論文 / 著書情報
Article / Book Information

Title	A Path-sequence Based Discrimination for Subtree Matching in Approximate XML Joins
Author	Wenxin Liang, Haruo Yokota
Journal/Book name	Proc. of International Special Workshop on Databases For Next Generation Researchers (SWOD 2006), Vol. , No. , pp. 24-28
Issue date	2006, 4
DOI	10.1109/ICDEW.2006.15
URL	http://www.ieee.org/index.html
Copyright	(c)2006 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Note	このファイルは著者（最終）版です。 This file is author (final) version.

A Path-sequence Based Discrimination for Subtree Matching in Approximate XML Joins

Wenxin Liang † Haruo Yokota ‡
†‡ Department of Computer Science
‡ Global Scientific Information and Computing Center
Tokyo Institute of Technology
2-12-1 Oh-Okayama, Meguro-ku Tokyo 152-8552, Japan
† wxliang@de.cs.titech.ac.jp, ‡ yokota@cs.titech.ac.jp

Abstract

In this paper, we discuss the one-to-multiple matching problem in leaf-clustering based approximate XML join algorithms and propose a path-sequence based discrimination method to solve this problem. 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 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.

1. Introduction

XML has been recognized as a widely important standard for data representation and exchange on the Internet. Recently, more and more data, for example bioinformatics data such as Swiss-Prot [7] and TrEMBL [8], and bibliography data such as DBLP [12] and ACM SIGMOD Record [1], are published and shared by XML on the Internet. However, XML documents from different data sources may represent nearly or exactly the same information but may be constructed by different structures. In addition, even the two XML documents convey the same information, each of them may have some extra information what the other does not do.

Figure 1 shows an example of two XML document trees with different DTDs. Although the two document trees are different on structures, they represent very similar information. Besides, each document has some information what the other does not do. For example, `volume` and `number` in Figure 1 (a); and `initPage` and `endPage` in Figure 1 (b).

In previous work [4, 5], 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 algorithms. 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 discrimination 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 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 ones. 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.

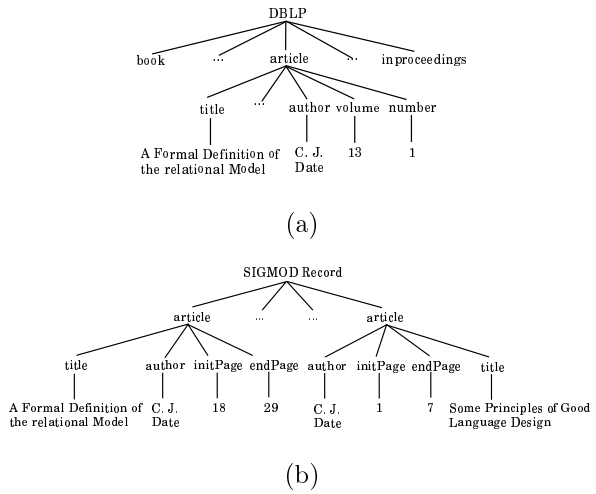


Figure 1. Example XML document trees

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

2. Related and Previous Work

A well formed XML document can be parsed into an ordered labeled tree [9]. 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.

The edit distance between two ordered labeled trees is defined as the minimum cost edit operations (insertions, deletions and substitutions) required to transform one tree to another [13]. The tree edit distance is recognized as a traditional metric for measuring the structural similarity between XML documents [2, 3, 6]. 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 as much as possible, S. Guha, et al. proposed the lower and upper bound as inexpensive filters for the tree edit distance operation [3]. 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 [4,5], 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.

3. One-to-multiple Matching Problem

In the leaf-clustering based approximate XML join algorithms [4,5], the two XML documents to be joined are segmented into subtrees representing independent information units[1]. Then, the subtree matching is determined by the *subtree similarity degree* 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}.*

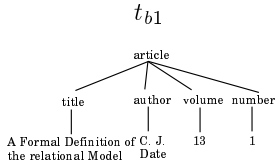
$$SSD(t_{bi}, t_{tj}) = \frac{n}{n_{bi}} \times 100\,(\%) \qquad (1)$$

