Measuring the similarity of PML documents with RFID-based sensors

WANG Zhong-qin[1], YE Ning[1], Reza Malekian[2], ZHAO Ting-ting[1] ,WANG Ru-chuan[1]

[1] Institute of Computer Science, Nanjing University of Post and Telecommunications, Nanjing, China
[2]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa, 0002

Abstract

The Physical Mark-Up Language (PML) is to represent and describe data related to objects in EPC Network. The PML documents of each component to exchange data in EPC Network system are XML documents based on PML Core schema. For managing theses huge amount of PML documents, it is inevitable to develop the high-performance technology, such as compressing the amount of data, filtering and integrating these tag data. For above purpose, one of the effective methods is clustering, which could depend on the structure and semantics of these data. Indeed, the similarity computation, which measures the similarity of the compared PML documents, is the foundation of the clustering method. So in this paper, we propose an approach for measuring the similarity of PML documents based on Bayesian Network. With respect to the features of PML, while measuring the similarity, we firstly reduce the redundancy data except information of EPC. On the basis of this, the Bayesian Network model derived from the structure of the PML documents being compared is constructed. And this model has taken into consideration not only the EPC values contained in the PML but also their internal structure. Then the similarity between two PML documents could be deduced. Finally, the experiments evaluate the value range of similarity, timing result and the effectiveness of the similarity measure.

Keywords: Similarity, PML, Bayesian Network, EPC Network, RFID

1 Introduce

The Physical Mark-Up Language (PML) is a collection of common, standardized XML vocabularies to represent and describe information related to EPC Network enabled objects[1]. The documents of each component to exchange data in EPC Network system are XML documents based on PML Core schema, and this type document is called the PML document [2]. Among them, the purpose of the PML Core schema is to provide a standardized format for the exchange of the data captured by the sensors in an Auto-ID infrastructure, e.g. RFID readers. PML is to be regarded as the complementary vocabularies for business transactions or any other XML application libraries, which include a new library composed of relevant definitions about EPC Network system, rather than to replace the XML to be a new markup language.

EPC tags to identify each of objects are adapted by Auto-ID Center. Different sorts of sensors which equipped on shops, warehouses, workshops and so on [3], are to acquire EPC data and other information, such as temperature and geography location. It is essential for EPC network to process these data signed by PML documents at speed of hundreds of millions per second. For managing theses huge data stream and reduce network traffic, it is inevitable to develop the high-performance technology for managing these PML documents, such as compressing the amount of data, filtering and integrating these tag data.

For above purpose, one of the effective methods is clustering[4], which could depend on the structure and semantics of these data. Indeed, the similarity computation, which measures the similarity of the compared PML documents, is the foundation of the clustering method. In this paper, we mainly focus on the similarity computation of PML documents.

Many researches have proposed a wide range of algorithms for XML similarity computation, the kind of technique being used mainly include ED-based (Tree Edit Distance), IR-based (Information Retrieval) and others (e.g., edge matching, path similarity, etc.) to measure similarity of the XML documents.

Some of above methods to XML similarity mainly concern on the structural properties of XML data and disregard element/attribute values of XML[5], but many others consider values in their similarity computations. With respect to XML documents which are less structurally disparate (they might originate from the same data source, and might even conform to the same grammar), similarity computation based on structure and content is a favorable method [6]. As follow, we introduce algorithms of structure-and-content method.

Liang and Yokota[7] provided an approximate XML similarity method based on leaf nodes (leaf node values in particular), entitled LAX (Leaf-clustering based Approximate XML join algorithm). Kade and Heuser[8] develop a method for comparing XML documents as documents lists. Weis and Naumann[9] in put forward a method entitled Dogmatix for comparing XML elements (and consequently documents) based on their direct values, as well as corresponding parent and children similarities. An approach for document/pattern comparison, developed in the context of data integration and XML querying, is proposed by Dorneles et al. [10]. Leitao[11]provided a probabilistic approach, using a Bayesian network to combine the probabilities of children and descendents being duplicates, for a given pair of XML elements in the documents being compared. The similarity between two XML documents corresponds to the probabilities of their root nodes being duplicates.

From Leitao's study, we improve the method of XML Fuzzy Duplicate Detection proposed by Leitao in accordance with the features of PML document, and propose the method of measuring the similarity of PML documents based on Bayesian Network.

The remainder of the article is organized as follows. In Sec.2, we describe the background of PML documents and Bayesian network. In Sec.3, we present the Bayesian network for PML similarity computation, including the relationship between PML documents similarity and Bayesian network probability, redundancy reduction of PML documents, Bayesian network model for PML documents and the algorithm of constructing Bayesian network model, and elucidate how PML similarity measure is performed using the proposed Bayesian network. Section 4 presents our prototype and experimental tests. Section 5 concludes the paper and outlines future research directions.

2. Background

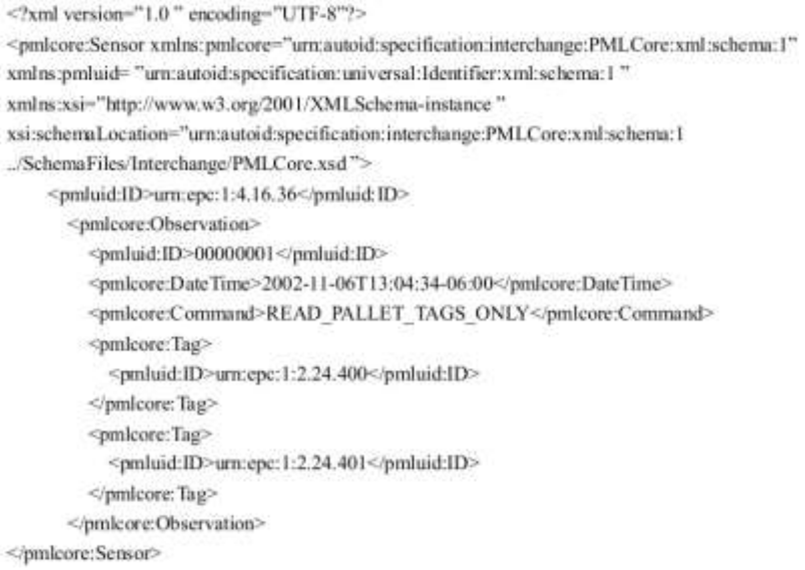
2.1 PML document

In order to stress the need for relatedness assessment in PML document comparisons, consider the example in Fig. 1 and Fig. 2. It depicts PML document of data captured by RFID readers. RFID readers capture the Electronic Product Code stored on the individual Auto-ID compliant tags (e.g. 1:2.24.404 and 1:12.8.128).

PML document should enable to elaborate the process of that RFID readers acquire data, including where is the certain RFID reader, which is identified by a unique identifier (e.g. 1:4.16.36), when certain tags in its read range are observed (e.g. 2002-11-06T13:04:34-06:00) and so on. Each such observation might need to be labeled with the command that was issued to trigger the observation (e.g. READ_PALLET_TAGS_ONLY) and a unique label to reference a certain observation (e.g. 00000001).

Within the EPC Network, RFID readers are one of the main components. The data they capture are routed within the EPC Network from readers to Savant[12] (the Savant is a middleware system which requests from upper application and receives data from sensors.) ,from one Savant to other, from Savant to the EPC Information Service. To standardize the mark-up of those captured data, PML document needs to adequately represent the observed values.

