JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1343-1358 (2017)
DOI: 10.6688/JISE.2017.33.5.14

Semantic Similarity Measure of Fuzzy XML DTDs With Extreme Learning Machine*

ZHEN ZHAO[1,2] AND ZONG-MIN MA[3,+]
[1]*College of Computer Science and Engineering*
Northeastern University
Shenyang, 110819 P.R. China
[2]*College of Information Science and Technology*
Bohai University
Jinzhou, 121013 P.R. China
[3]*College of Computer Science and Technology*
Nanjing University of Aeronautics and Astronautics
Nanjing, 211106 P.R. China

Data integration for distributed and heterogeneous XML data sources is still an open challenging, and XML DTD matching is crucial task in this process. A considerable amount of algorithms for comparing XML DTDs have been proposed in the literature. Yet, the existing approaches fall short in ability to identify semantic similarities in fuzzy XML DTDs. To fill this gap, in this paper, we provide an approach to cope with semantic similarities in the fuzzy XML DTDs. The present paper makes two major contributions. First, we propose a novel fuzzy XML DTD tree model to represent fuzzy XML DTD. Second, based on the proposed tree model, we present an effective algorithm based on Extreme Learning Machine (ELM) to synthesize the semantic similarities between fuzzy XML DTDs. The corresponding computational experimental results demonstrate that our proposed approach has a prominent high performance.

Keywords: data integration, fuzzy XML, semantic similarity, extreme learning machine (ELM), heterogeneous data

1. INTRODUCTION

With the development of the Internet, the management of data available on the Internet becomes ever more significant. XML (Extensible Markup Language) has become a de-facto standard for representing and manipulating rapidly increasing amount of Web data in numerous applications. As the number of applications that utilize heterogeneous and distributed data source grows, the importance of XML data integration mechanisms increases greatly [1, 2]. Due to a valid XML data is one that has a XML Document Type Definitions (DTD) and conforms to it. It was natural that there is a growing demand in the research of XML DTD matching before XML data integration. In the case of measuring the similarity of two XML DTDs, we consider the problem of similarity of two sets of regular expressions, typically called a XML DTD matching problem. Various systems and approaches have been designed to perform XML DTD matching automatically [3-9]. The majority of the approaches exploit functions which evaluate similarity of a particular feature of the given XML DTDs, such as similarity of labels [3-5], similarity of context

Received June 21, 2016; revised July 17, 2016; accepted August 14, 2016.
Communicated by Shyi-Ming Chen.
+ Corresponding author: zongminma@nuaa.edu.cn.
* This work was supported by the National Natural Science Foundation of China (61370075 and 6177051086).

[6, 7] or similarity of paths [4, 8].

Previous similarity measuring methods of XML DTDs assume that data is deterministic. In fact, fuzzy information is often included in some practical application domains. Some simple model for representing and querying XML with fuzzy information are proposed in [10, 11]. Although XML DTD matching has been extensively investigated, the proposed approaches cannot be applied to process fuzzy XML DTDs matching due to the lack of an effective fuzzy XML DTD model. To our best knowledge, so far, there are not any reports discussing the similarity measure of the fuzzy XML DTDs. This motivates us to fill this gap. Concentrating on the similarity comparison of heterogeneous fuzzy XML DTDs which are collected from different data sources, this paper devotes to develop an integrate approach to support the identification of semantic similarity measure in the fuzzy XML DTDs. For this purpose, we take a first step in construction of a new fuzzy XML DTD tree model, which makes it easier to describe fuzzy data and capture the feature information in fuzzy XML DTDs. Based on the proposed fuzzy XML DTD tree model, we develop an effective algorithm based on Extreme Learning Machine (ELM) to synthesize the individual similarity measure of nodes. Here the ELM-based algorithm is a kind of fuzzy XML clustering algorithms, which is categorized as fuzzy web mining [12]. A comprehensive survey on fuzzy web mining is given in [13, 14]. Actually many useful techniques for fuzzy data mining have been proposed in the literature [15].

Being a machine learning approach, ELM is invented for classification, regression and so on [16, 17]. In particular, ELM has been applied in XML documents classification [9] and even in probabilistic XML documents classification [18]. But this paper is the first effort to apply ELM in the semantic similarity measure of fuzzy XML DTDs. In addition, the fuzzy XML DTD tree proposed in the paper is different from the common fuzzy XML DTD tree model given in [19]. The major difference is that the fuzzy XML DTD tree proposed in the paper is a kind of simplified fuzzy XML DTD tree model, which removes the redundant nodes so that the complexity of similarity comparison can be reduced. To sum up, the main contributions of this paper include: (1) a novel fuzzy XML DTD tree model is proposed, which can concisely represent the feature information in fuzzy XML DTDs; (2) an ELM-based algorithm is developed to synthesize semantic similarities by combining multiple individual similarities of fuzzy XML DTDs tree nodes. The experimental results validate our approach and show the practicability of algorithm.

The rest of the paper is organized as follows. After a presentation of fuzzy XML DTD tree model in Section 2, we introduce Extreme Learning Machine in Section 3. The semantic similarity measure approach in the fuzzy XML DTD is proposed in Section 4. Experimental evaluations are given in Section 5, and Section 6 concludes this paper.

