AUTOMATYKA • 2011 • Tom 15 • Zeszyt 3

Marcin Pietroń*, Kazimierz Wiatr**

The Ontology Alignment System Based on Algorithms Using Data Similarity Metrics

1. Introduction

Matching of data from different sources is a very important task in complex IT systems and global web applications. In many applications it is a necessary issue to be dealt with, because of increasing amount of data from various sources, but concerning the same scope. Therefore, there is a lot of research in this area to create algorithms which could conduct these operations automatically [1, 4, 5, 6, 13, 15]. Such algorithms have two main criteria: precision of alignment and computional complexity. The ontology alignment and integration algorithms have the same goals and have to deal with the similar problems.

The ontology in information technology is used to conceptualize domain knowledge. It enables to create sharable data. The ontologies are heterogeneous and distributed because they are defined and created by different authors and organizations. Therefore they can have different characteristics. The concepts, relations, or instances with the same meaning may have different labels in different ontologies or the same label may represent different meanings.

A lot of research has been made on automating the process of ontology alignment [6, 7, 9, 12, 18]. It focused on specific applications or providing universal algorithms for various applications. Most of them are based on calculating similarities between entities of two ontologies by using following information in ontologies: entity names, taxonomy structures, entity comments, entities instances etc. These methods can be classified into two categories:

– single strategy – all available information are used to compute a single similarity function;
– combining multiple strategies – similarity functions are defined based on different types of information, then proper method of alignment is chosen after combining the results of different similarities.

* AGH University of Science and Technology, ACK Cyfronet, ul. Nawojki 11, 30-950 Krakow, Poland
** AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronic, Department of Electronics, al. A. Mickiewicza 30, 30-059 Krakow, Poland

In recent years, the combination method builds on multiple strategy becomes most popular, due to its ease of extension and flexibility [4, 13]. Multistrategy algorithm means that depends on values of metrics different branch of alignment algorithm could be computed. Despite that fact several problems for ontology alignment are still needed to further investigation. There is no effective method to automatically determine when a single strategy or multiple strategy is best solution for alignment task.

In a combined method (multi-strategy), there is no algorithm to define how each strategy should impact the alignment issue, which alignment methods to be used, what ontology and entities information to use in the similarity calculation and how to combine methods. The publication presents the ontology alignment based on OWL format. It describes ontology alignment framework build on similarity metrics. It is mainly structural and linguistic based similarity. These global similarities could decrease complexity of algorithm and increase alignment quality by determine proper strategy for alignment task.

It is worth to say that only few systems have explored how to automatically combine multiple strategies to improve the matching effectiveness.

The rest of this paper is organized as follows, section 2 provides a description of the formal definition of ontology, section 3 presents data matching and ontology alignment processes. Section 4 elaborates on architecture of our system, especially metrics computation and results of using these metrics. Section 5 provides conclusions and further research.

2. Ontology

In computer science, an ontology is a formal representation of knowledge as a set of concepts, and the relationships between those concepts [10]. An ontology provides a shared vocabulary, which can be used to model a domain. Ontologies are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics etc. as a form of knowledge representation about the world ,some part of it or some domain. Common components of ontologies include:

- Individuals: instances or objects.
- Classes: collections, concepts, classes in programming, types of objects.
- Attributes: properties, features, parameters that objects (and classes) can have.
- Relations: ways in which classes and individuals can be related to one another.
- Function terms: complex structures formed from certain relations that can be used in place of an individual term in a statement.
- Restrictions: formally stated descriptions of what must be true for some assertion to be accepted as input.
- Rules: sentence that describe the logical inferences between concepts and properties.

The ontology in formal mathematical language is described as a 6-tuple:

$$O = \{C, P, H^C, H^P, A^O, I\} \tag{1}$$

where:

C is the set of concepts;

P is the set of properties;

H^C defines the hierarchical relationships $H^C \subset C \times C$ $(c_i, c_j) \in H^C$ means that conept c_i is subconcept of c_j;

H^P, $H^P \subset C \times C$ defines the hierarchical relationships between properties;

H^O is set of axioms;

I is set of instances of concepts and properties.