XML documents represent hierarchically structured information and can be modeled as Ordered Labeled Trees (OLTs) [13]. In the OLTs, nodes represent XML elements and are labeled with corresponding element tag names. Element attributes mark the nodes of their containing elements. Some studies have considered OLTs with distinct attribute nodes, labeled with corresponding attribute names[14]. Attribute nodes appear as children of their encompassing element nodes, sorted by attribute name, and appearing before all sub-element siblings[15]. So we reference the XML document's OLTs and describe the PML document's OLTs in Fig. 1 and Fig. 2. Element/attribute values are also considered in the comparison process following the application of structure-and-content.



```
<?xml version="1.0 " encoding="UTF-8"?>
<pmlcore:Sensor xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
xmlns:pmluid= "urn:autoid:specification:universal:Identifier:xml:schema:1 "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance "
xsi:schemaLocation="urn:autoid:specification:interchange:PMLCore:xml:schema:1
../SchemaFiles/Interchange/PMLCore.xsd ">
      <pmluid:ID>urn:epc:1:4.16.36</pmluid:ID>
        <pmlcore:Observation>
          <pmluid:ID>00000001</pmluid:ID>
          <pmlcore:DateTime>2002-11-06T13:04:34-06:00</pmlcore:DateTime>
          <pmlcore:Command>READ_PALLET_TAGS_ONLY</pmlcore:Command>
          <pmlcore:Tag>
            <pmluid:ID>urn:epc:1:2.24.400</pmluid:ID>
          </pmlcore:Tag>
          <pmlcore:Tag>
            <pmluid:ID>urn:epc:1:2.24.401</pmluid:ID>
          </pmlcore:Tag>
        </pmlcore:Observation>
</pmlcore:Sensor>
```

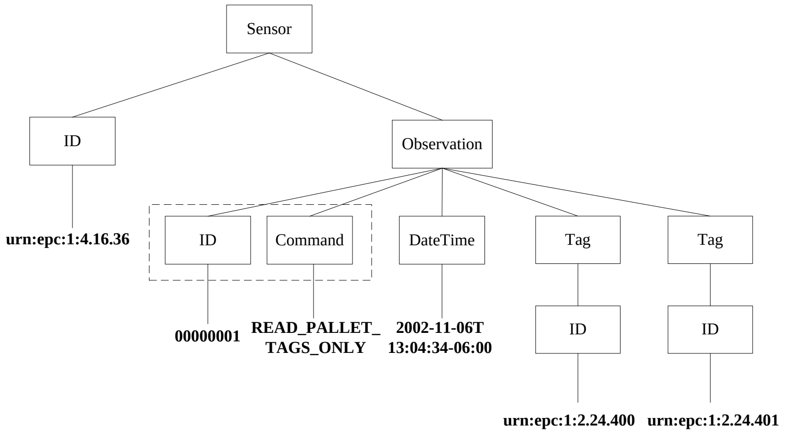
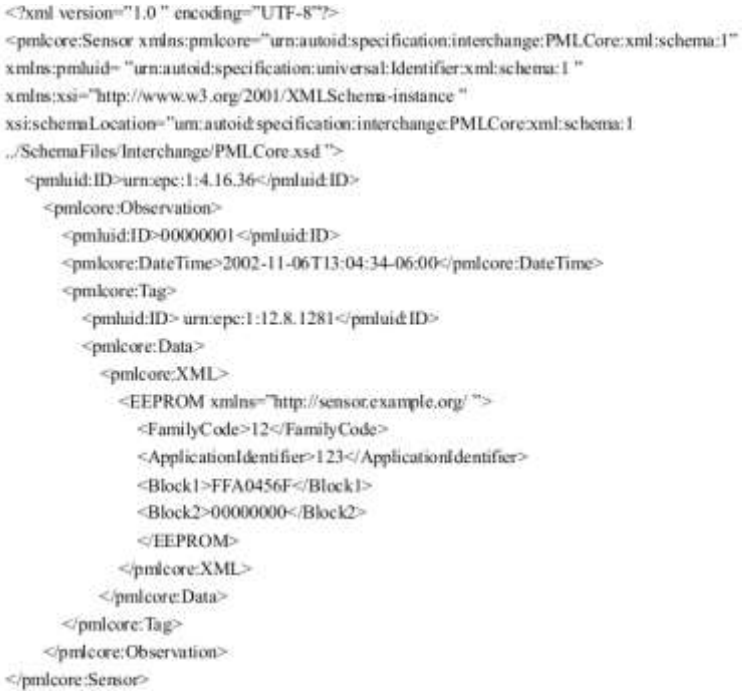


Fig. 1 PML document of tags captured by RFID readers and OTLs with its values



```
<?xml version="1.0 " encoding="UTF-8"?>
<pmlcore:Sensor xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
xmlns:pmluid= "urn:autoid:specification:universal:Identifier:xml:schema:1 "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance "
xsi:schemaLocation="urn:autoid:specification:interchange:PMLCore:xml:schema:1
../SchemaFiles/Interchange/PMLCore.xsd ">
   <pmluid:ID>urn:epc:1:4.16.36</pmluid:ID>
      <pmlcore:Observation>
         <pmluid:ID>00000001</pmluid:ID>
         <pmlcore:DateTime>2002-11-06T13:04:34-06:00</pmlcore:DateTime>
         <pmlcore:Tag>
            <pmluid:ID> urn:epc:1:12.8.1281</pmluid:ID>
            <pmlcore:Data>
               <pmlcore:XML>
                  <EEPROM xmlns="http://sensor.example.org/ ">
                     <FamilyCode>12</FamilyCode>
                     <ApplicationIdentifier>123</ApplicationIdentifier>
                     <Block1>FFA0456F</Block1>
                     <Block2>00000000</Block2>
                     </EEPROM>
                  </pmlcore:XML>
               </pmlcore:Data>
         </pmlcore:Tag>
      </pmlcore:Observation>
</pmlcore:Sensor>
```

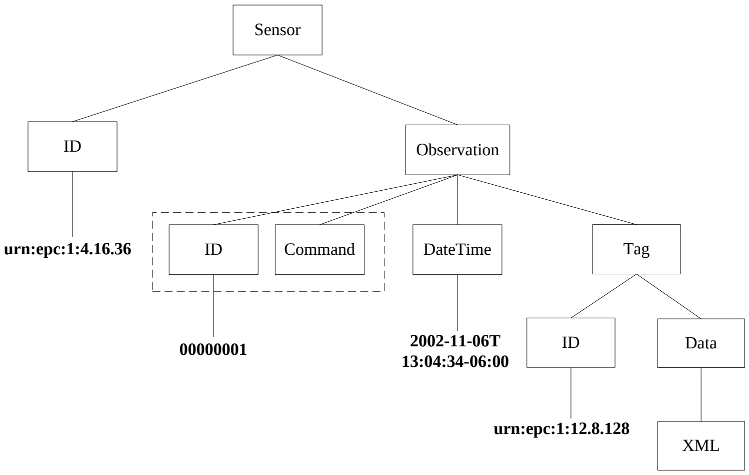


Fig. 2 PML document of tags with data captured by RFID readers and OTLs with its values