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

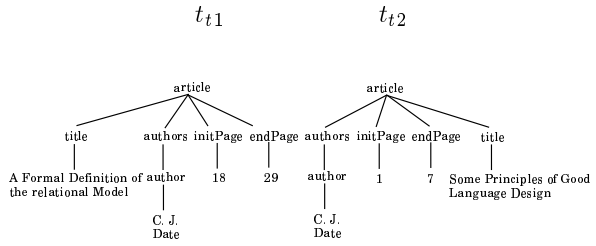
Problem 1 (One-to-multiple Matching) *For a base subtree t_{bi}, if there are two or more target subtrees t_{tj} having the same subtree similarity degree with t_{bi}, the base subtree t_{bi} will be matched with those target subtrees in one join loop.*

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 "A Formal Definition of the relational Model" and "C. J. Date", while those for (t_{b1}, t_{t2}) are "C. J. Date" and "1". However, according to Definition 1, $SSD(t_{b1}, t_{t1}) = SSD(t_{b1}, t_{t2}) = \frac{2}{4} \times 100\% = 50\%$. That means both t_{t1}*

1 The details of the segmentation algorithm are available in [4].



(a) Base subtree



(b) Matched target subtrees

Figure 2. One-to-multiple matching problem

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 an important issue.

4. Path-sequence Based Method

In Section 3, 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 based discrimination method to solve this one-to-multiple matching problem.

4.1. 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 $t_{bi}(1 \leq i \leq k_b)$ and $t_{tj}(1 \leq j \leq k_t)$, 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.*

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

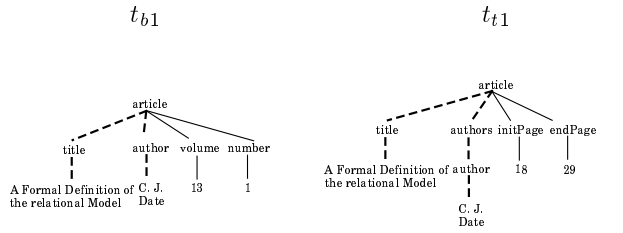
Then, the similarity between the matched subtrees can be determined by calculating the *path-sequence based subtree similarity degree.*

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.*

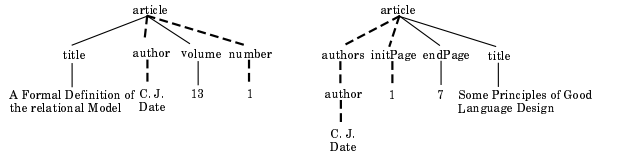
$$PSSD(t_{bi}, t_{tj}) = \frac{\sum_{i=1}^{K} PSD(i)}{K} \times SSD(t_{bi}, t_{tj}) \tag{3}$$

4.2. 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 $L_M(1) =$ "A Formal Definition of the relational Model" and $L_M(2) =$ "C. J. Date". Then, the base path sequence for $L_M(1)$, $P_b(1) = \{$"article", "title", "A Formal Definition of the relational Model"$\}$, and the target path sequence path for $L_M(1)$, $P_t(1) = \{$"article", "title", "A Formal Definition of the relational Model"$\}$ 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", "C. J. Date"$\}$ and $P_t(2) = \{$"article", "authors", "author", "C. J.*



(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

Date"}, $PSD(2)$ is also 100%. Therefore, the path-sequence based subtree similarity degree for t_{b1} and t_{t1}, $PSSD(t_{b1}, t_{t1}) = \frac{100\%+100\%}{2} \times 50\% = 50\%$. For the second matched subtree $T_M(2)$, the dashed lines in Figure 3 (b) show 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}, $PSSD(t_{b1}, t_{t2}) = \frac{100\%+66.7\%}{2} \times 50\% = 41.7\%$. Because $PSSD(t_{b1}, t_{t1}) = 50\% > PSSD(t_{b1}, t_{t2}) = 41.7\%$, the matched target subtree t_{t1} is considered to be more similar with the base subtree t_{b1}.

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.

```
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) ≥ Max) {
      Max = PSSD(t_bi, t_tj);
      output = (t_bi, t_tj);
    }
  }
  return output;
}
```

Figure 4. Path-sequence based Algorithm

5. Experiment

5.1. Experimental Environment

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

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

Table 1. Experimental Environment

5.2. Data Used

For bibliography data, we use six fragment documents of DBLP.xml [12], 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 [10], named sprot1-6.xml, 3MB per document (about 30,000 nodes), as the base documents, and a fragment document of uniprot_trembl.xml [11], named trembl.xml, 1MB (about 25,000 nodes), as the target one.

