A Path-sequence Based Method for Solving the One-to-multiple Matching Problem in Leaf-Clustering Based Approximate XML Join Algorithms

Wenxin LIANG[†] and Haruo YOKOTA[††,†]

[†] Department of Computer Science, Graduate School of Information Science and Engineering,
Tokyo Institute of Technology
2–12–1 Ookayama, Meguro-ku, Tokyo 152–8552, Japan
[††] Global Scientific Information and Computing Center, Tokyo Institute of Technology
2–12–1 Ookayama, Meguro-ku, Tokyo 152–8550, Japan

Abstract In previous work, we have proposed approximate XML join algorithms based on the clustered leaf nodes for measuring the approximate similarity between XML documents and integrating them at the subtree classes. However, in a join loop, one base subtree may infrequently happen to be matched with multiple target ones. In this paper, we propose a path-sequence based method to solve the one-to-multiple matching problem in leaf-clustering based approximate XML join algorithms. In our method, each path sequence from the top node to the matched leaf in the base and target subtree is extracted, and the most similar target subtree for the base one is determined by the path-sequence based subtree similarity degree. We conduct experiments to evaluate our method by using both real bibliography and bioinformatics XML documents. The experimental results show that our method can effectively solve the one-to-multiple matching problem for both bibliography and bioinformatics XML data, and hence improve the precision of the leaf-clustering based approximate XML join algorithms.

Key words XML, Semi-structured data, Information Integration

1. Introduction

XML has become a widely important standard for data representation and exchange on the World Wide Web. Recently, a large number of data, for example bioinformatics data such as TrEMBL [18] and Swiss-Prot [16], and bibliography data such as ACM SIGMOD Record [1] and DBLP [23], are published and shared by XML on the Internet. However, XML documents from different data sources may convey nearly or exactly the same information but may be different on structures. Besides, even the two XML documents represent the same information, each of them may have some extra information what the other does not do.

A well formed XML document can be parsed into an ordered labeled tree [20]. The Document Type Descriptor (DTD) is regarded as an effective tool for detecting the structural information from XML documents [2, 6, 10, 14]. However, even if XML documents have the same DTDs, they may not be constructed by identical tree structures because of the repeating and optional elements and attributes [8, 9, 14]. Figure 1 shows an example of two XML document trees with different DTDs. Although the two document trees are structurally different, they represent very similar information. In addition, both of the two document trees have some information what the other does not do. For example, `volume` in Figure 1 (a); and `initPage` in Figure 1 (b).

In previous work [11, 12], we have proposed leaf-clustering based approximate XML join algorithms for measuring the approximate similarity between XML documents and integrating them at subtree classes. However, in a join loop, a one-to-multiple matching problem might occasionally occur for a base subtree; that is one base subtree may happen to be matched with multiple target ones. Although the one-to-multiple matching problem occurs very infrequently, it still affects the precision of the leaf-clustering based approximate XML join algorithm. Therefore, how to select the most proper matched target subtree for the base one in the case of one-to-multiple matching becomes a critical issue.

In this paper, we propose a path-sequence based method to solve the one-to-multiple matching problem. In the proposed method, each path sequence from the top node to the matched leaf in the base and target subtree is extracted, and the most similar target subtree for the base one is determined by the path-sequence based subtree similarity degree. We

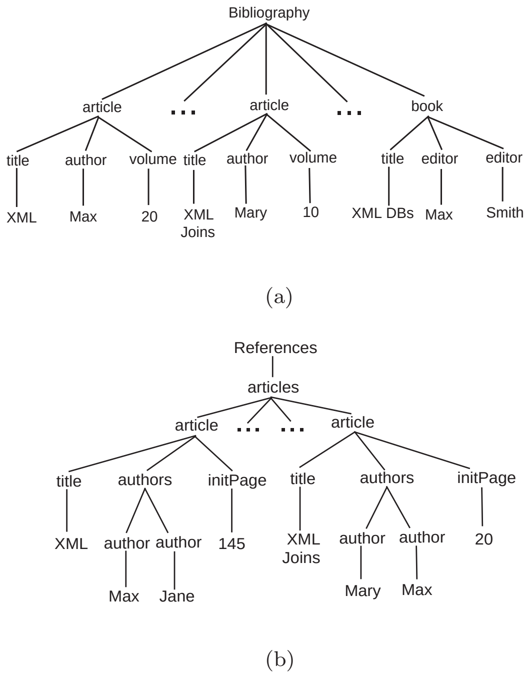


Figure 1 Example XML document trees

conduct experiments using both real bibliography and bioinformatics XML documents to compare the occurrence rate of one-to-multiple matching and the precision of matching for the proposed algorithm comparing with those for the original one. The experimental results show that the path-sequence based method can effectively decrease the occurrence rate of one-to-multiple matching for both bibliography and bioinformatics XML data, and hence improve the precision of the leaf-clustering based approximate XML join algorithms.