2.2 Bayesian network

Bayesian networks (BNs) provide a graphical formalism to explicitly represent the dependencies among the variables of a domain, thus providing a concise specification of a joint probability distribution [16][17]. The network structure of the Bayesian network (belief networks or Bayes nets for short), belonging to the family of probabilistic graphical models (GMs), is an DAG (Directed Acyclic Graph), where each node represents an attribute or data variables and the arcs represent the probabilistic dependency relation between attribute nodes. The relationship of complex variables in specific issues is represented by a network structure, reflecting dependency relationship between variables in the problem areas. In addition to the DAG structure, which is often considered as the “qualitative” part of the model, one needs to specify the “quantitative” parameters of the model. The parameters are described in a manner which is consistent with a Markovian property, where the conditional probability distribution (CPD) at each node depends only on its parents[18]. A mathematic model is used to express Bayesian network as follows:

$$B = (V, E, P) \tag{1}$$

The set of collection of random variables is defined as:

$$V = \{V_1, V_2, \ldots V_n\} \tag{2}$$

The collection of directed edges is defined as:

$$E = \{V_i V_j \mid V_i, V_j \in V\} \tag{3}$$

The set of Conditional probability distribution, namely Conditional probability table is defined as:

$$P = \{P(V_i \mid V_1, V_2, \ldots, V_{i-1}), V_i \in V\} \tag{4}$$

Consider the following example that illustrates some of the characteristics of BNs. The example shown in Fig. 3 presents the Bayesian network of two PML documents being the rooted node of sensor, which have the same data structure but different value. Firstly, it considers Tag similarity, represented by the variable Tag (denoted by ST) might result from ID' similarity，represented by the variable ID' (denoted by SI'). Secondly, Observation similarity represented by the variable Observation (denoted by SO) might result from DateTime similarity represented by the variable DateTime (denoted by SD). In the final case, it is reasonable to assume that sensor similarity represented by the variable Sensor (denoted by SS) will be determined by SO and ID similarity, represented by the variable ID (denoted by SI). All variables are binary; thus, they are either true (denoted by "T") or false (denoted by "F").

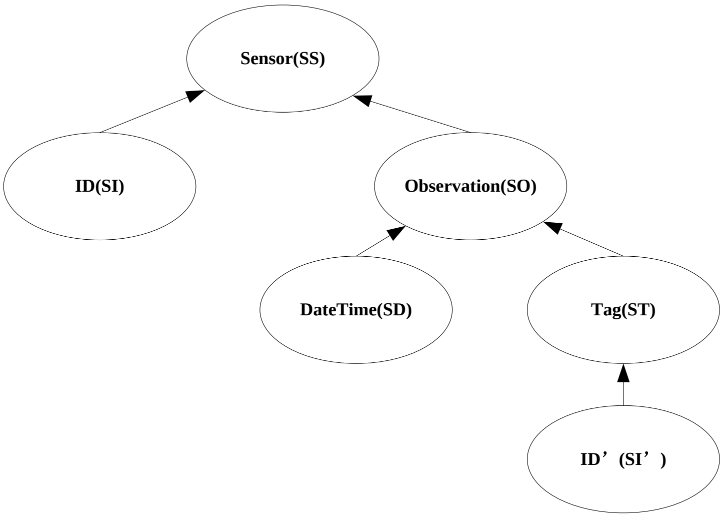


Fig. 3 Bayesian network of two sensors

For example, the CPTs of Tag and ID' are listed in Table 1 and Table 2. P(SI') presents the same probability of two ID's and P(ST) presents the same probability of two Tags.

Table 1 same probability of two ID's

P(SI'=F)	P(SI'=T)
0.5	0.5

Table 2 same probability of two Tags with the same probability of two ID's

SI'	P(ST=F)	P(ST=T)
F	1	0
T	0	1

From total probability formula,

$$\mathrm{P}(B) = \sum_{i=1}^{n} P(A_i) P(B|A_i) \tag{5}$$

We enable to demonstrate that the different probability of two tags P(ST=T) is 0.5, while the probability is defined as:

$$\begin{aligned} \mathrm{P(ST = T) = P(ST = T|SI' = T)P(SI' = T) + P(ST = T|SI' = F)P(SI' = F)} \\ = 1 \times 0.5 + 0 \times 0.5 \\ = 0.5 \end{aligned} \tag{6}$$

Then we have:

$$\mathrm{P(ST = F) = 1 - P(ST = T)} \tag{7}$$

Similarly, by applying Eq. (1), probability P (SS=T) is defined as:

$$\begin{aligned} \mathrm{P(SS = T) = P(SS = T|SI = T)P(SI = T) + P(SS = T|SI = F)P(SI = F)} \\ \mathrm{+ P(SS = T|SO = T)P(SO = T) + P(SS = T|SO = F)P(SO = F)} \end{aligned} \tag{8}$$

And we have:

$$\mathrm{P(SS = F) = 1 - P(SS = T)} \tag{9}$$

3. Bayesian network for PML similarity computation

3.1 Redundancy reduction of PML documents

The aim of this phase is to reduce the redundant nodes in the original tree before construction of Bayesian network.

After researching the PML Core specification defined in 'PMLCore.xsd' XML schema file, we know that the rooted element Sensor is main comprised of two subordinate ID element and Observation element. And the Observation element consists of the following:

- an optional ID element
- an optional Command element
- DateTime element
- zero or more Data elements
- zero or more Tag elements

Among them, the Tag element consists of the following elements:

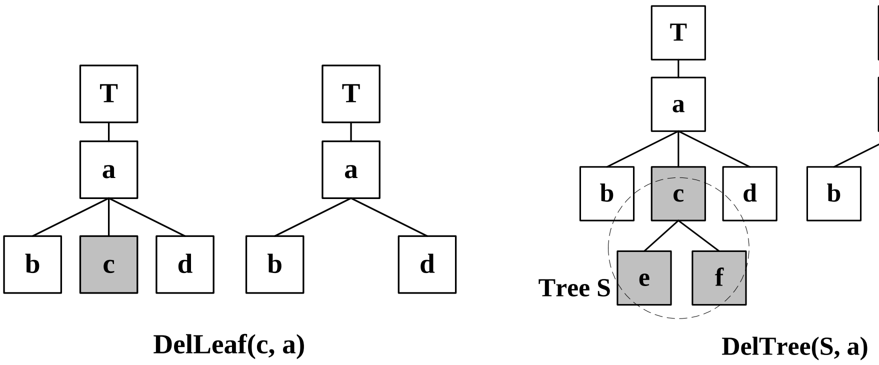
- ID element
- optional Data element
- zero or more Sensor elements

A sensor is considered any device that makes measurements and observations, such as an RFID reader or a temperature sensor. As mentioned earlier, each of objects, including different sensors, have the unique ID, namely EPC, to identify their information in EPC network. EPC regarded as a point enable to inquiry and retrieve information from supply chains. In the paper, we mainly concern on the similarity of PML documents rather than the concrete information that each PML document contain. For example in Fig. 2, the information stored in tag EEPROM is not important for PML comparison similarity. If the client wants to acquire these data, they enable to receive the EPC by RFID reader, which finding IP address to get the object information stored in EPC IS from internet. So redundancy is the data in addition to be able to identify EPC, including ID, Command, DateTime and Data in Observation element and Data , Sensor in Tag element, only retaining the ID in Tag element.

Redundancy reduction of tree deletion operations between two rooted ordered labeled trees that represent two PML documents are defined as follows:

(1) Given a leaf node x and a tree T, T containing node p with first level sub-trees and x being the ith child of p, e.g. {P_1,…, P_{i-1},x,P_{i+1},…,P_m}, DelLeaf(x, p) is the deletion operation applied to node p that yields x with first level sub-trees {P_1,…, P_{i-1},P_{i+1},…,P_m} (Fig.4).