5.3. 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 ($\mathcal{N}_B$) as the following*

equation.

$$\mathcal{R} = \frac{\mathcal{N}}{\mathcal{N}_B} \times 100\,(\%) \quad (4)$$

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 ($\mathcal{N}_C$) out of the number of total multiple matched subtrees ($\mathcal{N}_M$) using the original algorithm as the following equation.*

$$\mathcal{P} = \frac{\mathcal{N}_C}{\mathcal{N}_M} \times 100\,(\%) \quad (5)$$

Table 2 and Table 4 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, $\mathcal{N}_B$ denotes the number of total subtrees in the base document, 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 3 and Table 5 show the precision of matching for the six pairs of bibliography and bioinformatics documents, where $\mathcal{N}_{CO}$ and $\mathcal{N}_{CP}$ denote the number of correctly matched subtrees using the original algorithms and the path-sequence based algorithm, respectively, $\mathcal{N}_M$ denotes the number of total multiple matched subtrees using the original algorithm, and $\mathcal{P}_O$ and $\mathcal{P}_P$ indicate the precision of matching for the original algorithms and the path-sequence based algorithm, respectively.

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.

	$\mathcal{N}_O$	$\mathcal{N}_P$	$\mathcal{N}_B$	$\mathcal{R}_O$	$\mathcal{R}_P$
$dblp1 \times sigmod$	15	3	1599	0.94%	0.19%
$dblp2 \times sigmod$	9	0	1451	0.62%	0.00%
$dblp3 \times sigmod$	8	0	1538	0.52%	0.00%
$dblp4 \times sigmod$	16	4	1584	1.01%	0.25%
$dblp5 \times sigmod$	28	5	1680	1.67%	0.30%
$dblp6 \times sigmod$	0	0	1474	0.00%	0.00%
Mean value	12.67	2.00	1554.33	0.79%	0.12%

Table 2. Occurrence rate of one-to-multiple matching for bibliography data

	$\mathcal{N}_{CO}$	$\mathcal{N}_{CP}$	$\mathcal{N}_M$	$\mathcal{P}_O$	$\mathcal{P}_P$
$dblp1 \times sigmod$	6	13	15	40.00%	86.70%
$dblp2 \times sigmod$	4	9	9	44.40%	100%
$dblp3 \times sigmod$	3	8	8	37.50%	100%
$dblp4 \times sigmod$	5	13	16	31.25%	81.25%
$dblp5 \times sigmod$	7	25	28	25.00%	89.30%
$dblp6 \times sigmod$	-	-	-	-	-
Mean value	5.0	13.6	15.2	36.00%	91.00%

Table 3. Precision of matching for bibliography data

	$\mathcal{N}_O$	$\mathcal{N}_P$	$\mathcal{N}_B$	$\mathcal{R}_O$	$\mathcal{R}_P$
$sprot1 \times trembl$	10	1	335	2.99%	0.30%
$sprot2 \times trembl$	7	2	337	2.08%	0.59%
$sprot3 \times trembl$	5	0	324	1.54%	0.00%
$sprot4 \times trembl$	0	0	309	0.00%	0.00%
$sprot5 \times trembl$	6	0	350	1.71%	0.00%
$sprot6 \times trembl$	12	3	360	3.33%	0.83%
Mean value	6.67	1.00	335.83	1.94%	0.29%

Table 4. Occurrence rate of one-to-multiple matching for bioinformatics data

	$\mathcal{N}_{CO}$	$\mathcal{N}_{CP}$	$\mathcal{N}_M$	$\mathcal{P}_O$	$\mathcal{P}_P$
$sprot1 \times trembl$	4	9	10	40.00%	90.00%
$sprot2 \times trembl$	2	5	7	28.57%	71.43%
$sprot3 \times trembl$	1	5	5	20.00%	100%
$sprot4 \times trembl$	-	-	-	-	-
$sprot5 \times trembl$	2	6	6	33.33%	100%
$sprot6 \times trembl$	3	9	12	25.00%	75.00%
Mean value	2.4	6.8	8.00	29.38%	87.29%

Table 5. Precision of matching for bioinformatics data

6. Conclusion

A one-to-multiple matching problem may occasionally occur in leaf-clustering based approximate XML join algorithms. 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 discriminating 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 discrimination method using the path-sequence based subtree similarity degree to solve the one-to-multiple matching problem in leaf-clustering based approximate XML join algorithms. We have conducted 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 of Tsukuba University for his valuable 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".

References

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

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

[3] 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.

[4] 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.

[5] 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.

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

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

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

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

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

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

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

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