Ontology Matching using Multiple Similarity Measures

Thi Thuy Anh Nguyen and Stefan Conrad
Computer Science, Heinrich-Heine-University Düsseldorf, Universitätsstr. 1, 40225, Düsseldorf, Germany

Keywords: Ontology Matching, Lexical, Structure, Semantic.

Abstract: This paper presents an automatic ontology matching approach (called LSSOM - Lexical Structural Semantic-based Ontology Matching method) which brings a final alignment by combining three kinds of different similarity measures: lexical-based, structure-based, and semantic-based techniques as well as using information in ontologies including names, labels, comments, relations and positions of concepts in the hierarchy and integrating WordNet dictionary. Firstly, two ontologies are matched sequentially by using the lexical-based and structure-based similarity measures to find structural correspondences among the concepts. Secondly, the semantic similarity based on WordNet dictionary is applied to these concepts in given ontologies. After the semantic and structural similarities are obtained, they are combined in the parallel phase by using weighted sum method to yield the final similarities. Our system is implemented and evaluated based on the OAEI 2008 benchmark dataset. The experimental results show that our approach obtains good F-measure values and outperforms other automatic ontology matching systems which do not use instances information.

1 INTRODUCTION

Ontologies are applied in various application domains, for example, Semantic Web, information integration, e-commerce, and so on. Each ontology includes of sets of features such as names of concepts, properties, and relationships. Ontology matching is an operation taking two ontologies as input and returning a set of the correspondent relations between entities (called alignment) as output (Euzenat and Shvaiko, 2013). In general, a single measure can perform well (Tumer and Ghosh, 1995), however, it is not enough for determining the final alignment because the accuracy of results is not good for all kinds of domains (Kittler et al., 1998). For example, techniques based on lexical-based approach work well in ontologies in which class names having the same meaning are similar strings; however, they do not return satisfying final match results when class names use different strings for the same object having similar meanings (called synonym) or the same string for different objects (called polysemy). Therefore, to improve this situation, the matching systems should combine the results of several single similarity methods in order to achieve the final matching results instead of only one technique. Many ontology matching systems have been proposed so far based on lexicon, structures, instances, semantic, and combination of the above approaches (Anchor-Flood (Seddiqui and Aono, 2009), DSSim (Nagy et al., 2008), MapPSO (Bock and Hettenhausen, 2008), TaxoMap (Hamdi et al., 2008), GLUE (Doan et al., 2004), iMAP (Dhamankar et al., 2004), AROMA (David et al., 2006), NOM (Ehrig and Sure, 2004), QOM (Ehrig and Staab, 2004), SAMBO (Lambrix and Tan, 2006)). However, the efficiency of these systems depends on how and what the similarity methods between entities are applied. This paper takes into account the combination of different matching strategies including lexical-based, structure-based, and semantic-based methods to match ontologies and then obtains a final alignment. However, our approach does not consider instances data and user's feedback. In particular, it focuses on names, labels, comments, positions of concepts in the hierarchy, relationships between these concepts, and semantics based on WordNet. Each measure (e.g. lexical, semantic, and structure similarities) gets a similarity value and then these results are integrated together to yield the overall similarity. We use a weighted sum method to combine these measures in which a weight is assigned to each component. Our matching process uses sequential and parallel strategies in which the sequential phase is based on combining lexical and structural measures and parallel phase is relied on combining semantic measure and structural similarity values obtained in the previous step. The process of the manual ontology matching is usually consumptive and expensive

Nguyen, T. and Conrad, S..
Ontology Matching using Multiple Similarity Measures.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 603-611
ISBN: 978-989-758-158-8
Copyright © 2015 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

(Nezhadi et al., 2011; Noy and Musen, 2000). To reduce computational costs, it is really needed to propose an automatic ontology matching solution. In this paper, a framework (called LSSOM - Lexical Structural Semantic-based Ontology Matching method) is developed to align ontologies automatically.

The remainder of this paper is organized as follows. Section 2 overviews well-known systems. In section 3, the proposed ontology matching framework and a detailed description of our approach are provided. Section 4 discusses and evaluates the results. Finally, conclusions and future work are presented in section 5.

2 RELATED WORK

Normally, ontology matching systems can be produced by combining some different techniques. In this section, some of the most systems that have been applied so far to the task of matching based on structures in the hierarchy are discussed. In general, structural-based ontology matching systems consider information of structure in the hierarchy to find the matching entities of given ontologies, in which our proposed approach is also concentrated on. Another reason to chose these systems is that these systems are evaluated based on the same benchmark, the OAEI 2008 test set, which is convenient and fair in comparison.

CIDER (Gracia and Mena, 2008) applies ontology matching techniques to determine similarities between classes and properties based on the labels, structures, instances, and semantic in OWL or RDF ontologies. This system extracts terms based on their semantic by using an external resources such as WordNet up to a fixed depth. These terms are then computed the similarities based on lexical, taxonomical and relational techniques. In particular, the system employs Levenshtein edit distance metric for calculating similarities between labels and descriptions, a vector space model to achieve structural similarities, and an artificial neural network to integrate similarities. CIDER uses thresholds to extract one-to-one alignments.

Spider (Sabou and Gracia, 2008) combines two subsystems: CIDER and Scarlet where Scarlet investigates online ontologies automatically to obtain different types of relations between two concepts, for example, equivalence, subsumption, disjointness, and named relationships by applying derivation rules.

GeRoMeSuite (Kensche et al., 2007b) is a flexible model management tool using the metamodel GeRoMe (Kensche et al., 2007a). This system executes a number of matching techniques, for example, string-based, semantic-based, and structure-based methods. Additionally, GeRoMeSuite approach can load XML Schema and OWL ontologies and then performs alignment task.

MLMA+ (Alasoud et al., 2009) implements a matching algorithm in two levels where the structure-based method at the second level is followed by the name and linguistic similarities at the first level to obtain the final matching results. Besides, MLMA+ suggests a list of similarity measures which should be used to improve the overall similarity results. The final alignment of this system is a many-to-many cardinality.

Similar to the MLMA+ system, Anchor-Flood (Seddiqui and Aono, 2009) combines lexical-based, structure-based, and semantic-based similarity measures to calculate the correspondences between fragments in RDFS and OWL ontologies and then returns one-to-one alignments. However, this approach computes the similarity between terms through the Winkler-based string metric, which is different from MLMA+.

DSSim (Nagy et al., 2008) is an ontology matching framework using the structures in the hierarchy to find the confidence degrees between concepts and properties in the two large scale ontologies. In addition, the Monge-Elkan and Jaccard similarity measures are used for calculating similarities between strings and WordNet dictionary, which can be employed in determining semantics. DSSim system utilizes inputs as OWL and SKOS ontologies and gives outputs as one-to-one alignments.

Lily (Wang and Xu, 2008) combines three ontology matchers including Generic ontology matching method (GOM), Large scale ontology matching (LOM), and Semantic ontology matching (SOM) to compute one-to-one alignments. After preprocessing step, Lily applies measures to determine the similarity between entities in given ontologies including string-based, structure-based, semantic-based, and instance-based comparison algorithms. Then ontology mapping debugging technique is applied for the post-processing step to find the best possible matching solution.

MapPSO (Bock and Hettenhausen, 2008) combines the SMOA string distance, structure-based, WordNet-based and vector space similarity approaches, and ordered weighted average method to achieve one-to-one matching between concepts and properties in large OWL ontologies. In addition, the MapPSO approach considers the finding of the correspondences as an optimization problem.

TaxoMap (Hamdi et al., 2008) develops its previous version presented in (Zargayouna et al., 2007). In this new implementation, TaxoMap applies ontology matching techniques including the linguistic, 3-grams, structural similarity methods, and heuristic rules to obtain one-to-many cardinality between concepts. Besides, TaxoMap approach only concentrates on the labels and the relationships between the concepts in the hierarchy. The difference from the old version is that TaxoMap system runs on large scale ontologies.

Akbari&Fathian (Akbari and Fathian, 2010) is a combined approach to identify correspondences between entities in the source and target ontologies. This system computes the lexical similarities of class names, object properties and data properties, and the structural similarities of class names and then integrates similarity matrices to produce the final alignment by using the weighted mean.

AgreementMaker system (Cruz et al., 2009) matches concepts in the given ontologies by comparing their information available, for example, labels, comments, annotations, and instances. This system can deal with XML, RDFS, OWL, and N3 ontologies and then applies lexical, syntactic, structural, and semantic methods. The total values are aggregated through the weighted average method to match one entity to one entity.

ASCO (Bach et al., 2004) is an automatic ontology matching system. It uses RDF(S) ontologies and implements the linguistic and structural phases for finding the corresponding matches between entities in the considered ontologies. Besides, this approach applies a several well-known measures, for example, Jaro-Winkler, Levenshtein, Monger-Elkan, and computes the semantic similarities based on WordNet dictionary. The weighted sum method is then used in integrating the partial similarities to yield one-to-one or one-to-many alignments. ASCO2 (Bach and Dieng-Kuntz, 2005) is developed to work with OWL ontologies.

3 A COMBINED APPROACH FOR ONTOLOGY MATCHING

In this section, a framework for automatic ontology matching is described. Ontology matching is divided into two main strategies including the sequential and parallel compositions (Euzenat and Shvaiko, 2013) to obtain alignments between input ontologies. Our framework supports some matching approaches and also applies both strategies. Figure 1 shows the two phases in our framework. For the sequential phase, the lexical similarity values are applied to structural method to create a similarity matrix while the parallel composition phase is the combination of structure-based and semantic-based measures. The processes of similarity calculation return values between all pairs of concepts in two ontologies. All these values are stored in the structure and semantic matrices, respectively. Each pair of concepts from these two matrices is combined by using weights, then the overall similarity values are produced. Based on these results and a threshold *th*, the alignment is finally obtained. The similarity between two entities in the given ontologies depends on the similarities of their components and structures.

In this study, the considered components including names, labels, comments as well as relations and structures among entities in two ontologies are taken to calculate the similarity among these entities. There are many techniques to aggregate similarities, for example weighted product, weighted sum, weighted average, fuzzy aggregation, voting, and arguing (Doan et al., 2002; Euzenat and Shvaiko, 2013). Thanks to parameters, a various matching systems are composed by a set of individual measures to produce good alignments in an optimal and flexible way (Doan et al., 2004; Ehrig and Staab, 2004; Hariri et al., 2006; Jean-Mary et al., 2009; Lambrix and Tan, 2006; Madhavan et al., 2001). In fact, each ontology has its own characteristic. Therefore, depending on the features of ontologies and application domains are chosen these parameters should be changed.

Similar to AgreementMaker's (Cruz et al., 2009) and ASCO's (Bach et al., 2004) systems, the component and the combined similarity results in our work are computed by using weighted average and weighted sum methods in case they have more than one similarity degree, respectively. The details of our approach are explained in the subsequent sections.

3.1 Related Definitions

Let two ontologies be O_1 and O_2, entities belonging to these ontologies are e_1 and e_2, respectively. Entities usually consist of their names, denoted as $name(e_1)$ and $name(e_2)$, their labels, denoted as $label(e_1)$ and $label(e_2)$, and their comments, denoted as $comm(e_1)$ and $comm(e_2)$. The overall similarity value between two entities e_1 and e_2 is defined as $Oral\_Sim(e_1, e_2)$. This value results from the fundamental similarities achieved in two following phases.

- The sequential phase: in this phase, the structural similarities depend on the lexical similarities calculated in the previous step and the positions of entities in ontologies. The structural similarity between entities is defined as $Struct\_sim(e_1, e_2)$.

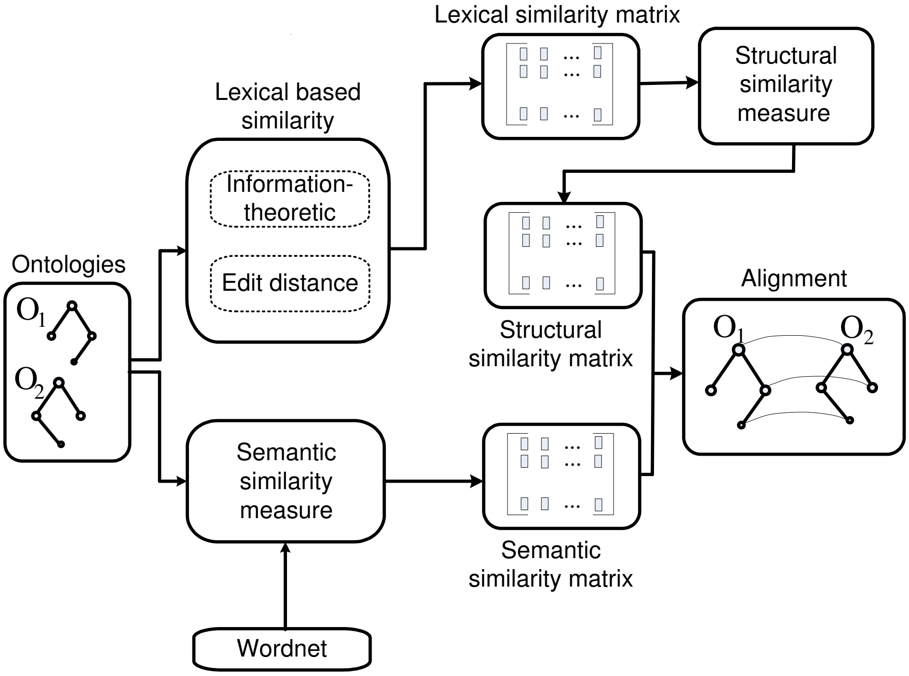

Figure 1: Framework for ontology matching.

The lexical similarity (as also called string-based similarity), $Lex\_sim(e_1, e_2)$, comes from cooperation between information-theoretic and edit distance approaches.

