A Hybrid Method to Evaluate Similarity of XML Document

Yubiao Dai
School of Computer Science and Engineer
Qujing Normal University
Qujing,China
abiaodai@126.com

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

Abstract— XML is an important standard of information representation and data exchange over the Internet, document classification is an important way to get useful information from the mass of information solutions, a method of XML document classification is proposed based on fuzzy matching path in the paper. First, the information that has no influence on the classification is removed; Then a mixed method is used to compute XML document similarity, XML document is expressed as a collection of path, deleting the recurring and matching fuzzy path in order to improve efficiency, Hungarian algorithm to calculate the similarity between documents; Finally, 2 experiments are done and the results show that the method is effective.

Keywords-XML; Path; Semantic; Fuzzy Longest common subsequence; Hungarian

I. INTRODUCTION

With the development of Internet and resource enriched on the Web, all kinds of information retrieval services based on XML have emerged. In order to improve the efficiency and accuracy of information retrieval, an important means to solve this problem is that XML documents are classified as different categories, where the same category of documents is similar in intrinsic characteristics, and different category of documents are not similar in the inherent characteristics of the document. There are 2 methods to divide into XML documents in accordance with the characteristics; they are XML document classification based on structure and the XML document classification based on the structure and content. The method based on structure concern only document structure information, without regard to the content of the document. Garboni et al. proposed the classification using distance in [1]; Knijf use decision tree to classify in [2]; Zaki et al. proposed in [3,4] a rule-based classification model called XRules. The method based on the structure and content take into account not only the structure but also contents of the document. The classification methods are used commonly which are K- nearest neighbor [5], Bayesian classifier [6], SVM [7] and the bottom-up classification [8] and so on. Regardless of which type of document classification method is used, whose accuracy depends on document similarity measure, so the problem to compute similarity of XML document is studied.

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 so on. Tag matching 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,. Nonetheless, using tag matching, the structure of the documents is completely ignored, thus attaining low clustering quality. Edge matching match the edges connecting XML nodes, thus taking into account the father-son relations in the comparison process, it’s more accurate than tag similarity. But the result of similarity uses exact match, and not consider the semantic information of nodes. The authors in [10] describe the structure of an XML document as a set of paths, then compute similarity by taking into account all the paths in between the path set of the second XML tree. If the more paths two XML documents share in common, then the more similar they are. It’s more accurate than edge similarity, as it considers not only father-son relation but also Grandchild relationship. But the result of similarity uses exact match, and not consider the semantic information of nodes. Viewing XML documents as trees, Nierman and Jagadish [11] use the graph edit distance measure to compute the structural similarity between two XML documents. The algorithm for this distance measure was derived from one for the edit distance between strings. 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. In practice, a cost may be assigned to each individual operation to reflect its importance. Typical tree distance algorithms include [11] and [12]. Flesca et al. [13] 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

In general, XML document should follow a given syntax, DTD or XML Schema, which define the corresponding XML elements appearing in this document, related properties, and rules of each element or attribute should be followed. However, XML documents are often found on the web no corresponding syntax documents, especially XML documents created based on valid HTML, so it is necessary to analyze and determine whether the XML document has the same structure. In order to improve the efficiency and accuracy of the calculation, firstly, the documents are preprocessed, and the information which has no effect on the structure are deleted; then a hybrid similarity calculation method is used, that considers not only the structure of the document, but also the elements of semantics.

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. Document Similarity

The same nodes in the XML document have effect on efficiency. In this paper, so the problem of duplicate nodes is solved firstly; in addition, As the XML document is compiled by different mechanisms. The same content is described by the different elements, the fuzzy match is used to solve the problem. In this paper, the fuzzy path comparison method is used to computing the similarity of documents, whose process is shown in Fig. 1.

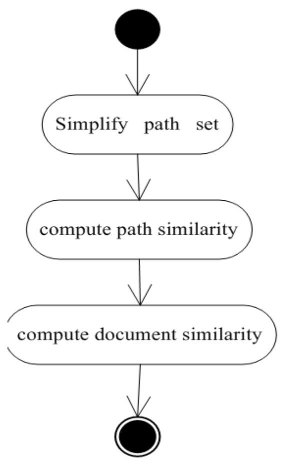

Figure 1. The steps of estimate similarity

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

Definition 1 XML document tree: an XML document tree is an ordered labeled tree parsed from an XML document where content isn't included in the tree. 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 , bookstore is the root, category,lang,author and price are leave.

Definition 2 Path: a path is a element sequence from the root element to leave element. As an example, bookstore/book/title/lang is a path in Fig .2.

Definition 3 Path set : PS is a set which includes all of the paths in XML document tree. which removes repeat same element from XML document tree. As an example, a set {bookstore/book/category, bookstore/book/title/lang, bookstore/book/author, bookstore/book/price, bookstore/book/category, bookstore/book/title/lang, bookstore/book/author, bookstore/book/price } is the path set of Fig .2.

Definition 4 Simplify Path Set: it is a set where the repeat path are deleted from the path set. As an example, a set {bookstore/book/category, bookstore/book/title/lang, bookstore/book/author, bookstore/book/price} is the path set of Fig .2.

