JIS Accepted for Publication
By the Journal of Information Science: http://jis.sagepub.co.uk

An Empirical Comparison of Ontology Matching Techniques

Ahmed Alasoud, Volker Haarslev, and Nematollaah Shiri[1]

Dept. of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada

Abstract

Ontology matching aims to find semantic correspondences between a pair of input ontologies. A number of matching techniques have been proposed recently, however, we may benefit more from a combination of such techniques as opposed to just a single method. This is more appropriate, but very often the user has no prior knowledge about which technique is more suitable for the task at hand. However, it remains a labour intensive and expensive task to perform. Further, the complexity of the matching process as well as the quality of the result is affected by the choice of the applied matching techniques. We study this problem and propose a framework for finding suitable matches. A main feature of this is that it improves the structure matching techniques and the end result accordingly. We have developed a running prototype of the proposed framework and conducted experiments to compare our results with existing techniques. While being comparable in efficiency, the experimental results indicate our proposed technique produces better quality matches.

Keywords: Ontology matching; similarity measures; performance; semantic web.

1. Introduction

Ontology matching is an important problem in sharing information and integrating ontology sources in numerous applications, examples of which are as follows.

- Ontology engineering

  Ontology engineering is the environment where users design, implement and maintain ontology-based applications. Therefore, they need ontology matching to deal with multiple ontologies. For instance, some users keep updating their ontologies, which very often lead to having more than one version for the same ontology. As a result, ontology matching is important to help users investigate what entities have been changed (added, deleted, or renamed) between different versions of ontologies [1,2,3,4].

[1] Corresponds to: Nematollaah Shiri, Dept. of Computer Science & Software Engineering, Concordia University, 1455 De Maisonneuve West, Montreal, QC, H3G 1M8, Canada. Email: shiri@cse.concordia.ca.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

- Information Integration

  Information integration is a popular application, where matching is considered as a necessary solution. There are different problems of information integration, such as schema integration [5,6,7,8], data warehousing [9], data integration [10,11,12,13,14], and catalogue integration [15,16,17,18]. One of the most popular scenarios in information integration is the use of an integrated view. For instance, suppose there is a company that has branches, dealers, etc, distributed over the world. The main branch needs to access information such as customers, sellers, and statistics about employees, sales, etc. In this case, one can provide a unified view (or global ontology) for the main branch, which can query the ontologies at various branches through proper mappings and wrappers. All in all, the matching step in such a scenario is to relate the correspondences between the entities in both the global ontology and the local ontologies (source ontologies).

Due to the popularity of the semantic web, we witness a continuous growth in both the number and size of available ontologies. This has resulted in an increased heterogeneity in the available information. For example, the same entity could be given different names in different ontologies or it could be modeled or described in different ways. The Ontology Matching Problem (OMP) can be described as follows: given two ontologies, each describing a collection of discrete entities such as classes, properties, individuals, etc., users want to identify semantic correspondences between the components of these entities. This problem has been the subject of numerous studies, and a number of techniques have been proposed. These matching techniques are often domain-dependent, for being mainly based on a single similarity measure, such as names, structures, logic satisfiability, etc. This makes them useful and efficient in specific domains. For example, matching techniques which are based on syntactic similarity provide good results in domains, where there is a high probability that whenever the matched entities agree on their syntax, they also agree on their semantics. However, such techniques which are solely based on name similarity might not work well in application domains, where similar entity names are used with different meanings. Consequently, some researchers consider using a number of matching techniques, and then aggregating the results of individual matching methods in order to compute the final matching result.

These matcher composition systems (matching systems that use more than one similarity technique) are not always clear about the suitability of their reused matching techniques for different kinds of matching domains. It is therefore not easy for a regular user to decide, among the vast number of matching techniques, which one is preferred for matching the given ontologies. Consequently, the choice of the user might affect the matching process in both time and quality.

In the context of ontology matching, we proposed a multi-level matching algorithm (MLMA) [19] that allows combining different measures within one framework to improve the matching results. Fig. 1 illustrates the main idea of the multi-level method when there are two levels. It shows that different similarity measures $\{m_1, m_2 \ldots m_l\}$ are divided into two partitions and applied at two levels. For instance, the name and linguistic similarity measures are applied in the first level, and then, at the second level, the structural similarity measure is applied to the candidate results' states $\{e_1, e_2 \ldots e_n\}$. As a result, we could obtain the state $\{e_f\}$, which has the highest confidence.

The MLMA method assumes that the collection of similarity measures is partitioned by the user into several groups, and that there is a partial order on the partitions, also defined by the user. However, very often, the user is not an expert and does not have enough knowledge about a most suitable order on the groups for the matching task in order to achieve a satisfactory result in reasonable time.

The contributions of this paper are as follows:

1. We study the impact of selected matching techniques on both time and quality.
2. We propose a recommendation analysis method that improves the structure-based measure on one side, and improves the efficiency and the matching quality on the other side.

As our objective in this study has been to improve and support the matching techniques we introduced earlier, these ideas can be also applied in ontology integration applications.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

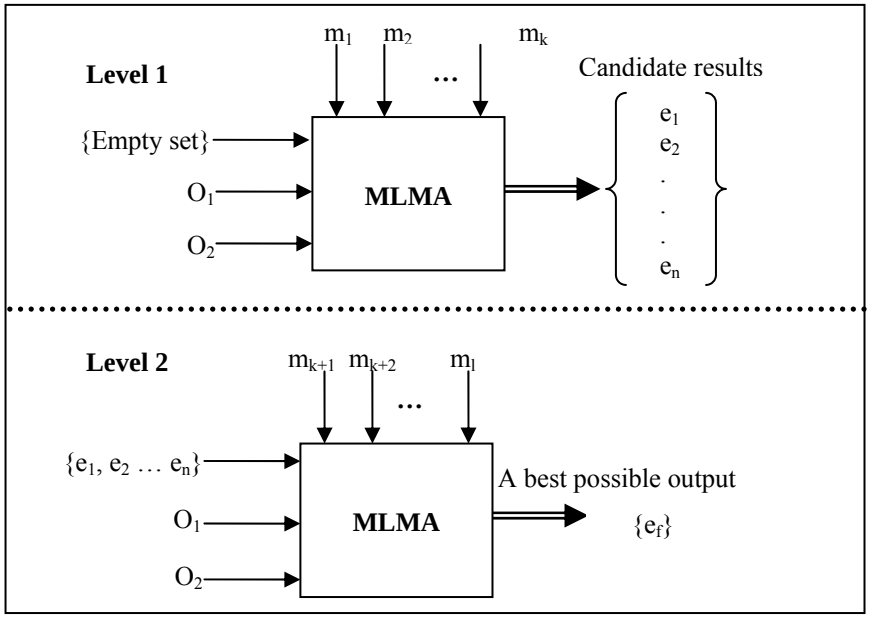


Figure 1. Schematic description of the MLMA approach

2. Motivating example

Through the following example, we illustrate the main ideas of the proposed technique. Fig. 2 shows two sample taxonomies for two computer ontologies O_1 and O_2. For ease of presentation, we use simple and small taxonomies.

We have to integrate the ontolgies into a single ontology. For reducing the manual work involved, we use a matching algorithm to identify the matching entities, and then help the middleware to integrate the ontologies. As can be seen in Fig. 2, entities S_1, S_2, S_3, and T_1, T_2, T_3 are *concepts*, which are high-level entities in the input ontologies. The goal is to find the corresponding matches among the entities in the two input ontologies.

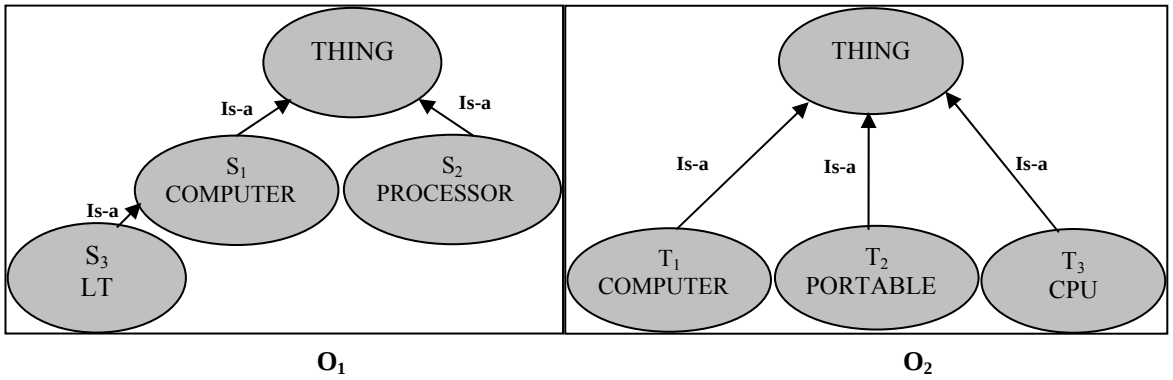


Figure 2. Computer Ontologies

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

Many methods exist to measure similarities between two entities, such as string similarity, linguistic similarity, etc. However, when we use a single matching measure for an input pair of ontologies, we may not be satisfied with the final match result. For instance, if we use a string similarity measure only, the concepts *LT* and *PROCESSOR* in O_1 have no matches in O_2. On the other hand, a string similarity measure is the basis for some other methods of measuring similarities between entities, and it works well in some domains where a match based on their syntax would most probably mean agreement on their semantics.