2. REPRESENTATION MODEL OF FUZZY XML DTD

2.1 Fuzzy XML DTD

A fuzzy XML DTD serves as a grammar for a fuzzy XML document, determining its internal structure. A fuzzy XML DTD is a structure composed of a set of elements and attributes, linked together via the containment relation. In addition, in order to represent

fuzzy information in fuzzy XML data, a representation model based on "membership degree and possibility distributions" is developed in [19]. In this model, an element may be involved in a membership degree which indicates the possibility of being its parent's child element. The attribute values of elements may be presented as possibility distributions in this representation model. The *Type* attribute as a child of element *Dist* is used to indicate the type of possibility distribution, having values of disjunctive or conjunctive. In addition, each *Dist* element has a *Val* element as its child. The *Poss* attribute as child of element *VAL* indicates the membership degrees of a given element.

```
1.  <!ELEMENT College (Val+)>
2.  <!ATTLIST College Cname  IDREF #REQUIRED>
3.  <!ELEMENT Val (Department*)>
4.  <!ATTLIST Val Poss CDATA "1.0">
5.  <!ELEMENT Department (Teacher*, Student*)>
6.  <!ATTLIST Department Dname IDREF #REQUIRED>
7.  <!ELEMENT Teacher (Dist)>
8.  <!ATTLIST Teacher TID IDREF #REQUIRED>
9.  <!ELEMENT Dist (Val+)>
10. <!ATTLIST Dist Type (disjunctive)>
11. <!ELEMENT Val (Tname?, Title?)>
12. <!ATTLIST Val Poss CDATA "1.0">
13. <!ELEMENT Student (Age?, (Email|Phone)?)>
14. <!ATTLIST Student Sid IDREF #REQUIRED>
15. <!ELEMENT Age (Dist)>
16. <!ELEMENT Dist (Val+)>
17. <!ATTLIST Dist Type (disjunctive)>
18. <!ELEMENT Email (Dist)>
19. <!ELEMENT Dist (Val+)>
20. <!ATTLIST Dist Type (conjunctive)>
21. <!ELEMENT Phone (Dist)>
22. <!ELEMENT Dist (Val+)>
23. <!ATTLIST Dist Type (conjunctive)>
24. <!ELEMENT Val (#PCDATA)>
25. <!ATTLIST Val Poss CDATA "1.0">
```

Fig. 1. A fragment of fuzzy XML DTD.

Here we do not present the detailed definitions of fuzzy XML DTD representation model and only give an example fragment of fuzzy XML DTD shown in Fig. 1. One can refer to [19] for more details.

2.2 Fuzzy XML DTD Tree Model

Generally speaking, a fuzzy XML DTD which represents hierarchically structured information also can be presented as a rooted ordered labeled tree. But the fuzzy XML DTD is clearly different from the crisp XML DTD because the fuzzy XML DTD con-

tains several special fuzzy construction (*Poss* and *Type*, *VAL* and *Dist*).

In order to reduce the complexity of the similarity comparison, the redundant elements/attributes are deleted and the pertinent information of the elements/attributes is encapsulated in the tree node. We note that *Type* always appears as the first child attribute of the *Dist* and *Poss* always appears as the first child attribute of the *VAL*. They are considered redundant data and increase the computational complexity. Therefore all of *Type* and *Poss* attribute are no longer reserved when a fuzzy XML DTD is mapped into a fuzzy XML DTD tree. But *Type* values (conjunctive/disjunctive) are remained because they need to be considered while calculating the nodes similarity degree (cf. Section 4.1). So we need to copy *Type* values into its sibling (element/attribute) nodes in processing of transformation. Similarly, we can disregard the *Val* and *Dist* elements if it is not affecting the tree structure and the depth of the other node. Based on the discussion above, in order to improve the efficiency of comparison, we present a new **F**uzzy **X**ML **D**TD **T**ree model (FXDT for short). It is defined as follows.

Definition 1 (Fuzzy XML DTD tree): Formally, we model a fuzzy XML DTD as a rooted ordered labeled tree FXDT = {*N*, *E*, *L*, *T*, *FT*, *CC*, *AC*}. Here

- *N* is the set of nodes in tree FXDT.
- *E* is the set of edges, which reflect the hierarchical structure of the tree FXDT.
- *L* is the set of labels of the elements/attributes corresponding to the nodes in *N*.
- *T* is the set of data types, including the basic element/attribute data types.
- *FT* is the set of fuzzy data types of nodes *N*.
- *CC* is the set of cardinality constraints associated with the elements/attributes of FXDT ('?', '*', '+' and Null).
- *AC* is the set of alternative constraints associated with the elements/attributes of FXDT (',', '|').

We need to hold related information of elements/attributes (*e.g.* label, constraint, the fuzzy value) for computing of the similarity (cf. Section 4.1) when the fuzzy XML DTD is transformed into the fuzzy XML DTD tree model (FXDT). Hence, following our tree representation model, a fuzzy XML DTD tree node is modeled as follows:

Definition 2 (Fuzzy XML DTD tree node): A node $n \in N$ of FXDT is represented by a sextuplet n = {*NodeLabel*, *NodeDepth*, *NodeDataType*, *NodeFuzzyType*, *NodeCardConstraint*, *NodeAlterConstraint*}. Here

- $NodeLabel \in L$ is the label name of the node.
- *NodeDepth* is the nesting depth of the node in the fuzzy XML DTD. The depth of the root node is defined to be 1.
- $NodeDataType \in T$ is data-types. It may be "#PCDATA", "String", "Decimal", and so on.
- $NodeFuzzyType \in FT$ denotes the type of possibility distribution, disjunctive or conjunctive distributions. For a crisp node, its value is equal to Null.
- $NodeCardConstraint \in CC$ is the cardinality constraints of the node.
- $NodeAlterConstraint \in AC$ is the alternative constraints of the node.

To sum up, FXDT is a rooted ordered tree, in which the nodes represent the elements/attributes. Nodes are ordered following their order of appearance in the fuzzy XML DTD. And FXDT representation model that we propose here considers the most common characteristics of fuzzy XML DTDs. After the fuzzy XML DTD mentioned in Section 2.1 is converted into the FXDT tree model in this way, we can measure the individual similarity of nodes and apply the approach based on ELM to synthesize the semantic similarities of nodes. Fig. 2 shows a FXDT tree instance that basically comes from the corresponding fragment of a fuzzy XML DTD is described in Fig. 1.

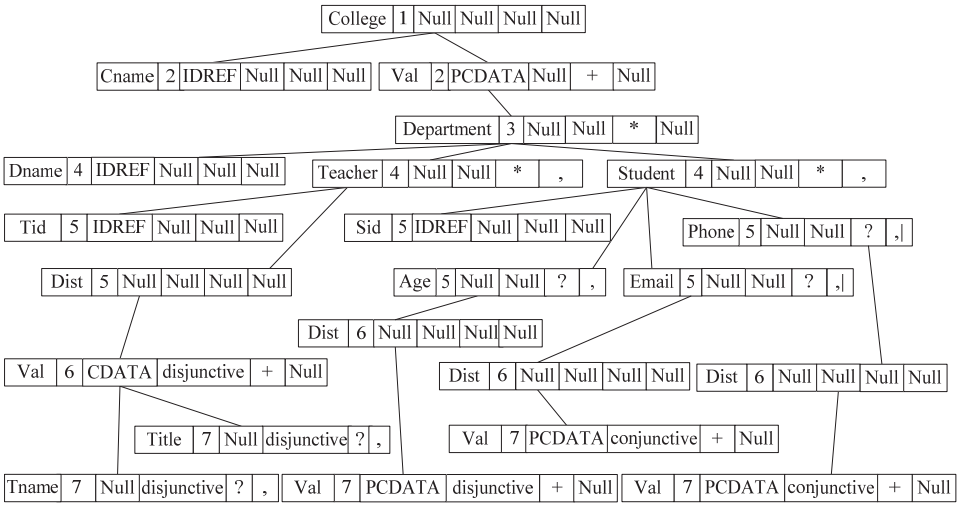


Fig. 2. A FXDT tree corresponding to the fuzzy XML DTD in Fig. 1.

3. EXTREME LEARNING MACHINE (ELM)

The Extreme Learning Machine (ELM) model was originally proposed in [16] and developed in [17], which implements a single-hidden layer feed forward neural network (SLFN) with N mapping neurons. The biggest advantage of ELM is that it can provide extremely fast learning speed and good generalization performance compared with the traditional neural network. The essence of ELM is that the network's hidden layer weights and bias values can be randomly initialized.

Consider N arbitrary samples $(x_i, t_i) \in \mathbb{R}^{n\times m}$. Then ELM is modeled as

$$\sum_{i=1}^{L} \beta_i g(W_i \cdot x_j + b_i) = o_j \quad (1 \le j \le N). \tag{1}$$

Here L is the number of hidden layer nodes, $g(.)$ is activation function, W_i is the weight vector between the ith hidden node and the input nodes, β_i is the weight vector between the ith hidden node and the output nodes, and b_i is the bias of the ith hidden node, and o_j is the actual output value of the SLFN. There must exist W_i, β_i, b_i such that

$$\sum_{i=1}^{L} \beta_i g(W_i \cdot x_j + b_i) = t_j \quad (1 \le j \le N). \tag{2}$$

The equation above can be expressed compactly as follows:

$$H\beta = T. \tag{3}$$

Training of a single hidden layer neural network is equivalent to simply finding the least-squares solution of the linear system. The output weight β is calculated as

$$\beta = H^{\dagger}T. \tag{4}$$

Here $H^{\dagger} = (H^T H)^{-1} H^T$ is the Moore–Penrose generalized inverse of input matrix H.

After the calculation of the network output weights β, the response of the corresponding to a new data points (vector x_t) can be predicted by

$$O_t = \beta h(x_t). \tag{5}$$

4. SEMANTIC SIMILARITY MEASURE OF FUZZY XML DTDS USING ELM

Semantic similarity measure in the fuzzy XML DTDs can be viewed as detecting the similarity matching of fuzzy XML DTD trees. So how to calculate similarity matching of the fuzzy XML DTD trees is a key task. To match the fuzzy XML DTD trees, firstly, we need to synthesize semantic similarity of nodes by utilizing ELM and to find out the similar nodes being compared from the fuzzy XML DTD trees, and then to calculate the similarity of the fuzzy XML DTDs according to the number of matching nodes. In other words, we need to first identify the similarity of nodes before comparing the similarity of fuzzy XML DTD trees.

4.1 Node Similarity Measures

Given a fuzzy XML DTD tree, a node (element/attribute) is a fundamental data item for the similarity measures. We use Sim_{Node} (N_i, N_j) to represent the similarity degree of two nodes N_i and N_j, where come from different fuzzy XML DTDs. The characteristics associated with each node in a fuzzy XML DTD tree are called the node features. *NodeLabel*, *NodeDepth*, *NodeDataType* are the most commonly used features for elements/ attributes. To accurately assess the similarity between node-pairs, a similarity measure should exploit the features of nodes. According to the different exploited features, some of the commonly used similarity measures were proposed.

• Label name similarity measure

The label name (*NodeLabel*) is important for node matching. Label similarity measures take the advantage of the strings representation of label names to deal with the similarity of two nodes. There are many commonly used methods to compare strings, for

example, the Jaro similarity measure [20]. Here we adopt Lin's similarity measure method in [21] based on the edit distance. The Lin's similarity measure between two strings is given by the minimum cost of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. To compute the similarity degree between the nodes based on values of *NodeLabel* L_i and L_j, the following formula is used:

$$Sim_{Label} = (N_i, N_j) = 1/(1 + editDistance(L_i, L_j)). \quad (6)$$

Here $editDistance(L_i, L_j)$ is the minimum number of character insertion, deletion, and substitution operations that is needed to transform L_i to L_j.

• Depth similarity measure

The depth of a node is also a feature must be considered. Depth similarity measure has been proposed to assess the depth of nodes and their nearest common ancestor node. The similarity of two nodes is defined by how closely they are in the hierarchy and it can be calculated using the following equation:

$$Sim_{Depth}\ (N_i, N_j) = (d_{ci}+d_{cj})/(d_i+d_j) \quad (7)$$

Here d_i and d_j are the depth of N_i and N_j in local fuzzy XML DTD tree. N_c is the most specific common ancestor of N_i and N_j, and d_{ci} and d_{cj} is the depth of N_c in the fuzzy XML DTD tree respectively.

• Data type similarity measure

The node data type is another information source that makes a contribution in determining the node similarity [22]. Nodes having the same data types or belonging to the same data type category have the possibility to be similar and their data type similarity measure ($Sim_{DataType}$) is high. Table 1 illustrates that the data type similarity of nodes having the different data types is different.

Table 1. Data type similarity table.

Type1	Type2	$Sim_{DataType}$
#PCDATA	#PCDATA	1.0
CDATA	CDATA	1.0
#PCDATA	CDATA	0.5
CDATA	NMTOKEN	0.8
ID	IDREF	0.9
…	…	…

• Fuzzy type similarity measure

The node values of *NodeFuzzyType* are necessary feature information that contributes to determining the node similarity. Note that the *NodeFuzzyType* value of a node in a fuzzy XML DTD tree is of either disjunctive or conjunctive. Fuzzy type similarity is represented as $Sim_{FuzzyType}$ (N_i, N_j). It can be shown in following formula.

$$Sim_{FuzzyType}(N_i,N_j)=\begin{cases}1 & \text{if} \quad T_i = T_j \\ 0.5 & \text{if} \quad T_i \neq T_j\end{cases} \tag{8}$$

• Cardinality constraints similarity measure

Another available feature of the node that makes a contribution to assessing the node similarity is its cardinality constraint [3]. Nodes having the same cardinality constraints have the more high possibility similarity measure. We denote the cardinality constraint similarity of two nodes as $Sim_{CardConstraint}$ (N_i, N_j), which can be determined from the table of cardinality constraints similarity (Table 2).

Table 2. Cardinality constraints similarity table.

	*	+	?	NULL
*	1.0	0.9	0.8	0.5
+	0.9	1.0	0.8	0.7
?	0.8	0.8	1.0	0.8
NULL	0.5	0.7	0.8	1.0

• Alternative constraints similarity measure

Alternativeness constraint operators specify a node's disposition with regard to its parent and siblings [3]. For instance, in declaration ((*a* | *b*), *c*), Alternative constraints ', |' is associated with both elements *a* and *b*, while Alternative constraints ',' is associated with element *c*. We denote the Alternative constraint similarity of two nodes as $Sim_{Alter\text{-}Constraint}$ (N_i, N_j), which can be determined from the Table 3.

Table 3. Alternative constraints similarity table.

	,	|	,|	|,	…
,	1.0	0	0.8	0.5	…
|	0	1.0	0.5	0.8	…
,|	0.8	0.5	1.0	0	…
|,	0.5	0.8	0	1.0	…
…	…	…	…	…	…

There are a lot of node similarity measure approaches [23-25] and we simply sign corresponding individual similarity measure which is calculated by those approaches as S_1, S_2, $\cdots$, S_n. It is clear that each individual similarity measure of node exploits a specific feature of the node. But, it is necessary to consider a variety of node features along with a variety of measures to assess the similarity. This multi-measure nature is potent in that it makes a matching system highly flexible and adaptable to a particular application domain. However, it results in considerable challenges on how to combine these measures [26]. In general, with regard to the purpose of the similarity measure, all the individual results are then combined into the resulting similarity value, usually using a kind of weighted sum [3]. To get node semantic similarity value between pairs of nodes, in the next section, we present strategies of combining individual similarity measures values resulted from different features similarity measures.

