

A Label-based Metadata for Schema Clustering

Pitsanu Lousangfa

Naiyana Sahavechaphan

Large Scale Simulation Research Laboratory
National Electronics and Computer Technology Center
112 Thailand Science Park, Pahon Yothin Rd.
Klong 1, Klong Luang, Pathumthani 12120 Thailand
Tel: 662-564-6900 Ext.2279, Fax: 662-564-6772
Email: pitsanu.lousangfa|naiyana.sahavechaphan@nectec.or.th

Abstract

Clustering elements defined in schemas has been recognized for reducing the number of element-to-element comparisons and hence improving the schema matching performance. In this paper, we propose a label-based metadata that facilitates the schema clustering such that a cluster contains elements representing *semantically similar* information. Our experiments showed that a *relation-based* metadata (*RM*) is a potential metadata that not only facilitated the schema clustering but also supported the reduction of the number of comparisons in the schema matching process.

1 Introduction

Schema matching (Madhavan et al., 2001; Rahm and Bernstein, 2001) has been recognized in several database application domains such as schema integration, data warehouse, e-commerce and semantic query processing. Its fundamental operation is *match* which takes two schemas as input and produces a mapping between elements of the two schemas that correspond semantically to each other. In addition, the *match* operation (Li and Clifton, 1994) is typically done in a pairwise fashion in which all possible pairs of elements are compared. This results in unreasonably large tasks especially since most pairs do not represent the same information.

Today, some approaches ranging from element-based to content-based heuristics have been proposed in the literature (Li and Clifton, 1994; Madhavan et al., 2001) to reduce the number of element-to-element comparisons

and hence improve the schema matching performance. In the absence of data instances (or content), Cupid (Madhavan et al., 2001) clusters elements and defines cluster metadata based on either data types of elements or concepts of tokens encapsulated in element names. Here, all possible pairs of elements, e_s and e_t , in compatible categories of two schemas, S_s and S_t , are compared to find the matching pairs. This approach potentially reduces the number of element comparisons. However, a big overhead occurs in finding compatible categories which are determined by comparing all tokens in two given categories. In addition, schema clustering is heavily dependent on concepts predefined by experts, prohibiting it to be done in an automatic manner. SemInt (Li and Clifton, 1994) trains neural networks to classify elements according to element specifications (i.e. type, length and value constraint) and data values (i.e. patterns and statistics). Each category is represented by the *numeric* metadata, an average vector over numeric vectors of elements¹ in the category. The match between the source element (e_s) and the numeric metadata (the representative of target elements e_{t_1}, \dots, e_{t_j}) implies that e_s potentially matches with all e_{t_1}, \dots, e_{t_j} . This approach infers element relationships unrecognized by programmer and hence provides an automatic schema clustering. However, it potentially produces too many clusters wherein each cluster contains only elements representing *the same* information. In particular, the number of clusters is likely equivalent to the number of elements, making it hard to reduce the number of comparisons.

While these approaches take steps into the

¹The numeric vector of an element is created based on numeric convertible element specification and data values.

right direction, each approach individually is limited to an *efficient* comparison. To address the previous drawbacks, we believe that the application of *filtering* methodology enables the reduction of comparisons. Specifically, target elements of the target schema S_t are first clustered wherein each cluster should contain elements representing *semantically similar* (not the same) information. For example, element names *telephone*, *fax* and *email* represent telecommunicating information, while element names *county*, *city* and *state* are address information. A source element e_s of the source schema S_s is then compared to target elements e_{t_i}, \dots, e_{t_j} in the clusters whose metadata matches the source element e_s itself to further find its mapping.

To achieve such cluster, in this paper, we thus propose a simple but novel and practical metadata created based on element names (or labels). This is mainly because element label is the only information that enables the creation of clusters containing semantically similar information. Particularly, we have defined three metadata, namely *distinct-based* (DM), *synonym-based* (SM) and *relation-based* (RM) metadata. Our experiments through realistic databases showed that: (i) these three metadata facilitated the schema clustering in an automatic manner wherein they provided an approximate 60% clustering accuracy; and (ii) as expected, RM metadata yielded less number of tokens than DM and SM metadata, significantly reducing the number of token comparisons in the schema matching process.