Definition 5 The longest common subsequence: 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 condition: 1)S $\subset$ P1 and S $\subset$ P2 ; 2) S is the longest sequence which Satisfies the condition 1.for example :the subsequence bookstore/book is the longest common subsequence of two path bookstore/book/author and bookstore/book/price.

Definition 6 the fuzzy longest common subsequence: it is a longest subsequence having fuzzy match.

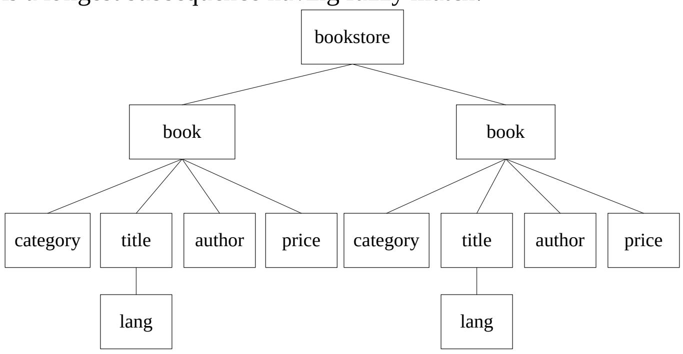

Figure 2. XML document tree

1) Simplify path set

Fundamental content of an xml document contains declaration, comment, tag, element, attribute and text. The xml document is modeled as an orderd labeled tree, tree node is the element or attribute in the document ,and leaf node is content of element and attributes. As +,* appear in DTD make repeat node which make similarity computing complexity. To improve accuracy and decrease complexity,the two steps are taken for the tree of XML document ,firstly,the attribute node becomes the sub node of the element node, as some information of xml document may be denoted using attribute or sub element, so all of the attribute information transform to sub element to compute simple. Secondly, paths are extract for the tree, the repeat path are deleted from the path collect to decrease complexity. The process is realized using the following pseudo code.

Input: path set $P = \{p_1, p_2 \cdots p_n\}$

output：simpliest path set P

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

$p_{k-1} = p_k$

Next
P.length=P.length-1
end if
Next j
Next i

2) Compute Fuzzy path similarity

A path is taken out separately from two simplest path set, the fuzzy longest common subsequence is determined based on semantic similarity measure. And the similarity of LCS is the sum of the product that the weight of node level and the degree of matching. The level is determined by definition 1 whose weight is defined $q_i = \frac{1}{2^i}$ in this paper, the degree of matching divide into three kinds of cases:

a) If they are same, then the degree of matching equals to 1;

b) If they are synonym in WordNet, then the degree of matching equals to 0.8;

c) If they don't include in (1) and (2), then the degree of matching equals to 0.

As there are common feature between the Fuzzy longest common sub sequence and the longest common sub sequence, and dynamic programming is an effective method to solve the longest common sub sequence, dynamic programming is used to solve the most common sub sequence in the paper and calculate the similarity.

3) Compute Document similarity

The task allocation is a maximum (or minimum) matching problem in the weighted two map, Hungarian algorithm is one of the most effective algorithms to solve the linear assignment problem that can solve the problem in polynomial time[14].After the path similarity of two documents is calculated, the similarity between documents is a problem to find the optimal matching between paths, therefore, 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 document 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, we let its similarity to each path in the opposite set be 0.5. Thus, if the original m × n matrix is M, and if m > n, the resulting matrix M′ will be m × m and have the m − n additional rows filled with 0.5. The similarity between the two path sets is then defined as Hungarian(M′) / m. The process is realized using the following pseudo code.

Input : path sets of tree T_1 and T_2

Output : tree similarity between T_1 and T_2

For all $p_i \in T_1, 1 \le i \le |T_1|$

For all $p_j \in T_2, 1 \le j \le |T_2|$

$s_{ij} = sim(p_i, p_j)$

End for

End for

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1|T_2|} \\ s_{21} & s_{22} & \cdots & s_{2|T_2|} \\ \cdots & \cdots & \cdots & \cdots \\ s_{|T_1|1} & s_{|T_1|2} & \cdots & s_{|T_1||T_2|} \end{bmatrix}$$

If $|T_1| > |T_2|$ then

$$S^{'} = \begin{bmatrix} 1-s_{11} & 1-s_{12} & ... & 1-s_{1|T_2|} & 0.5 & ... & 0.5_{1|T_1|} \\ 1-s_{21} & 1-s_{22} & ... & 1-s_{2|T_2|} & 0.5 & ... & 0.5_{2|T_1|} \\ ... & ... & ... & ... & ... & ... & ... \\ 1-s_{|T_1|1} & 1-s_{|T_1|2} & ... & 1-s_{|T_1||T_2|} & 0.5 & ... & 0.5_{|T_1||T_1|} \end{bmatrix}$$