(2) Given a sub-tree A and a tree T, T containing node p with first level sub-trees, e.g.{P_1,…,P_{i-1},A, P_{i+1},…,P_m}, DelTree(A, p) is the deletion operation applied to node p that deletes sub-tree A in T from among the children of p {P_1,…, P_{i-1},P_{i+1},…,P_m} (Fig. 4).



DelLeaf(c, a)

DelTree(S, a)

Fig. 4 Delete leaf node and Delete sub-tree

In our model, we first simplify a PML tree using the algorithm Pred. The description of algorithm is as follows.

```
Algorithm: Pred（PMLtree T）
    foreach node Ni in NodeList do
        if Ni==Observation then
            foreach childnode t of Observation do
                if $t \in$ {ID, Command, DataTime} then
                    DelLeaf(Observation, t);
                else if t==Tag then
                        foreach childnode s of Tag do
                            if s==Data then
                                DelTree(Tag, Data);
```

The input of the algorithm is a PML tree, as shown in Fig.1 and Fig. 2. We assume that all nodes are stored in a dynamic list NodeList in accordance with the gradation in the tree. And the parent-child relationship between the nodes is also shown in the list. The algorithm traverses the list NodeList and at the same time, using the functions of DelLeaf and DelTree to respectively delete the redundant nodes and subtrees. Definition of DelLeaf (c, a) is to delete a leaf node c that is eligible for deleting and is parented at node a. What's more, DelTree (S, a) is used to delete a eligible subtree S that is parented at a.

The result of the algorithm is to obtain a new NodeList made of the remaining nodes by the way of deleting those redundant nodes and subtrees. Of course, the deleting operation will not change the original gradation relationship. The output is shown in Fig.5.

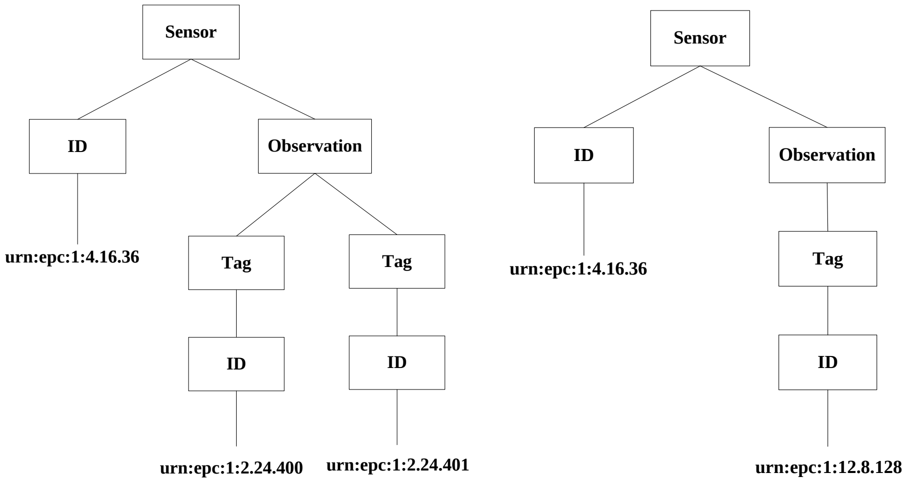


Fig. 5 Result of redundancy reduction from PML documents in Fig.1 and Fig. 2

3.2 Bayesian network model for PML documents

For measuring the similarity of PML documents, we construct a Bayesian network model as illustrated in Fig. 6. The network model has a rooted node labeled Sensor representing the possibility of node sensor in two compared PML trees. If a tree A and tree B is two compared trees, the node Sensor represents the possibility of node Sensor in tree A being a duplicate of the node Sensor in tree B. The probability of the Sensor nodes being duplicates depends on the probability of each pair of children nodes being duplicates. Then the node ID represents the possibility of node ID in tree A being a duplicate of the node ID in tree B; node Observation represents the possibility of node Observation in tree A being a duplicate of node Observation in tree B. Similarly, we enable to repeat the process of other two nodes.

However, it is a slightly different procedure of PML nodes labeled ID of the children of Tag node. In this case, we wish to compare the full set of nodes, instead of each node independently. In this case, the set of ID nodes of the children of Tag nodes being duplicate depends on each ID node in tree A being a duplicate of any ID node in tree B. It is presented by nodes ID_{M*N}, ID_{MN} and ID_{in} in Fig. 6. Because the nodes ID_{in} have no children, their probability of being duplicates only depends on their values ID_{in}[Value].

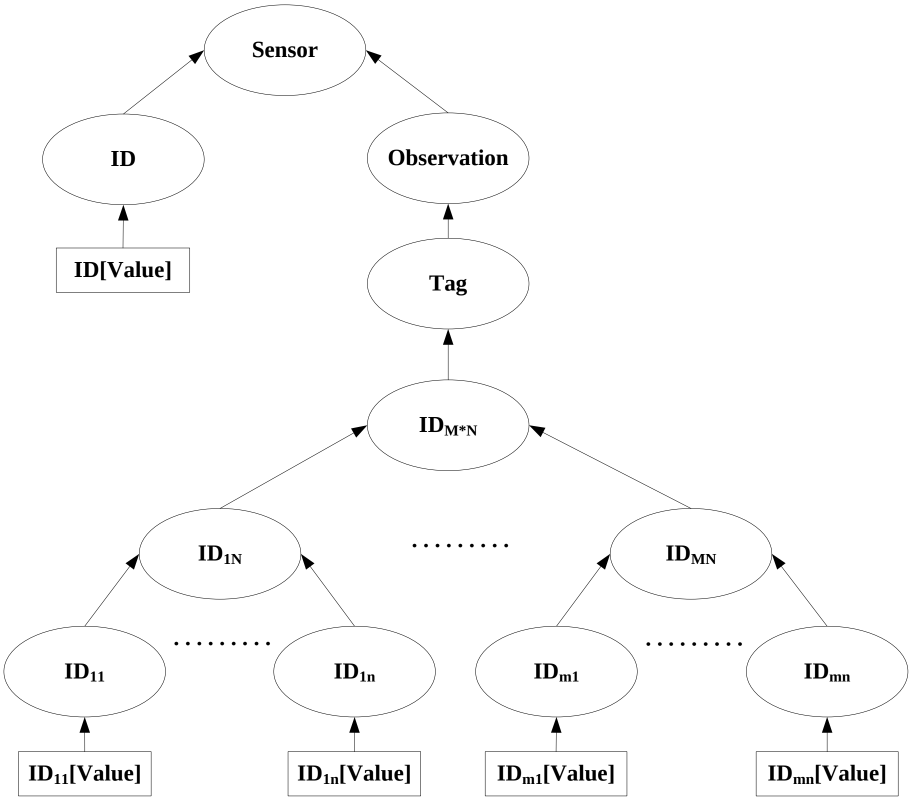


Fig. 6 Bayesian network model

We know that elements of Sensor, ID and Observation are contained in each of PML document from the PML Core schema. And the probability of the two PML nodes being duplicates depends on (1) whether or not their values of nodes are duplicates, and (2) whether or not their children of nodes are duplicates. The node is assigned a binary random variable. If a node exists in the same location of two PML trees, this variable takes the value 1 to present. Otherwise, the variable takes the value 0 to express.