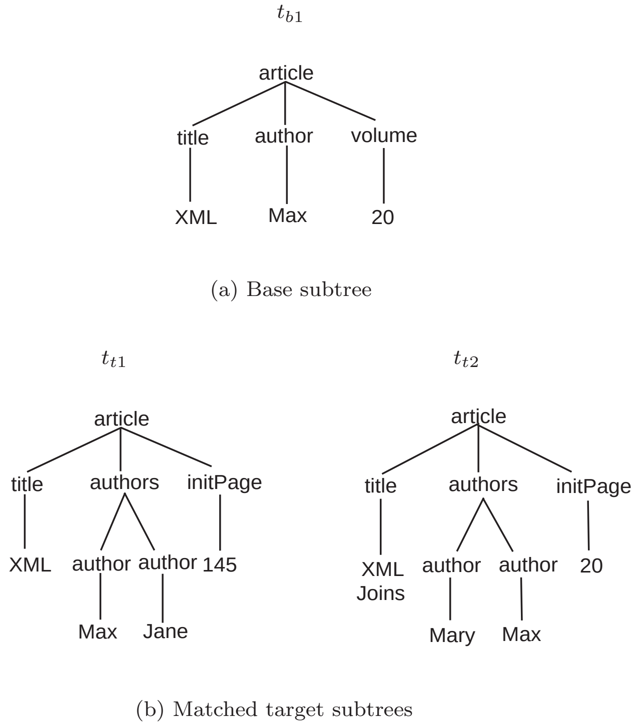
The remainder of the paper is organized as follows. Section 2. states the problem of one-to-multiple matching. In Section 3., we propose the path-sequence based method. Section 4. conducts experiments using both real bibliography and bioinformatics data and discusses the experimental results. In Section 5., we briefly introduce the related work and our previous proposed algorithms. Finally, Section 6. concludes this paper.

2. One-to-multiple Matching Problem

In the leaf-clustering based approximate XML join algorithms [11, 12], the two XML documents to be joined are segmented into subtrees representing independent information units(注1). Then, the subtree matching is determined by the *subtree similarity degree* which is defined as follows.

〔Definition 1〕 **(Subtree Similarity Degree (SSD))** For a base subtree t_{bi} and a target one t_{tj}, the subtree similarity degree between them, $SSD(t_{bi}, t_{tj})$ is defined by Equation (1) as the percentage of the number of matched leaf nodes (the pair of leaf nodes that has the same PCDATA value) out of the number of leaf nodes in the base subtree t_{bi}, where n and n_{bi} denote the number of matched leaf nodes and the number of leaf nodes in the base subtree t_{bi}.

$$SSD(t_{bi}, t_{tj}) = \frac{n}{n_{bi}} \times 100\ (\%) \qquad (1)$$



(a) Base subtree

(b) Matched target subtrees

Figure 2 One-to-multiple matching problem

However, for a base subtree, a one-to-multiple matching problem may infrequently occur as stated as follows.

〔Problem 1〕 **(One-to-multiple Matching)** For a base subtree t_{bi}, if there are two or more target subtrees t_{tj} having the same subtree similarity degree with t_{bi}, the base subtree t_{bi} will be matched with those target subtrees in one join loop.

〔Example 1〕 Figure 2 shows a base subtree t_{b1} and two target ones t_{t1} and t_{t2} segmented from the document trees in Figure 1. The matched leaves for (t_{b1}, t_{t1}) are `XML` and `Max`, while those for (t_{b1}, t_{t2}) are `Max` and `20`. However, according to Definition 1, $SSD(t_{b1}, t_{t1}) = SSD(t_{b1}, t_{t2}) = 66.7\%$. That means both t_{t1} and t_{t2} will be matched with t_{b1} because of the same subtree similarity degree.

Therefore, how to select the most similar one from the multiple matched target subtrees becomes a critical issue.

3. Path-sequence Based Method

In Section 2., we have discussed the one-to-multiple matching problem in the leaf-clustering based approximate XML join algorithms. In this section, we propose a path-sequence

(注1)：The details of the segmentation algorithm are available in [11].

based method to solve this one-to-multiple matching problem.

3. 1 Key Definition

Let T_b and T_t be two XML document trees (b denotes *base*, and t denotes *target*). Assume T_b and T_t are segmented into k_b and k_t subtrees $t_{bi}(1 \leqq i \leqq k_b)$ and $t_{tj}(1 \leqq j \leqq k_t)$, respectively. Before we treat of the path-sequence based method, we present the following key definitions.

［Definition 2］ **(Matched Leaf)** For each pair of base subtree t_{bi} and target one t_{tj}, the matched leaf $L_M(i)$ is the pair of leaf nodes l_{bi} and l_{tj} that has the same PCDATA value.