$$S_{\min} = Hungarian(S^{'})$$

$$S_{\max} = 1 - \frac{Hungarian(S^{'})}{|T_1|} Hungarian(S^{'})$$

Else

$$S^{'} = \begin{bmatrix} 1-s_{11} & 1-s_{12} & ... & 1-s_{1|T_2|} \\ 1-s_{21} & 1-s_{22} & ... & 1-s_{2|T_2|} \\ ... & ... & ... & ... \\ 1-s_{|T_1|1} & 1-s_{|T_1|2} & ... & 1-s_{|T_1||T_2|} \\ 0.5 & 0.5 & ... & 0.5 \\ ... & ... & ... & ... \\ 0.5_{|T_2|1} & 0.5_{|T_2|2} & ... & 0.5_{|T_2||T_2|} \end{bmatrix}$$

$$S_{\min} = Hungarian(S^{'})$$

$$S_{\max} = 1 - \frac{Hungarian(S^{'})}{|T_2|}$$

End if

Return $S_{\max}$

IV. Experiment

To test the performance of the approach, two experiments are done.

A. *Experiment 1*

Given the DTD documents, then the sets including in 15 XML documents are generated by XML Generator automatically, the similarity of the document 1 and others are computed using Tag matching, Edge matching, Path matching, Edit Distance and the method in the paper. In the following, the DTD documents used in the experiment is [16]. The result of Similarity is displayed in Fig .3.

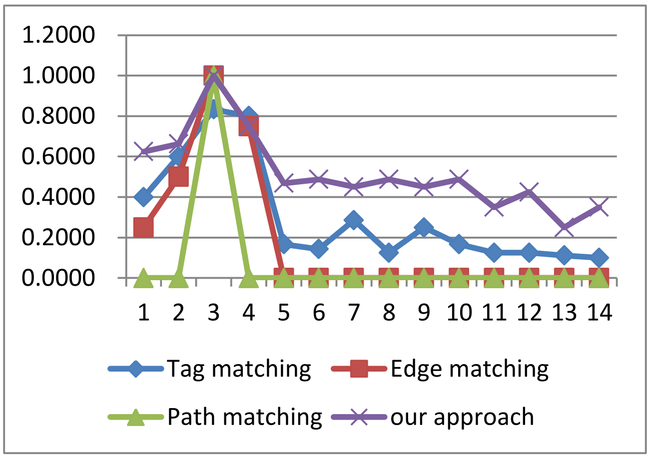

Figure 3. The result of similarity

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, 50 and 50 XML documents are chosen respectively .

80% documents are selected from each category DTD as already classified documents, 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 1, and show that: the accuracy of classification arrives at 100%.

TABLE I. RESULT OF EXPERIMENT

DTD	Number of test document	number of correct classification	Number of false classification
Proceesings Page	2	2	0
IndexTermsPage	10	10	0
OrdinaryIssuePage	10	10	0

V. SUMMARY

XML document classification is the basis of the information retrieval; similarity is the key problem in document classification. A similarity method is proposed which compute similarity considering structure and semantic of XML document. The XML document is represented as path set, and deletes duplicate paths in order to resolve repeat matching impact on efficiency; then do fuzzy path matching; finally, the similarity of the document is computed using the Hungarian algorithm. Two experiments are done and results show that the method is effective.

REFERENCES

[1] Garboni C, Masseglia F, Trousse B. Sequential Pattern Mining for Structure-Based XML Document Classification[C]. The 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, Springer , 2006:458-468.

[2] Knijf J D. FAT-CAT: Frequent Attributes Tree Based Classification[C]. The 5th International Workshop of the Initiative for the Evaluation of XML Retrieval , Springer, 2007:485-496.

[3] Zaki M, Aggarwal C. XRules: An effective algorithm for structural classification of XML data[J]. Machine Learning, 2006, 62(1):137-170.

[4] Zaki M J, Aggarwal C C. XRules: an effective structural classifier for XML data[C]. The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003: 316-325.

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

[6] Yi J, Sundaresan N. A classifier for semi-structured documents[C]. The 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2000: 340-344.

[7] Ghosh S, Mitra P. Combining Content and Structure Similarity for XML Document Classification using Composite SVM Kernels[C]. The 19th International Conference on Pattern Recognition, Tampa, FL, 2008: 1-4.

[8] Wu J, Tang J. A bottom-up approach for XML documents classification[C]. The 2008 International Symposium on Database engineering and applications, ACM, 2008: 131-137.

[9] Andrew Nierman,H.V. Jagadish.Evaluating Structual Similarity in XML Document[EB/OL]. http://db.ucsd.edu/webdb2002/papers/44.pdf.2013.12

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

[11] Nierman, A., Jagadish, H.V.: Evaluating Structural Similarity in XML Documents. In: Mary, F., Fernandez, Y.P. (eds.) WebDB 2002, Madison, Wisconsin, USA, pp. 61–66 (2002)

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

[13] S. Flesca,F. Furfaro,S. Greco. Querying and Repairing Inconsistent XML Data[C]. Web Information Systems Engineering – WISE 2005, 175-188

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

[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] Chawathe S.,Comparing Hierarchical data in external memory.proceedings of the 20th international conference on very large dada base,98~100,1999

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

[18] k-nearest neighbors algorithm[EB/OL]. http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm