GMO: A Graph Matching for Ontologies

Wei Hu, Ningsheng Jian, Yuzhong Qu, Yanbing Wang

Department of Computer Science and Engineering
Southeast University
Nanjing 210096, P. R. China

{whu, nsjian, yzqu, ybwang}@seu.edu.cn

ABSTRACT

Ontology matching is an important task to achieve inter-operation between semantic web applications using different ontologies. Structural similarity plays a central role in ontology matching. However, the existing approaches rely heavily on lexical similarity, and they mix up lexical similarity with structural similarity. In this paper, we present a graph matching approach for ontologies, called GMO. It uses bipartite graphs to represent ontologies, and measures the structural similarity between graphs by a new measurement. Furthermore, GMO can take a set of matched pairs, which are typically previously found by other approaches, as external input in matching process. Our implementation and experimental results are given to demonstrate the effectiveness of the graph matching approach.

Categories and Subject Descriptors

D.2.12 [**Software**]: Interoperability; I.2.6 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Graph*; I.5.3 [**Pattern Recognition**]: Clustering—*Similarity measures*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Semantic Web, Ontology Alignment, Graph Matching, Structure Similarity

1. INTRODUCTION

Web ontologies written by RDF Schema [7] or OWL [13] play a crucial role in the emerging Semantic Web, and ontology matching (or alignment) is necessary for establishing inter-operation between semantic web applications using different ontologies. Ontology matching can be seen as an operation that takes two graph-like structures and produces a mapping between elements of the two graphs that correspond semantically to each other. Due to the hardness of subgraph matching, ontology matching is a difficult issue. Some similarity-based approaches to ontology matching have been proposed in the literatures [3, 4, 8, 10, 12, 14]. As we know, structural similarity plays a central role in ontology matching. However, the existing approaches rely heavily on lexical similarity between labels of nodes and similarity of labels brought from thesaurus, e.g. WordNet [11]. And these approaches mixed lexical similarity with structural similarity.

In this paper, we present a new approach to ontology matching called GMO (Graph Matching for Ontologies). It uses bipartite graphs to represent ontologies, and measures the structural similarity between graphs by a new measurement. Usually, GMO takes a set of matched pairs, which are typically found previously by other approaches, as external input in the matching process, and output additional matching pairs by comparing the structural similarity. The input mapping given to GMO can be gained by variant approaches available, and may have big variance in size. So, our structural similarity is designed to be independent to lexical similarity, and the effectiveness of GMO is tested with variant sized input mapping. The rest of this paper is organized as follows: Ontology representation based on Bipartite Graph is presented in Section 2. A measure of structural similarity between a pair of web ontologies is proposed in Section 3. Our implementation is described in Section 4. Experimental results are reported in Section 5, and some comparison to related work is discussed in Section 6. Finally, Section 7 summarizes our work and outlines some of future work.

2. ONTOLOGY REPRESENTATION BASED ON BIPARTITE GRAPH

RDF model, a foundation of Semantic Web, has the nature of graph structure. OWL ontology can be mapped to an RDF Graph as stated in the fourth section of

Figure 1: The RDF graph (upper) and RDF bipartite graph (lower) of $\mathcal{O}_\mathcal{A}$

OWL S&AS [13]. Thus, in order to compute structural similarity between ontology entities with semantic correspondence, we adopt the graph structure approach to represent web ontology in this paper.

2.1 RDF Graph vs RDF Bipartite Graph

According to the well-established methods from graph theory, we utilize the RDF Bipartite Graph model [6] to represent web ontology instead of the RDF Graph model. In this paper, we use $\mathcal{O}_\mathcal{A}$, $\mathcal{O}_\mathcal{B}$, ... to denote web ontology. As an example, Fig. 1 shows the RDF graph and RDF bipartite graph of ontology $\mathcal{O}_\mathcal{A}$.

2.2 Matrix Representation of Ontology

Definition 1. Let $\tilde{G}_A$ be the RDF bipartite graph of $\mathcal{O}_\mathcal{A}$. The directed bipartite graph of ontology $\mathcal{O}_\mathcal{A}$, denoted by G_A, is a derivation of $\tilde{G}_A$ by replacing the "s" edges with edges pointing to statement nodes, and the "p" and "o" edges with edges pointing from statement nodes. The adjacency matrix of G_A is called the matrix representation of ontology $\mathcal{O}_\mathcal{A}$, denoted by A.