［Definition 3］ **(Matched Subtree)** In the ith join loop, the matched subtree $T_M(i)$ is the pair of subtrees t_{bi} and t_{tj} that has the maximum subtree similarity degree.

［Definition 4］ **(Path Sequence)** For a pair of matched subtrees $T_M(i)$, a path sequence $P(i)$ is defined as the path from the root node to the matched leaf $L_M(i)$ in the base or target subtree.

For the matched subtrees (t_{b1}, t_{t1}) and (t_{b1}, t_{t2}) in Figure 2, the path sequences for them are shown by the dashed lines in Figure 3. The similarity between the path sequences for each pair of matched leaves is determined based on the *path-sequence similarity degree.*

［Definition 5］ **(Path-sequence Similarity Degree (PSD))** For a pair of matched leaves $L_M(i)$, the path-sequence similarity degree $PSD(i)$ is defined by the following equation, where N denotes the number of nodes in the base path sequence that have the same labels (non-leaf nodes) or values (leaf nodes) with those in the target path sequence, and N_{bi} denotes the total number of nodes in the base path sequence.

$$PSD(i) = \frac{N}{N_{bi}} \times 100\ (\%) \tag{2}$$

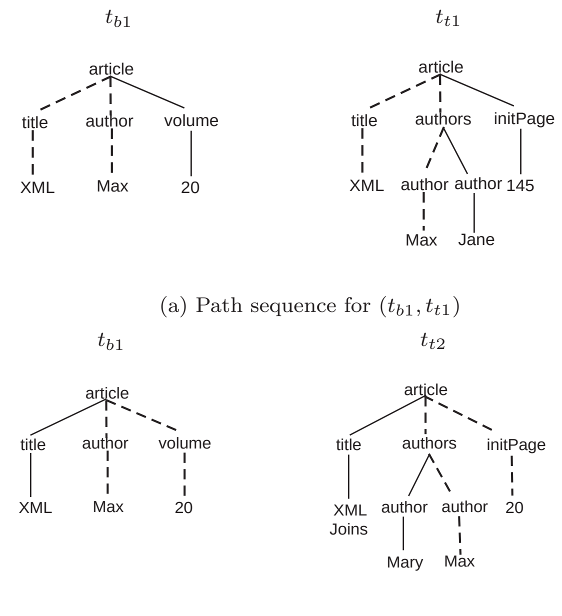
Then, the similarity between the matched subtrees can be determined by calculating the *path-sequence based subtree similarity degree.*

［Definition 6］ **(Path-sequence based Subtree Similarity Degree (PSSD))** For a pair of matched subtrees $T_M(i)$, assume the number of matched leaves is K, the path-sequence based subtree similarity degree $PSSD(t_{bi}, t_{tj})$ is determined by the following equation.

$$PSSD(t_{bi}, t_{tj}) = \frac{\sum_{i=1}^{K} PSD(i)}{K} \times SSD(t_{bi}, t_{tj}) \tag{3}$$

3. 2 Algorithm *PathSeq*

［Example 2］ For the base subtree t_{b1} and target ones t_{t1} and t_{t2} in Figure 2. The matched subtree pairs are $T_M(1) = (t_{b1}, t_{t1})$, and $T_M(2) = (t_{b1}, t_{t2})$. For the first matched subtree $T_M(1)$, the matched leaves are $L_M(1) =$ "XML" and $L_M(2) =$ "Max". Then, the base path sequence for $L_M(1)$, $P_b(1) = \{$"*article*", "*title*", "XML"$\}$, and the target path sequence path for $L_M(1)$, $P_t(1) = \{$"*article*", "*title*", "XML"$\}$ as shown by the dashed lines in Figure 3 (a). According to the Equation (2), the path-sequence similarity degree between $P_b(1)$ and $P_t(1)$, $PSD(1) = \frac{3}{3} \times 100\% = 100\%$. Similarly, the path-sequence similarity degree between $P_b(2) = \{$"*article*", "*author*", "Max"$\}$ and $P_t(2) = \{$"*article*", "*authors*", "*author*", "Max"$\}$, $PSD(2)$ is also 100%. Therefore, the path-sequence based subtree similarity degree for t_{b1} and t_{t1}, $PSSD(t_{b1}, t_{t1}) = \frac{100\%+100\%}{2} \times 66.7\% = 66.7\%$. For the second matched subtree $T_M(2)$, Figure 3 (b) shows the path sequences for t_{b1} and t_{t2}. We can similarly figure out the path-sequence based subtree similarity degree for t_{b1} and t_{t2}, $PSSD(t_{b1}, t_{t2}) = \frac{100\%+66.7\%}{2} \times 66.7\% = 55.6\%$. Because $PSSD(t_{b1}, t_{t1}) = 66.7\% > PSSD(t_{b1}, t_{t2}) = 55.6\%$, the matched target subtree t_{t1} is considered to be more similar with the base subtree t_{b1}.