Another example is when we use a stronger semantic measure, such as a linguistic based measure. For instance, we find out that the concept *COMPUTER* in O_1 is mapped to concepts *PORTABLE* and *COMPUTER* in O_2. So, this will not help the user to focus on his/her intention. As a result, if we use both measures (string and linguistic), the concept *COMPUTER* in O_1 will be mapped to the concept *COMPUTER* in O_2 with a very high confidence, and concept *PROCESSOR* in O_1 will be mapped to the *CPU* in O_2.

Given the input ontologies and the matching techniques, it is difficult to specify that concept *LT* in O_1 corresponds to concept *PORTABLE* in O_2. Suppose that the input ontologies O_1 and O_2 are represented in description logic language [20], as follows:

$$O_1:\ \begin{array}{l} LT \subseteq COMPUTER \\ COMPUTER \subseteq THING \\ PROCECCOR \subseteq THING \end{array} \qquad O_2:\ \begin{array}{l} PORTABLE \subseteq \exists has\_cpu.CPU \\ \exists has\_cpu.CPU \subseteq COMPUTER \\ CPU \subseteq THING \end{array}$$

where $\subseteq$ denotes subsumption relationships such as is-a, $\exists$ denotes the existential quantification, and *has_cpu* is a binary relationship.

Now, using description logic (DL) reasoning techniques about these ontologies before matching them can help infer useful information in order to be used by a matching technique. For instance, applying the DL reasoning technique on O_2 yields RO_2, which is shown in Fig. 3. However, there are no further inferences which can be obtained from O_1. In other words, RO_1 is O_1. As a result, matching RO_1 to RO_2 assists the similarity matching technique (structure-based technique) to identify the relationship between the concept *LT* in RO_1 and concept *PORTABLE* in RO_2.

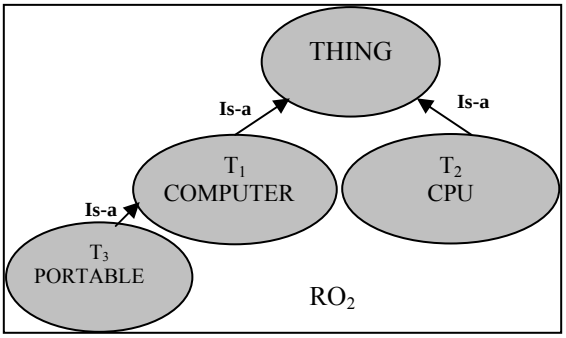


Figure 3. Result of reasoning about O_2

Fig. 4 shows the taxonomy of ontology number 232 from the benchmark test samples suite of the Ontology Alignment Evaluation Initiative OAEI-07 (Sec. 6 gives more details). After applying the DL reasoning technique, we get more structural information. Therefore, it supports the structure-based matching techniques to provide better matching results when the ontology 232 is matched to the reference ontology 101. Fig. 5 shows the results of applying reasoning techniques to the ontology 232.

All in all, the ontology matching process facilitates the interoperation between the knowledge and data expressed in the matched ontologies. It is thus of utmost importance for some applications, such as ontology merging, query answering …etc, whose interoperability is jeopardized by heterogeneous ontologies.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

As a result, the user should be given support in deciding which underlying technique, or combination of techniques, is the best suited for the matching task at hand.

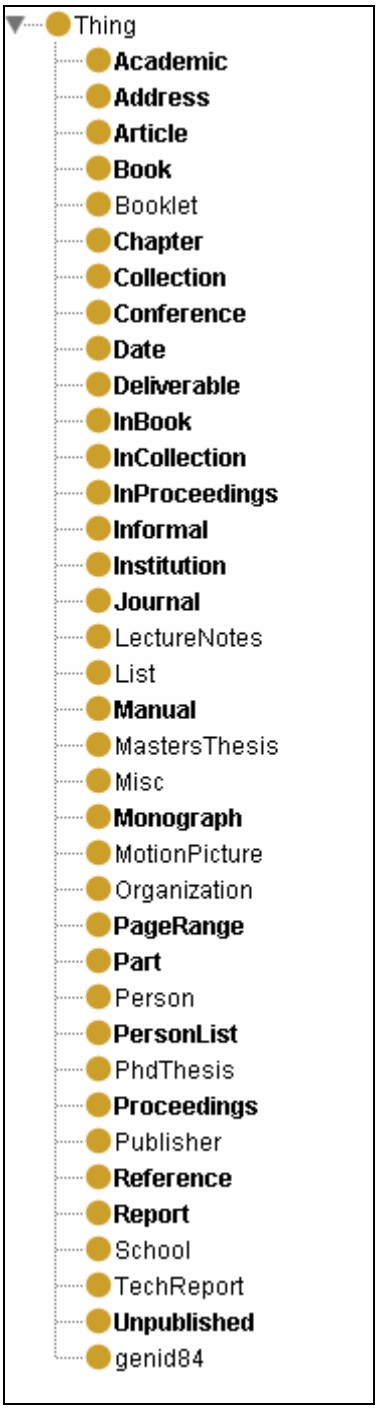


Figure 4 Taxononmy of notology 232

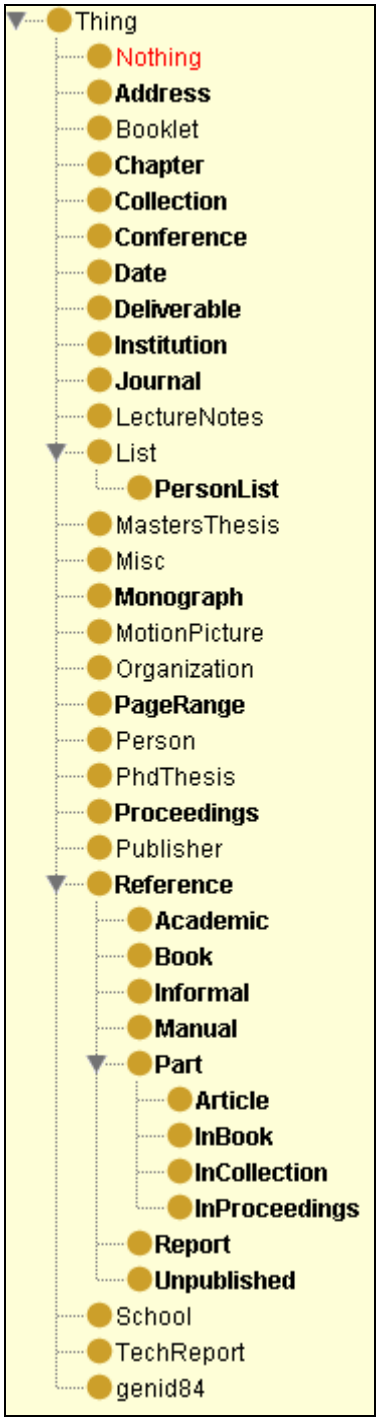


Figure 5 Result of reasoning about ontology 232

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

3. Our proposed framework

We propose a technique for the analysis and reuse of matching methods in order to identify and recommend matching methods for a given pair of ontologies. Furthermore, it assists the structure similarity measuring methods, optimizes the matching process by omitting the unpractical matching methods for the matching task at hand, and therefore, improves the end result's matching quality and efficiency.

Fig. 6 illustrates the main idea of our technique. It shows the different similarity measures $\{m_1, m_2 \ldots m_k\}$ together with RO_1 and RO_2 that are fed to the recommendation process, which will offer the user a rank of the similarity measures considered (M_i). Moreover, users have the option to use the recommended similarity measures list (M_i) or ignore it and use their own ranked similarity measure list (user's list).