4.2 Synthesizing Node Similarity using ELM

Target of this paper is that designs a matching framework based on semantic similarity between two fuzzy XML DTDs. To get the similarity of fuzzy XML DTDs, firstly, we have to measure the similarity of nodes in fuzzy XML DTD trees. But single measurement method is biased and partial. These similarity measure value need to be integrated. In some previous studies [3, 22], the method based on weight is commonly used. Duo to the allocation of weight is provided by the human expert, so that the ultimate integration value may be a relatively large deviation. So, we introduce a synthesizing similarity measure approach using ELM with regard to the similarity integration. Specifically, it presents an integrated similarity measure approach between nodes (elements/attributes) come from different fuzzy XML DTD trees based on ELM, which judges the latent semantic similarity. More accurate results can be obtained comparing with those approaches based on weight given by a human expert mentioned in research [22].

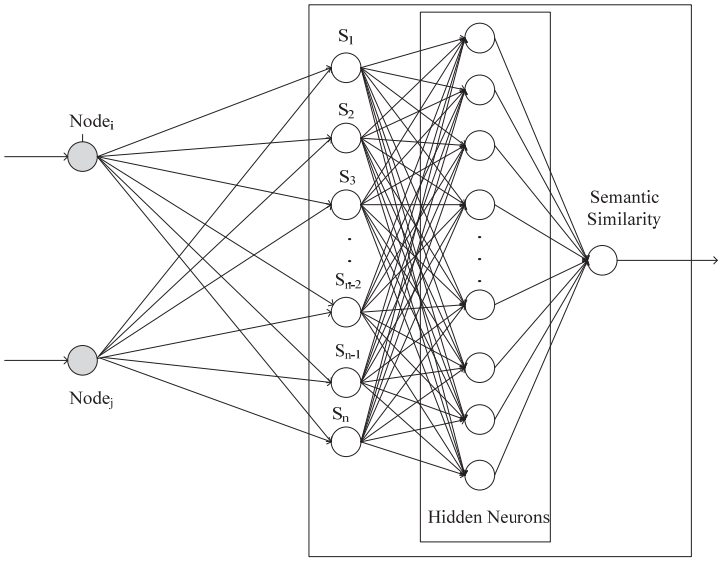


Fig. 3. ELM-based similarity synthesis.

Node matching is a process of measuring similarity among nodes and then selecting the most plausible nodes based on the similarity measure. In this perspective, the node matching procedure is as follows:

Step 1: It is needed to extract features of nodes (*e.g.*, label or depth) coming from two different fuzzy XML DTD trees, and represent them in an internal format can be applied to individual similarity measure like mentioned in Section 2.2.

Step 2: It is needed to compute individual similarity measures value respectively like mentioned in Section 4.1.

Step 3: It is needed to synthesize the individual similarity measures using ELM. This synthesis can be viewed as a prediction.

In Step 3, we need to synthesize various similarity measures (cf. Section 4.1) values to a single semantic similarity value. The ELM model of synthesizing various similarity measures in node matching is depicted in Fig. 3. $Node_i$ and $Node_j$ are node-pairs come from different fuzzy XML DTDs trees ($FXDT_p$, $FXDT_q$) respectively. S_1, S_2, …, S_n are individual similarity measures values. The processing which synthesizes various similarity measures using ELM is divided into two phases, namely, training phase and prediction phase. In order to obtain reliable training data and to construct a robust prediction model, the training phase supports the prediction-phase by providing a supervised learning that predicts semantic similarity from various measures. On the other hand, the prediction-phase is semantic similarity computing of node-pairs. This prediction model based on ELM is built to map the relationship between the input variables (various similarity measures values) and output variables (semantic similarity value) with the sampling data. The processing of sampling is presented as follows. At first, randomly selecting a number of node-pairs ($Node_i$, $Node_j$) from different fuzzy XML DTD trees ($FXDT_p$, $FXDT_q$) to be used as a training data set. Secondly, compute individual similarity measures value S_1, S_2, …, S_n between two nodes of node-pairs using the approach mentioned in section 4.1 respectively. It is an input variable. Thirdly, label samples by human experts to determine semantic similarity measure between the two nodes. It is an output variable. The use of the ELM model dramatically reduces the time required for building a prediction model. A similarity synthesizing algorithm based on ELM described in above is presented as follows.

Algorithm 1: Sim_{Node}
Input: $Node_i$, $Node_j$ //node-pairs of different fuzzy XML DTD trees
Output: *SimNodeValue* //node similarity measure value
Begin
1. Node Feature extracting;
2. Compute individual similarity measure values S_1, S_2, …, S_n, respectively;
3. Training ELM for calculate $\beta = H^{\dagger}T$;
4. Calculate output $SimNodeValue = \beta H$;
5. Return *SimNodeValue*

End

First, the feature value of each node in the fuzzy XML document tree is extracted (with several linguistic terms) based on the FXDT model in Section 2.2 (Line 1). Then individual similarity measure values S_1, S_2, …, S_n between two nodes of node-pairs are computed using the approach proposed in Section 4.1, respectively (Line 2). After obtaining enough node-pairs training sample, the value of β can be calculated by training ELM (Line 3). At last, the semantic similarity of node-pairs is calculated (Line 4). In the semantic similarities computation process, ELM algorithm is used to construct the prediction model in order to map the relationship between the various similarity measures values and semantic similarity value. This approach can thus speed up the later synthesizing process compared to the common similarity combining approach.