(a) Path sequence for (t_{b1}, t_{t1})

(b) Path sequence for (t_{b1}, t_{t2})

Figure 3 Path sequence for the base and target subtree

When one base subtree is matched with multiple target subtrees, the most proper matched subtree pair can be determined based on the path-sequence similarity degree. The details of the path-sequence based algorithm is illustrated by Algorithm *PathSeq* shown in Figure 4.

4. Experiment

In this section, we conduct experiments using both bibliography and bioinformatics data to observe how often the one-to-multiple matching occurs in the original algorithms and how effectively the proposed method solves this problem.

4. 1 Experimental Environment

Our experiments have been done under the environment shown in Table 1.

```
Algorithm $PathSeq$ {
Input: pairs of matched subtrees $(t_{bi}, t_{tj})$
Output: pair of matched subtrees
  $max = 0$;
  $output = null$;
  for each pair of matched subtrees $(t_{bi}, t_{tj})$ {
    for each pair of matched leaves $L_M(i)$ {
      calculate $PSD(i)$;
    }
    calculate $PSSD(t_{bi}, t_{tj})$;
    if $(PSSD(t_{bi}, t_{tj}) \geqq Max)$ {
      $max = PSSD(t_{bi}, t_{tj})$;
      $output = (t_{bi}, t_{tj})$;
    }
  }
  return $output$;
}
```

Figure 4 Path-sequence based Algorithm

Table 1 Experimental Environment

CPU	Intel Pentium 4 (2.80GHz)
Memory	1.0 GB
OS	MS Windows XP Professional
Programming Environment	Sun JDK 1.4.2

4. 2 Data Used

For bibliography data, we use six fragment documents of DBLP.xml [23], named dblp1-6.xml, 600KB per document (about 30,000 nodes), as the base documents, and the XML version of SIGMOD Record [1], named sigmod.xml, 482KB (about 20,000 nodes), as the target one. And for bioinformatics data, we use six fragment documents of uniprot_sprot.xml [21], named sprot1-6.xml, 3MB per document (about 30,000 nodes), as the base documents, and a fragment document of uniprot_trembl.xml [22], named trembl.xml, 1MB (about 25,000 nodes), as the target one. The number of segmented subtrees in each document is shown in Table 2.

4. 3 Result and Discussion

We join the six pairs of bibliography and bioinformatics XML documents and observe the occurrence rate of one-to-multiple matching in the original algorithms and the proposed algorithm, respectively. The *occurrence rate of one-to-multiple matching* is defined as follows.

［Definition 7］ **(Occurrence Rate of One-to-multiple matching)** The occurrence rate of one-to-multiple matching ($\mathcal{R}$) is the percentage of the number of multiple matched subtrees ($\mathcal{N}$) out of the number of total subtrees in the base document (N_b) as the following equation.

$$\mathcal{R} = \frac{\mathcal{N}}{N_b} \times 100\ (\%) \tag{4}$$

In order to observe how effectively the path-sequence improves the precision of subtree matching in the situation of one-to-multiple matching, we define *precision of matching* as follows.

［Definition 8］ **(Precision of Matching)** The precision of matching ($\mathcal{P}$) is the percentage of the number of correctly selected subtrees (N_c) out of the number of total multiple matched subtrees (N_m) using the original algorithm as the following equation.

$$\mathcal{P} = \frac{N_c}{N_m} \times 100\ (\%) \tag{5}$$

Table 3 and Table 5 show the occurrence rate of one-to-multiple matching for the six pairs of bibliography and bioinformatics documents, where $\mathcal{N}_O$ and $\mathcal{N}_P$ denote the number of multiple matched subtrees in the base document by the original algorithms and by the path-sequence based algorithm, respectively, and $\mathcal{R}_O$ and $\mathcal{R}_P$ indicate the occurrence rate of one-to-multiple matching in the original algorithms and in the path-sequence based algorithm, respectively. Table 4 and Table 6 show the precision of matching for the six pairs of bibliography and bioinformatics documents, where N_{cO} and N_{cP} denote the number of correctly matched subtrees using the original algorithms and the path-sequence based algorithm, respectively, and $\mathcal{P}_O$ and $\mathcal{P}_P$ indicate the precision of matching for the original algorithms and the path-sequence based algorithm, respectively.

Figure 5 and Figure 7 indicate the differences of the occurrence rates of one-to-multiple matching in the original algorithms and the path-sequence based algorithm for the six pairs of bibliography and bioinformatics documents, respectively. Figure 6 and Figure 8 show the differences of the precision of matching for the original algorithms and the path-sequence based algorithm for the six pairs of bibliography and bioinformatics documents, respectively.