Note: The matrix has the following block structure,

$$A = \begin{pmatrix} 0 & 0 & A_{ES} \\ 0 & 0 & A_S \\ A_E & A_{OP} & 0 \end{pmatrix}, \qquad (MR)$$

where A_{ES} is a matrix representing the connections from external entities to statements; A_S is a matrix representing the connections from ontology entities within $\mathcal{O}_\mathcal{A}$ (internal entities) to statements; A_E is a matrix representing the connections from statements to external entities of $\mathcal{O}_\mathcal{A}$; A_{OP} is a matrix representing the connections from statements to internal entities.

Figure 2: The directed bipartite graph of ontology $\mathcal{O}_\mathcal{A}$

The external entities for an ontology are usually those vocabulary defined by RDFS or OWL, built-in data types, and data literals. In some cases, the external entities may include some common ones (e.g. FOAF terms) used in two ontologies to be compared, and the ones specified in an input mapping. Therefore, the separation of "external" and "internal" is relative to the current matching process. In addition, when external entities of $\mathcal{O}_\mathcal{A}$ are not used as subject in $\mathcal{O}_\mathcal{A}$ (as in Fig. 2), A_{ES} is a zero matrix.

As an example, Fig. 2 shows the directed bipartite graph of ontology $\mathcal{O}_\mathcal{A}$. The matrix representation of ontology $\mathcal{O}_\mathcal{A}$ is as follows:

rdfs : subClassOf													
rdfs : domain													
rdfs : range													
ex1 : supervise									0	0	0	1	1
ex1 : Scholastics									0	0	0	0	0
ex1 : Graduate									1	0	0	0	0
ex1 : Supervisor									0	1	0	0	0
ex1 : PhD_Candidate									0	0	1	0	0
stA1	1	0	0	0	1	0	0	0					
stA2	1	0	0	0	1	0	0	0					
stA3	1	0	0	0	0	1	0	0					
stA4	0	1	0	0	0	0	1	0					
stA5	0	0	1	0	0	1	0	0					

3. STRUCTURAL SIMILARITY MEASURE FOR ONTOLOGY

We propose in this section a measure of structural similarity for ontology. With this measure, the structural similarity matrix is illustrated by an example. Then, the updating equations for structural similarity matrix are refined according to entity classification. Finally, a process for similarity computing is outlined.

3.1 A Measure of Structural Similarity for Ontology

Here, we conduct the structural comparison between entities in the two directed bipartite graphs for given ontologies. The idea of our measure is as follows. Similarity of two entities from two ontologies comes from the accumulation of similarities of involved statements (triples) taking the two entities as the same role (subject, predicate, object) in the triples, while the similarity of two statements comes from the accumulation of similarities of involved entities (including external entities) of the same role in the two statements being compared. Our formulation of structural similarity (3.1) and (3.2) is based on the similarity measurement between directed graph vertices proposed in the literature

[1].

The formulation in [1] (the equation (1.2) on page 650), uses the following updating equations for similarity matrix:

$$X_{k+1} = BX_kA^T + B^TX_kA, \quad k = 0, 1, \ldots \quad (3.0)$$

where X_k is the $n_B \times n_A$ matrix of entries x_{ij} at iteration k, and A and B are the adjacency matrices of G_A and G_B respectively. It is proved in the literature [1] that the normalized even and odd iterations of this updating equation converge, and that the limit $Z_{even}(1)$ is among all possible limits the only one with largest 1-norm. This limit is taken as the similarity matrix. By making use of the mentioned work, we define our similarity formulation for ontology as follows.

Definition 2. (A measure of structural similarity for ontology) Let A and B be the matrix representation of ontologies $\mathcal{O}_A$ and $\mathcal{O}_B$ respectively. Let O_k represent the similarity matrix of ontology entities within B to ontology entities within A at iteration k, S_k represent the similarity matrix of statements within B to statements within A at iteration k, and E_{BA} mean the similarity matrix of the external entities of B to the external entities of A. Suppose A, B and X_k (the structural similarity matrix of B to A at iteration k) has the following block form respectively .

