarity in XML documents have been proposed in the literature. They are Tag matching, Edge matching, Path matching, Edit Distance and Fourier Transform [7-12].

A. Tag Matching

It is known as the simplest measure for XML similarity, as it only considers the intersection of the sets of tags over the union between the documents being compared.

B. Edge Matching

It considers the edges connecting XML nodes, the father-son relations in the comparison process.

C. Path Matching

The authors in [10] describe the structure of an XML document as a set of paths, compute similarity by taking into account all the paths in the path set of the second XML tree. If the more paths two XML documents share in common, then the more similar they are.

D. Edit Distance

Viewing XML documents as trees, Nierman and Jagadish [11] use the graph edit distance to compute the structural similarity between two XML documents. Given a set of graph edit operations, such as deletion, insertion, and substitution, the edit distance is defined as the shortest sequence of edit operations that transform one tree into the other. Typical tree distance algorithms include [11] and [12].

E. Fourier Transform

Flesca et al. represent XML documents as time series and compute the structural similarity between two documents by exploiting Discrete Fourier Transform of the corresponding signals.

III. OUR APPROACH

The structure of the documents is completely ignored in tag matching, thus attaining low clustering quality. Path matching considers not only father-son relation but also grandchild relationship, so it is more accurate than edge matching. But the result of similarity uses exact match, and not consider the semantic information of nodes.

As stated in the above, a new method to estimate similarity is proposed which determines similarity considering both the structure and sematic of XML documents. The structure similarity between the two XML documents is computed by path and frequency, and the sematic similarity is computed by WordNet[13]. The similarity between document A and document B has the following characteristics: (1) $\mathrm{Sim}(A, B) \in [0,1]$; (2) $\mathrm{Sim}(A, B) = Sim(\mathrm{B}, A)$; (3) $\mathrm{Sim}(A, \mathrm{A}) = 1$. The processes of similarity computing are shown in Fig.1.

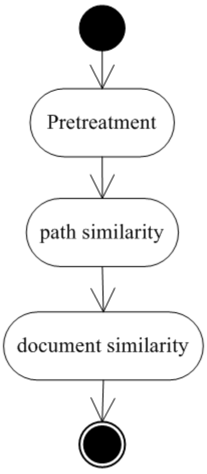


Figure 1. The process of similarity computing

The following definitions are given firstly for the convenience of description.

Definition 1 XML document tree: an XML document tree is an unordered labeled tree parsed from an XML document not including content, and the attribute looks as the sub node of the element. The top node is root, and the lowest node is leave, every level in XML document tree from root to leave is numbered using integer from 1.For example, Fig 2 is a tree , a is the root, c and d are leave.

Definition 2 tuple: the document tree is traversed by depth-first and describe as tuples with the form <path, frequency >. In that a path is element sequence from the root to leaf; frequency is 1 for every path. As an example, < a/b/c,1 > is a tuple in Fig 2.

Definition 3 the longest common subsequence (LCS): it is to find the longest subsequence common to all sequences in a set of sequences. Suppose that S is a common subsequence of two sequences P1 and P2, then s has 2 conditions: 1) $S \subset P1$ and $S \subset P2$; 2) S is the longest sequence which Satisfies the condition 1.

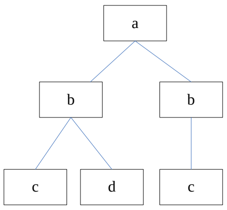


Figure 2 XML document tree

A. Pretreatment

XML document is composed of elements, attributes, XLink, comments and etc., Xlink is very important in data usage, but it is not the case in the impact of the document structure; in addition, comments are added in order to facilitate understanding of the document and information, they don't impact on document structure; therefore they are not considered in the calculation process of structural similarity.

B. Path similarity

The same paths in the XML document have effect on computing efficiency, so the problem of duplicate paths is solved firstly; in addition, as the XML document is compiled by different mechanisms. The same content is described by different tags; the fuzzy match is used by WordNet to solve the problem. In order to improve the efficiency and accuracy of the calculation, firstly, paths are extracted from tuple set; then tags similarity are computed by WordNet; finally, path similarity is computed by the dynamic programming . The process is shown in Fig.3.