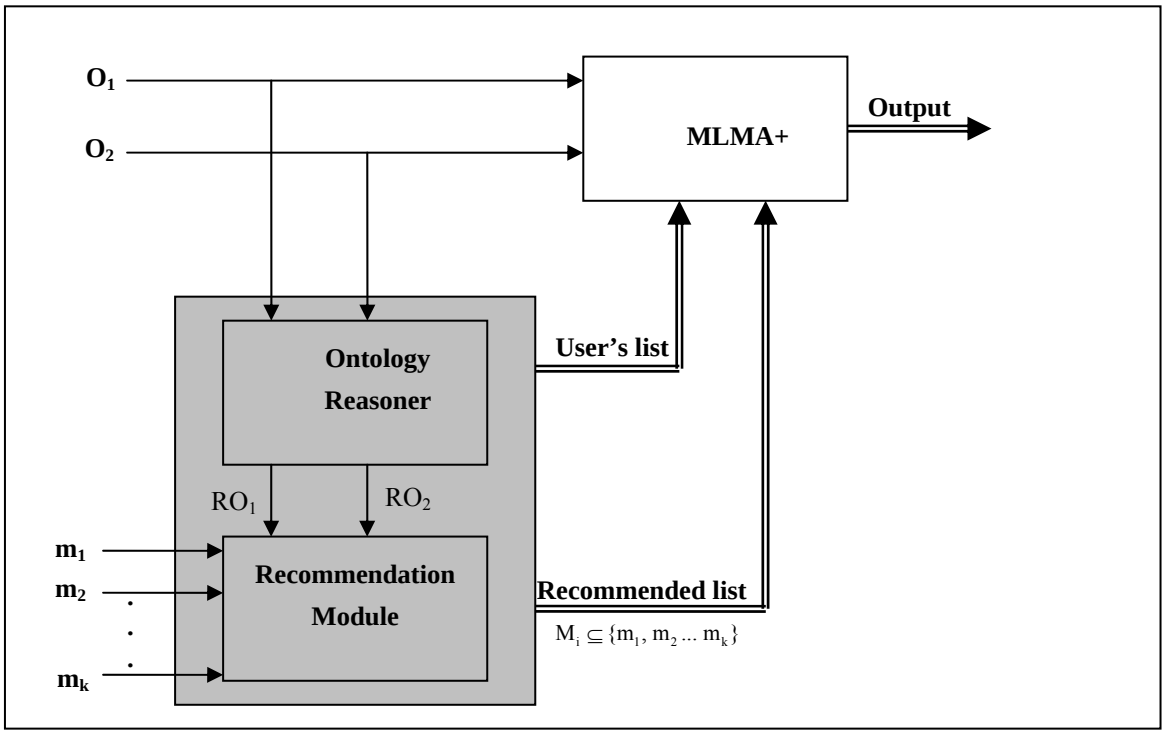


Figure 6. The proposed framework

M_i is based on the appropriate similarity methods considered for matching the entities of O_1 to the entities of O_2. Further, RO_1 and RO_2 are obtained by applying RACER [21] to O_1 and O_2.

As a result, the Multi-Level Matching Algorithm (MLMA+) [22] that performs a neighbour search takes these recommendations into account in order to find the correspondences between the entities in the given ontologies.

4. Description of the framework

In this section, we describe the main components of our approach to the solution. We describe the ontology mapping problem as identifying pairs of similar nodes (also called vertices) in the input ontologies modelled as labelled directed graphs. The nodes in the input graph correspond to entities in the ontologies, and the edges indicate the relationships between the pairs of nodes they connect. The labels indicate the kind of relationship, e.g. "domain" or "range."

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

Before introducing the multi-level match approach (MLMA) and the neighbour search by the multi-level match approach (MLMA+), we provide some notations and definitions.

Definition 1 (Entity-relationships) Let S be a source ontology, T be a target ontology. We use $E^S = \{s_1, s_2,\ldots, s_n\}$ and $E^T = \{t_1, t_2,\ldots, t_m\}$ to denote the set of entities in S and T, respectively. Entity refers to classes, properties, or individuals for which we want to find matches in the input ontologies. We use $R(r_{ij})$, defined below, to denote the relationship between entities s_i and t_j. We use r_{ij} to denote a matching degree between s_i and t_j.

Definition 2 (Relationship Matrix) This relational matrix, denoted $R(r_{ij})$, represents the relationship between ontologies S and T, i.e., r_{ij} indicates the similarity between concept s_i in S and concept t_j in T. Using R, we define another relational matrix, called the *similarity matrix*, which captures a different relationship between S and T. In the matrix $R(r_{ij})$, $s_i\ r\ t_j$ says that entity s_i in the source ontology S matched with entity t_j in the target ontology T based on relationship r, where r could be any of the existing similarity measuring method, such as string similarity measure, linguistic similarity measure, … etc.

$$R(r_{ij}) = \begin{bmatrix} s_1 r t_1 & s_1 r t_2 & \ldots & s_1 r t_j \\ s_2 r t_1 & s_2 r t_2 & \ldots & s_2 r t_j \\ \ldots & \ldots & \ldots & \ldots \\ s_i r t_1 & s_i r t_2 & \ldots & s_i r t_j \end{bmatrix}$$

Definition 3 (Similarity Matrix) This relational matrix, denoted $L(l_{ij})$, includes entries in [0,1], which are called the *similarity coefficients* and denote the degree of similarity between s_i and t_j. R and L are n×m matrices.

$$L(l_{ij}) = \begin{bmatrix} l_{11} & l_{12} & \ldots & l_{1j} \\ l_{21} & l_{22} & \ldots & l_{2j} \\ \ldots & \ldots & \ldots & \ldots \\ l_{i1} & l_{i2} & \ldots & l_{ij} \end{bmatrix}$$

Moreover, the similarity matrix $L(l_{ij})$ captures the similarity coefficients between E^S and E^T based on the defined relationship matrix $R(r_{ij})$. For example, if $R(r_{ij})$ is defined to be a string similarity relationship between E^S and E^T, then the similarity coefficient l_{ij} in the similarity matrix $L(l_{ij})$ says that entity s_i in the source ontology S matched with entity t_j in the target ontology T based on a string similarity measure with a similarity coefficient l_{ij}. As a result, for each $R(r_{ij})$, we compute its $L(l_{ij})$.

Definition 4 (Matching Matrix) A matching matrix, denoted $Map_{0\text{-}1}$, is a 0-1 matrix with dimension n×m and with entries $r_{ij} \in \{0,1\}$. If $r_{ij} = 1$, it means that S_i and t_j are "matchable." They are unmatchable if $r_{ij} = 0$.

Definition 5 (Matching Space) All the possible assignments for the matching matrix form a *matching space*, also called the *mapping space*. Every assignment is a state in the matching space. The state represents a solution of ontology matching.

The following example illustrates the above concepts and terms.

Example 1. Let S and T be a given pair of ontologies, and $E^S = \{s_1, s_2,\ldots, s_n\}$ and $E^T = \{t_1, t_2,\ldots, t_m\}$ be the sets of entities. A matching matrix $Map_{0\text{-}1}$ indicates the similarity relation between the elements of E^S and E^T. The number of relationship matrices $Map_{0\text{-}1}$ is 2^{nxm}, i.e., the matching space has 2^{nxm} states. These matrices form the matching space. For instance, when $Map_{0\text{-}1}$ is 2×2, the matching space would have 16 states. Some of these mapping states are as follows, in which the rows are entities in S and the columns are entities in T.

$$\left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right).$$

The first matrix indicates no mapping. The third matrix indicates that entity s_1 is matched with t_1 or t_2, and that s_2 is matched with t_2, etc.

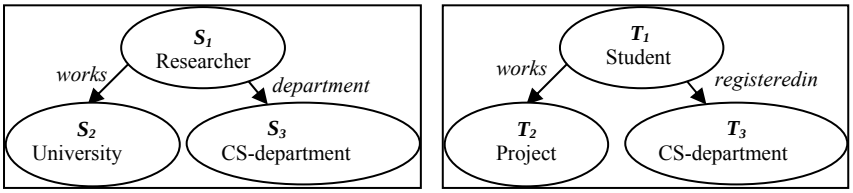
Definition 6 (Multiple matching spaces) Matching spaces are distinguished by diverse similarity measures. Moreover, different kinds of similarity measures use different methods to compute and compare the similarity of two ontologies. Accordingly, we construct the similarity matrices and matching spaces. Furthermore, different

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

relation spaces are built by using different methods of measuring similarity (see [23] for the classification of similarity methods).

4.1. Multi-level matching approach (MLMA):

This is a matching approach which uses different similarity measures at different levels. It assumes that the collection of similarity measures is partitioned by the user, and that there is a partial order on the partitions, also defined by the user. In this section, we describe the main idea of the MLMA. Fig. 7 shows two sample taxonomies for Researchers (O_1) and Students (O_2) of different universities.



O_1 O_2

Figure 7. Researchers (O_1) and Students (O_2) ontologies

We have to integrate these ontologies into a single ontology. To reduce the manual work involved, we use a matching algorithm to identify the matching entities, and then help the middleware to integrate the schemas.