Roadmap: The rest of this paper is organized as follows: Section 2 describes three metadata types. Section 3 gives the linguistic similarity measure. Experimental evaluation is given in Section 4. We conclude in Section 5.

2 Label-based Cluster Metadata

In this work, we consider metadata as a representative of a group of element labels (not data contents) defined in a schema. For example, the metadata *address* refers to *street*, *city*, *state* and *zipcode* elements. Particularly, we have defined three metadata types, namely *distinct-based* metadata (DM), *synonym-based* metadata (SM) and *relation-based* metadata (RM). Each metadata M' of choice is created based on a current metadata M and an ele-

ment label L to be added into a cluster. Details of each metadata are provided in the following sections and (Lousangfa et al., 2007). For simplicity, let T_M a set of normalized tokens t_m in M , T_L a set of normalized tokens t_l in L , and T'_M a set of normalized tokens t'_m in M' .

2.1 Distinct-based (DM) Metadata

Definition 1 $T'_M = T_M \cup T_L$

Description: A distinct-based metadata (DM) is simply a collection of *lexically different* tokens defined in T_M and T_L .

Consider Table 1 that shows two metadata created based on: (1) the current metadata $\{street, state\}$ and label *country*; and (2) the current metadata $\{street, state\}$ and label *city*. With the DM type, in the first example, the new metadata $\{street, state, country\}$ is created by combining all tokens defined in the current metadata and label. Similarly, in the second example, the new metadata $\{street, state, city\}$ is created.

2.2 Synonym-based (SM) Metadata

Definition 2 $T'_M = (T_M \cup T_L) - S$

where $S = \{t_l \mid t_l \in T_L \text{ and } t_l \text{ is synonym to } t_m \in T_M\}$

Description: A synonym-based metadata (SM) is a collection of tokens defined in T_M and T_L where such tokens must be *lexically different* and are not *synonym* to each other. In particular, it is an extension to a distinct-based metadata with the exception that one of two lexically different but synonym tokens, t_m and t_l , is removed to reduce the number of tokens in metadata.

Again, consider Table 1 with the SM type. In the first example, the new metadata $\{street, state\}$ remains the same as *country* is synonym to *state*. In the second example, the new metadata $\{street, state, city\}$ is created as there is no identical or synonym token pairs.

2.3 Relation-based (RM) Metadata

Definition 3 $T'_M = ((T_M \cup T_L) - S - R) \cup A$

where $S = \{t_l \mid t_l \in T_L \text{ and } t_l \text{ is synonym to } t_m \in T_M\}$, $R = \{t_m, t_l \mid t_l \in T_L, t_m \in T_M \text{ and } t_l \text{ is related } t_m\}$ and $A = \{t \mid t \text{ is the nearest ancestor of } t_m \text{ and } t_l \text{ where } t_m \text{ is related to } t_l\}$

Description: A relation-based metadata is a collection of tokens defined in T_M and T_L

Table 1: Example of Metadata Types

	street	state	DM	SM	RM
Example 1: country	0.60	1.00	street_state_country	street_state	street_state
Example 2: city	0.74	0.89	street_state_city	street_state_city	street_region

where such tokens must be *lexically different* and are not *related*. Two tokens, t_m and t_l , are related if their match is above threshold. In particular, it is an extension to a synonym-based metadata with the exception that two lexically different but related tokens, t_m and t_l , are replaced with their nearest ancestor.

Again, consider Table 1 with the the RM type. In the first example, the new metadata $\{street, state\}$ is created in a similar manner to the SM type. In the second example, the new metadata $\{street, region\}$ is created wherein *region* is the nearest ancestor of two related tokens, *state* and *city*.

3 Linguistic Similarity Measure

An element name (or label), typically represented by natural language, can be classified as either (i) an atomic label - composed of a single word; or (ii) a composite label - composed of multiple words, where the start of each word is distinguished generally by punctuations (for example, *purchase-order*), case distinction (for example, *purchaseOrder*), or numeric digits (for example, *street1*). Typically, no restrictions are applied on the words themselves – they can be a fully-defined dictionary word, an abbreviation, an acronym, or a substring. For example, *qty* is an abbreviation of *quantity*; *wom* an acronym of *unitOfMeasure*; and *addr* a substring of *address*.