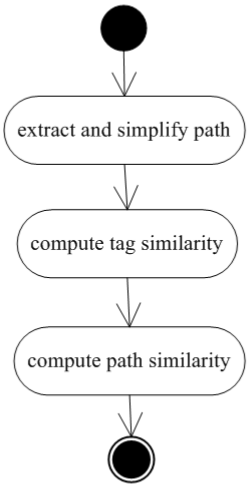


Figure 3. The process of path similarity

1) Extract and simplify path

The xml document is modeled as a tuple set where there are the same paths. To decrease complexity, the three steps are taken for the tuple set of XML document. Firstly, these paths are extracted from every tuple of tuple set form path set; Secondly, the tuple with the same paths are deleted, and the frequency value is cumulative with the same path; finally, all of the paths are extracted from the tuple set to compute structure similarity. Simplify tuple set is constructed using the following pseudo code.

```
Input：Tuple set TS $\mathrm{TS} = \{\langle p_1, f_1\rangle, \langle p_2, f_2\rangle \cdots \langle p_n, f_n\rangle\}$
Output：Simply tuple set
Extract path from every tuple called $P = \{p_1, p_2 \cdots p_n\}$
FOR I=2 TO P.LENGTH
FOR J=1 TO I-1
IF $p_i = p_j$ THEN
FOR K=J+1 TO P.LENGTH
DELETET TUPLE I
```

```
$f_j = f_i + f_j$
NEXT
P.LENGTH=P.LENGTH-1
END IF
NEXT J
NEXT I
```

2) *Tag similarity*

The XML documents are edited by different people and organizations, so these phenomenon will appear:

- Different words are used to describe the same thing. For example: car and automobile.
- Abbreviated forms and words combined are used in tags. For example student_id and stu_id.
- Different forms of writing are used in tags of XML documents. for example: Student and student.

Some pretreatments are required before computing path similarity. They are:

- All tags are converted in lowercase;
- The composite words are transformed into a collection of all words.as an example, student_id is converted to{student , id};
- The abbreviations are extension by WordNet, and then they are converted using the above rule.

The semantic similarity between tags are semantic similarity between words set, it is computed using the following 3 steps for tag N1 and N2.

- If N1 is described as the set $\{e_{11}\ ,\ \cdots\ ,e_{1m}\}$,and N2 is $\{e_{21}\ ,\ \cdots\ ,e_{2n}\}$,then The similarity between terms is computed by formula (2).

  $$Ssim(e1,e2)=\frac{2*level(LCS)}{level(e1)+level(e2)} \qquad (2)$$

  In that , $level(LCS)$ is the level of the lowest common father for $e1$ and $e2$ in WordNet , $level(e1)$ is the level of $e1$, $level(e2)$ is the level of $e2$.

- A matrix is constructed to compute set similarity where a term n in word set is as a row of matrix, a term m of word set is as a column of matrix, and the value in the matrix is similarity between words, the matrix is constructed with the form of formula (3)

  $$SSim(N_1,N_2)=\begin{bmatrix} sim(e_{11},e_{21}) & \cdots & sim(e_{11},e_{2n}) \\ \vdots & \ddots & \vdots \\ sim(e_{1m},e_{21}) & \cdots & sim(e_{1m},e_{2n}) \end{bmatrix} \qquad (3)$$

- If m>n then The node similarity is computed by the following formula (4); Otherwise the matrix is transposed firstly, then similarity is computed.

$$SSim(\mathrm{N}_1,\mathrm{N}_2)=\frac{\sum_{j=1}^{m}\max_{k=1}^{n}(sim(n_{1j},n_{2k}))}{m} \qquad (4)$$

3) *Path similarity*

The path similarity is relate to not only these nodes in the path and levels of nodes in XML documents, and the root node is the most influential to the whole similarity, and then decreased, so the path similarity is defined as the sum of product between the matching degree of all the nodes in the path and the weights of the hierarchy. In that the matching degree of the nodes is computed by WordNet, and the weights of the hierarchy is defined as $q_i=\frac{1}{2^i}$. The degree of node matching follows the rule:

If the similarity of semantic is larger than 0.7, then the degree of matching equals to the similarity of semantic; otherwise the value equals to 0.