$$A = \begin{pmatrix} 0 & 0 & A_{ES} \\ 0 & 0 & A_S \\ A_E & A_{OP} & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 0 & B_{ES} \\ 0 & 0 & B_S \\ B_E & B_{OP} & 0 \end{pmatrix},$$

$$X_k = \begin{pmatrix} E_{BA} & & \\ & O_k & \\ & & S_k \end{pmatrix}$$

The updating equations for structural similarity matrix are defined as follows:

$$
\begin{aligned}
O_{k+1} &= B_S S_k A_S^T + B_{OP}^T S_k A_{OP} & (3.1) \\
S_{k+1} &= B_E E_{BA} A_E^T + B_{ES}^T E_{BA} A_{ES} \\
&\quad + B_{OP} O_k A_{OP}^T + B_S^T O_k A_S & (3.2)
\end{aligned}
$$

If the limits of normalized even of iterations with $O_0 = \mathbf{1}$ and $S_0 = \mathbf{1}$ (we denote by $\mathbf{1}$ the vector or matrix whose entries are all equal to 1) of this updating equations exist, we take the limit of O_k as the structural similarity matrix of ontologies $\mathcal{O}_B$ to $\mathcal{O}_A$.

Our formulation of structure similarity, (3.1) and (3.2), differ from the one in [1] in three aspects: (i) We use directed bipartite graph instead of directed graph, (ii)
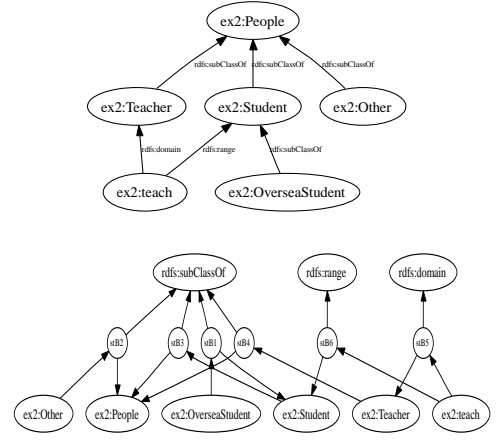
Figure 3: The RDF graph (upper) and directed bipartite graph (lower) of ontology $\mathcal{O}_B$

Nodes are classified in different categories, (iii) The similarities between external entities are kept unchanged during updating.

3.2 Structural Similarity Matrix by Example

Let $\mathcal{O}_A$ be the ontology described in section 2, $\mathcal{O}_B$ be the ontology depicted in Fig. 3.

The similarity matrix E_{BA} between external entities used in $\mathcal{O}_B$ and $\mathcal{O}_A$ is set in advance as

$$
\begin{array}{c}
\text{rdfs : subClassOf} \\
\text{rdfs : domain} \\
\text{rdfs : range}
\end{array}
\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
$$

By using the updating equations (3.1) and (3.2), we get the structural similarity matrix of $\mathcal{O}_B$ to $\mathcal{O}_A$ (after 12 iterations), as follows

	supervise	Supervisor	Graduate	Scholastics	PhD_Candidate
ex2 : teach	1	0	0	0	0
ex2 : Other	0	0.132	0.132	0	0.040
ex2 : People	0	0.001	0.220	1	0
ex2 : Student	0	0.579	0.884	0.025	0.040
ex2 : OsStudent	0	0.007	0.007	0	0.107
ex2 : Teacher	0	0.502	0.579	$9.05E-5$	0.040

3.3 Refinement of Structural Similarity

For most cases, we can classify the entities described in a given ontology as properties, classes, instances (individuals and data literals). In fact, this kind of classification is guaranteed to be success for OWL DL and FOL subset of RDFS.

After successful classification, we can refine the matrix representation form (MR) of ontology $\mathcal{O}_A$, in section 2,

43

as follows:

$$A_{ES} = \begin{pmatrix} A_{EPS} \\ A_{ECS} \\ A_{EIS} \end{pmatrix}, \quad A_S = \begin{pmatrix} A_{PS} \\ A_{CS} \\ A_{IS} \end{pmatrix},$$

$$A_E = (A_{EP}, \ A_{EC}, \ A_{EI}),$$

$$A_{OP} = (A_{POP}, \ A_{COP}, \ A_{IOP}).$$

where A_{EPS}, A_{ECS} and A_{EIS} represent the connections from external properties, classes and individuals to statements, respectively; A_{PS}, A_{CS} and A_{IS} represent the connections from internal properties, classes and individuals to statements, respectively; A_{EP}, A_{EC} and A_{EI} represent the connections from statements to external properties, classes and instances (including data literals); A_{POP}, A_{COP} and A_{IOP} represent the connections from statements to internal properties, classes and instances, respectively. As shown in above, we can also make the refinement of ontology $\mathcal{O}_{\mathcal{B}}$:

$$B_{ES} = \begin{pmatrix} B_{EPS} \\ B_{ECS} \\ B_{EIS} \end{pmatrix}, \quad B_S = \begin{pmatrix} B_{PS} \\ B_{CS} \\ B_{IS} \end{pmatrix},$$

$$B_E = (B_{EP}, \ B_{EC}, \ B_{EI}),$$

$$B_{OP} = (B_{POP}, \ B_{COP}, \ B_{IOP}).$$

The similarity matrix of external entities and the structure similarity matrix of ontologies have the diagonal structure as follows:

$$E_{BA} = \begin{pmatrix} EP_{BA} & & \\ & EC_{BA} & \\ & & EI_{BA} \end{pmatrix},$$

$$O_k = \begin{pmatrix} P_k & & \\ & C_k & \\ & & I_k \end{pmatrix}.$$

where EP_{BA}, EC_{BA} and EI_{BA} represent the similarity matrices of external properties, classes and individuals, respectively; P_k, C_k and I_k represent the similarity matrices of inner properties, classes and individuals, respectively.

The updating equations for structural similarity matrix are refined as follows:

$$
\begin{aligned}
P_{k+1} &= B_{PS} S_k A_{PS}^T + B_{POP}^T S_k A_{POP} & (3.3) \\
C_{k+1} &= B_{CS} S_k A_{CS}^T + B_{COP}^T S_k A_{COP} & (3.4) \\
I_{k+1} &= B_{IS} S_k A_{IS}^T + B_{IOP}^T S_k A_{IOP} & (3.5) \\
S_{k+1} &= B_{EPS}^T EP_{BA} A_{EPS} + B_{ECS}^T EC_{BA} A_{ECS} \\
&\quad + B_{EIS}^T EI_{BA} A_{EIS} + B_{EP} EP_{BA} A_{EP}^T \\
&\quad + B_{EC} EC_{BA} A_{EC}^T + B_{EI} EI_{BA} A_{EI}^T \\
&\quad + B_{POP} P_k A_{POP}^T + B_{COP} C_k A_{COP}^T \\
&\quad + B_{IOP} I_k A_{IOP}^T + B_{PS}^T P_k A_{PS} \\
&\quad + B_{CS}^T C_k A_{CS} + B_{IS}^T I_k A_{IS}. & (3.6)
\end{aligned}
$$

Note: The refined formulation of structure similarity has two advantages: (1) good computing performance due to the matrix computation with blocks; (2) avoiding the unnecessary computing of similarity between different kinds of entities, e.g. the ones between classes and properties.

4. IMPLEMENTATION

The graph matching for ontologies is implemented as an important component of our tool for aligning ontology, which is called Falcon-AO. In Falcon-AO v0.3, the input mapping to GMO is a set of matched pairs generated by another component, called LMO (A Linguistic Matching for Ontologies). In this section, the implementation of GMO is explained, followed by a brief introduction of LMO.

4.1 Matching Process of GMO

Suppose the ontologies to be matched are denoted by $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$. Given an initial mapping as input, by applying GMO, additional matching pairs will be generated. The implemented process of GMO is outlined as follows.

1. Parse $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$, and transform them to corresponding RDF bipartite graphs.

2. Classify entities (including anonymous ones) in $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$ as classes, properties and instances.

3. Coordinate $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$ using coordination rules described in 4.2.

4. Determine external entities for $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$ and setup external similarity matrix. In our implementation, the external entities are made of two parts: one includes built-in vocabularies of web ontology language, datatypes, data literals and URIs used in both $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$, and their similarity is pre-assigned; the other is identified by the input mapping.

5. Setup matrix representation for $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$.

6. Initialize the similarity matrices P_k, C_k, I_k, S_k with **1** (we denote **1** the matrix whose entries are all equal to 1, with corresponding rows and columns suitable to the context).

7. Run the even steps of iterations with updating equations (3.3)-(3.6) till some pre-defined convergence precision is reached.

8. Find a one-one mapping by means of the similarity matrices P_k, C_k and I_k.

9. Output additional matching pairs.

In the current implementation, the iteration times of updating structural similarity matrix is set to 12, which

is based on our primary experience. The finding of one-to-one mapping is finished when an estimated low similarity is reached.

4.2 Coordinating Ontologies with GMO

Ontologies to be matched may be represented differently, due to the heterogeneous ways in expressing semantics and the inference capability brought from ontology languages. Therefore, it is necessary to coordinate the two ontologies before mapping them.

Here, we outline several coordination rules, which are implemented in GMO. These rules can be classified into four categories presented as follows:

- Discarding: Some statements (triples) within an ontology may become redundant and/or worthless for computing structural similarity. For example, some typing statements such as (ex:A rdf:type owl:Class) become redundant after we successfully classify entities, and ontology header is worthless to structural comparing. Some rules are designed in GMO to discard such kinds of statements.

- Merging: Two entities could be stated to be same or equivalent to each other, e.g. (ex:A owl:equival entClass ex:B), then these entities should be merged in the RDF bipartite graph. There are some coordination rules to deal with this issue.

- Inference: In some situations, adding some inferred triples to the RDF bipartite graph with some inference rules would be helpful to structural comparing. For example, if there exist two triples, (ex:p owl:inverseOf ex:q) and (ex:q rdfs:domain ex:A), then, we could add one triple, (ex:p rdfs:range ex:A), if there is no triple to state the range of ex:p.

- List: To avoid heterogeneous in expressing a list using rdf:List, a List rule is presented. All members of a list are collected, and we use rdfs:member property to express the relation between the list and each of its members, instead of using RDF collection vocabularies(rdf:first, rdf:rest and rdf:nil).

More coordination rules will be introduced in later version of GMO. It is also worthy of note that there is a tradeoff between the cost of inference and the quality of mapping.

4.3 LMO – A Linguistic Matching for Ontologies

As is presented above, our GMO can be fed by an input mapping. In Falcon-AO v0.3, the input mapping to GMO is generated by LMO (A Linguistic Matching for Ontologies).

LMO includes two parts, one is based on string comparison, and the other is based on VSM (Vector Space Model). For string comparison, we use edit distance approach to calculate similarities between entities. For VSM, we treat the ontology entities (classes, properties and instances) as virtual documents. These virtual documents are constructed as "bags of terms" by using entity names, labels and comments, as well as neighbors' names or labels. Then, we can use VSM to gain the similarity matrix between entities. The details of LMO are out of the scope of this paper.

LMO brings some effectiveness to Falcon-AO, as demonstrated by the experimental results shown in section 5.2.

5. EXPERIMENTAL RESULTS

We have so far performed the GMO approach on OAEI 2005 benchmark test suite [1] and used standard information retrieval metrics to assess the results of our tests:

$$
\begin{aligned}
\text{Precision} &= \frac{\sharp\text{correct\_found\_alignments}}{\sharp\text{found\_alignments}}, \\
\text{Recall} &= \frac{\sharp\text{correct\_found\_alignments}}{\sharp\text{existing\_alignments}}, \\
\text{F} - \text{Measure} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\end{aligned}
$$

5.1 Effect of GMO

We tested the effectiveness of GMO on OAEI 2005 benchmark test cases, by taking some percentage of standard matched pairs as input mapping to GMO. The experimental results are demonstrated in Fig.4 by average precisions and recalls of all the test cases.

As shown below, with input matched pairs being fed increasingly, the GMO can find more additional correct matching pairs. The average precisions and recalls of test case #101-304 are indicated in y-axis, and the percentages of matched pairs as input mapping are shown in x-axis. It is worth noting that even with no input mapping, GMO still performs well, and the overall average precision and recall are 0.62 and 0.59 respectively.

We have categorized all the test cases into four groups: test case #101-104, #201-210, #221-266 and #301-304. Their average F-Measure are shown in Fig.5.

The results of test case #101-104 and #201-210 demonstrate that GMO is more suitable for those ontologies with similar structure than others. For these two categories of test cases, GMO still performs very nice without input mapping.