As can be seen in Fig. 7, entities S_1, S_2, S_3, and T_1, T_2, T_3 are *concepts*, which are high-level entities in the input ontologies. For ease of explanation, we only use two different similarity measures to compare the entities in S and T, name similarity (Levenshtein distance) [24] and linguistic similarity (WordNet) [25]. We thus obtain the following similarity matrices for the *concepts*.

$$L_{name\_concept} = \begin{bmatrix} 0.0 & 0.2 & 0.308 \\ 0.2 & 0.2 & 0.0 \\ 0.308 & 0.308 & 1.0 \end{bmatrix}. \qquad L_{ling\_concept} = \begin{bmatrix} 0.75 & 0.181 & 0.307 \\ 0.4 & 0.181 & 0.0 \\ 0.307 & 0.166 & 1.0 \end{bmatrix}.$$

This produces two similarity spaces: name space and linguistic space. When an assignment is found for the matching space, we check the similarities of entities to see whether they exceed a user-defined threshold, denoted *th*. The choice of threshold values is application dependent and should be suitably chosen for each space. We define the following evaluation function, which measures the threshold value for the states obtained by the first phase of the MLMA algorithm. *k indicates the number of matched pairs*.

$$v = \left(Map_{0-1} \cdot L\right) / k = \sum_{i=1}^{n} \sum_{j=1}^{m} Map_{0-1}(i,j).L(i,j) \Big/ \sum_{i=1}^{n} \sum_{j=1}^{m} Map_{0-1}(i,j) \text{ , and } v \geq th.$$

The multi-matching algorithm (MMA) aims to find the correspondences between the entities of the input ontologies. The important features of this method are that it benefits from existing individual matching techniques and "combines" their match results to provide enhanced ontology matching, and it matches a collection of n elements in the source ontology S to a collection of m elements in the target ontology T. MMA is considered to be a core search algorithm for its succesor algorithms, such as MLMA and MLMA+.

Furthermore, the multi-level extension of MMA, called MLMA, assumes that the collection of similarity measures are portioned by the user, and that there is a partial order on the partitions, also defined by the user. The main characteristic of the MLMA technique is that applying this method will not decrease the number of matching concepts (size), but will increase the similarity measure of states that have high structural similarity among their

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

concepts (structure). Also, our similarity measure ensures that MLMA works even in the case that there are no structural similarities in the given input ontologies.

In addition, MLMA+ improves the efficiency of MLMA considerably, due to its use of the neighbor search algorithm. It proceeds by computing an initial state and then performing a search in its neighboring states.

We now provide a brief description of the multi-match algorithm (MMA) search process. The initial state of the mapping matrix is a zero matrix. Then, if the search process exceeds the maximum number of iterations, the obtained states (Map_{max}) will be offered as the final mapping result. Also, we need to set the additive constraints in the search process. For this example, since the number of concepts in S is equal to that in T, we consider the ontologies S and T to have been fully matched. So, the mapping states of concepts include 6 entries now, e_1, e_2, …, e_6 as shown in Fig. 8.

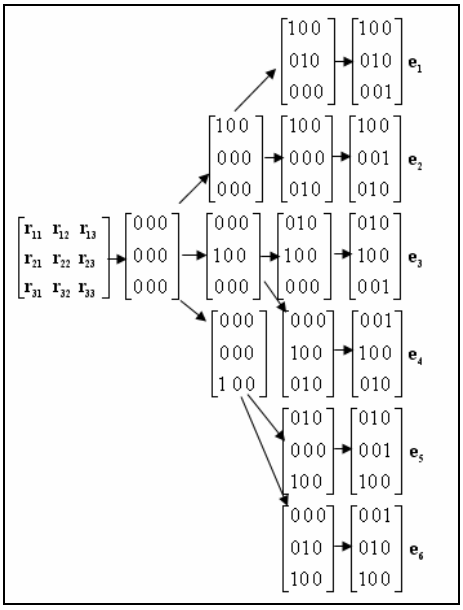


Figure 8. Searching in the matching space

The outputs of MMA are states e_1, e_2, …, e_6 shown in Fig. 9, which are represented as labelled directed graphs. It shows that e_1 has obtained one common edge, and no common edges have been obtained by the other states.

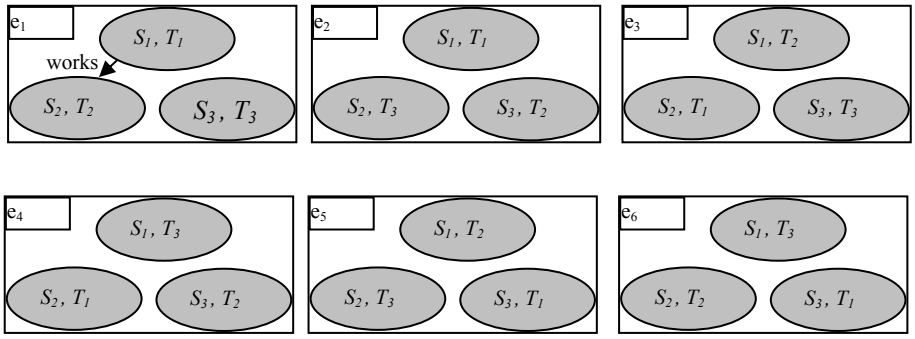


Figure 9. The states determined by MMA

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

As shown in Table 1, e_1 is the "best" match found. Using the formula for computing the threshold values for the name and linguistic similarity matrices $L_{name\_concept}$ and $L_{ling\_concept}$ above, we obtain values 0.4 and 0.64 for name similarity v_1 and linguistic similarity v_2, respectively. Each entry is determined as follows. We show this for e_1:

$$Map_{0-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad v_1 = \begin{bmatrix} 0.0 & 0.2 & 0.308 \\ 0.2 & 0.2 & 0.0 \\ 0.308 & 0.308 & 1.0 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Big/ 3 \text{ , and } \quad v_2 = \begin{bmatrix} 0.75 & 0.181 & 0.307 \\ 0.4 & 0.181 & 0.0 \\ 0.307 & 0.166 & 1.0 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Big/ 3$$

Then, v is computed by normalizing the cost of v_1 and v_2 as follows:

$$V(e_m) = \sum_{i=1}^{n} w_i v_i(e_m) \text{ , and for } e_1, \ V(e_1) = (w_1 * v_1) + (w_2 * v_2)$$

where v_i is the matching score obtained by the similarity measuring technique i, w_i is the weight of the similarity measuring technique i, and $v(e_m)$ is the score for state e_m. Consequently, in this example we used $w_1 = w_2 = 0.5$.

To measure S_{strc} for the mapping state e_1 we have:

- The number of common relationships that connect common concepts to other common concepts is 1.
- The number of relationships in O_1 with at least one end belonging to the common concepts is 2.
- The number of relationships in O_2 with at least one end belonging to the common concepts is 2. As a result, we obtain $S_{strc} = ((2*1)/(2+2)) = 0.5$.

Table 1 shows the individual and combined similarity matching results for each state e_i. Note that, using only the name similarity space, the mapping result would be e_3. In the same way, using only the linguistic space, we would obtain e_1. Also, using $Map_{name\_concept}$, $Map_{ling\_concept}$, and the threshold value *th,* we obtain S_{MMA}. Consequently, the output result state e_1 means that we matched the *n* concepts in the source ontology S to the m concepts in the target ontology T. That is, s_1 is matched with t_1, s_2 with t_2, and s_3 with t_3. Accordingly, the algorithm matches the properties and/or instances of each pair of matched concepts.

Level 1				**Level 2**	
State	**Name** v_1	**Concept** v_2	**S_{MMA}** Normalized cost $v = (v_1 + v_2)/2$	**S_{strc}**	$S = S_{MMA} + (x * S_{strc})$
e_1	0.4	**0.64**	**0.52**	0.5	**0.77**
e_2	0.103	0.305	0.204	0.0	0.204
e_3	**0.466**	0.527	0.497	0.0	0.497
e_4	0.272	0.291	0.282	0.0	0.282
e_5	0.169	0.163	0.166	0.0	0.166
e_6	0.269	0.265	0.267	0.0	0.267

Table 1. Individual and combined similarity match results

We can also notice the recognized performance of the measure and how the similarities S_{MMA} and S_{strc} are combined to compute the final measure *S*. The scenario indicates that S is always greater than or equal to S_{MMA} for our similarity measures. This leads to the fact that *S* increases the weight of states with connected common concepts, rather than those that are not connected.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

JIS Accepted for Publication
By the Journal of Information Science: http://jis.sagepub.co.uk

As a result, using *S*, we gain the following:

- *S* maintains as many matched concepts as possible.
- *S* can improve the performance of S_{MMA}, if the ontologies that are to be matched are structurally similar. However, it will not affect S_{MMA} at all if there is no structure similarity in the given input ontologies.

In the combined similarity S, suppose S_{strc}= 0. This means S depends only on the similarity measure of MMA. On the other hand, if S_{strc} = 1, the neighborhood of the concepts matched by MMA is the same, and consequently S will take the maximum value. Since $S_{MMA} + x = 1$, we have that $x = 1 - S_{MMA}$, representing the complementary part of the information described in the relationships among concepts in a desired state found by MMA.

As we do not want to miss a final matching state which includes a large number of concepts matched, S_{MMA} provides possible good matches in the input ontologies together with the similarity degrees. The MLMA method determines the same collection of matched states, but differentiates them better by taking into account the structural measures in the second level. The extension of this two level method to a multi-level method is straightforward, if the user can identify which measures could or should be applied at which level.

4.2. Neighbour search by multi-level matching algorithm (MLMA+)

A neighbor search strategy uses the MLMA as a backbone and performs a neighbor search to find correspondences between the entities in the given ontologies. An important feature of this algorithm is its fast convergence, while providing quality results obtained by a search in the neighborhood of some initial match result. We introduce a neighbor search algorithm, with a proper initialization, as an optimization for the multi-level matching algorithm. This improves the computation time. This process includes three main phases:

1. A partial set of similarity measures $\{m_1, m_2, \ldots, m_k\}$ has been applied to input ontologies to suggest an initial result.
2. The neighbour search algorithm will search the neighbours of the initial result. By neighbours we mean the mapping states that can be computed either by adding to or removing from the initial matching state some vertices, obtained by toggling a bit in the similarity matrix $L(l_{ij})$.
3. The algorithm will apply the next level similarity techniques, in order to find the final matching result.

Example 2. Consider the simple taxonomy examples in Fig. 10 for the computer ontologies O_1 and O_2.

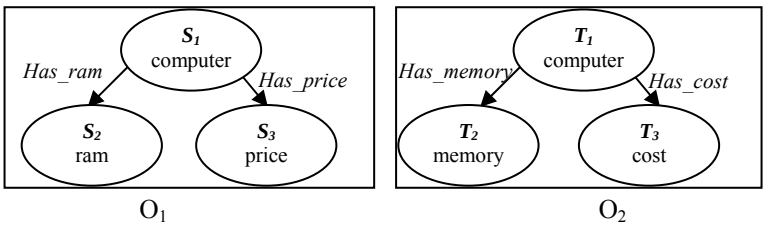


Figure 10. Computer Ontology Examples

In this example, we only use three different similarity measures, applied in two phases. There are two similarity measures applied in the first phase to compute the initial state St_0: name similarity (Levenshtein distance) and linguistic similarity (WordNet). This yields two similarity matrices for the concepts. The first matrix is based on name similarity, and the second matrix is based on linguistic similarity. Assuming that $th \geq 0.45$, and normalizing the cost of the two similarity matrices, we get matrix L. Then, L is transformed into the matching matrix $Map_{0\text{-}1}$. Note that we are using $Map_{0\text{-}1}$ and St_n as synonyms.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

$$L = \begin{bmatrix} 1.0 & 0.4 & 0.265 \\ 0.463 & 0.534 & 0.083 \\ 0.363 & 0.158 & 0.5 \end{bmatrix} \qquad \text{Map}_{0\text{-}1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The binary matrix $\text{Map}_{0\text{-}1}$ above corresponds to state $St_0=\{(s_1, t_1), (s_2, t_1), (s_2, t_2), (s_3, t_3)\}$, which means entity s_1 is matched to t_1, s_2 is matched to both t_1 and t_2, and s_3 is matched to t_3. Table 2 indicates the binary matrix for other neighboring states, together with their score values. In the search phase, 9 neighbors of St_0 will be evaluated to pick the best candidate(s) for the next level. To reduce the cost of the evaluation phase, we filter the neighbor states by keeping $\lceil x\% \rceil$ of the top weighted states for the next level. The reasons for using x%, rather than, for example, using a threshold value for filtering the candidate states, are as follows. First, this ensures that there will be some candidate states in the next level to evaluate. This may not be possible in general, when we consider a high threshold value, which would leave no candidate for the next run. A second reason is that users in general may have no knowledge of the computed score values to pick a correct threshold value. Now, choosing x=50%, the candidate states for the next level will include St_{n2}, St_{n4}, St_{n5}, St_{n7}, and St_{n9}. In phase three, we applied our structure similarity measure. Finally, the search algorithm will produce St_4 as output, which has the highest overall score value for having better structural similarity.

Neighbor number	**Matched pairs**	**Score value based on our score function V_{stn}**
St_{n1}	$\{(s_2, t_1), (s_2, t_2), (s_3, t_3)\}$	0.499
St_{n2}	$\{(s_1, t_1), (s_1, t_2)\ , (s_2, t_1), (s_2, t_2), (s_3, t_3)\}$	0.5794
St_{n3}	$\{(s_1, t_1), (s_1, t_3)\ , (s_2, t_1), (s_2, t_2), (s_3, t_3)\}$	0.5524
St_{n4}	$\{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}$	**0.678**
St_{n5}	$\{(s_1, t_1), (s_2, t_1), (s_3, t_3)\}$	0.6543
St_{n6}	$\{(s_1, t_1), (s_2, t_1), (s_2, t_2), (s_2, t_3)\ , (s_3, t_3)\}$	0.516
St_{n7}	$\{(s_1, t_1), (s_2, t_1), (s_2, t_2), (s_3, t_1), (s_3, t_3)\}$	0.572
St_{n8}	$\{(s_1, t_1), (s_2, t_1), (s_2, t_2), (s_3, t_2), (s_3, t_3)\}$	0.531
St_{n9}	$\{(s_1, t_1), (s_2, t_1), (s_2, t_2)\}$	0.6656

Table 2. Score value for each state neighbour

4.3. Specific similarity measures used in the proposed framework

For ease of presentation, we focus on the techniques we have so far implemented in our framework. The proposed framework, however, is flexible and, thus, we can incorporate any other matching techniques. In our work, we considered a string based technique (Levenshtein distance) [24], linguistic based technique (WordNet) [25], and structure based technique [19].

The string and linguistic based techniques evaluate the given entities by analyzing their names, labels and comments. They consider both the lexical and linguistic features as terms of comparison. Moreover, the structure based techniques take into account the structural layout of the ontologies considered, e.g., graph matching. In this work, we are improving our structure similarity presented in [19] by considering the inferred input ontologies (RO_1, and RO_2) by using a DL reasoner, i.e., RACER on the input pair of ontologies (O_1 and O_2). Consequently, our structure similarity measure will be updated as follows:

$$S_{strc} = 2\ \left| r(O_{output}) \right| \Big/ \left[\left| r(O_{output}(RO_1)) \right| + \left| r(O_{output}(RO_2)) \right| \right]$$

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

where $|r(O_{output})|$ is the number of relationships in the output ontology (a neighbour/candidate result), and $|r(O_{output}(RO_i))|$ is the number of relationships in the immediate neighborhood of O_{output} in the inferred input ontology RO_i. This neighbourhood of O_{output} consists of the relationships of RO_i with at least one end (one of the edge's end) belonging to O_{output}. In other words:

- $|r(O_{output})|$ is the number of *common* relationships that connect *common* concepts to other *common* concepts (immediate neighbour). The resulting correspondences between entities (concepts/relationships) in RO_1 and RO_2 are what is meant by *common*.
- $|r(O_{output}(RO_1))|$ is the number of relationships in RO_1 with at least one end belonging to the *common* concepts belonging to O_{output}.
- $|r(O_{output}(RO_2))|$ is the number of relationships in RO_2 with at least one end belonging to the *common* concepts belonging to O_{output}.

4.4. Similarity recommendation technique

In this subsection, we illustrate the heuristic technique we used in our framework, adopted from [26], in order to offer users a ranked list (M_i) of appropriate techniques for the matching task at hand. The string/linguistic based techniques are evaluated as follows:

M = (number of concept pairs with the same label/synonym) / (max (C_1, C_2))

where the number of concept pairs with the same label/synonym represents the number of concepts' pairs that have the same label for the name based techniques and the same synonym for the linguistic based technique, such that $\{(c_1, c_2) | \, c_1 \in RO_1 \; and \; c_2 \in RO_2\}$.

We use labels for string based techniques and synonyms for linguistic based techniques. Further, max (C_1, C_2) stands for the maximum number of concepts, either in RO_1 or RO_2.

The structure based techniques are evaluated as follows:

M = (number of common concepts) / (max_number_of_nonleaf (C_1, C_2))

Where "number of common concepts" represents $\{(c_1, c_2) | \, c_1 \in RO_1 \; and \; c_2 \in RO_2\}$, such that both c_1 and c_2 have the same number of sub-concepts and the same depth. The max_number_of_nonleaf (C_1, C_2) denotes the maximum number of concepts that have sub concepts either in RO_1 or RO_2.