［Example 3］ Figure 9 shows the real source codes of a base subtree t_{b1} in dblp1.xml and two matched target subtrees t_{t1} and t_{t2} for it in sigmod.xml. According to Equation (1), the subtree similarity degrees for the both matched subtrees, $SSD(t_{b1}, t_{t1}) = SSD(t_{b1}, t_{t2}) = \frac{2}{10} \times 100\% = 20\%$. For the matched subtrees (t_{b1}, t_{t1}), the path sequences in t_{b1} are $P_b(1)$ ={ “article”, “author”, “C. J. Date”} and $P_b(2)$ = {“*article*”, “*number*”, “1”}, and the corresponding path sequences in t_{t1} are $P_t(1)$ ={“article”, “authors”, “author”, “C. J. Date”} and $P_t(2)$ = {“*article*”, “*initPage*”, “1”}. Therefore, the path-sequence similarity degrees for $(P_b(1), P_t(1))$ and $(P_b(2), P_t(2))$, $PSD(1) = \frac{3}{3} \times 100\% = 100\%$, and $PSD(2) = \frac{2}{3} \times 100\% = 66.7\%$. According to Equation (3), the path-sequence based subtree similarity degrees for t_{b1} and t_{t1}, $PSSD(t_{b1}, t_{t1}) = \frac{100\%+66.7\%}{2} \times 20\% = 16.7\%$. Similarly, the path-sequence based subtree similarity degree for t_{b1} and t_{t2}, $PSSD(t_{b1}, t_{t2}) = \frac{100\%+100\%}{2} \times 20\% = 20\%$. Because

Table 2 Number of subtrees in each document

	sigmod	dblp1	dblp2	dblp3	dblp4	dblp5	dblp6
N	1504	1599	1451	1538	1584	1680	1474
	trembl	sprot1	sprot2	sprot3	sprot4	sprot5	sprot6
N	396	335	337	324	309	350	360

$PSSD(t_{b1}, t_{t1}) = 16.7\% < PSSD(t_{b1}, t_{t2}) = 20\%$, the path-sequence based method will select t_{t2} as the final matched target subtree which is exactly the same article as the base subtree t_{b1} as shown in Figure 9.

According to the experimental results, we can draw the following conclusions:

- For both bibliography and bioinformatics data, the mean occurrence rates of one-to-multiple matching are less than 2% in the original algorithms. Namely, the one-to-multiple matching in the original algorithms does not frequently occur. However, even for the infrequent occurrence of one-to-multiple matching, it can still reduce the overall precision of the leaf-clustering based approximate XML join algorithms.
- The mean occurrence rate of one-to-multiple matching in the original algorithm for bioinformatics data is 1.94%. It is larger than that for bibliography data, 0.79%. That means the one-to-multiple matching occurs more frequently for bioinformatics data because each subtree of the bioinformatics document contains much more information than that of bibliography document does.
- For bibliography documents, the mean occurrence rate of one-to-multiple matching decreases from 0.79% to 0.12% by using the path-sequence based method. And for bioinformatics documents, the mean occurrence rate of one-to-multiple matching reduces from 1.94% to 0.29%. Thus, the mean occurrence rate of one-to-multiple matching using the path-sequence based method is less than one sixth of that using the original algorithms for both bibliography and bioinformatics documents.
- The mean precision of matching increases from 36.00% to 91.00% by using the path-sequence based method for bibliography documents, and it increases from 29.83% to 87.29% for bioinformatics documents. Therefore, the mean precision of matching for the path-sequence based method becomes about three times larger than that for the original algorithms.

5. Related and Previous Work

A well formed XML document can be parsed into an ordered labeled tree [20]. The tree structure is constructed by the nesting of its elements, and the node labels record the contents of the elements by element tags, attribute names, attribute values and PCDATA values.

Table 3 Occurrence rate of one-to-multiple matching for bibliography data

	$\mathcal{N}_O$	$\mathcal{N}_P$	$\mathcal{R}_O$	$\mathcal{R}_P$
dblp1 × sigmod	15	3	0.94%	0.19%
dblp2 × sigmod	9	0	0.62%	0.00%
dblp3 × sigmod	8	0	0.52%	0.00%
dblp4 × sigmod	16	4	1.01%	0.25%
dblp5 × sigmod	28	5	1.67%	0.30%
dblp6 × sigmod	0	0	0.00%	0.00%
Mean value	12.67	2.00	0.79%	0.12%

Table 4 Precision of matching for bibliography data

	N_{cO}	N_{cP}	$\mathcal{P}_O$	$\mathcal{P}_P$
dblp1 × sigmod	6	13	40%	86.70%
dblp2 × sigmod	4	9	44.40%	100%
dblp3 × sigmod	3	8	37.5%	100%
dblp4 × sigmod	5	13	31.25%	81.25%
dblp5 × sigmod	7	25	25.00%	89.30%
dblp6 × sigmod	-	-	-	-
Mean value	5.0	13.6	36.00%	91.00%