- The parallel phase: the result of this phase is the overall similarity integrated by structural and semantic measures multiplied by weights. The semantic similarity between entities (as also called knowledge-based similarity), $Sem\_sim(e_1, e_2)$, is determined by relationships, semantics, and structures of these entities in hierarchy of WordNet.

Both lexical and semantic similarity degrees depend on three component similarities including class names, labels, and comments of entities. In general, by assigning a weight to each of the component similarity, lexical and semantic similarities are described as follows (Cruz et al., 2009):

$$Sim(e_1, e_2) = \frac{\sum_{k=1}^{3} w_k * sim_k(e_1, e_2)}{\sum_{k=1}^{3} w_k} \quad (1)$$

where w_k are weights corresponding to features, $sim_k(e_1, e_2)$ are component similarities.

If two entities e_1 and e_2 do not contain any feature (for example, comments), the similarity of that feature is ignored. In this case, its corresponding weight w_k is assigned to 0. If one feature belongs to only one entity, its corresponding weight is set to 0 and then the similarity between two entities is defined as

$$Sim(e_1, e_2) = \max(Sim(e_1, e_2) - 0.05, 0) \quad (2)$$

The Eq. 2 is generated because of the reasons as follows. The first reason is that, if two concepts have the same features and the component similarity values between these concepts equal to 1, these concepts are chosen as centroid concepts and will be used for calculating the structural similarity values. On the other hand, these concepts do not match perfectly. Secondly, according to our intuition, the similarity degree of two concepts is based on a various features such as class names, labels, comments, and so on. Therefore, for non-existence feature, the similarity value $Sim(e_1, e_2)$ between two concepts is reduced to 0.05. Moreover, maximum function is applied to yield the nonnegative similarity values.

3.2 Measuring Structural Similarity

At the first time, when the lexical measure is used, the similarity for each component of each pair of entities in two ontologies is obtained. After getting the lexical similarity values between entities, the combination of lexical-based and structure-based metrics together is implemented.

3.2.1 Lexical-based Similarity

Lexical-based method is separately applied to names, labels, and comments of entities in two ontologies to achieve the similarities of each component of these entities.

- The similarities of class names and labels: normally, class names and labels are text chains such as words, the combination of a few words together without blank spaces, so they are short. The lexical similarity measure proposed in (Nguyen and Conrad, 2014) was applied for calculating the similarities of these class names and labels.

$$Lex\_sim(e_1,e_2) = \frac{\alpha(\max(|e_1|,|e_2|)-ed(e_1,e_2))}{\alpha(\max(|e_1|,|e_2|)-ed(e_1,e_2))+\beta(|e_1|+|e_2|-2\max(|e_1|,|e_2|)+2ed(e_1,e_2))} \quad (3)$$

where $ed(e_1,e_2)$ is Levenshtein measure. Let us consider the following example.

Example 1. Given names of two entities:
$name(e_1)$="Proceedings" and
$name(e_2)$="InProceedings".
The Levenshtein distance between these strings is 2.
In addition, $|Proceedings| = 11$,
$|InProceedings| = 13$,
$\max(|Proceedings|,|InProceedings|) = 13$.
By applying Eq. 3, the similarity between two strings "Proceedings" and "InProceedings" is:
$Lex\_name(Proceedings,InProceedings) = 0.733$

- The similarities of comments: classes usually contain comments describing these classes. However, comments are usually short texts too. To determine the similarity between two comments, two steps including normalization and comparison steps were executed. In the normalization step, we broke each comment into the ordered sets of tokens and then removed stop-words (for example, *the*, *a*, *and*, *of*, *to*), blank spaces, punctuation, symbols, replaces abbreviations (for example, *PC* → *Personal Computer*, *OS* → *Operating System*), and so on. Let *Comm*1 and *Comm*2 are two ordered sets of tokens of comments of two entities e_1 and e_2 in input ontologies O_1 and O_2, respectively. *Comm*1 and *Comm*2 can be presented as

  $Comm1 = \{comm(e_1)_1, comm(e_1)_2, ..., comm(e_1)_n\}$,
  $Comm2 = \{comm(e_2)_1, comm(e_2)_2, ..., comm(e_2)_m\}$.

  In the comparison step, these similarities are calculated in the same way as the similarities of class names and labels but applied to tokens. We will illustrate this idea with the following example.