The longest fuzzy common subsequence of two paths is determined based on the degree of tag matching; the path similarity is computed by the longest fuzzy common subsequence. As there are the common features between the longest fuzzy common subsequence and LCS, and dynamic programming is an effective method for solving LCS[14], so the longest fuzzy common subsequence may be solved by dynamic programming and compute similarity.

C. Document similarity

Document similarity of documents is the Optimal matching problem between paths, so it is similarity with the task allocation, Hungarian algorithm is one of the most effective algorithms to solve the linear assignment problem that can solve the problem in polynomial time [15]. The Hungarian algorithm is used to calculate the optimal path matching of two documents in this paper, and then the average value is calculated as the structural similarity. For the purposes of our similarity measure, it is desired that any such unmatched paths contribute also to the overall similarity between the two path sets. To this end, we add some virtual paths to the smaller set, so that both sets have the same size, and for each such virtual element, its similarity is defined as 0.5 to satisfy the definition of similarity. The original m ×n matrix is M where the number of different paths in document 1 is m, and the number of different paths in document 2 is n. The matrix is $\begin{bmatrix} sim(p_{11},p_{21}) & \cdots & sim(p_{11},p_{2n}) \\ \vdots & \ddots & \vdots \\ sim(p_{1m},p_{21}) & \cdots & sim(p_{1m},p_{2n}) \end{bmatrix}$. If m < n, the resulting matrix M′ will be m × m and have m − n additional rows filled with 0.5;If m > n, the resulting matrix M′ will be m ×m and have m − n additional columns filled with 0.5.The structural similarity is computed by formula (5).

$$sim(D1,D2)=\frac{\sum_{j=1}^{m}\max_{k=1}^{n}(sim(p_{1j},p_{2k}))\times(f_{1j}+f_{2k})}{2\times\max(\sum_{j=1}^{m}f_{1j},\sum_{k=1}^{n}f_{2k})} \qquad (5)$$

In that: f_{1j} is the path p_{1j} times in D1, f_{2k} is the frequency of the path p_{2k} in D2, $\sum_{j=1}^{m} f_{1j}$ is the sum of frequency for the whole path p_{1j} in D1, $\sum_{k=1}^{n} f_{2k}$ is the sum of frequency for the whole path f_{2k} in D2.

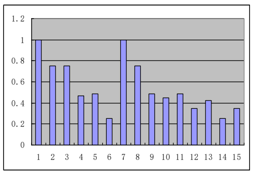
To illustrate, consider two documents A ={<a/b/c,2>,<a/b/d,1>} and B={<a/b/c,1>,<a/b/d,1>}. The path a/b/c in A with a/b/c in B and a/b/d in A with a/b/d in B yielding the maximal path similarity 1. The similarity between these two path sets are thus sim(A,C) = ((2+1)*1+(1+1)*1) /2*3 = 0.83.

IV. EXPERIMENT

Two experiments are done to show the accuracy of the method.

A. Experiment 1

Given the 3 DTD documents in [16], then 5 XML documents are generated for every DTD by XML Generator automatically, then a document is selected out randomly in the document set called Ⅰ,Ⅱ,Ⅲ, finally, the similarity of document Ⅰ, Ⅱ,Ⅲ and all documents are computed using the method in the paper. The result of similarity are displayed in Fig.4, Fig.5,Fig.6.