The weakness of GMO is also explicit. It performs not so well when the ontologies to be matched have a great
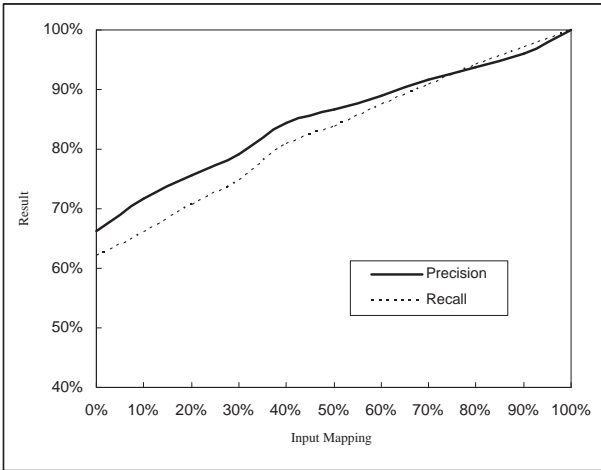
---

[1] http://oaei.inrialpes.fr/2005/benchmarks/

Figure 4: Average precision and recall

Figure 5: Average F-Measure of four categories

	101-104	201-210	221-266	301-304	Total
Prec.	1.0	0.96	0.86	0.93	0.89
Reca.	1.0	0.95	0.82	0.81	0.85
F-M.	1.0	0.95	0.83	0.86	0.87

210, and performs pretty good for other two categories of test cases.

6. RELATED WORK

Our presented work falls into the scope of similarity-based approaches to ontology matching. Logic based approach, e.g. Semantic Matching [5], and some others are beyond the scope of this paper. Here we present the closed-related work on similarity-based approaches. Among them, QOM [3] has a distinguished feature in efficiency with an emphasis on the alignment of RDFS ontologies. Anchor-PROMPT (included in PROMPT [12]) can produce new concept mapping by analyzing similar paths between a set of anchor matches, which are identified earlier (manually or automatically). OLA [4] and ASCO [8] are dedicated to the alignment of OWL ontologies (with an emphasis on OWL-Lite), and try to use as much as possible all of the information extracted from two given ontologies. In the literature [14], semantic-neighborhood matching is combined with word matching for class comparison. SF [10] is based on the idea that elements of two distinct models are similar when their adjacent elements are similar. The principle of our approach is similar to the basic idea of SF, but with very different measurement. In general, with these approaches and some others [9], entity features are setup based on labeled graphs or RDF graphs, and entity similarity is computed by counting feature-matches based on Tversky's contrast model [16], and then entity mapping is established based on (aggregated) similarities comparison and some specific heuristics rules (or user's interaction). Usually, those approaches mixed up lexical similarity and structural similarity, and/or heavily rely on lexical similarity to proceed with structural comparison.

Compared with them, our presented GMO approach uses bipartite graphs to represent web ontologies instead of using labeled graph or RDF graph, and measures the structural similarity between graphs by a new measurement. Our similarity model emphasizes the structural similarity based on the connection similarity, and does not depend on or mix up with lexical similarity. In addition, GMO approach can make use of a set of matched pairs found previously by other approaches. In fact, our similarity model also makes use of connections to "external" entities as well as matches between "ex-

difference in structure (e.g. test case #221-266 and #301-304). In such cases, more matched pairs should provided as input.

5.2 Performance of Falcon-AO

The partial experiment results of our Falcon-AO are presented in Table 1, and you will see that Falcon-AO performs well for all these test cases.

The matched pairs generated by LMO are fed into GMO as input. In this step, we must make a decision between precision and effect, that is to say, the precision of input matched pairs should be as high as possible, and as is shown above, the amount will also affect the matching effect. The details of the decision will be presented in our experimental paper accompanied.

As can be seen from Table 1, our tool Falcon-AO works very well for test case #101-104 and test case #201-

ternal" entities identified earlier automatically or manually. This idea is similar to Anchor-PROMPT, but the method of similarity computing is very different. Furthermore, our work is targeted to web ontologies, including RDFS and OWL. Currently, with an emphasis on FOL subset of RDFS and OWL Lite.

The experimental results reported on EON Ontology Alignment Contest [15] also show that those reported ontology alignment tools rely heavily on lexical similarity between labels of nodes (node identifier, rdfs:label, and rdfs:comment). For the five tests with test number 201, 202, 204, 205 and 206, where property and class names were disturbed, the average f-measure of these tools is 0.61. When there is very little similarity found from lexical analysis, some tools will fail to proceed with structural comparison effectively. Our experimental results, as in Fig.5, show that GMO works very well for test #201-210 with average F-Measure more than 0.95, though some improvement is needed to enhance the overall effectiveness of GMO.