There are few languages in which it is possible to describe ontologies. Most popular of them is OWL language [19]. Using *rdfs:Class*, *rdfs:subClassOf* and *rdfs:Property*, *rdfs:subPropertyOf* we can define concepts with their subconcepts and properties with their subproperties, respectively. The *rdfs:domain* and *rdfs:range* enables define what concepts have which property and which concepts are values of property. Figure 1 describes example part of the ontology definition.

```
<owl:Class rdf:ID="Collection">
 <rdfs:subClassOf rdf:resource="#Book" />
 <rdfs:label xml:lang="en">Collection</rdfs:label>
 <rdfs:comment xml:lang="en">A book that is collection of texts or articles.</rdfs:comment>
   <owl:Restriction>
     <owl:onProperty rdf:resource="#chapters" />
     <owl:allValuesFrom rdf:resource="#Chapter"/>
 </owl:Restriction>
 <owl:Restriction>
     <owl:onProperty rdf:resource="#parts" />
   <owl:allValuesFrom rdf:resource="#InCollection"/>
 </owl:Restriction>
</owl:Class>
<owl:DatatypeProperty rdf:ID="reviewed">
   <rdfs:domain rdf:resource="#Reference" />
   <rdfs:range rdf:resource="&xsd;string" />
   <rdfs:label xml:lang="en">howReviewed</rdfs:label>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="chapters">
   <rdfs:domain rdf:resource="#Reference"/>
   <rdfs:range rdf:resource="#Chapter" />
   <rdfs:label xml:lang="en">chapters</rdfs:label>
   <rdfs:comment xml:lang="en">The chapters of a book (monograph or collection).
   </rdfs:comment>
</owl:ObjectProperty>
<Journal rdf:about="#a246119474">
   <rdfs:label>Journal of Web Semantics</rdfs:label>
   <foaf:name rdf:datatype="&xsd;string">Journal of Web Semantics</foaf:name>
   <shortName rdf:datatype="&xsd;string">JWS</shortName>
</Journal>
```

Fig. 1. Part of ontology definition

From the Figure 1 we can read following information: ‘Collection’ is a concept and subconcept of ‘Book’, has properties ‘chapters’ and ‘parts’, ‘reviewed’ is datatype property with domain ‘Reference’ and range ‘string’, ‘chapters’ is object property with domain ‘Reference’ and range ‘Chapter’, ‘#a246119474’ refers to a ‘Journal’ instance and rest of the tags like comment, label etc. are metadata of each entity.

The family of OWL language is divided into three main types: OWL Lite, OWL DL and OWL Full [23]. Our system and algorithms support OWL DL specification.

3. Data matching and ontology alignment

Finding correspondences between elements of data schemas is required in many applications. This task is named as *matching* process. Schema and instance matching is needed for a variety of types of schemas including UML, SQL schemas and ontologies. Matching problems often are different because of various matched schemas or instances. Because of this diversity, applications that rely on matching are often built from scratch. The main goal is to build effective universal algorithm managed to match various source of schemas. Therefore some of these algorithms can be adapted to ontology alignment process.

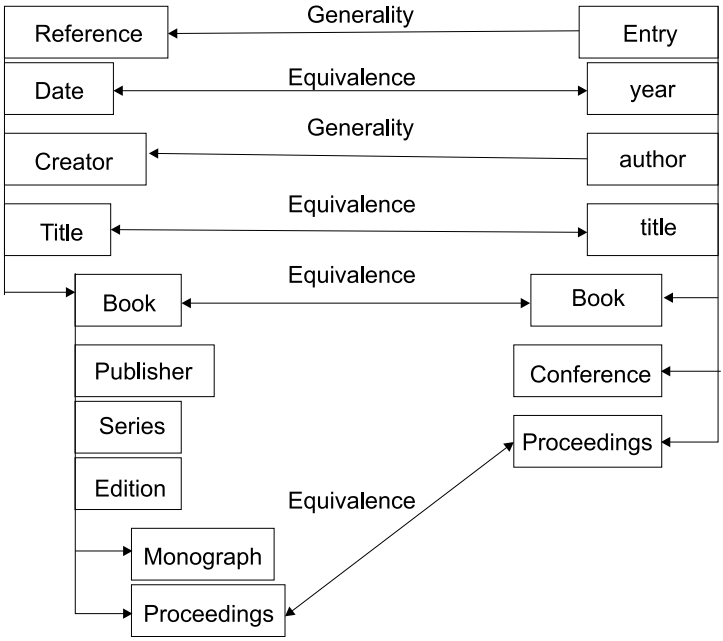

Fig. 2. Example of alignment process

Ontology alignment is process that takes two ontologies as input and as an output delivers alignment result between entities of the input ontologies. In formal description ontology alignment problem is described as follows (Fig. 2):

$$Align(O_1,O_2)=\begin{pmatrix} (e_{i1},e_{i2},con_i,relation_i) \\ e_{i1}\in O_1,e_{i2}\in O_2,con_i\in[0,1] \\ relation_i\in\{exact,narrower,broader,overlap\} \end{pmatrix} \quad (2)$$

The 4-tuple $(e_{i1},e_{i2},con_i,relation_i)$ in $Align(O_1,O_2)$ represents that entity e_{i1} in O_1 is aligned to entity e_{i2} in O_2 with the confidence con_i and alignment type *relation*. The alignment can have four types: exact (1:1 alignment), narrower (subentity alignment), broader (superentity alignment), overlap (partially overlapping alignment), con_i is value of alignment. In our work, we deal with 1:1 alignment.

Table 1 describes two aligned ontologies from Figure 3. It shows aligned pair of entities and types of alignment.

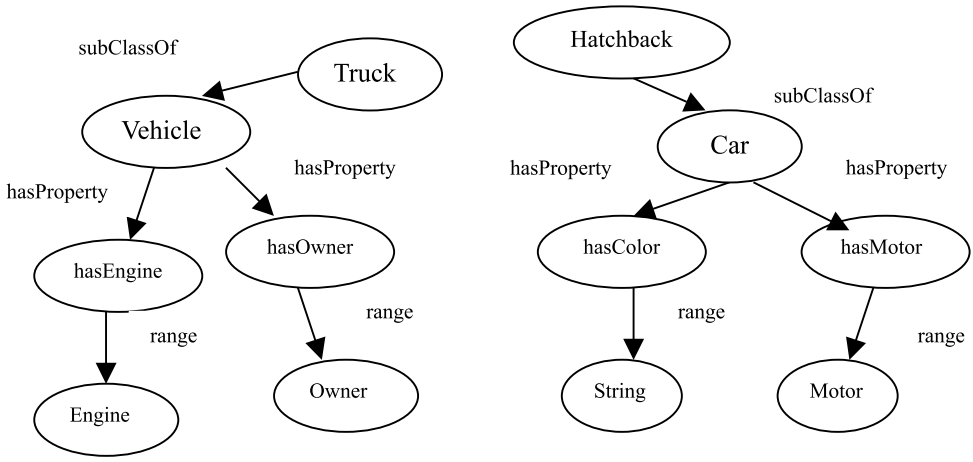

Fig. 3. Two simple ontologies to align

Table 1
Aligned two ontologies from Figure 3

Entity	Mapped entity	Confidence	Relation
Truck			
Vehicle	car	1.0	exact
hasOwner			
hasEngine	hasMotor	1.0	exact
Engine	Motor	1.0	exact
Owner			

4. Prototype of algorithm based on similarity metrics

4.1. Similarity metrics

The similarity metrics is new approach to ontology alignment. The ontologies can differ on two levels of their architecture. First is entity level, including its label, comment etc. This is called text or linguistic based level. The second one is whole structure of entities in ontologies. It is called structure or taxonomy level.

The benchmarks of testing quality of alignment can be divide to four main groups:

- similar in label description and hierarchy structure;
- similar in hierarchy structure, label description are random, in different languages etc.;
- similar in label description, different hierarchy and structure;
- different in both label description and hierarchy structure.

The proposed metrics in this article are designed to detect degree of similarity in each level. After that it is possible to estimate to which class alignment task belongs to and which type of algorithm should be taken or what combination of algorithms. The additional knowledge is with which weights final alignment between entities should be computed.

Our framework implements following text metrics:

$$lbl\_sim = \frac{|lbl\_sim\_concepts| + |lbl\_sim\_properties|}{\max(|C_1| + |P_1|, |C_2| + |P_2|)} \tag{3}$$

where *lbl_sim_concepts* and *lbl_sim_properties* are number of similar concepts and properties labels.

$$meta\_sim = \frac{|meta\_sim\_concepts| + |meta\_sim\_properties|}{\max(|C_1| + |P_1|, |C_2| + |P_2|)} \tag{4}$$

meta_sym is similarity of metadata (additional knowlegde about entities such comments etc.). *Meta_sim_concepts* (number of concepts with similar metadata) and *meta_sim_properties* (number of properties with similar metadata) are computed by comparing same words in metadata:

$$meta\_sim\_entity(e_1, e_2) = \frac{|same\_words(e_1, e_2)|}{|words(e_1)| + |words(e_2)|} \tag{5}$$

The metadata of entities are similar if *meta_sim_entity* factor is higher then treshold set by the user.

$$syns\_sim = \frac{|syns\_sim\_concepts| + |syns\_sim\_properties|}{\max(|C_1| + |P_1|, |C_2| + |P_2|)} \tag{6}$$

The *syns_sim* is number of synonyms ocurred between concepts and properties in two ontologies. *Syns_sim* informs that is necessary to use WordNet database during alignment. For example if *lbl_sim* and *meta_sim* are low but *syns_sim* factor is higher it is information that ontologies has high linguistic similarity despite label and entity metadata similarity not indicating it.