With respect to the Bayesian network model, we could compute the probability in Fig. 1 and Fig. 2. Three types of conditional probabilities are defined as follows:

(1) The probability of the values of the nodes being duplicates depends on each individual pair of values being duplicates;

(2) The probability of two nodes being duplicates depends on their values and their children being duplicates or each pair of children nodes being duplicates (i.e. Sensor).

(3) The probability of a set of nodes of the same type being duplicates depends on each pair of individual nodes in the set are duplicates.

In our example, these two types of conditional probabilities correspond to the respective probabilities listed in Tab. 3 (a), (b) and (c).

Table 3 Conditional Probabilities

Conditional Probability
P (ID| ID [Value])
P (ID_{mn} |ID_{mn}[Value])

(a)

Conditional Probability
P (Tag| ID_{M*N})
P (Observation| Tag)
P (Sensor| ID, Observation)

(b)

Conditional Probability
P (ID_{iN}| ID_{i1},…, ID_{in})
P (ID_{M*N}| ID_{1N},…, ID_{MN})

(c)

3.3 The algorithm of constructing Bayesian network model

In this paper, a PML tree is defined as a triple T=(S, V, W), where

- S is a root node label, e.g., for tree T in Fig.5, S=Sensor.
- V is a set of (attribute, v) pairs, where v is the value of this node. If the node itself has a value, we define it as a special (attribute, v) pair. For tree T in Fig.5, we have
- W is a set of PML trees, means that W is the set of subtrees of T. These subtrees are again each described as a triple. For tree T in Fig.5, W contains subtree rooted at observation.

```
Algorithm: Merg（PTree T，PTree T'）
Input: T=(S, V, W)
        T'=(S', V', W')
Output: A directed graph G=(N,E)
/* ------------- Initialization -------------- */
X= Y =0;
/* ------------------------------------------ */
if S==S' then
    Insert a node S into N;
    if $V \cup V' \neq \varnothing$ then
        if V == V' then
            Insert a node V into N;
            Insert an edge into E from this node to S;
            Insert a node v into N;          // v represents value.
            Insert an edge into E from this node to V;
            Insert a node v' into N;
            Insert an edge into E from this node to V;
    if $W \cup W' \neq \varnothing$ then
        foreach $W_i \in W$ do
            foreach $W_j' \in W'$ do
            if $W_i \neq$ Tag and $W_j' \neq$ Tag then      // None of them owns a Tag.
                R=S;
                G'=（N'，E'）←Merg（$W_i$，$W_j'$）
                foreach node n $\in$ N' do
                    Insert n into N;
                foreach edge e $\in$ E' do
                    Insert e into E;
                foreach node n $\in$ N' without outgoing edges do
                    Insert an edge into E from this node to R;
            else          // Any of them owns at least a Tag.
                  Insert a node Tag into N;
                  Insert an edge into E from this node to S;
                  if $W_i$==Tag then
                      X++;
                  if $W_j'$==Tag then
                      Y++;
        Insert a node $ID_{X*Y}$ into N;
        Insert an edge into E from this node to Tag;
        foreach Tag $t_i(1\leq i\leq X) \in$ W do
              P=ID value of Tag $t_i$;
              Insert a node $ID_{i*Y}$ into N;
              Insert an edge into E from this node to $ID_{X*Y}$;
              foreach Tag $t_j(1\leq j\leq Y) \in$ W' do
                      Q=ID value of Tag $t_j$;
                      Insert a node $ID_{i*j}$ into N;
                      Insert an edge into E from this node to $ID_{i*Y}$;
                      Insert a node P into N;
                      Insert an edge into E from this node to $ID_{i*j}$;
                      Insert a node Q into N;
                      Insert an edge into E from this node to $ID_{i*j}$;
```

The idea of designing Algorithm is to merge two PML trees into one tree, which is starting from the root nodes. We assume that two trees can only be merged in the case of that the root nodes are the same. In our example, the root nodes are the identical S=Sensor, while V has only one element (ID, value). And W is the subtree rooted at Observation. There are two variables, X and Y, which are respectively used to store the number of Tags in the two trees. It is clear that they are initialized as Null.

In this algorithm, we define the structure of the input PML tree, as described above, a triple. The algorithm takes as input two sets of PML trees T and T'. We only deal with the case of root node S=S', otherwise we will exit the algorithm and the output is Null. In the former case, we judge V = V' whether to set up. Under the condition of V = V', we respectively construct new nodes named as the values v of elements of V and V'. After that the function will construct a new edge pointed to the root node. In the subtree, we recursively invoke the merging function Merg. In the case of meeting with Tag, it is inevitable to make a one-to-one comparison for which requires a new node

$\mathrm{ID}_{i*j.}$ So it is necessary to generate nodes with the number of X*Y. The result of this algorithm is a directed graph G=(N,E), where N is the set of nodes in G while E represents the set of edges between these nodes. This graph is initialized as NULL. When applying this algorithm to the PML tree T and T' of Fig. 5, we can obtain the directed graph in Fig. 6.

3.4 Defining the probabilities

As illustrated in previous section, we describe how to construct the Bayesian network model, so we need to define the conditional probabilities to inner nodes and prior probabilities to leaf nodes. Here we also define the notion P(x) to mean P(x=1), presenting the probability of two same nodes occurring at the same time.

3.4.1 Conditional probabilities

Conditional Probability CP1: CP1 denotes that the probability of the values of the nodes being duplicates depends on each individual pair of values being duplicates. In this case, we enable to define P ($\mathrm{ID}_{t_{ij}}$ | $t_{ij}[n_1]$, $t_{ij}[n_2]$,…) to correspond to above presentation, where $\mathrm{ID}_{t_{ij}}$ is a leaf node ID of parent node t_{ij}, $t_{ij}[n]$ the value of attribute n of the i-th node with tree t in the PML tree.

If all values of attribute n are duplicates, we consider that the value of leaf node ID of parent node t_{ij} as duplicates, and this value represents the importance of the corresponding attribute in determining whether the nodes are duplicates. For instance, if the attribute ID_{11}[Value] is equal to 1, then we consider the leaf node ID_{11} values are duplicates.

This definition is represented in Eq. (10), and we determine that the probability of the PML nodes being duplicates equals a given value, w.

$$\mathrm{P}\ (\mathrm{ID}_{t_{ij}} \mid t_{ij}[n_1],\ t_{ij}[n_2],\ldots)= \sum_{1\le k\le n,\, t_{ij}[a_k]=1} w_{a_k} \tag{10}$$

Subject to $\sum_{1\le k\le n} w_{a_k}=1$

In this case, since all of leaf nodes have only an attribute value, Equation (11) is represented as follows:

$$\mathrm{P}\ (\mathrm{ID}_{t_{ij}} \mid t_{ij}[n_1])= \sum_{1\le k\le n,\, t_{ij}[a_k]=1} w_{a_k} =1 \tag{11}$$

For instance, P(ID| ID[Value])=1 and P(ID_{11}| ID_{11}[Value])=1.