These heuristic techniques are not a precise measure of the real matching similarities of the entities for the input pair of ontologies. However, they can estimate the features of the two ontologies and provide a ranked list of the appropriate matching techniques.

All in all, for the matcher composition systems, (matching systems that use more than one similarity technique) using a recommended subset of their similarity measures list should improve the final matching results in terms of time and quality. Moreover, the recommendation techniques improve the overall running time, as it is unnecessary to reuse and combine all their underlying similarity measuring methods. Using only a recommended subset should decrease the average running times. Furthermore, the recommendation techniques can enhance the matching quality by excluding the unworkable similarity matching methods for a task at hand. For instance, if there is no string, linguistic, or structure similarity between a given pair of input ontologies, then including, combining, and aggregating the matching results retrieved by string, linguistic, or structure similarity measuring methods would affect the overall matching result quality in a negative manner.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

5. Related work

In [22], a partial set of similarity measures was applied to the input ontologies to suggest an initial result. Then, a neighbour search algorithm was introduced to investigate the neighbours of the initial result and suggest a final result from them. The algorithm can execute one or a combination of strategies, which were based on string-based, linguistic-based, and structure-based strategies. Finaly, the results were aggregated and suggested to the user.

The RiMOM system [26] integrates multiple strategies, such as edit distance, statistical learning, and three similarity propagation based strategies. It applies a strategy selection method in order to decide the strategy it will rely on to a greater extent. As a result, RiMOM combines the results using linear interpolation. The work proposed in [27] describes a tool which enhances ontology matching based on probabilistic inferences. The work in [28] describes an engine, which contains diverse libraries that support many matching algorithms and strategies. The matching results were combined by aggregating the results of the matchers applied to the given input ontologies. The final results were selected using a threshold value. In addition, a number of other systems use machine learning techniques for finding class similarity from instances [29]. Falcon-AO [30] has three elementary matchers: two linguistics matchers and one structural matcher. The results of Falcon-AO were mainly derived from the alignments generated from either linguistic or structural matchers, resulting in better matches. Otherwise, the Falcon-AO results will be generated by combining both linguistic and structural matchers with a weighting scheme. Some researchers propose similarity metrics between concepts in different ontologies based on their relationships to other concepts. For instance, a similarity metric between concepts in OWL ontologies [31] is a weighted combination of similarities of various features in OWL concept definitions, including their labels, domains and ranges of properties, restrictions on properties (such as cardinality restrictions), types of concepts, subclasses and superclasses, and so on. Algorithms such as the one proposed in [32] make use of derived graphs or alternative representations like pair-wise connectivity graphs.

It is less clear, however, which matching techniques are more suitable for a given pair of ontologies, or in which order they should be applied for best results. In [33] PROMPT, FCA-Merge, ODEMerge, and Chimaera were evaluated. Functionality, interoperability, and visualization were considered as part of the evaluation. However, the matching quality has not been considered. The research in [34] and [35] was focused on identifying the appropriate matching strategies. In [34], the idea of choosing appropriate strategies was based on the previous use of strategies. A number of related features, such as input, output, approach, usage, cost and documentation, were recognized. Then, the Analytical Hierarchy Process (AHP) was employed to identify suitable matching techniques. The fundamental idea of the AHP is that, for a characterized pair of ontologies to be matched, having a definition of the problem to be solved (merge ontologies to create a new one, match ontologies to compare profiles, match data etc.) along with particular requirements regarding the final application, one must decide which matching algorithms should be applied to satisfy these specifications and to obtain the desired output. [35] dealt with the problem of recommending an alignment strategy for a given alignment problem. The idea was based on an evaluation of existing alignment strategies on a number of small selected pieces from ontologies, and employs the evaluation results to offer recommendations. The main difference between our technique and the one in [34] is that their method requires knowledge about the suitability of each technique, which might not be available, while we rank the similarity techniques based on analyzing real ontologies. Moreover, we differ from [35] in that they consider small selected pieces of ontologies, while we consider entire ontologies.

All in all, there are four features, described as follows, which make our approach distinct from the aforementioned algorithms and systems:

1. Our matching results are guided by the fact that *n* entities at a time are matched to *m* entities.
2. The way similarities are transformed into mappings and measured using our multi-match technique in order to deal with a many-to-many match problem.
3. The neighbour search method we introduced can be viewed as an optimization algorithm to improve the efficiency of our multi-level match algorithm (MLMA).
4. The use of description logic reasoning techniques, in order to infer extra knowledge to be used for improving the structure similarity, as well as the process of choosing suitable matching techniques.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

6. Experiments and results

We have evaluated the performance of our proposed framework using two factors: quality and time. For the quality of matching results, we compare our result with 10 algorithms presented in the Ontology Alignment Evaluation Initiative OAEI-06 [36] and OAEI-07 [37]. For the running time, we conducted numerous experiments to show the impact of the proposed framework on the overall performance. We use MLMAR to refer to MLMA+, with the proposed recommendation analysis technique included.

All the tests have been performed on a Pentium 4, 2800, with 768 MB of RAM, running Windows XP, and with no applications running but a single matcher. To measure a match quality, we used the following indicators: *precision*, *recall*, *and F-measure*. *Precision* is a value in the [0, 1] range; the higher the value, the smaller the set of wrong mappings returned (false positives). *Recall* is a value in [0, 1]; the higher this value, the smaller the set of correct mappings not found (true positives). *F-measure* varies in the [0, 1] range, which is a global measure of the matching quality. The version computed here is the harmonic mean of precision and recall [38].

$$\text{precision} = \frac{\text{number\_of\_correct\_found\_alignments (by tools)}}{\text{number\_of\_found\_alignments (by tools)}}$$

$$\text{recall} = \frac{\text{number\_of\_correct\_found\_alignments (by tools)}}{\text{number\_of\_existing\_alignments (by experts)}}$$

$$\text{F\_Measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

In our comparison study, we used the OAEI 2007 benchmark test samples suite [39]. The test numbers of the ontologies we used from this benchmark suite included: 101, 103, 104, 205, 206, 209, 224, 228, 230, 232, and 239. The ontology 101 is the reference ontology, and, hence, in test case 101, the ontology number 101 is matched to itself, and in test 103 the ontology 101 is matched to ontology 103, etc.

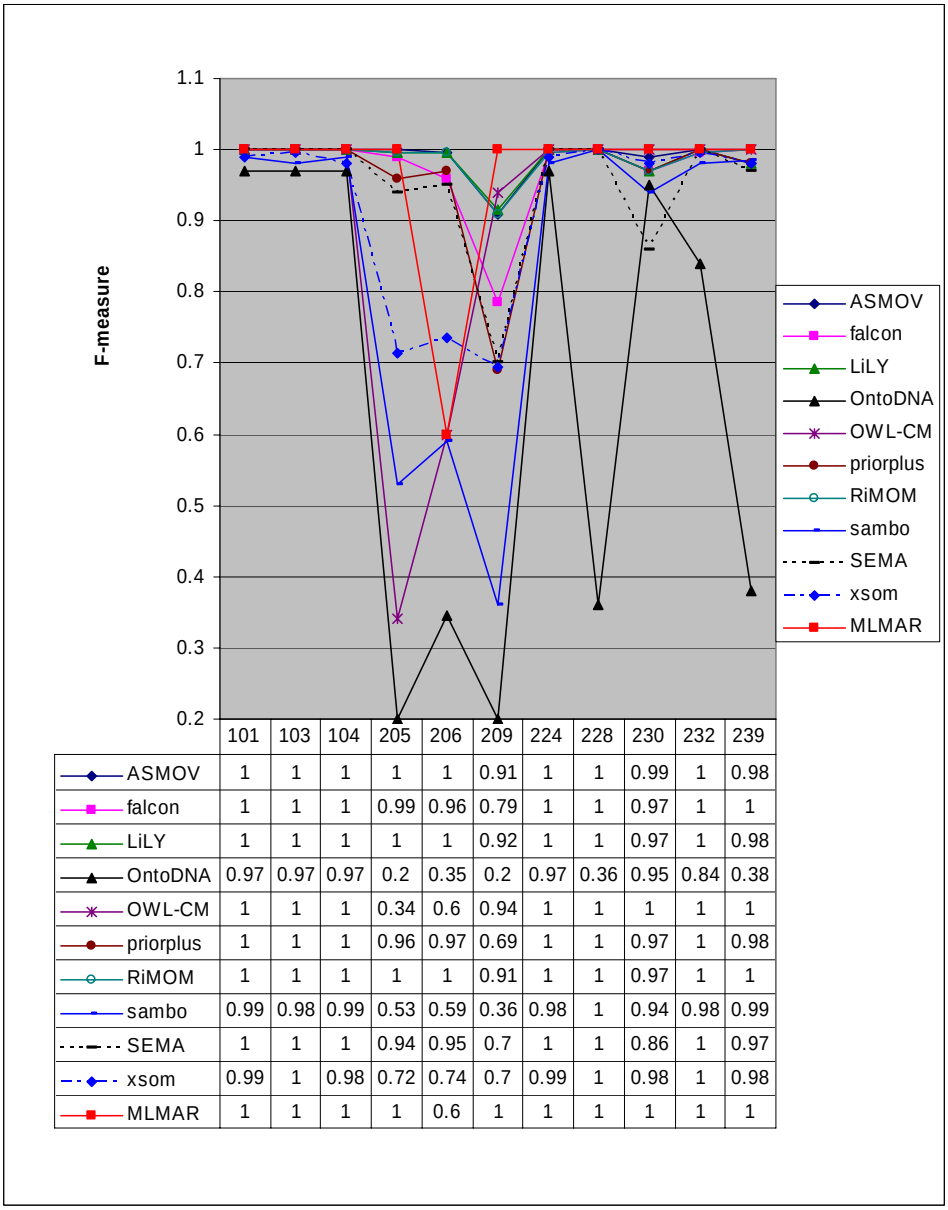
We noticed that all the systems we considered produced all the correct mappings, together with some additional unwanted mappings. The precision of our framework, on the other hand, did not fall below the *recall* value, such that no extra unwanted mappings were returned by our framework. The matching quality of the framework was mainly affected by the reuse of underlying matching techniques. For instance, if the reused matching techniques were not able to discover some matching entities, then the quality of matching results was affected accordingly. For instance, in test case 206, the reason that the matching results of our framework were not fulfilled was that it did not use translating techniques as one of its underlying techniques. So, the results were mainly obtained by both string and structure similarity measures, which were not able to obtain all the expected mapping sets. Fig. 11 compares the matching quality of our algorithm with the other 10 systems. In addition, Fig. 12 shows a time comparison indicating the scalability of our framework (please note the logarithmic scale).