4.3 Fuzzy XML DTD Similarity Measure

Given a set of fuzzy XML DTDs $FD = \{FDTD_1, FDTD_2, \ldots, FDTD_N\}$, we compute

the similarity of their corresponding FXDT trees. For any two FXDTs, we sum up the number of all match pairs of nodes which similarity values are bigger than a given threshold (θ), and normalize the result. Algorithm 2 gives the algorithm to compute the similarity matrix of a set of fuzzy XML DTDs.

Algorithm 2: FDTDSimilarity
Input: $FD=\{FXDT_1,\ldots,FXDT_N\}$ //fuzzy XML DTD trees set
Output: *FDSimMatrix* //FDTD similarity matrix FDSimMatrix
Begin
1. **For** (p = 1 to N–1)
2. { **For** (q=p+1 to N)
3. { **For** each node $Node_{pi} \in FXDT_p$, $Node_{qj} \in FXDT_q$
4. {if ($Sim_{Node}(Node_{pi}, Node_{qj}) > \theta$)
5. $NumSimNode_{pq} = NumSimNode_{pq}+1$; //number of similarity node }
6. $FDSimMatrix_{pq} = NumSimNode_{pq}$ /min ($|FXDT_p|, |FXDT_q|$)
7. }
8. }
9. Return *FDSimMatrix*
End

The FDTDSimilarity algorithm is a fuzzy XML DTDs clustering algorithm. In this algorithm, we select any of the two fuzzy XML DTDs in the fuzzy XML DTDs set *FD* (Lines 1–2), traveling each node in the two fuzzy XML DTDs (Lines 3) and computing node similarities (Line 4). If the similarity degree of any two nodes is greater than a given threshold (say θ), the number of similar nodes adds 1 (Lines 5). Finally a similarity matrix that represents the semantic similarity values between all nodes is obtained (Line 6). After we obtain the fuzzy XML DTDs similarity matrix, clustering of fuzzy XML DTDs can be carried out. Fuzzy XML DTDs from the same application domain tend to be clustered together and form different clusters.

5. EXPERIMENTAL EVALUATIONS

To evaluate the effectiveness of our approach for fuzzy XML DTDs matching, we present experiments conducted to compare the performance of the proposed approaches and report the results. We present the similarity measures evaluation criteria adopted in our experimental evaluation process in Section 5.1. Section 5.2 details our matching experiments results and performance analysis.

5.1 Evaluation Metrics

There are two main performances that should be considered with regard to the matching process: *matching effectiveness* and *matching efficiency*.

First let us look at the effectiveness measures. Owing to the proficient use of predecessors, we make use of the *precision* and *recall* metrics defined in [23, 27], to evaluate the effectiveness of our approach. Following Dalamagas *et al*. [23, 27], it is needed to define some terms that are used in computing match effectiveness as follows. A is the

number of fuzzy XML DTDs that are the correct matches and correctly identified by the system. B is the number of fuzzy XML DTDs that are the false matches but identified by the system. C is the number of fuzzy XML DTDs that should have been matches but not identified by the system.

Hence, we have:

$$P = A/(A+B),\ R = A/(A+C),\ F-measure = (2\times P\times R)/(P+R). \tag{9}$$

The effectiveness of matching is commonly determined with the standard measures precision (P), recall (R) and *F-measure* with respect to a manually determined "perfect" result. Efficiency of a matching system is usually determined by using two aspects: response time and space. In this study, we make use of the response time as an indicator for the matching efficiency.

5.2 Experimental and Results

In this section, the simulation results are discussed in detail to illustrate the effectiveness of the proposed method in evaluating fuzzy XML DTDs similarity. And we use a real-world data set[1] from four different domains as original datasets, from which we synthesized synthetic fuzzy datasets for the experimental evaluation.

We firstly add fuzzy nodes to the XML DTDs in real-world data sets. That is to say, we use our fuzzy XML DTDs generator to randomly generate the multiple corresponding fuzzy XML DTDs. After doing this, each domain has corresponding 50 fuzzy XML DTDs which show different characteristics and they represent different application domains, as shown in Table 4.

Table 4. Characteristics of the synthetic data sets.

Domain	No. of fuzzy XML DTDs	No. of element	Avg-depth
Auction	50	1365	4.76
University	50	975	4. 44
Protein Sequence	50	4732	5.35
Publication	50	2650	7. 14

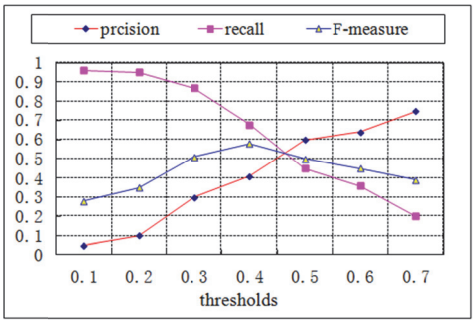


Fig. 4. Matching quality in synthetic data sets.