7. CONCLUSION

The GMO approach (a Graph Matching for Ontologies) presented in this paper has two distinguished features from early works as follows:

(i) It uses bipartite graphs to represent ontologies instead of using labeled graph or RDF graph. The bipartite graph model can reveal the real structure of web ontologies to be compared.

(ii) A new measure of structural similarity for web ontology. This measure will play an important role in ontology matching, especially when lexical similarity could not be gained.

Our GMO approach has been implemented in our ontology matchers. The experimental results demonstrated the feasibility and the effectiveness of GMO. As illustrated in Section 5, GMO is irreplaceable when there is little gain from lexical comparison. In addition, GMO-based matcher can be integrated with other matchers. Therefore, GMO is also a complement to other related work in the area of ontology matching.

As we pointed out in Section 3, ontologies should be coordinated before comparison due to the heterogeneous ways in expressing semantics and the inference capability brought from ontology languages. However, it is not easy to select appropriate coordination rules due to the tradeoff between the cost of inference and the quality of mapping. Another issue is the interaction between similarities at conceptual layer and instance layer. In current stage of our implementation, we separate these two layers, and use matches at conceptual layer as inputs to compute similarities at instance layer.

In the case of comparing instance-intensive ontologies, machine learning (e.g. GLUE [2]) is a promising approach to make use of instance information in aligning classes or properties.

As part of future research, we are going to improve the GMO approach and related algorithms in some aspects, e.g. coordination issue and layers issue. We plan to integrate GMO with techniques in machine learning and natural language processing to realize more powerful ontology matchers.

Acknowledgments

8. REFERENCES

[1] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, P. Van Dooren. A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. *SIAM Review, 46(4) (2004) 647-666*

[2] A. Doan, J. Madhavan, P. Domingos, A. Halevy. Learning to Map between Ontologies on the Semantic Web. *The Eleventh International World Wide Web Conference (WWW'2002), Hawaii, USA (2002)*

[3] M. Ehrig, S. Staab. QOM-Quick Ontology Mapping. *In Proc. of the 3rd International Semantic Web Conference (ISWC2004), November, 2004, Hiroshima, Japan, LNCS 3298, Springer (2004)*

[4] J. Euzenat, P. Valtchev. Similarity-based Ontology Alignment in OWL-Lite. *Proc. of ECAI, 333-337, 2004*

[5] F. Giunchiglia, P. Shvaiko. Semantic Matching. *The Knowledge Engineering Review Journal 18(3) (2004) 265-280*

[6] J. Hayes, C. Gutiérrez. Bipartite Graphs as Intermediate Model for RDF. *International Semantic Web Conference (2004) 47-61*

[7] P. Hayes (ed.). RDF Semantics. *W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/rdf-mt/*

[8] B. T. Le, R. Dieng-Kuntz, F. Gandon. On Ontology Matching Problems for Building a Corporate Semantic Web in a Multi-communities Organization. *ICEIS (4) (2004) 236-243*

[9] A. Maedche, S. Staab. Measuring Similarity between Ontologies. *EKAW (2002) 251-263*

[10] S. Melnik, H. Garcia-Molina, E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. *In Proc. of ICDE (2002) 117-128*

[11] A. G. Miller. Wordnet: A Lexical Database for English. *Communications of the ACM 38(11) (1995) 39-41*

[12] N. F. Noy, M. A. Musen. The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *International Journal of Human-Computer Studies 59 (2003) 983-1024*

[13] P. F. Patel-Schneider, P. Hayes, I. Horrocks (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. *W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/owl-semantics/*

[14] M. A. Rodriguez, M. J. Egenhofer. Determining Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Trans. Knowl. Data Eng. 15(2) (2003) 442-456*

[15] Y. Sure, O. Corcho, J. Euzenat, T. Hughes (eds.). Proceedings of the 3rd International Workshop on Evaluation of Ontology Based Tools (EON2004). *Vol. 128, CEUR-WS Publication. November (2004)*

[16] A. Tversky. Features of Similarity. *Psychological Review, 84(4) (1977) 327-352*