Table 3 shows the initial estimation for the similarity measures, as well as a description of each test number. As it can be noted in test numbers 101, 103, 104, 224, 228, and 230, the modifications made to the reference ontology did not affect the string, linguistic, and structure similarities, and hence all the matchers obtained the highest similarity value. Accordingly, our framework will take advantage of not running all the matchers and will offer only a single matcher to the user. In such scenarios, we can often use string similarity because it is the most efficient one and it is the backbone for other matchers. Furthermore, in test numbers 205 and 209, the framework offers both the linguistic and the structure measures. In test 206, both string and structure similarity measures were used. Moreover, test 232 shows the best scenario, where there is no hierarchy, and using the DL reasoning technique (RACER), the structure similarity jumps from 0.0 to 0.7. Consequently, we applied both the string and structure similarities to this test. Lastly, in test number 239, the string similarity was applied.

In general, these recommendations greatly affected the performance time and placed our framework (MLMAR) at the top of the compared algorithms, based on average time. Also, they considerably improved the efficiency of MLMA+ by using only the recommended similarity techniques, rather than using all of them. The reason that MLMAR may not perform as a first rate system is that, in test cases 205, 206, 209, and 232, a combination of low

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

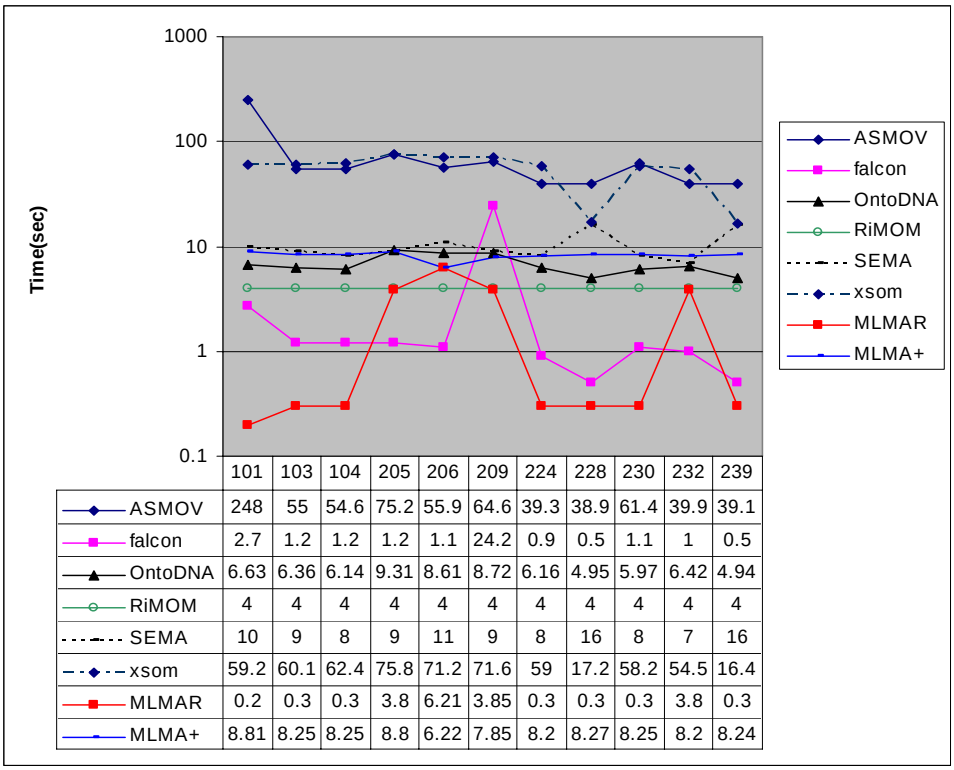
efficiency similarity measure techniques, such as linguistic and structure, was used. These ontologies, in some sense, were considered as a worse case scenario, where all matching techniques needed to be applied. However, in general, matching tools are equipped with numerous underlying similarity measuring techniques and using the recommended techniques will reduce the number of candidate techniques for a matching task at hand. Accordingly, the matching process time decreases remarkably.



	101	103	104	205	206	209	224	228	230	232	239
ASMOV	1	1	1	1	1	0.91	1	1	0.99	1	0.98
falcon	1	1	1	0.99	0.96	0.79	1	1	0.97	1	1
LiLY	1	1	1	1	1	0.92	1	1	0.97	1	0.98
OntoDNA	0.97	0.97	0.97	0.2	0.35	0.2	0.97	0.36	0.95	0.84	0.38
OWL-CM	1	1	1	0.34	0.6	0.94	1	1	1	1	1
priorplus	1	1	1	0.96	0.97	0.69	1	1	0.97	1	0.98
RiMOM	1	1	1	1	1	0.91	1	1	0.97	1	1
sambo	0.99	0.98	0.99	0.53	0.59	0.36	0.98	1	0.94	0.98	0.99
SEMA	1	1	1	0.94	0.95	0.7	1	1	0.86	1	0.97
xsom	0.99	1	0.98	0.72	0.74	0.7	0.99	1	0.98	1	0.98
MLMAR	1	1	1	1	0.6	1	1	1	1	1	1

Figure 11. Quality Comparison

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008



	101	103	104	205	206	209	224	228	230	232	239
ASMOV	248	55	54.6	75.2	55.9	64.6	39.3	38.9	61.4	39.9	39.1
falcon	2.7	1.2	1.2	1.2	1.1	24.2	0.9	0.5	1.1	1	0.5
OntoDNA	6.63	6.36	6.14	9.31	8.61	8.72	6.16	4.95	5.97	6.42	4.94
RiMOM	4	4	4	4	4	4	4	4	4	4	4
SEMA	10	9	8	9	11	9	8	16	8	7	16
xsom	59.2	60.1	62.4	75.8	71.2	71.6	59	17.2	58.2	54.5	16.4
MLMAR	0.2	0.3	0.3	3.8	6.21	3.85	0.3	0.3	0.3	3.8	0.3
MLMA+	8.81	8.25	8.25	8.8	6.22	7.85	8.2	8.27	8.25	8.2	8.24

Figure 12. Time Comparison

Test No.	String Similarity	Linguistic Similarity	Structure Similarity	Test Description
101	1	1	1	Ontology 101 is matched to itself
103	1	1	1	The generalization basically removes owl:unionOf and owl:oneof and the Property types (owl:TransitiveProperty).
104	1	1	1	This test compares the ontology with its restriction in OWL Lite (where unavailable constraints have been discarded).
205	0.125	0.85	1	Labels are replaced by synonyms. Comments have been suppressed.
206	0.1	0.1	1	The ontology translated into French
209	0.125	0.85	1	Synonyms are used
224	1	1	1	All individuals have been suppressed from the ontology.
228	1	1	1	Properties and relations between objects have been suppressed.
230	1	1	0.78	Some components of classes are expanded in the class structure (e.g., year, month, day attributes instead of date).
232	1	1	0.7	No Hierarchies and no instances
239	1	1	0.55	Flattened Hierarchy and no properties

Table 3. Initial estimations for the similarity measures

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

7. Conclusions and future work

In this paper, we studied the impact of different choices of strategies for matching ontologies. We proposed a framework for analysing the reused matching techniques. The study shows the importance of assisting the user with appropriate matching strategies. The user often has little or no idea about the suitability of matching strategies for a given matching task. As a result, the quality of matching results and processing times will be affected. The main advantages of the proposed framework are that (1) it is independent from other matching techniques, (2) it infers a hidden structure relationship among the entities of the input ontologies, and consequently makes the structure based similarity measure more precise, and (3) it considerably improves the matching process time.