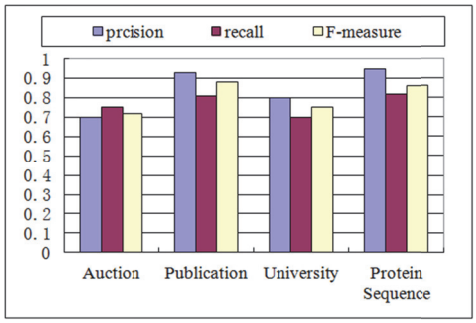


Fig. 5. Matching quality comparison for difference domain from synthetic data sets.

[1] All of data sets are collected from http://www.cs.washington.edu/research/xmldatasets/

• *Matching effectiveness*

The first scenario is to compare the quality to individual domain (Auction) in synthetic data sets. In the first step, the similarity degree between every fuzzy XML DTD pairs is first computed by using our method after sampling. And we obtain the fuzzy XML DTDs similarity matrix. Second Step, we started a series of classification tasks by varying the classification similarity threshold in the [0, 1] interval. The fuzzy XML DTDs which similarity degrees are greater than a given threshold will be grouped together. Lastly, according to the clustering results, we compute precision (*P*), recall (*R*) and *F-measure* for each of the classification sets in the multilevel classification phases, the results of these evaluations are reported in Fig. 4. From Fig. 4, we can see that inconsistent fuzzy XML DTDs are gradually filtered from the classification sets, while varying the classification threshold from 0 to 1. Results demonstrate that our algorithm yields optimal classes at a very early stage of the multilevel classification process (with classification thresholds < 0.5). The second scenario is to compare the quality with difference domain from synthetic data sets. In this scenario, the quality obtained in different threshold between [0, 1] for all fuzzy XML DTDs in the same domain is then averaged to obtain the ultimate quality for the domain. Results are summarized in Fig. 5, and we find that matching quality over the publication and protein sequence higher than the matching quality over the auction and university domains. This is mainly due to the fact that fuzzy XML DTDs in publication and protein sequence domains are more homogeneous than fuzzy XML DTDs in other domains.

• *Matching efficiency*

We experimented with synthetic data sets (chose all 200 fuzzy XML DTDs with a total size of 1MB). The timing experiments were implemented in JDK 1.6 and MATLAB R2014a, and performed on a system with Intel Core i7 processor, 8GB RAM and running on Windows 7. The results in Fig. 6 reflect that the time of matching fuzzy XML DTDs grows in a linear dependency on the size of fuzzy XML DTDs being compared. This figure indicates that the FDTD measure performs the worst among the other similarity measures. The reason behind this can be explained as follows: The FDTD measure is based on the other similarity measures. The synthesizing process consumes much time.

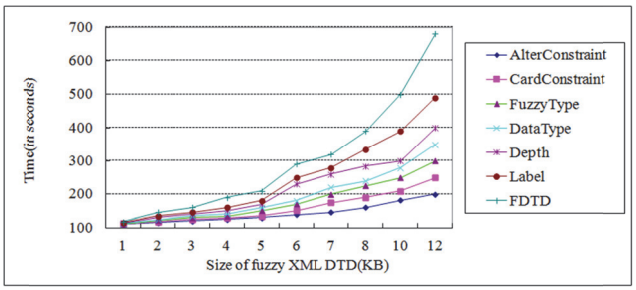


Fig. 6. Timing results to compare similarity measure for different size of fuzzy XML DTDs.

6. CONCLUSION

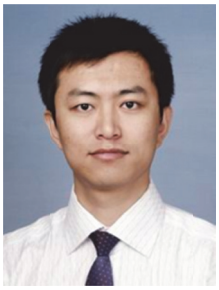
In order to deal with the issue of semantic similarities in the fuzzy XML DTDs ef-

fectively, in this paper, we first propose a novel tree representation model to capture the node information of fuzzy XML DTDs, and then a effective solution based on Extreme Learning Machine is proposed, which synthesize the semantic similarities of fuzzy XML DTDs tree nodes represented with the proposed model. The experimental results show that our algorithms can efficiently perform matching on the fuzzy XML DTDs. This study provides two contributions: (1) a novel fuzzy XML DTD tree model is proposed, which eliminates the redundancy node in order to reduce the complexity of the similarity comparison; (2) an ELM-based algorithm is proposed for synthesizing semantic similarities of fuzzy XML DTDs trees by combining multiple individual similarities of fuzzy XML DTDs tree nodes. Note that the ELM-based algorithm proposed in the paper is mainly for synthesizing semantic similarities of fuzzy XML DTDs trees. At this point, granular computing can play an essential role. Currently granular computing has been widely investigated in the literature [28, 29]. It is worth of future research to apply granular computing techniques to solve the semantic similarity measure of fuzzy XML DTDs problem.