Conditional Probability CP2: CP2 denotes that the probability of two nodes being duplicates depends on their values and their children being duplicates or each pair of children nodes being duplicates. i.e., P(Sensor| $\mathrm{ID}_{\mathrm{Sensor}}$, $\mathrm{Ob}_{\mathrm{Sensor}}$), if both ID and Observation values and their children are duplicates, we could consider the nodes as duplicates. So this definition is represented in Eq. (12).

$$P(t_{ij} \mid ID_{t_{ij}}, Ob_{t_{ij}}) = \begin{cases} 1 & iff\ \ ID_{t_{ij}} = Ob_{t_{ij}} = 1 \\ 0 & Otherwise \end{cases} \tag{12}$$

Conditional Probability CP3: CP3 denotes that the probability of a set of nodes of the same type being duplicates depends on each pair of individual nodes in the set are duplicates, i.e., P (ID_{M*N}|ID_{1N}, ID_{2N},...) and P (ID_{1N}|ID_{11}, ID_{12},...), the set of nodes ID depends on that each of its nodes is a duplicate. We also assume that the more nodes ID are duplicates, the higher the probability that the whole set of nodes is a duplicate. So this definition is represented in Eq. (13.

$$P(t_{M*N} \mid t_{1*N}, t_{2*N}, ...) = \frac{1}{n}\sum_{k=1}^{n} t_{kN} \tag{13}$$

And the probability P (ID$_{1N}$|ID$_{11}$, ID$_{12}$,...), which reflects the fact that a node ID in an PML tree is a duplicate if it is a duplicate of at least one node of the same type in the other PML tree. This is represented in Eq. (14).

$$P(t_{iN} \mid t_{i1}, t_{i2}, ..., t_{iN}) = \begin{cases} 1 & iff\ \exists_j \mid t_{ij} = 1 \\ 0 & Otherwise \end{cases} \tag{14}$$

3.4.2 Prior probabilities

Note that the P(t$_{ij}$[n]) can be defined based on the similarity between values, the higher probability, the more the similarity they are. For instance, the probability of the ID attributes in two Sensor elements being the same can be similar between both ID nodes. We normalize this similarity to a value between 0 and 1. Thus, we define

$$P(t_{ij}[n]) = \begin{cases} sim(ID_i[n], ID_j[n]) & if\ similarity\ was\ measured \\ 1 - sim(ID_i[n], ID_j[n]) & Otherwise \end{cases} \tag{15}$$

Where sim (·) is a similarity function, normalized to fit between 0 and 1.

For instance, for the ID attribute in the Sensor nodes, we can define sim(ID, ID’)=1 if ID[Value]= ID’[Value], and otherwise sim(ID, ID’)=0.

3.4.3 Finally probability

All conditional and prior probabilities are defined, so we could depend on the knowledge of Bayesian network to compute the probability of two PML trees. And the Bayesian network model has been described in sec. 3.2. According to the network, and applying Eq. (12), the probability is defined as:

$$\begin{aligned} P(Sensor) &= \sum_{ID,Ob} P(Sensor \mid ID_{Sensor}, Ob_{Sensor}) P(ID_{Sensor}, Ob_{Sensor}) \\ &= \sum_{ID,Ob} P(Sensor \mid ID_{Sensor}, Ob_{Sensor}) P(ID_{Sensor}) P(Ob_{Sensor}) \\ &= P(ID_{Sensor}) P(Ob_{Sensor}) \end{aligned} \tag{16}$$

Similarly, by applying Eq. (10), probability P (ID$_{Sensor}$) is defined as:

$$\begin{aligned} P(ID_{Sensor}) &= P(ID_{Sensor} \mid ID_{Sensor}[value]) P(ID_{Sensor}[Value]) \\ &= w_{value} P(ID_{Sensor}[Value]) \\ &= 1 \times P(ID_{Sensor}[Value]) \\ &= P(ID_{Sensor}[Value]) \end{aligned} \tag{17}$$

Since w$_{value}$ = 1, according to Eq. (10).

As for probability P (Ob$_{Sensor}$), according to Eq. (14), we have:

$$\begin{aligned} P(Ob_{Sensor}) &= P(Tag_{Ob}) = P(ID_{M*N}) \\ &= \frac{P(ID_{1N}) + ... + P(ID_{MN})}{M} \end{aligned} \tag{18}$$

Using Eqs.(12) and (10) we can compute probability P(ID_{1N}) as:

$$P(ID_{1N}) = 1 - \prod_{i=1}^{n}(1 - P(ID_{1i}[Value])) \qquad (19)$$

A similar equation can be obtained from P (ID_{2N}) to P (ID_{MN}).
Finally, join Eqs. (16) through (19), we have:

$$P(Sensor) = P(ID_{Sensor})P(Ob_{Sensor}) \qquad (20)$$

$$= P(ID_{Sensor}[Value]) \times \frac{1 - \prod_{i=1}^{n}(1 - P(ID_{1i}[Value])) + ... + 1 - \prod_{i=1}^{n}(1 - P(ID_{mi}[Value]))}{M}$$

4. Experiment

We measure the PML similarity in terms of timing results and effectiveness on data, which is followed XML Schemas of PmlCore.xsd and Identifier.xsd and is generated randomly by PML generator. Our evaluation covers (1) timing result for various sizes of PML documents, (2) the impact of various sizes of PML documents on effectiveness, and (3) the impact of various sizes of the same elements in PML documents on effectiveness.

4.1 Data Sets

We use four different data sets.

- Data Set 1: 500 random PML documents.
- Data Set 2: 20% of the same 500 PML documents.
- Data Set 3: 50% of the same 500 PML documents.
- Data Set 4: 80% of the same 500 PML documents.

And theses data sets are extracted from PML data generator designed by our project team, which enable to generate different PML documents in accordance with our needs.

In the generator, the parameters of self-definition include (1) amount of Tag element, (2) type of Tag element, and (3) value of ID element. Hence, Dataset 1 represents the scenario where we don’t understand the structure and duplicate of PML documents, and all of theses PML documents are randomly generated. Dataset 2, 3, 4 are used to show the impact of different degree of duplicates to our algorithm’ timing result and effectiveness.

4.2 Computing Environment

These tests were done on a Thinkpad X220i computer with dual processor CPU of Core i3 2370M Processors, running at 2.4 GHz. All experiment approaches, include measure of timing result and effectiveness, were implemented by us in Matlab. And we know that the timing results of algorithm could be influence by different computer

4.3 Experimental Setup

Firstly, we define the prior probability as follow.

$$P(ID_{ij}[Value]) = Sim(ID_i, ID_j) = 1 - \frac{Compare(ID_i, ID_j)}{Max(|\,ID_i\,|, |\,ID_j\,|)} \qquad (21)$$

Where Compare(ID_i, ID_j) presents the comparison of strings ID_i and ID_j and the result is the integer value of difference of two strings. |ID| is the length of string ID. So the result of Max(|ID_i |, | ID_j |) is the maximum value of two strings.

To measure effectiveness, we use the commonly used precision and recall[19]. Precision measures the percentage of correctly identified duplicates contained over the total set of objects determined as duplicates by the system. Recall measures the percentage of duplicates correctly identified by the system over the total set of duplicate objects [20].

4.4 Experiments

Experiment 1 to measure what the value of final probability could be considered duplicates for two PML documents by using Data Set 1.

Firstly, we should determine whether a distribution of statistics follows a normal distribution compared with the probability density function of normal distribution graph. The frequency histograms are constructed in Fig. 7 (a).

Secondly, a normal distribution could be verified by Fig. 7 (b). With the increase of Data Set, the discrete points close to the inclined straight line segments. So the conclusion is that the values of final probability approximate normal distribution.

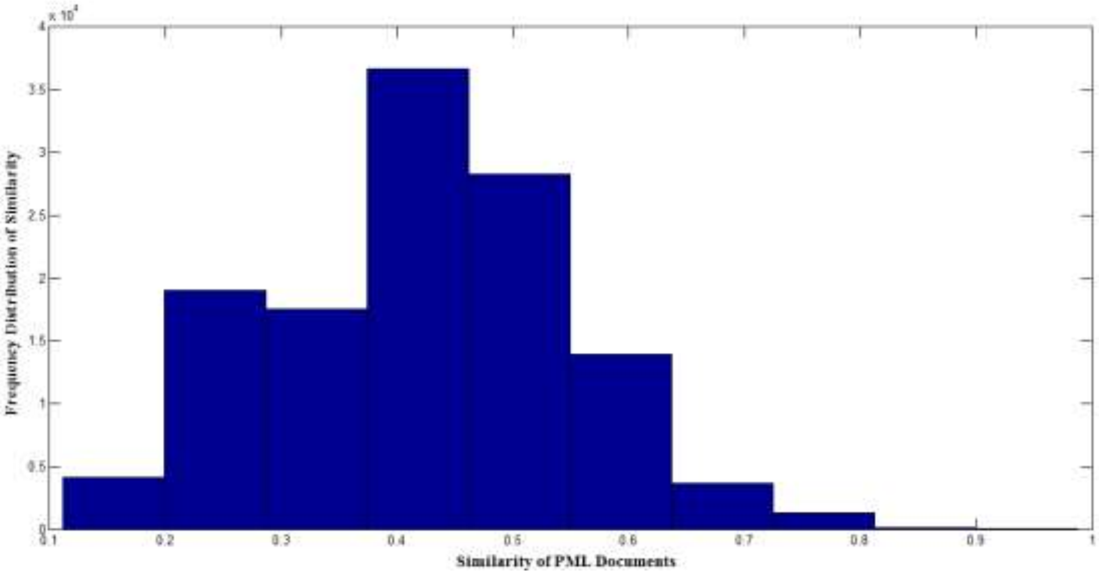


Fig. 7 (a) Frequency histogram

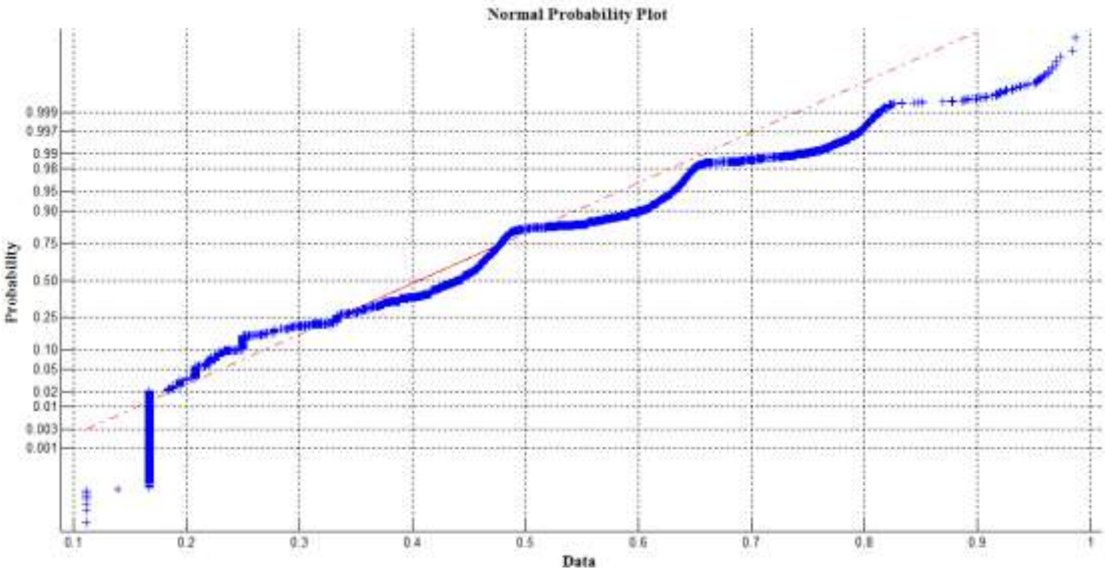


Fig. 7 (b) Distribution normality test

Finally, we perform three sets of random experiments, the average of the data show as follows.

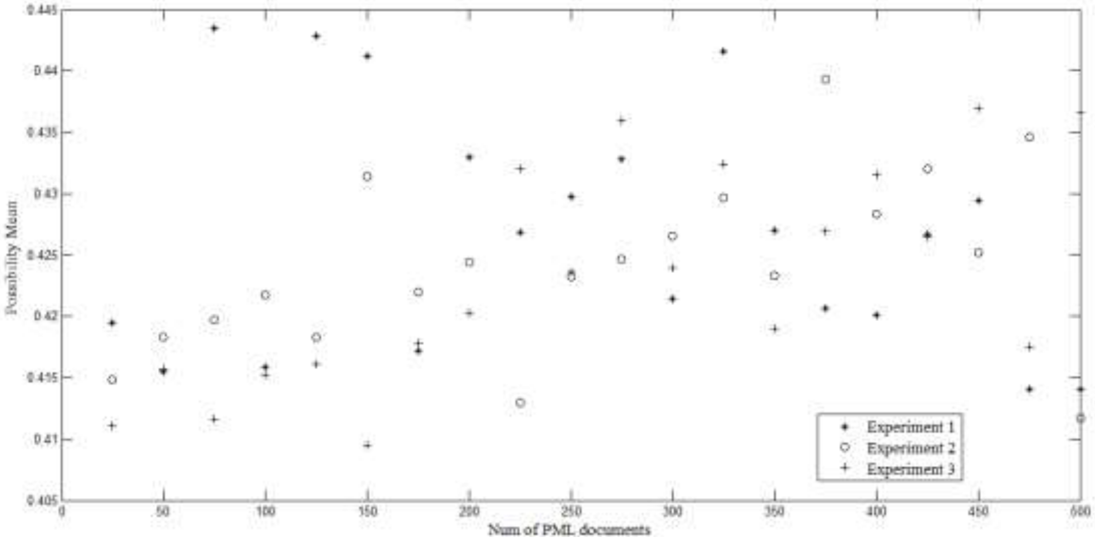


Fig. 8 Range of probability mean in three random experiments

From the Fig. 8, the average mostly concentrates in the [0.4095, 0.4435]. So only objects whose duplicate probability is above or equal to the value range [0.4095, 0.4435] are considered similarity.

Experiment 2 to evaluate the timing result of our algorithm in accordance to different scenarios by using Dataset 1-4. From the Fig. 9, the time to compare pairs of PML documents of various sizes grows in an almost perfect linear fashion with size and duplicate of PML documents.