Table 5 Occurrence rate of one-to-multiple matching for bioinformatics data

	$\mathcal{N}_O$	$\mathcal{N}_P$	$\mathcal{R}_O$	$\mathcal{R}_P$
sprot1 × trembl	10	1	2.99%	0.30%
sprot2 × trembl	7	2	2.08%	0.59%
sprot3 × trembl	5	0	1.54%	0.00%
sprot4 × trembl	0	0	0.00%	0.00%
sprot5 × trembl	6	0	1.71%	0.00%
sprot6 × trembl	12	3	3.33%	0.83%
Mean value	6.67	1.00	1.94%	0.29%

Table 6 Precision of matching for bioinformatics data

	N_{cO}	N_{cP}	$\mathcal{P}_O$	$\mathcal{P}_P$
sprot1 × trembl	4	9	40.00%	90.00%
sprot2 × trembl	2	5	28.57%	71.43%
sprot3 × trembl	1	5	20.00%	100%
sprot4 × trembl	-	-	-	-
sprot5 × trembl	2	6	33.33%	100%
sprot6 × trembl	3	9	25.00%	75.00%
Mean value	2.4	6.8	29.38%	87.29%

There is considerable previous work on measuring the edit distance between ordered labeled trees [3–5, 7, 13, 15, 17, 19, 24–26]. The edit distance between two ordered labeled trees is defined as the minimum cost edit operations (insertions, deletions and changes) required to transform one tree to another [26]. The tree edit distance is recognized as an traditional metric for measuring the structural similarity between XML documents [7, 8, 14]. However, the computational cost of the tree edit distance is extremely expensive; in the worst case, it is an $O(n^4)$ operation for the XML documents of size n.

In order to avoid the expensive tree edit distance operation

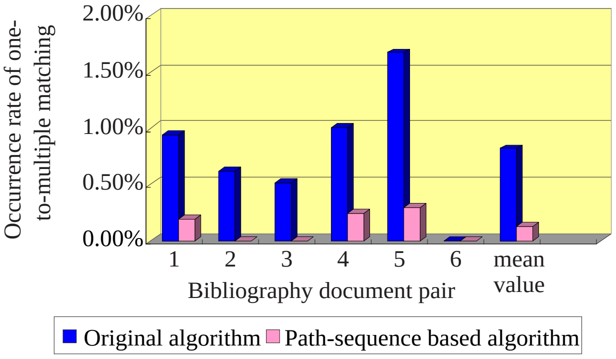


Figure 5 Occurrence rate of one-to-multiple matching for bibliography documents

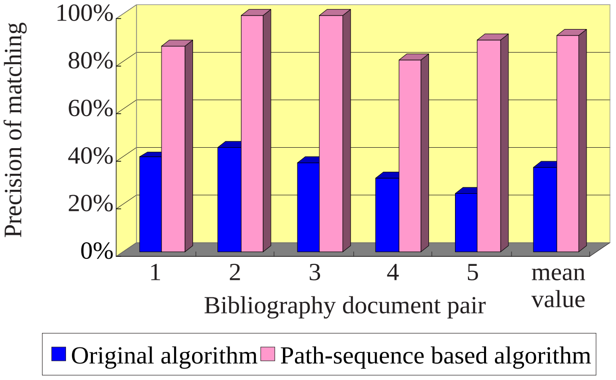


Figure 6 Precision of matching for bibliography documents

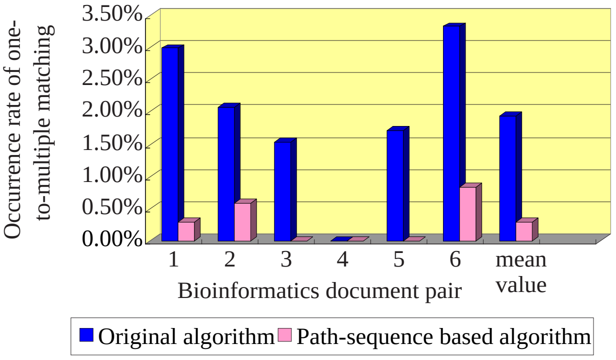


Figure 7 Occurrence rate of one-to-multiple matching for bioinformatics documents

as much as possible, S. Guha, et al. proposed the lower and upper bound as inexpensive filters for the tree edit distance operation [8]. However, when the upper bound is greater than the threshold distance τ and at the same time the lower bound is less than τ , the tree edit distance still can not be avoided.

In [11,12], we have proposed leaf-clustering based approximate XML join algorithms for measuring the approximate similarity between XML documents and integrating them at subtree classes. Our previous experimental results show that the leaf-clustering based approximate XML join algorithms perform more efficiently for computing the approximate similarity between XML documents than the tree edit distance does. In the worst case, they are $O(n^2)$ operations for the XML documents of size n. Besides, the previous proposed algorithms can effectively integrate the XML documents containing similar information from different sources at subtree classes.