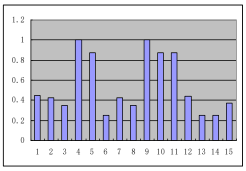


Figure 4. Similarity of Ⅰ and all documents Figure 5. Similarity of Ⅱ and all documents

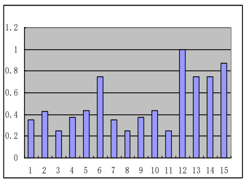


Figure 6. Similarity of Ⅲ and all documents

B. Experiment 2

The 3 DTDs are chosen from the data sets of ACM SIGMOD[17], they are Proceesings Page.DTD, IndexTermsPage.DTD and OrdinaryIssuePage.DTD. 17 , 40 and 40 XML documents are chosen respectively .80% documents are selected from each category DTD as training set, the rest are as a test set, similarity is calculated using the method in the paper, and then the KNN is used for document classification [18], the results from the experiment is shown in Table 4, and show that: the accuracy of classification arrives at 100%.

TABLE I. THE RESULT OF DOCUMENT CLASSIFICATION

DTD	Number of test document	number of correct classification	Number of false classification
Proceesings Page	3	3	0
IndexTermsPage	8	8	0
OrdinaryIssuePage	8	8	0

V. CONCLUSIONS

The similarity of XML documents is the basis of XML classification. A similarity method is proposed which compute similarity considering structure and semantic of XML document. The structure similarity of XML document is computed by the frequency path similarity, and the semantic similarity is computed by WordNet. Two experiments are done by the method in the paper and results show that it is effective.

REFERENCES

[1] Joe Tekli, Richard Chbeir, Kokou Yetongnon.An overview on XML similarity: background, current trends and future directions. www.sciencedirect.com，2015

[2] J. Rijsbergen Van. Introduction to Information Retrieval,In Proceedings of EARIA 06 , 2006

[3] A.M. Kade and C.A. Heuser. Matching XML Documents in Highly Dynamic Applications. In Proceeding of the 8th ACM Symposium on Document Engineering (DocEng'08), Brazil,2008:191-198.

[4] Bertino E., Guerrini G., Mesiti M., Rivara I. and Tavella C., Measuring the Structural Similarity among XML Documents and DTDs.http://www.disi.unige.it/person/MesitiM.,2015

[5] Bouchachia A, Hassler M. Classification of XML Documents[C]. IEEE Symposium on Computational Intelligence and Data Mining, 2007:390-396.

[6] J. Tekli, R. Chbeir and K. Yetongnon. Structural Similarity Evaluation between XML Documents and DTDs. In Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE'07), Springer-Verlag Berlin Heidelberg (LNCS 4831), Nancy, France,2007：196-201.

[7] Yao, J.T., Varde, A., Rundensteiner, E., Fahrenholz, S.: XML Based Markup Languages for Specific Domains. In: Web-based Support Systems. Advanced Information and Knowledge Processing, Springer，2010：215－238.

[8] Zhang Na,Zhang Dongzhan.An improved method for classifying XML documents based on structure and content[C].ISCSC10,2010.8:427-429.

[9] Andrew Nierman,H.V. Jagadish.Evaluating Structual Similarity in XML Document[EB/OL]. http://db.ucsd.edu/webdb2002/papers/44.pdf.2014.12

[10] K.C.Tai.Tree to tree editing problem[J].ACM,1979:422-423

[11] T. Schlieder and H. Meuss. Querying and Ranking XML Documents. Journal of the American Society for Information Science, Spec. Top. XML/IR 53(6),2002:489-503.

[12] Alsayed Algergawy,Marco Mesiti.XML data clustering:An overview[J].ACM,2011:14

[13] WordNet[EB/OL]. http://wordnet.princeton.edu/wordnet/download/current-version/#win,2014.1

[14] Thomas H.Cormen,Charles E.Leiserson,Ronald L.Rivest. Introduction to algorithm [M]. Machinery Industry Press,2006:208-2014

[15] Task allocation problem [EB/OL]. http://zh.wikipedia.org/wiki/%E4%BB%BB%E5%8A%A1%E5%88%86%E9%85%8D%E9%97%AE%E9%A2%98.2014.3

[16] Joe Tekli,Richard Chbeir,Kokou Yetongnon. A Hybrid Approach for XML Similarity[EB/OL]. http://www.researchgate.net,2014.1

[17] Sigmod XML data sets[EB/OL]. http://www.sigmod.org/publications/sigmod-record/1312, 2014.2