Example 2. Given comments of two classes:
$comm(Proceedings)$="The proceedings of a conference." and
$comm(InProceedings)$="An article in a conference proceedings.".
The sets of ordered tokens of comments are:
*Comm*1={proceedings, conference} and
*Comm*2={article, conference, proceedings}.
The Levenshtein distance between two comments $comm(Proceedings)$ and $comm(InProceedings)$ is 2.
In addition, $|comm(Proceedings)| = 2$, $|comm(InProceedings)| = 3$,
$\max(|comm(Proceedings)|,|comm(InProceedings)|) = 3$.
Applying Eq. 3, the similarity between these two comments is:
$Lex\_comm(comm(Proceedings),comm(InProceedings)) = 0.143$

In our approach, each concept in the ontologies is represented by its descriptive information including its name, label, and comment. Applying the lexical similarity measure achieves the similarities between names, labels, and comments, respectively. After calculating lexical similarities between each concept in source ontology to all concepts in target ontology, three similarity matrices of classes, labels, and comments are obtained. By applying Eq. 1, the lexical similarity between e_1 and e_2 is presented as

$$\begin{aligned} Lex\_sim(e_1,e_2) = & \frac{w_n * Lex\_name(e_1,e_2)}{w_n+w_l+w_c} \\ & + \frac{w_l * Lex\_label(e_1,e_2)}{w_n+w_l+w_c} \\ & + \frac{w_c * Lex\_comm(e_1,e_2)}{w_n+w_l+w_c} \end{aligned} \quad (4)$$

where w_n, w_l, w_c, $Lex\_name(e_1,e_2)$, $Lex\_label(e_1,e_2)$, and $Lex\_comm(e_1,e_2)$ are weights and component similarities corresponding to features class names, labels, and comments, respectively.

The string-based measure shown in Eq. 4 is used for computing the similarity matrix representing lexical similarities between any two concepts with one from each ontology. This matrix is also employed to compute the similarity values of all pairs of concepts in ontologies based on the structure-based measure as discussed as follows.

3.2.2 Structure-based Method

The structural information is very important in ontologies because it contains the semantics of entities (Madhavan et al., 2001; Mitra and Wiederhold, 2004; Noy and Musen, 2001) and indicates the relationships between entities in these ontologies where these relationships are taken into account. Therefore, ontology matching based on structures in the hierarchy should be concerned. In this phase, a structure-based similarity metric proposed in (Nguyen and Conrad, 2013a) is applied for calculating similarity of each pair of concepts. The initial matrix is the lexical similarity matrix introduced in the previous subsection.

In case each entity in an ontology matches perfectly to one in another ontology ($Lex\_sim(e_1,e_2) = 1$), these entities are picked as centroid concepts proposed in (Wang et al., 2010). At that time, a set of centroid concepts is obtained.

The process of similarity calculation gives values between all pairs of concepts between e_1 and e_2, where e_1 and e_2 belong to the ontologies O_1 and O_2, respectively. All these similarity values are then stored in a structural matrix.

In fact, the structure of entities refers to how an entity is related to other. In addition, the structure contains a lot of the semantics but not the whole of the entities that they express as well as the similarity degree value between two arbitrary entities. Therefore, semantic measure described hereafter will be integrated with our structural technique together.

3.3 Semantic Similarity Measure

WordNet is considered as a background knowledge source to take semantics of terms. In this section, the proposed measure in (Nguyen and Conrad, 2013b) and the method in (Bach et al., 2004) were applied to calculate the semantic similarity. Class names, labels, and comments are conventionalized to sets by tokenizing them based on upper case, punctuation, symbols, and so on. Each token in a set (for example, comments) is compared with all tokens from another same type of set (any two tokens with one from each of set), and then the best similarities are chosen. The average of all best similarities of the same type is the semantic similarity between two objects. For example, the semantic similarity between two comments is described as:

$$Sem\_comm = \frac{\sum_{i=1}^{n} \max(comm(e_1)_i, Comm2)}{n+m} + \frac{\sum_{j=1}^{m} \max(comm(e_2)_j, Comm1)}{n+m} \quad (5)$$

where n and m are the numbers of tokens in the sets of comments $Comm1$ and $Comm2$, respectively.

Example 3. Using the two entities from the previous section:

$name(e_1)$=“Proceedings” and
$name(e_2)$=“InProceedings”.

The similarity between the two strings “Proceedings” and “InProceedings” is:

$Sem\_name(Proceedings, InProceedings) = 0.767$

Example 4. Given comments of two entities:

$comm(Proceedings)$=“The proceedings of a conference.” and

$comm(InProceedings)$=“An article in a conference proceedings.”.

The sets of ordered tokens of comments are

$Comm1$={proceedings, conference} and

$Comm2$={article, conference, proceedings}, respectively.