REFERENCES

1. A. Thomo and S. Venkatesh, "Rewriting of visibly pushdown languages for XML data integration," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008, pp. 521-530.
2. S. Guha, H. V. Jagadish, N. Koudas, and D. Srivastava, and T. Yu, "Integrating XML data sources using approximate joins," *ACM Transactions on Database Systems*, Vol. 31, 2006, pp. 161-207.
3. J. Tekli and R. Chbeir, "Minimizing user effort in XML grammar matching," *Information Sciences*, Vol. 210, 2012, pp. 1-40.
4. M. L. Lee, L. H. Yang, W. Hsu, and X. Yang, "XClust: clustering XML schemas for effective integration," in *Proceedings of the 11th International Conference on Information and Knowledge Management*, 2002, pp. 292-299.
5. A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, "Algorithmic detection of semantic similarity," in *Proceedings of the 14th International Conference on World Wide Web*, 2005, pp. 107-116.
6. A. Wojnar, I. Mlýnková, and J. Dokulil, "Structural and semantic aspects of similarity of document type definitions and XML schemas," *Information Sciences*, Vol. 180, 2010, pp. 1817-1836.
7. H. Su, S. Padmanabhan, and M. L. Lo, "Identification of syntactically similar DTD elements for schema matching," in *Proceedings of International Conference on Advances in Web-Age Information Management*, 2001, pp. 145-159.
8. N. Tansalarak and K. T. Claypool, "QMatch – using paths to match XML schemas," *Data and Knowledge Engineering*, Vol. 60, 2007, pp. 260-282.
9. X. Zhao, G. Wang, X. Bi, P. Gong, and Y. Zhao, "XML document classification based on ELM," *Neurocomputing*, Vol. 74, 2011, pp. 2444-2451.
10. K. Turowski and U. Weng, "Representing and processing fuzzy information-an XML-based approach," *Knowledge-Based Systems*, Vol. 15, 2002, pp. 67-75.
11. B. Oliboni and G. Pozzani, "Representing fuzzy information by using XML schema," in *Proceedings of the 19th International Conference on Database and Expert Systems Application*, 2008, pp. 683-687.
12. G. E. Tsekouras and D. Gavalas, "An effective fuzzy clustering algorithm for web

document classification: A case study in cultural content mining," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 23, 2013, pp. 869-886.
13. C. W. Lin and T. P. Hong, "A survey of fuzzy web mining," *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, Vol. 3, 2013, pp. 190-199.
14. T. P. Hong, C. H. Chen, and J. C. W. Lin, "A survey of fuzzy data mining techniques," *Fuzzy Statistical Decision-Making*, Vol. 343, 2016, pp. 329-354.
15. J. C. W. Lin, T. P. Hong, *et al*., "A CMFFP-tree algorithm to mine complete multiple fuzzy frequent itemsets," *Applied Soft Computing*, Vol. 28, 2015, pp. 431-439.
16. G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, Vol. 70, 2006, pp. 489-501.
17. G. B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, Vol. 6, 2014, pp. 376-390.
18. X. Zhao, X. Bi, G. Wang, *et al*., "Uncertain XML documents classification using extreme learning machine," *Neurocomputing*, Vol. 174, 2016, pp. 375-382.
19. Z. M. Ma and L. Yan, "Fuzzy XML data modeling with the UML and relational data models," *Data and Knowledge Engineering*, Vol. 63, 2007, pp. 972-996.
20. W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proceedings of KDD Workshop on Data Cleaning and Object Consolidation*, 2003, pp. 73-78.
21. D. Lin, "An information-theoretic definition of similarity," in *Proceedings of International Conference on Machine Learning*, 1998, pp. 296-304.
22. A. Algergawy, R. Nayak, and G. Saake, "Element similarity measures in XML schema matching," *Information Sciences*, Vol. 180, 2010, pp. 4975-4998.
23. J. Tekli, R. Chbeir, *et al*., "Approximate XML structure validation based on document-grammar tree similarity," *Information Sciences*, Vol. 295, 2015, pp. 258-302.
24. M. Szymczak, S. Zadrożny, A. Bronselaer, and G. D. Tré, "Coreference detection in an XML schema," *Information Sciences*, Vol. 296, 2015, pp. 237-262.
25. S. Agreste, P. D. Meo, E. Ferrara, and D. Ursino, "XML Matchers: approaches and challenges," *Knowledge-Based Systems*, Vol. 66, 2014, pp. 190-209.
26. H. Köpcke and E. Rahm, "Frameworks for entity matching: A comparison," *Data and Knowledge Engineering*, Vol. 69, 2010, pp. 197-210.
27. T. Dalamagas, T. Cheng, K. J. Winkel, and T. Sellis, "A methodology for clustering XML documents by structure," *Information Systems*, Vol. 31, 2006, pp. 187-228.
28. G. Peters and R. Weber, "DCC: A framework for dynamic granular clustering," *Granular Computing*, Vol. 1, 2016, pp. 1-11.
29. A. Skowron, A. Jankowski, and S. Dutta, "Interactive granular computing," *Granular Computing*, Vol. 1, 2016, pp. 95-113.



Zhen Zhao received the B.S. degree in Computer Science Education from Bohai University in 2002, and he obtained his M.S. degree in Computer Application Technology from Dalian Jiaotong University in 2009. He is currently working towards the Ph.D. degree in the field of Computer Science at Northeastern University. His research interests include fuzzy data management and machine learning.

Zong-Min Ma is currently a full Professor at Nanjing University of Aeronautics and Astronautics, China. He received his Ph.D. degree from the City University of Hong Kong, China. His research interests include databases, the Semantic Web, and knowledge representation and reasoning with a special focus on information uncertainty. He has published more than one hundred and seventy papers on these topics. He is also the author of four monographs published by Springer. He is a senior member of the IEEE.