To determine the similarity distance between two given atomic labels, we use the linguistic similarity measure defined in WordNet::Similarity (Pedersen et al., 2004). The similarity distance between two given composite labels, on the other hand, is computed on the basis of the similarity distance of two given atomic labels. Specifically, three essential steps are needed: (1) the two given labels are first parsed and normalized into tokens (words); (2) the similarity of all possible token pairs is measured; and (3) the similarity of two labels is finally computed based on the similarity results on return from step 2. In particular, the similarity of a metadata and an element label (in step 3), denoted as $sLSim(M,$

$L)$ or $sLSim(T_M, T_L)$, is computed as the average of the best similarity of each element token t_l with a metadata token t_m and and the best similarity of each metadata token t_m with an element token t_l , a *symmetric* computation. $sLSim(T_M, T_L)$ is formally given as:

$$sLSim(T_M, T_L) = \frac{lSim_M(T_M, T_L) + lSim_L(T_M, T_L)}{|T_M| + |T_L|} \quad (1)$$

where

$$lSim_M(T_M, T_L) = \sum_{t_m \in T_M} [\max_{t_l \in T_L} lingSim(t_m, t_l)] \quad (2)$$

$$lSim_L(T_M, T_L) = \sum_{t_l \in T_L} [\max_{t_m \in T_M} lingSim(t_l, t_m)] \quad (3)$$

where *lingSim* is the linguistic similarity measure between two given tokens, and $|T_M|$ as well as $|T_L|$ are the number of tokens in the metadata and element label, respectively.

4 Preliminary Experimental Evaluation

The goal of label-based metadata is to (i) facilitate schema clustering; (ii) improve schema matching process; and (iii) represent element members. In this evaluation, we showed the benefit of label-based metadata on schema clustering and schema matching, while the representative of element members can be found in (Lousangfa et al., 2007).

4.1 Experimental Setup and Methodology

Figure 1 illustrates the overall architecture of label-based clustering system, namely *lCluster*. The *lCluster* system takes a set of schema elements as input and produces a set of clusters as output. Specifically, *lCluster* is developed on top of two modules: *LabelMatching* – providing the match value of two labels as defined in Section 3. In this evaluation, we use the *wup* measurement (Wu

and Palmer, 1994); and *LMC* (Lousangfa et al., 2007) – providing the label-based cluster metadata as defined in Section 2.

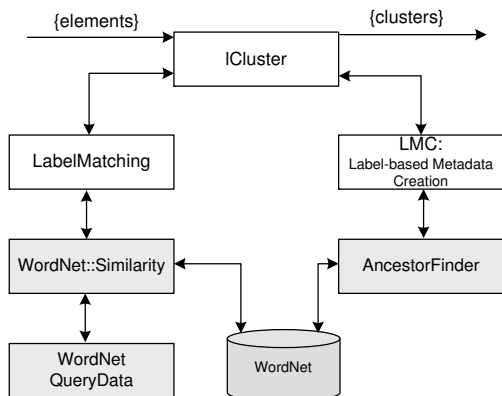


Figure 1: The Overall Architecture of *ICluster* system.

For each schema element e , the process of *ICluster* would discover for its appropriate clusters determined by the difference distance² between the label of element e itself and the cluster metadata. If the difference distance is less than threshold, that element e is decided to be in the cluster. Once there is a new element put into a cluster, the cluster metadata is updated to reflect it element members. Otherwise, the new cluster is created with the element label as the cluster metadata. Consider Figure 2 as an example with relation-based metadata (RM). Here, as there is no cluster in the system, a new cluster is thus created for the first element *personEmail* with the cluster metadata *person, email*. For the second element *personTelephone*, it is compared to the metadata of existing cluster – *person, e-mail*, and is determined to be in this cluster as their difference distance is less than threshold. The cluster metadata is then updated from *person, email* to *person, telecommunicate* to reflect its element members. Lastly, with the third element *address*, its difference distance with the cluster metadata *person, telecommunicate* is over threshold. The new cluster is thus created for it with the metadata *address*.

