Semi-Automatic Multimedia Metadata Integration

Samir Amir, Ioan Marius Bilasco, Taner Danisman, Thierry Urruty and Chabane Djeraba
LIFL UMR CNRS 8022, University of Lille1, Telecom-Lille1
Villeneuve d'Ascq, France
{samir.amir, marius.bilasco, taner.danisman, thierry.urruty, chabane.djeraba}@lifl.fr

ABSTRACT

The recent growing of multimedia in our lives requires an extensive use of metadata for multimedia management. Consequently, many heterogeneous metadata standards have appeared. In this context, several integration techniques have been proposed in order to deal with this challenge. These integrations are made manually which are costly and time-consuming. This paper presents a new system for a semi-automatic integration of metadata which is done by using several types of information on metadata schemas.

1. INTRODUCTION

Multimedia resources play an increasingly pervasive role in our lives. Thus, there is a growing need to enable the management of such resources. This is the origin of the appearance of several metadata standards [3] which are heterogeneous data since they have been created by independent communities. In order to resolve the heterogeneity problem, several solutions have been proposed to integrate heterogeneous metadata. However, These solutions are performed by human experts, which is costly and time-consuming. Besides, the integration process must be updated every time a new standard appears. In this context, an intelligent metadata integration solution is needed to address the interoperability problem by providing an automatic system for mapping between metadata. To do so, tools and mechanisms must resolve the semantic and the structural heterogeneity and align terms between metadata where schema matching plays a central role. Among the schema matching approaches that have been experienced, we can highlight the success of the work done in [5] [6]. However both use of them consider only one type of context which makes them not efficient in the case where schemas to be matched have a high structural heterogeneity. In this paper a new schema matching-based approach for XML metadata integration is proposed. In particular, we propose a new matching technique which exploits the semantic and structural information in a manner that increases the matching accuracy.

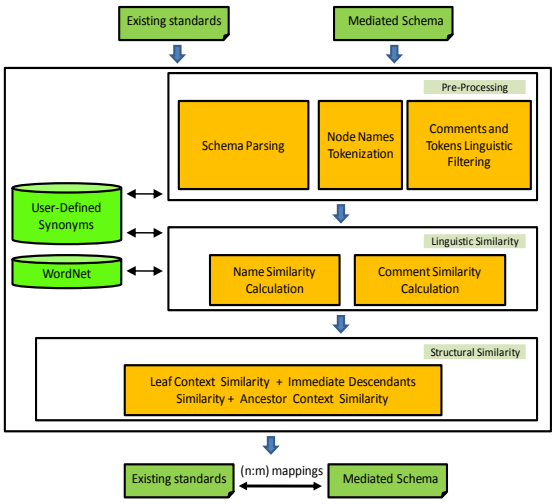

Figure 1: Matching process phases

2. THE PROPOSED APPROACH

In this section, we describe the different steps of the proposed matching system as shown in Figure 1: *pre-processing*, *linguistic* and *structural similarity computation*.

2.1 Pre-Processing

After modeling XML Schema as a directed labeled graph, we start by parsing all entities involved in the matching process (element, attributes and comments corresponding to these entities). Then, these entities are filtered and normalized using tokenization, lemmatisation and stopword list.

2.2 Linguistic Similarity Computation

This phase is concerned with the linguistic similarity computation between every XML Schema node pairs using their similarity names and comments.

2.2.1 Names Matching

We calculate the similarity distance between all node pairs in the two schemas. We first start with the explicitation of tokens by using WordNet. Each node n_i represented by a set of tokens M_i will have a set of synonyms *synset* for each token m_i. $M_i^{'}$ is the final result that regroups all synsets returned by M_i explicitation.

$$M_i^{'} = M_i \bigcup \{m_k | \exists m_j \in M_i \bigcap m_k \in synset(m_j)\} \quad (1)$$

We compute the similarity S_{name} between all node pairs. To do so, for each node pair (n_1, n_2) we calculate S_{name} by using Jaro-Winkler metric (JW) [1] between each token $m_i \in M_1^{'}$ and all tokens $m_j \in M_2$ (and vice versa). We take the maximum score (MJW) for each token m_i. Finally, the average of the best similarities is calculated:

$$S_{name}(n_1, n_2) = \frac{\sum_{m_i \in M_1} MJW(m_i, M_2^{'}) + \sum_{m_j \in M_2} MJW(m_j, M_1^{'})}{|M_1| + |M_2|} \quad (2)$$