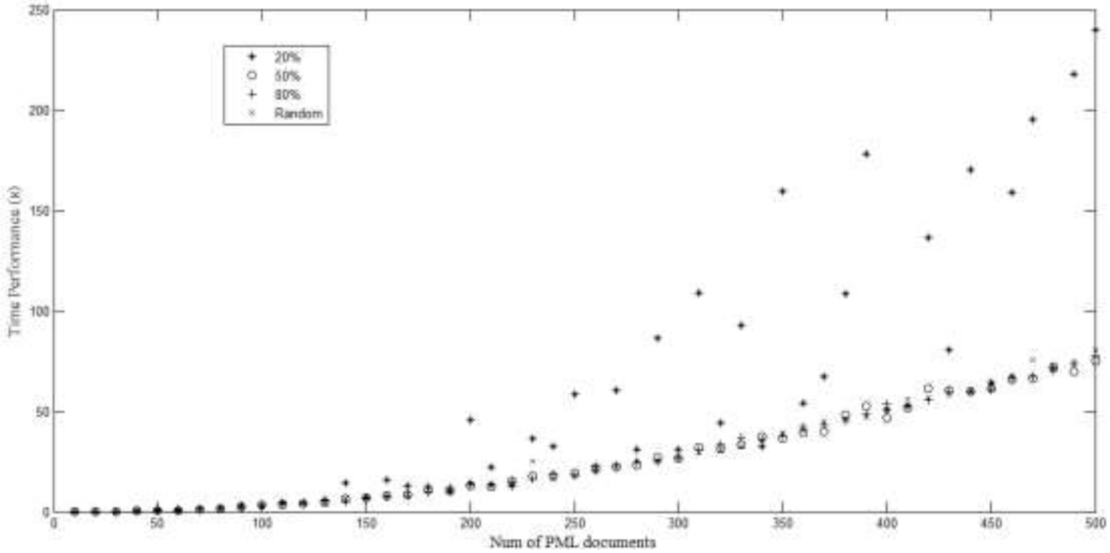


Fig. 9 Time performance for different amounts of duplicate data

Experiment 3 to evaluate the impact of various sizes and duplicates PML documents on effectiveness. The experiment was performed to determine the impact of the quality of the data being processed on the performance of the Bayesian network model. Fig. 10 shows the results for varying the probabilities of 20%, 50% and 80% respectively.

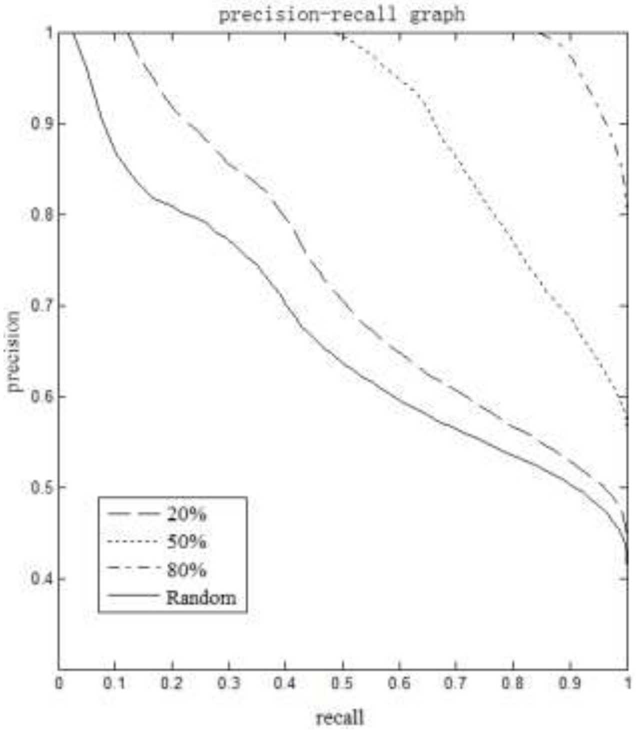


Fig. 10 Precision and recall values for different amounts of duplicate data

5. Conclusion

In this paper, we introduce the function and application area of PML documents and illustrate the necessity for computing the similarity of PML documents in EPC Network. Then we propose an approach for measuring the similarity of PML documents based on Bayesian Network. With respect to the feature of PML, while measuring the similarity, we firstly reduce the redundancy data except information of EPC. On the basis of this, the Bayesian Network model derived from the structure of the PML documents being compared is constructed. And this model has taken into consideration not only the ID values contained in the PML but also their internal structure. Then the similarity between two PML documents could be deduced. Finally, the experiments evaluate the value range of similarity, timing result and the effectiveness of the similarity measure.

We intend to further validate our similarity measures by considering Real-World Data, which could exist errors, such as missing data (e.g. lack of EPC) or incompleteness data (e.g. the EPC less than 96 bit) and so on, so we still need to validate this observation.

Another issue we should intend to consider is the scalability ether in space or in time. Scaling to large amounts of PML document with the help of external memory units also needs to be studied in the future.

6 Acknowledgment

The research is support by National Natural Science Foundation of P. R. China (Grant No. 61170065 and 61003039), Peak of Six Major Talent in Jiangsu Province (Grant No.2010DZXX026), Project sponsored by Jiangsu provincial research scheme of natural science for higher education institutions (Grant No.12KJB520009), Science & Technology Innovation Fund for higher education institutions of Jiangsu Province (Grant No.CXZZ11-0405).

References

[1] Floerkemeier C, Anarkat D, et al. PML core specification 1.0 [J]. Auto-ID Center Recommendation, 2003, 15.
[2] Clark S, Traub K, Council D A U C, et al. Auto-ID Savant Specification 1.0 2[J]. 2003.
[3] Brock D L, Milne T P, et al. The physical markup language [J]. Auto-ID Center White Paper MIT-AUTOID-WH-003, 2001.
[4] Dalamagas T, Cheng T, et al. A methodology for clustering XML documents by structure [J]. Information Systems, 2006, 31(3): 187-228.
[5] Manning C D, Raghavan P. Introduction to information retrieval [M]. Cambridge: Cambridge University Press, 2008.
[6] Tekli J, Chbeir R. An overview on XML similarity: background, current trends and future directions [J]. Computer science review, 2009, 3(3): 151-173.
[7] Liang W, Yokota H. LAX: An efficient approximate XML join based on clustered leaf nodes for XML data integration [J]. Database: Enterprise, Skills and Innovation, 2005: 82-97.
[8] A.M. Kade, C.A. Heuser. Matching XML documents in highly dynamic applicationsb, in: Proceeding of the 8th ACM Symposium on Document Engineering [C], DocEng'08, Brazil, 2008, pp. 191–198.
[9] M. Weis, F. Naumann. Dogmatix tracks down duplicates in XML [J], in: Proceedings of the ACM SIGMOD Conference, USA, 2005, pp. 431–442.
[10 C.F. Dorneles, C.A. Heuser, A.E.N. Lima, A.S. da Silva, E.S. de Moura, Measuring similarity between collections of values, in: Proceedings of the ACM international Workshop on Web Information and Data Management, USA, 2004, pp. 56–63.
[11] L. Leitao, P. Calado, M. Weis, Structure-based inference of XML similarity for fuzzy duplicate detection, in: Proceedings of the 16th ACM Conference on Information and Knowledge Management, CIKM'07, Portugal, 2007, pp. 293–302.
[12]Leong K S, Ng M L, Engels D W. EPC network architecture. Auto-ID labs research workshop. Zurich, Switzerland, 2004.
[13] WWW Consortium, The Document Object Model, http://www.w3.org/DOM.
[14] Z. Zhang, R. Li. Similarity metric in XML documents, in: Knowledge Management and Experience Management Workshop, Germany, 2003.
[15] A. Nierman, H.V. Jagadish. Evaluating structural similarity in XML documents, in: Proceedings of the 5th ACM SIGMOD International Workshop on the Web and Databases, WebDB, 2002, pp. 61–66.
[16] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference [M]. Morgan Kaufmann Pub, 1988.
[17]Silander T, Myllymaki P. A simple approach for finding the globally optimal Bayesian network structure [J]. arXiv preprint arXiv:1206.6875, 2012.
[18] Ben Gal I. Bayesian networks [J]. Encyclopedia of statistics in quality and reliability, 2007.
[19] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 233-240.
[20] Manning C D, Raghavan P, Schütze H. Introduction to information retrieval [M]. Cambridge: Cambridge University Press, 2008.