²In this evaluation, we applied the Euclidean distance to determine the difference distance between an element label and a metadata wherein it is based solely on label: $\sqrt{(1 - sLSim(T_M, T_L))^2}$

Measure of Efficiency. To evaluate the benefit of label-based metadata, we measured (i) the clustering accuracy between clusters produced by the *ICluster* against manually determined clusters. Particularly, we computed the clustering accuracy using final F-Measure (Zhao and Karypis, 2002) modified by Dan Walker (Walker,) which is calculated as per system clusters, not manually determined clusters. In addition, we modified the weight ratio such that the overall F-measure is normalized. The weight ratio is the number of matched elements between a manual and system clusters divided by the total number of elements in schema; and (ii) the number of comparisons in schema matching process based on clusters produced.

Databases. In this evaluation, we used 4 database schemas related to research information³:

- National Science and Technology Development Agency (NSTDA) research schema consists of 15 elements.
- National Electronics and Computer Technology Center (NECTEC) research schema consists of 38 elements.
- National Metal and Materials Technology Center (MTEC) research schema consists of 80 elements.
- National Center for Genetic Engineering and Biotechnology (BIOTEC) research schema consists of 76 elements.

4.2 Experimental Result

A series of experiment were conducted to evaluate the potential benefits of label-based metadata on schema clustering and schema matching.

4.2.1 Clustering Accuracy

The first set of experiments measured the accuracy of label-based metadata – *DM*, *SM* and *RM*. The accuracy was defined as the ability of metadata to facilitate schema clustering against manual clusters defined by experts. Particularly, the experiments were conducted for each previous research schema via *ICluster*

³Research information is information about researcher, publication and research project.

Elements	Current Clusters	New Clusters
personEmail	-	{personEmail} <i>person, email</i>
personTelephone	{personEmail} <i>person, email</i>	{personEmail, personTelephone} <i>person, telecommunicate</i>
address	{personEmail, personTelephone} <i>person, telecommunicate</i>	{personEmail, personTelephone} <i>person, telecommunicate</i> {address} <i>address</i>

Figure 2: The Clustering Example.

varying by different metadata types as well as various difference distances.

Figure 3 shows the best average clustering accuracy via the application of different metadata types on 4 research schemas. The x-axis has the metadata types and the y-axis is the clustering accuracy measured using F-Measure. Here, *DM*, *SM* and *RM* equivalently performed with the approximate clustering accuracy of 60%. In particular, all *DM*, *SM* and *RM* worked well for the difference distance threshold of 0.15.

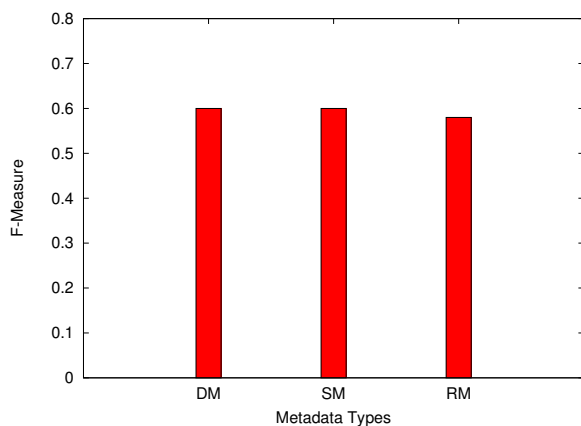


Figure 3: The Clustering Accuracy.

4.2.2 Schema Matching Improvement

The second set of experiments measured the effectiveness of label-based metadata – *DM*, *SM* and *RM*. The effectiveness was defined as the ability of schema clusters created to the improvement of schema matching performance. The improvement was determined by

the number of token comparisons – the less number of token comparisons implied the better schema matching performance. Particularly, the experiments were conducted on the source research schema that contained 16 elements with the overall 17 tokens against the target BIOTEC research schema that contained 76 elements with the total of 150 tokens. Initially, BIOTEC research schema was clustered based on each metadata type. Each source element was then compared with each individual metadata of BIOTEC clusters and further compared with a set of target elements in the matched clusters. The best difference distance was selected for each metadata type.