The similarity between two these comments is:

$Sem\_comm(Proceedings, InProceedings) = 0.870$

Semantic similarities between concepts result from the combination of component similarities.

$$Sem\_sim(e_1,e_2) = \frac{w_n * Sem\_name(e_1,e_2)}{w_n+w_l+w_c} + \frac{w_l * Sem\_label(e_1,e_2)}{w_n+w_l+w_c} + \frac{w_c * Sem\_comm(e_1,e_2)}{w_n+w_l+w_c} \quad (6)$$

where $Sem\_name(e_1,e_2)$, $Sem\_label(e_1,e_2)$, $Sem\_comm(e_1,e_2)$ are the semantic similarities of names, labels, and comments, respectively.

3.4 Combining Similarity Values

Applying weighted sum method presented in (Bach et al., 2004), the similarities are combined to get a overall similarity matrix representing the similarities of every pair of entities in given ontologies.

$$Oral\_Sim(e_1,e_2) = w_1 * Struct\_sim(e_1,e_2) + w_2 * Sem\_sim(e_1,e_2) \quad (7)$$

where $\sum_{t=1}^{2} w_t = 1$

In case the final similarity of two entities is equal or higher than the threshold, these entities are considered similarity. Consequently, one entity in an ontology can be similar to some entities in the other. It means, our system can output one-to-one and one-to-many alignments.

4 EVALUATION

The datasets was taken from OAEI benchmark 2008[1]

[1]http://oaei.ontologymatching.org/.

Table 1: Average Precision, Recall, and F-measure values of different approaches for three categories of ontologies in the benchmark OAEI 2008 (Pre.=Precision, Rec.=Recall).

Approaches	101-104		201-266		301-304		Average		F-measure
	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	
CIDER	0.99	0.99	0.97	0.57	0.90	0.75	0.97	0.62	0.76
Spider	0.99	0.99	0.97	0.57	0.15	0.81	0.81	0.63	0.71
GeRoMe	0.96	0.79	0.56	0.52	0.61	0.40	0.60	0.58	0.59
Anchor-Flood	1.0	1.0	0.96	0.69	0.95	0.66	0.97	0.71	0.82
Lily	1.0	1.0	0.97	0.86	0.87	0.81	0.97	0.88	0.92
DSSim	1.0	1.0	0.97	0.64	0.90	0.71	0.97	0.67	0.79
MapPSO	0.92	1.0	0.48	0.53	0.49	0.25	0.51	0.54	0.52
TaxoMap	1.0	0.34	0.95	0.21	0.92	0.21	0.91	0.22	0.35
MLMA+	0.91	0.89	0.57	0.52	0.68	0.65	0.69	0.65	0.67
Akbari&Fathian	0.98	0.95	0.78	0.74	0.87	0.84	0.86	0.83	0.84
LSSOM	1.0	1.0	0.90	0.72	0.98	0.74	0.96	0.80	0.87

to test and evaluate the performance of our system and other ones. Ontologies in this benchmark test were modified from the reference ontology 101 and can be divided into three categories: 101-104 (1xx), 201-266 (2xx), and 301-304 (3xx). Besides, ontologies 301-304 present real-life ontologies for bibliographic references found on the web. Since ontology 102 focus on wine which is irrelevant for the domain of bibliography, it is ignored. Ontology matching systems are chosen to compare including CIDER (Gracia and Mena, 2008), Spider (Sabou and Gracia, 2008), GeRoMe (Kensche et al., 2007a), Anchor-Flood (Seddiqui and Aono, 2009), Lily (Wang and Xu, 2008), DSSim (Nagy et al., 2008), MapPSO (Bock and Hettenhausen, 2008), TaxoMap (Hamdi et al., 2008), MLMA+ (Akbari et al., 2009), Akbari&Fathian (Akbari and Fathian, 2010), and ours (called LSSOM - Lexical Structural Semantic-based Ontology Matching method). The implementation of these approaches was evaluated based on the classical measures including Precision, Recall, and F-measure.

In our experimentation, the weights corresponding to features (class names, labels, and comments) and the partial similarity values (structural and semantic similarities) are assigned the fixed values 0.5, 0.25, 0.25, 0.5, and 0.5, respectively).

The Table 1 shows the average Precision, Recall, and F-measure values of categories and all ontologies in this benchmark test.

In the benchmark, ontologies in groups 1xx have good information, for instance, class names, labels, comments, and structures in the hierarchy. As a result, Precision and Recall values of all the systems are quite high, except that Recall of GeRoMe and MLMA+ are not good and Recall of TaxoMap is very low (0.34). Our approach and other systems (for example, Lily, Anchor-Flood, and DSSim) give Precision and Recall values of 1. Consequently, F-measure values are also equal to 1.