We evaluated our framework against other approaches using, different pairs of ontologies. Our results indicate an improved performance of our proposed framework in terms of both quality and time. As future work, we plan to test our framework with different kinds of input ontologies and matching strategies, and study the impacts of different choices of matching algorithms on the quality and performance of the proposed framework.

Acknowledgment: The authors would like to thank the anonymous reviewers for their valuable comments on previous versions of this paper.

8. References

[1] N. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428-440, 2004.

[2] N. Noy and M. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. In Proc. *18th National Conference on Artificial Intelligence (AAAI)*, pp. 744-750, Edmonton (CA), 2002.

[3] N. Noy and M. Musen. Ontology versioning in an ontology management framework. *IEEE Intelligent Systems,* 19(4):6-13, 2004.

[4] A. Miles and D. Brickley. SKOS core guide. Note, W3C, 2005. Available at: http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/. Last accessed Oct 28th , 2008.

[5] C. Batini, M. Lenzerini, and S. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323-364, 1986.

[6] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3): 183-236, 1990.

[7] S. Spaccapietra and C. Parent. Conflicts and correspondence assertions in interoperable databases. *SIGMOD Record* 20(4)-49-54 1991.

[8] C. Parent and S. Spaccapietra. Issues and approaches of database integration. *Communications of the ACM*, 41(5): 166-178, 1998.

[9] P. Bernstein and E. Rahm. Data warehouse scenarios for model management. In Proc. *19 International Conference on Conceptual Modeling(ER)*, LNCS 1920, pp. 1-15 Salt Lake City (UTUS), 2000.

[10] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In Proc. *16th Meeting of the Information Processing Society of Japan (IPSJ)*, pp. 7-18, Tokyo (JP), 1994.

[11] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt,G. Schuster, H. Neumann, and S. Hubner. Ontology-based integration of information - a survey of existing approaches. In Proc. *IJCAI Workshop on Ontologies and Information Sharing*, pp. 108-117, Seattle (WA US), 2001.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

[12] D. Draper, A. Halevy, and D. Weld. The nimble integration engine. In Proc. *20th International Conference on Management of Data SIGMOD,* pp. 567-568, Santa Barbara (CA US), 2001.

[13] A. Halevy, N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka. Enterprise information integration: successes, challenges and controversies. In Proc. *24th International Conference on Management of Data (SIGMOD)*, pp. 778-787, Baltimore (MD US), 2005.

[14] A. Alasoud, V. Haarslev, and N. Shiri. A hybrid approach for ontology integration. In Proc. *VLDB Workshop on Ontologies-based techniques for DataBases and Information Systems (ODBIS)*, Trondheim, Norway, September 2-3, 2005.

[15] R. Agrawal and R. Srikant. On integrating catalogs. In Proc. *10th International World Wide Web Conference (WWW)*, pp. 603-612 Hong Kong (CN), 2001.

[16] R. Ichise, H. Takeda, and S. Honiden. Integrating multiple internet directories by instance-based learning. *18th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 22-30, Acapuico (MX), 2003.

[17] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: A new approach and an application. In Proc. *2nd International Semantic Web Conference (ISWC)*, LNCS 2870, pp. 130-145, Sanibel Island (FL US), 2003.

[18] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In Proc. *13th International Conference on Cooperative Information Systems (CoopIS)*, LNCS 3761, pp. 347-365, Agia Napa (CY), 2005.

[19] A. Alasoud, V. Haarslev and N. Shiri. A Multi Level Matching Algorithm for Combining Similarity Measures in Ontology Integration, In: M. Collard, JL Cavarero (ed.), *Ontologies-based DataBases and Information Systems*, LNCS 4623, Springer-Verlag, Berlin, Heidelberg, pp. 1-17, 2007.

[20] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (eds). *The Description Logic Handbook: Theory, Implementation, and Applications* (Cambridge University Press, 2003).

[21] V. Haarslev, and R. Möller. RACER System Description. In proc. of *International Joint Conference on Automated Reasoning, IJCAR'2001*, R. Goré, A. Leitsch, T. Nipkow (Eds.), Siena, Italy, Springer-Verlag, Berlin,pp. 701-705. June 18-23, 2001.

[22] A. Alasoud, V. Haarslev and N. Shiri. An Effective Ontology Matching Technique. In: A. An, S. Matwin, Z. W. Ras, and D. Slezak (ed.), in Proc. *17th Int'l Symp. on Methodologies for Intelligent Systems (ISMIS'08)*, LNAI 4994, Toronto, Canada. 585-590. 2008.

[23] Y. Kalfoglou and B. Hu, CROSI Mapping System (CMS). In Proc. *Integrating Ontologies Workshop*, Banff, Canada. 79-84. Oct. 2, 2005.

[24] W. Cohen, P. Ravikumar and S Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In Proc. *18th International Joint Conference on Artificial Intelligence (IJCAI-03)*: 73-78, 2003.

[25] T. Pedersen, S. Patwardhan and S. Patwardhan. WordNet::Similarity – Measuring the Relatedness of Concepts. In Proc. *of 19th National Conf.on AI*, San Jose, CA. 1024-1025. 2004.

[26] Y. Li, J. Li, D. Zhang, and J. Tang. Results of ontology alignment with RiMOM. In Proc. *Int'l workshop on Ontology Matching (OM)*, Athens, Georgia, U.S.A. 216-224. November 5, 2007.

[27] P. Mitra, N. Noy, and A. R. Jaiswal. OMEN: A probabilistic ontology mapping tool. In Proc. *Workshop on Meaning Coordination and Negotiation.* Hisroshima, Japan. 71-82. 2004.

[28] S. Massmann, D. Engmann, E. Rahm, and J. Tang. Results of ontology alignment with COMA++. In Proc. *International workshop on Ontology Matching (OM)* , Athens, U.S.A, November 5. 107-114. 2006.

[29] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology matching: A machine learning approach. In: S. Staab and R. (ed.), Handbook on *Ontologies in Information Systems,* pages 397–416. Springer, Heildelberg (DE), 2003.

[30] W. Hu, G. Cheng, D. Zheng, X. Zhong, and Y. Qu. The results of Falcon-AO. In Proc. *Int'l workshop on Ontology Matching (OM)*, Athens, Georgia, U.S.A, November 5. 160-167. 2007.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008

[31] J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-Lite. In Proc. *16th European Conference on Artificial Intelligence (ECAI-04)*, Valencia, Spain. 333-337. 2004.

[32] W. Hu, N. S. Jian, Y. Z. Qu, and Y. B. Wang. GMO: A Graph Matching for Ontologies. In Proc. *K-Cap Workshop on Integrating Ontologies*, pages 43-50, 2005.

[33] OntoWeb Consortium, A survey on ontology tools, *Deliverable 1.3*, 2002. Available at: http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3.pdf. Last accessed: Oct 2008.

[34] M. Mochol, A. Jentzsch , and J. Euzenat. Applying an Analytic Method for Matching Approach Selection, Proc. of the Int. *Workshop on Ontology Matching*. Athens, Georgia, USA. 37-48. 2006.

[35] H. Tan, P. Lambrix. A method for recommending ontology alignment strategies. In Proceedings of the *6th International Semantic Web Conference*, Busan, Korea. 494-507. 2007.

[36] J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, O. Svab, V Svatek., W. Robert, V. Hage, and M. Yatskevich. Results of the ontology alignment evaluation initiative 2006. Proc. of *ISWC workshop on Ontology Matching*, Athens, pages 73–95, 2006.

[37] J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Šváb, V. Svátek, W. Robert, V. Hage and M. Yatskevich. Results of the ontology alignment evaluation initiative. Proc. of the *ISWC workshop on Ontology Matching*, Busan, Korea, Nov. 2007. Available at: http://www.dit.unitn.it/~p2p/OM-2007/OM-2007proc.pdf . Last accessed: Aug 2008.

[38] H. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In Proc. *workshop on Web and Databases*. 221-237. 2002.

[39] *Ontology Alignment Evaluation Initiative (2007)*. Available at: http://oaei.ontologymatching.org/2007/benchmarks/. Last accessed: March 5, 2008.

JIS-0759-v3 Received: 13th May 2008 Revised: 29th October 2008