Figure 4 shows the number of token comparisons in discovering the mapping of source elements with target elements. The x-axis has the metadata types and the y-axis is the number of token comparisons. Here, *RM* performed the best with 1890 token comparisons, 25% reduced comparisons comparing to matching without clusters. *SM* was slightly better than matching with no clusters. *DM* performed the worst requiring 2957 token comparisons. This is mainly because *DM* had more number of tokens in metadata than *SM* while its of *SM* was more than *RM* as shown in Figure 5. Consequently, there was many more clusters created based on *DM* and *SM* that matched each given target element than *RM* did.

5 Conclusion

We believe that the application of *filtering* methodology enables the reduction of element-to-element comparisons. Specifically, elements of the target schema S_t are first clus-

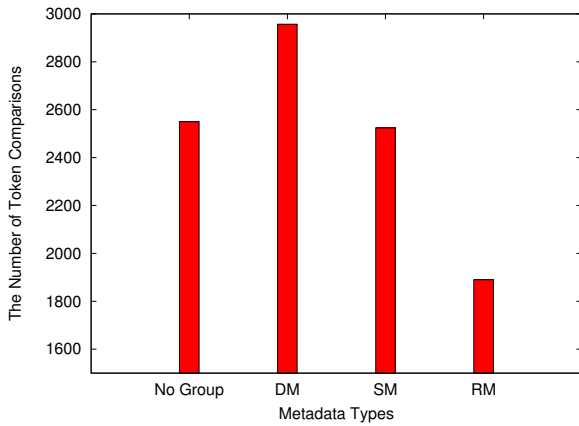


Figure 4: The Number of Token Comparisons.

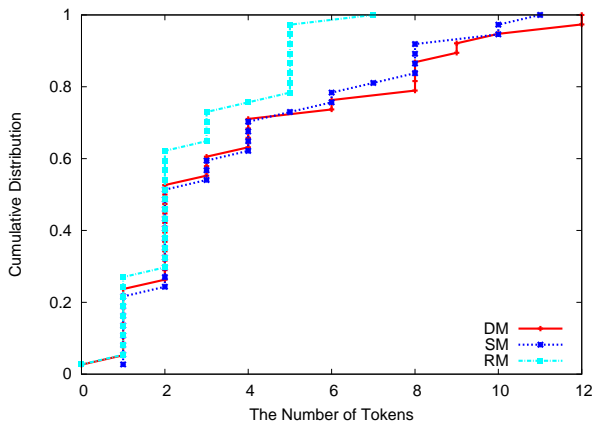


Figure 5: The Cumulative Distribution of The Number of Tokens.

tered wherein each cluster should contain elements representing *semantically similar* (not the same) information. A source element e_s of the source schema S_s is then compared to target elements e_{t_1}, \dots, e_{t_j} in the clusters whose metadata matches the source element e_s itself to further find its mapping. In this paper, we thus propose a label-based metadata that facilitates the creation of such clusters. Our experiments showed that a *relation-based* metadata (RM) is a potential metadata that not only facilitated the schema clustering but also supported the reduction of the number of comparisons in the schema matching process.

References

- Wen-Syan Li and Chris Clifton. 1994. Semantic Integration in Heterogeneous Databases Using Neural Network. In *Proceeding of the 20th International Conference of Very Large Data Bases (VLDB)*.
- Pitsanu Lousangfa, Naiyana Sahavechaphan, Jedsada Phengsuwan, and Seksit Suwan. 2007. LMC: Label-based Metadata Creation. In *Proceeding of the 4th International Conference of Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*.
- Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. 2001. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity - measuring the relatedness of concepts. In *AAAI*, pages 1024–1025.
- Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350.
- Dan Walker. CS 601R: Topics in NLP: Metrics for Clustering. faculty.cs.byu.edu/~ringger.
- Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133–138, New Mexico State University, Las Cruces, New Mexico.
- Y. Zhao and G. Karypis. 2002. Evaluation of hierarchical clustering algorithms for document datasets.