In the tests 2xx, ontologies have an absence of some features from the reference ontology. The tests 2xx include of three main groups: 201-210, 221-247, and 248-266. For tests 201-210, class names are arbitrary strings while some other information is lost such as labels and comments. In tests 221-247, the structures of the ontologies can be cut down to size or expanded. However, systems using structural technique also introduced good results even similar to the tests 1xx. Of course, Precision and Recall values of all the ontology matching systems are slightly worse than those for tests 101-104. The tests 248-266 have not good class names and structures, so the quality of the matchers becomes smaller in amount. As can be seen in the Table 1, Precision values in the tests 2xx are either higher than 0.90 or less than 0.6 while Recall values are quite low in general. There are only three systems having good values (Lily, Akbari&Fathian, and LSSOM).

For the real-world tests 301-304, Precision and Recall values are changed in the range between 0.15 (Spider) and 0.95 (Anchor-Flood) for Precision values and 0.21 (TaxoMap) and 0.84 (Akbari&Fathian) for Recall values.

In short, although average Precision value of TaxoMap system is high, its average F-measure value is the worst because its Recall value is also the worst. The MapPSO system is better than TaxoMap about the average F-measure, but it does not bring a good value. Anchor-Flood, Akbari& Fathian, and LSSOM approaches return average F-measure quite high: 0.82, 0.84, and 0.87, respectively. Lily is still considered the best ontology matching system. However, this system uses instances in matching. Our approach does not consider instances, which is different from Lily system. Our approach is highly significant compared to the other ontology matching systems which do not use instances data, and this is considered as one of the best ontology matchers on the OAEI 2008 benchmark test.

5 CONCLUSIONS AND FUTURE WORK

This paper presented an ontology matching approach to generate correspondences among entities of two input ontologies based on lexical-based, structure-based, and semantic-based measures in detail. In this work, our system implemented two phases which are sequential and parallel strategies. In the sequential phase, a structural similarity matrix applied the structure-based metric is produced by the following the lexical-based measure. Thanks to the weighted sum method, the combination of structural and semantic matchers in the parallel phase, and a certain threshold as well gives the final alignment. Consequently, our approach can induce one-to-one and one-to-many alignments. In addition, the results of our approach in the benchmark dataset of the 2008 OAEI were described. The experimental results demonstrate that our approach which automatically matches without instances achieves the high F-measure values.

Instances information of ontologies will be integrated in our approach in order to increase the accuracy of the final alignment. Moreover, machine learning techniques should be used to obtain a better quality of matching results. Our approach should also be tested on larger ontologies, evaluate its performance, and efficiency in the future work.

REFERENCES

Akbari, I. and Fathian, M. (2010). A Novel Algorithm for Ontology Matching. *Information Science*, 36(3):324–334.

Akbari, I., Fathian, M., and Badie, K. (2009). An Improved MLMA+ Algorithm and its Application in Ontology Matching. In *Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, pages 56–60. IEEE.

Alasoud, A., Haarslev, V., and Shiri, N. (2009). An Empirical Comparison of Ontology Matching Techniques. *Information Science*, 35(4):379–397.

Bach, T. L. and Dieng-Kuntz, R. (2005). Measuring Similarity of Elements in OWL DL Ontologies. In *The 1st International Workshop on Contexts and Ontologies (C&O) at the 20th National Conference on Artificial Intelligence (AAAI)*, pages 96–99.

Bach, T. L., Dieng-Kuntz, R., and Gandon, F. (2004). On Ontology Matching Problems for Building a Corporate Semantic Web in a Multi-Communities Organization. In *The 6th International Conference on Enterprise Information Systems*, pages 236–243.

Bock, J. and Hettenhausen, J. (2008). MapPSO Results for OAEI 2008. In *The 7th International Semantic Web Conference*.

Cruz, I. F., Antonelli, F. P., and Stroe, C. (2009). AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589.

David, J., Guillet, F., and Briand, H. (2006). Matching Directories and OWL Ontologies with AROMA. In *The 15th ACM International Conference on Information and Knowledge Management*, pages 830–831. ACM.

Dhamankar, R., Lee, Y., Doan, A., Halevy, A., and Domingos, P. (2004). iMAP: Discovering Complex Semantic Matches between Database Schemas. In *The 2004 ACM SIGMOD International Conference on Management of Data*, pages 383–394. ACM.

Doan, A. H., Madhavan, J., Domingos, P., and Halevy, A. (2002). Learning to Map between Ontologies on the Semantic Web. In *The 11th International Conference on World Wide Web*, pages 662–673. ACM.

Doan, A. H., Madhavan, J., Domingos, P., and Halevy, A. (2004). Ontology Matching: A Machine Learning Approach. In *International Handbooks on Information Systems*, pages 385–403. Springer.