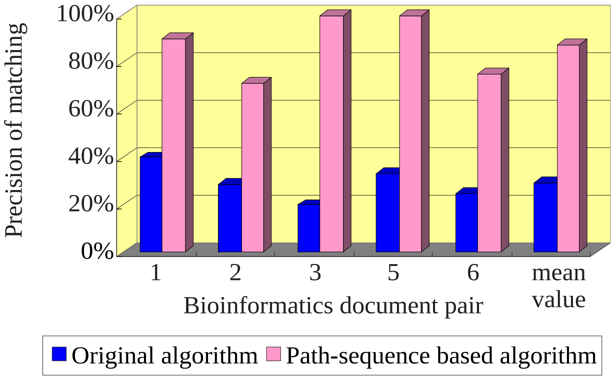
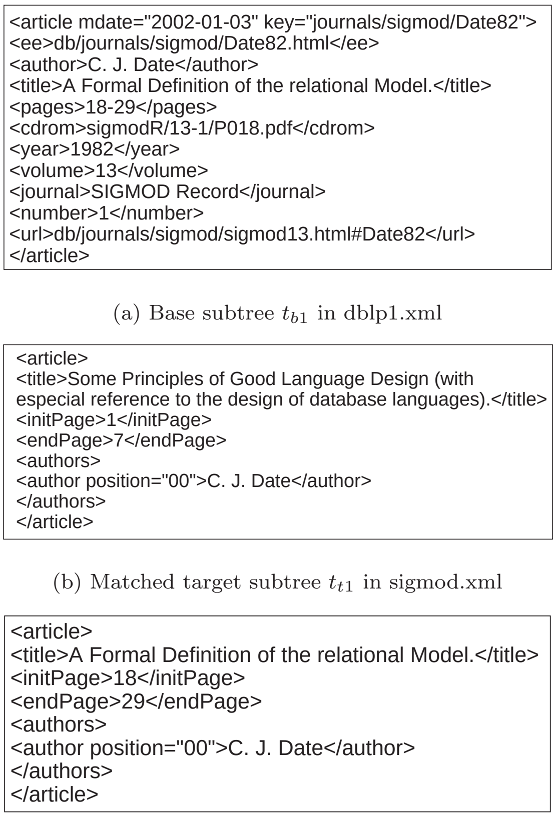


Figure 8 Precision of matching matching for bioinformatics documents



```
<article mdate="2002-01-03" key="journals/sigmod/Date82">
<ee>db/journals/sigmod/Date82.html</ee>
<author>C. J. Date</author>
<title>A Formal Definition of the relational Model.</title>
<pages>18-29</pages>
<cdrom>sigmodR/13-1/P018.pdf</cdrom>
<year>1982</year>
<volume>13</volume>
<journal>SIGMOD Record</journal>
<number>1</number>
<url>db/journals/sigmod/sigmod13.html#Date82</url>
</article>
```

(a) Base subtree t_{b1} in dblp1.xml

```
<article>
<title>Some Principles of Good Language Design (with
especial reference to the design of database languages).</title>
<initPage>1</initPage>
<endPage>7</endPage>
<authors>
<author position="00">C. J. Date</author>
</authors>
</article>
```

(b) Matched target subtree t_{t1} in sigmod.xml

```
<article>
<title>A Formal Definition of the relational Model.</title>
<initPage>18</initPage>
<endPage>29</endPage>
<authors>
<author position="00">C. J. Date</author>
</authors>
</article>
```

(c) Matched target subtree t_{t2} in sigmod.xml

Figure 9 Source codes for a base subtree and two matched target subtrees from dblp1.xml × sigmod.xml

6. Conclusion

In previous work, we have proposed leaf-clustering based approximate XML join algorithms for measuring the approximate similarity between XML documents and integrating them at subtree classes. However, in a join loop, a one-to-multiple matching problem may occasionally occur for a base subtree. Namely, one base subtree may happen to be matched with multiple target ones. Even the infrequent one-to-multiple matching may degrade the precision of the subtree matching. Therefore, an effective method for determining the most similar target subtree for the base one to overcome the one-to-multiple problem becomes important for leaf-clustering based approximate XML join algorithms.

In this paper, we have proposed a path-sequence based method to solve the one-to-multiple matching problem in leaf-clustering based approximate XML join algorithms. In our method, each path sequence from the top node to the matched leaf in the base and target subtree is extracted, and the most similar target subtree for the base one is determined by the path-sequence based subtree similarity degree. We have done experiments using both real bibliography and bioinformatics XML documents to compare the occurrence rate of one-to-multiple matching and the precision of matching for the proposed algorithm comparing with those for the original one. The experimental results show that the mean occurrence rate of one-to-multiple matching using the path-sequence based method is less than one sixth of that using the original algorithms, and the mean precision of matching for the path-sequence based method becomes about three times larger than that for the original algorithms. Therefore, we consider that our method can effectively decrease the occurrence rate of one-to-multiple matching for both bibliography and bioinformatics data, and hence improve the precise of the leaf-clustering based approximate XML join algorithms.