2.2.2 Comments Matching

We apply the tf/idf to calculate the similarity between comments. To do so, all comments on two schemas are considered as documents, each node will be represented by a vector whose coordinates are the results of tf/idf. Hence, the similarity between two nodes is the distance between vectors corresponding to their comments. Let us consider $v = (w_1, w_2, ..., w_P)$, a vector representing a certain node n. $P = |U|$ is the number of distinct words in all comments in two schemas. The i_{th} element w_i in the vector v, which represents the node n in a schema, is calculated as follow:

$$w_i = tf_i * idf_i \quad idf_i = \log_2 \frac{N}{b_i} \quad (3)$$

where tf_i is the term frequency. tf_i represents the number of times that the i_{th} word in U appears in the comment corresponding to n_i. idf_i is the inverse of the percentage of the concepts which contain the word w_i. N is the number of comments in U in both schemas. b_i is the number of comments which contain the word w_i at least one time. The similarity $S_{comment}$ is the distance between the vectors.

$$S_{comment}(v_i, v_j) = \frac{\sum_{k=1}^{P} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{P} (w_{ik})^2 * \sum_{k=1}^{P} (w_{jk})^2}} \quad (4)$$

The result of above processes is a linguistic similarity matrix *lSim*:

$$lSim(n_i, n_j) = \mu_1 * S_{name}(n_i, n_j) + \mu_2 * S_{comment}(n_i, n_j) \quad (5)$$

where $\mu_1 + \mu_2 = 1$ and $(\mu_1, \mu_2) \geq 0$

2.3 Structural Similarity Computation

Linguistic similarity computation may provide several false positive candidates. Thus, in order to eliminate the false candidates, the structural similarity is computed by considering three kinds of nodes contexts [4]: *ancestor context*, *immediate descendant context* and *leaf context*.

2.3.1 Ancestor Context

The ancestor context of a node n_i is defined as the path p_i extending from the root node of the schema to n_i. The ancestor context similarity *ancSim* between (n_i, n_j) is based on the resemblance measure between their paths (p_i, p_j). This is done by calculating three scores established in [2].

$$ancSim(n_i, n_j) = lSim(n_i, n_j) * (\delta LCS_n(p_i, p_j) - \theta GAP(p_i, p_j) - \lambda LD(p_i, p_j)) \quad (6)$$

$\delta + \theta + \lambda = 1$ and $(\delta, \theta, \lambda) \geq 0$

2.3.2 Immediate Descendants Context

To obtain the immediate descendants context similarity *immSim* (n_i, n_j), we compare their two immediate descendants context sets. This is done by using the linguistic similarity *lSim* between each pair of children in the two sets. We select the matching pairs with maximum similarity values. Finally, the average of best similarity values is taken.

2.3.3 Leaf Context

The leaf context of a node n_i is the set of leaf nodes of subtrees rooted at n_i. If $l_i \in$ *leaves*(n_i) is a leaf node, then the context of l_i is given by the path p_i from n_i to l_i.

$$leafSim(l_i, l_j) = lSim(l_i, l_j) * (\delta LCS_n(p_i, p_j) - \theta GAP(p_i, p_j) - \lambda LD(p_i, p_j)) \quad (7)$$

To obtain the leaf context similarity between two leaves $l_i \in$ *leaves*(n_i) and $l_j \in$ *leaves*(n_j), we compute the leaf similarity *leafSim* between each pair of leaves in the two leaf sets. We then select the matching pairs with the maximum similarity values. The average of the best similarity values is taken.

2.4 Node Similarity

The node similarity *nodeSim* is obtained by the combination of three context scores:

$$nodeSim(n_i, n_j) = \alpha * ancSim(n_i, n_j) + \beta * immSim(n_i, n_j) + \gamma * leafSim(n_i, n_j) \quad (8)$$

$\alpha + \beta + \gamma = 1$ and $(\alpha, \beta, \gamma) \geq 0$, once the structural similarity computation is made, the system returns the k node candidates per source n_i that have the maximum values of *nodeSim* and greater than a given threshold e.g. 0.7.

3. CONCLUSION

Due to the number of existing metadata standards and their heterogeneity, there has been a great interest to develop an automatic integration solution. The existence of such model makes the integration process faster and less expensive. We proposed a new XML Schema matching technique to automate the integration of multimedia metadata. We essentially proposed a linguistic and structural similarity measure linking metadata encoded in different formats. In our ongoing work, we plan to enhance the proposed matching system through a better use of the structural information by using the adjacency of nodes to detect other mappings.

4. REFERENCES

[1] M. Bilenko, R. J. Mooney, W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[2] D.Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching xml documents via xml fragments. In *SIGIR*, pages 151–158, 2003.

[3] M. Hausenblas. Multimedia vocabularies on the semantic web, July 2005.

[4] M.-L. Lee, L. H. Yang, W. Hsu, and X. Yang. Xclust: clustering xml schemas for effective integration. In *CIKM*, pages 292–299, 2002.

[5] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB*, pages 49–58, 2001.

[6] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE*, pages 117–128, 2002.