Ehrig, M. and Staab, S. (2004). QOM - Quick Ontology Mapping. In McIlraith, S. A., Plexousakis, D., and Harmelen, F. v., editors, *The Semantic Web - ISWC 2004*, volume 3298 of *LNCS*, pages 683–697. Springer.

Ehrig, M. and Sure, Y. (2004). Ontology Mapping - An Integrated Approach. In Bussler, C. J., Davies, J., Fensel, D., and Studer, R., editors, *The Semantic Web: Research and Applications*, volume 3053 of *LNCS*, pages 76–91. Springer.

Euzenat, J. and Shvaiko, P. (2013). *Ontology Matching*. Springer, 2nd edition.

Gracia, J. and Mena, E. (2008). Ontology Matching with CIDER: Evaluation Report for the OAEI 2008. In *The 7th International Semantic Web Conference*.

Hamdi, F., Zargayouna, H., Safar, B., and Reynaud, C. (2008). TaxoMap in the OAEI 2008 Alignment Contest. In *The 7th International Semantic Web Conference*.

Hariri, B. B., Abolhassani, H., and Sayyadi, H. (2006). A Neural-Networks-based Approach for Ontology Alignment. In *The Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on Advanced Intelligent Systems*, pages 1248–1252.

Jean-Mary, Y. R., Shironoshita, E. P., and Kabuka, M. R. (2009). Ontology Matching with Semantic Verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251.

Kensche, D., Quix, C., Chatti, M. A., and Jarke, M. (2007a). GeRoMe: A Generic Role based Metamodel for Model Management. *Journal on Data Semantics VIII*, pages 82–117.

Kensche, D., Quix, C., Li, X., and Li, Y. (2007b). GeRoMe-Suite: A System for Holistic Generic Model Management. In *The 33rd International Conference on Very Large Data Bases (VLDB'07)*, pages 1322–1325.

Kittler, J., Hatef, M., Duin, R. P., and Matas, J. (1998). On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Lambrix, P. and Tan, H. (2006). SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(3):196–206.

Madhavan, J., Bernstein, P. A., and Rahm, E. (2001). Generic Schema Matching with Cupid. In *The 27th International Conference on Very Large Data Bases*, pages 49–58. Morgan Kaufmann.

Mitra, P. and Wiederhold, G. (2004). An Ontology-Composition Algebra. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, pages 93–113. Springer.

Nagy, M., Vargas-Vera, M., and Stolarski, P. (2008). DSSim Results for OAEI 2008. In *The 7th International Semantic Web Conference*.

Nezhadi, A. H., Shadgar, B., and Osareh, A. (2011). Ontology Alignment Using Machine Learning Techniques. *International Journal of Computer Science and Information Technology*, 3(2):139–150.

Nguyen, T. T. A. and Conrad, S. (2013a). Combination of Lexical and Structure-based Similarity Measures to Match Ontologies Automatically. In Fred, A., Dietz, J. L. G., Liu, K., and Filipe, J., editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 415 of *LNCS*, pages 101–112. Springer.

Nguyen, T. T. A. and Conrad, S. (2013b). A Semantic Similarity Measure between Nouns based on the Structure of WordNet. In *The 15th International Conference on Information Integration and Web-based Applications & Services (iiWAS2013)*, pages 605–609. ACM.

Nguyen, T. T. A. and Conrad, S. (2014). Applying Information-Theoretic and Edit Distance Approaches to Flexibly Measure Lexical Similarity. In *The 6th International Conference on Knowledge Discovery and Information Retrieval*, pages 505–511. SciTePress.

Noy, N. F. and Musen, M. A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *The National Conference on Artificial Intelligence (AAAI)*, pages 450–455.

Noy, N. F. and Musen, M. A. (2001). Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 63–70.

Sabou, M. and Gracia, J. (2008). Spider: Bringing Non-Equivalence Mappings to OAEI. In *The 7th International Semantic Web Conference*.

Seddiqui, M. H. and Aono, M. (2009). An Efficient and Scalable Algorithm for Segmented Alignment of Ontologies of Arbitrary Size. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):344–356.

Tumer, K. and Ghosh, J. (1995). Classifier Combining: Analytical Results and Implications. In *The AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pages 126–132. AAAI Press.

Wang, P. and Xu, B. (2008). Lily: Ontology Alignment Results for OAEI 2008. In *The 7th International Semantic Web Conference*.

Wang, Y., Liu, W., and Bell, D. A. (2010). A Structure-based Similarity Spreading Approach for Ontology Matching. In *The 4th International Conference on Scalable Uncertainty Management*, pages 361–374. Springer.

Zargayouna, H., Safar, B., and Reynaud, C. (2007). TaxoMap in the OAEI 2007 Alignment Contest. In *The 6th International Semantic Web Conference*.