Acknowledgement

We thank Dr. Toshiyuki Amagasa from Tsukuba University for his helpful comments. This work is partially supported by the Grant-in-Aid for Scientific Research of MEXT Japan #16016232, by CREST of JST (Japan Science and Technology Agency), and by the TokyoTech 21COE Program "Framework for Systematization and Application of Large-Scale Knowledge Resources".

文献

[1] ACM SIGMOD Record in XML. Available at http://www.acm.org/sigmod/record/xml/

[2] M. Arenas and L. Libkin. A Normal Form for XML Documents. *ACM Transactions on Database Systems*, 29(1):195-232, March 2004.

[3] S. Chawathe and H. Garacia-Molina. Meaningful Change Detection in Structured Data. In *Proc. of ACM SIGMOD 1997*, pages 26-37, 1997.

[4] S. Chawathe, A. Tajaraman, H. Garacia-Molina and J. Widom. Change Detection in Hierarchically Structured Information. In *Proc. of ACM SIGMOD 1996*, pages 493-504, 1996.

[5] G. Cobena, S. Abiteboul and A. Marian. Detecting Changes in XML Documents. In *Proc. of ICDE 2002*, pages 41-52, 2002.

[6] W. Fan and L. Libkin. On XML Integrity Constraints in the Presence of DTDs. In *Proc. of PODS'01*, pages 114-125, 2001.

[7] M. Garofalakis and A. Kumar. Correlating XML data streams using tree-edit distance embeddings. In *Proc of PODS'03*, page 143-154, 2003.

[8] S. Guha, H.V. Jagadish, N. Koudas, D. Srivastava and T. Yu. Approximate XML Joins. In *Proc. of ACM SIGMOD 2002*, pages 287-298, 2002.

[9] S. Guha, N. Koudas, D. Srivastava and T. Yu. Index-Based Approximate XML Joins. In *Proc. of ICDE 2003*, pages 708-710, 2003.

[10] K. Ethier and S. Abel, Freely Available Structures: XML Document Type Definitions You Can Use Today, *Free Software Magazine*, Issue 6, pages 1-4, July 2005.

[11] W. Liang and H. Yokota. *LAX*: An Efficient Approximate XML Join Based on Clustered Leaf Nodes for XML Data Integration. In *Proc. of BNCOD 2005*, LNCS 3567, Springer, pages 82-97, July 2005.

[12] W. Liang and H. Yokota. *SLAX*: An Improved Leaf-Clustering Based Approximate XML Join Algorithm for XML Data Integration at Subtree Classes. In *Proc. of DB-Web 2005*, IPSJ Symposium Series, 2005(16):41-48, November 2005.

[13] A. Marian, S. Abiteboul, G. Cobena and L. Mignet. Change-Centric Management of Versions in an XML Warehouse. In *Proc. of 27th VLDB*, pages 581-590, 2001.

[14] A. Nierman and H. V. Jagadish. Evaluating Structural Similarity in XML Documents. In *Proc. of WebDB 2002*, pages 61-66, 2002.

[15] S. Selkow. The Tree-to-tree Editing Problem. *Information Processing Letters*, 6(6):184-186, December 1977.

[16] Swiss-Prot. http://www.ebi.ac.uk/swissprot/.

[17] Kuo-Chung Tai. The Tree-to-Tree Correction Problem. *Journal of the ACM*, 26(3): 422-433, 1979.

[18] TrEMBL. http://www.ebi.ac.uk/trembl/.

[19] Y. Wang, D. J. DeWitt and J. Cai. X-Diff: An Effective Change Detection Algorithm for XML Documents. In *Proc. of ICDE 2003*, pages 519-530, March 2003.

[20] World Wide Web Consortium (W3C). The Document Object Model (DOM). http://www.w3.org/DOM/.

[21] XML Version of Swiss-Prot. Avaibable at ftp://www.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.xml.gz

[22] XML Version of TrEMBL. Avaibable at ftp://www.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_trembl.xml.gz

[23] XML Version of DBLP. Available at http://dblp.uni-trier.de/xml/.

[24] K. Zhang. Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Recognition*, 28(3):463-474, 1995.

[25] K. Zhang and D. Shasha. Simple Fast Algorithm for the Editing Distance Between Trees and Related Problems. *SIAM Journal of Computing*, 18(6):1245-1262, December 1989.

[26] K. Zhang and D. Shasha. Tree Pattern Matching. *Pattern Matching Algorithms*, chapter 11. Oxford University Press, 1997.