For simple part of ontologies from Figure 4 text similarity factors are:*lbl_sim* = 1/3 and *syns_sim* = 1/3.

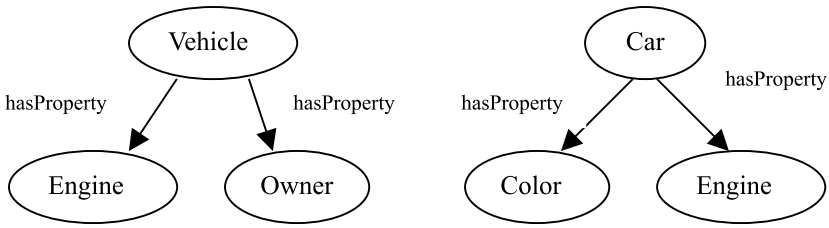

Fig. 4. Two part of ontologies on which text metrics is computed

The structure similarity metric is based on comparison between graph structures of matched ontologies. While reading input ontologies framework transforms OWL ontology to graph structure. Each node receives number which describes his level in graph structure. This data is necessary to compute structure similarity metric. Structure metric is computed by compare nodes of equal level value in both ontologies. The nodes of the same types are compared (class nodes with class nodes, properties nodes with properties). The following equation computes node structure similarity:

$$struct\_sim(N_1, N_2) = \frac{|match\_edges|}{|E_{N_1}| + |E_{N_2}|} \tag{7}$$

where:

$|match\_edges|$ is the number of matched outcome edges of compared nodes,
$|E_{N_1}|$ and $|E_{N_2}|$ are number of outcome edges of compared nodes.

The same nodes (with matching value equal to 1.0) are sum up and metrics is computed:

$$struct\_sim\_metric(O_1, O_2) = \frac{|struct\_sim(N_1, N_2) = 1.0|}{\max(|C_1| + |P_1|, |C_2| + |P_2|)} \tag{8}$$

4.2. Text-based algorithm

The text-based algorithms concentrates on linguistic context of ontologies' entities. It compares names (labels) of entities and whole metadata. Apart from computing text-based similarity, framework enables to get synonyms, hypernyms etc. from WordNet database.

It is necessary as was said in previous subsection to check if text-based similarity is low that some labels of entities between align ontologies are synonyms or they have another linguistic relation. If the text metrics are more than set treshold in framework then text_based-algorithm is run. Text similarity between entities is computed by following formula:

$$text\_sim(e_1,e_2) = \\ = \frac{meta\_sim \times ham\_meta\_dist(e_1,e_2) + lbl\_sim \times ham\_lbl\_dist(e_1,e_2)}{lbl\_sim + meta\_sim} \tag{9}$$

where:

$ham\_meta\_dist(e_1,e_2)$ – hamming distance of metadata,

$ham\_lbl\_dist(e_1,e_2)$ – hamming distance of labels.

4.3. Structure-based algorithms

The graph based similarity algorithm implemented in our framework is Similarity Flooding [16]. The similarity flooding (SF) algorithm is iterative graph based algorithm for matching schemas and instance as well. The engine of algorithm is adapted to deal with OWL ontologies. At the start ontologies are read from files by OWL-API to direct labeled graph (Fig. 3). The SF get two direct labeled graphs and connects them to one Pairwise Connectivity Graph (PCG). Figure 5 describes PCG graph created from ontologies from Figure 3. The formula of PCG graph is as follows:

$$((x,y),p,(x',y')) \in PCG(A,B) \Leftrightarrow (x,p,x') \in A \wedge (y,p,y') \in B \tag{10}$$

where: (x, p, x') is a triple where x is a source node, y is a target node and p is label of the direct edge connecting them (property). The node in PCG graph is candidate for an alignment pair. From PCG graph similarity propagationa graph calles IPG (Induced Pairwise Graph) is constructed. It is decribed on Figure 6. Each edge in IPG graph has weight that indicates how much similarity of source node will be propagated to target node. Similarities between entities are computed by following equation:

$$\begin{aligned} &\sigma^{i+1} = \frac{1}{z}(\sigma^0 + \sigma^i + \varphi(\sigma^0 + \sigma^i)) \\ &\varphi(\sigma^0 + \sigma^i) = \sum_{j=1}^{m} w_j \sigma_j^i \\ &z = \max(\sigma^{i+1}) \end{aligned} \tag{11}$$

where: σ^0, σ^i, σ^{i+1} are similarities at the start, the ith and (i+1) iterations. φ() is the function calculating increase the similarities in (i+1) iteration, z is normalization factor (value of maximal increase in iteration).

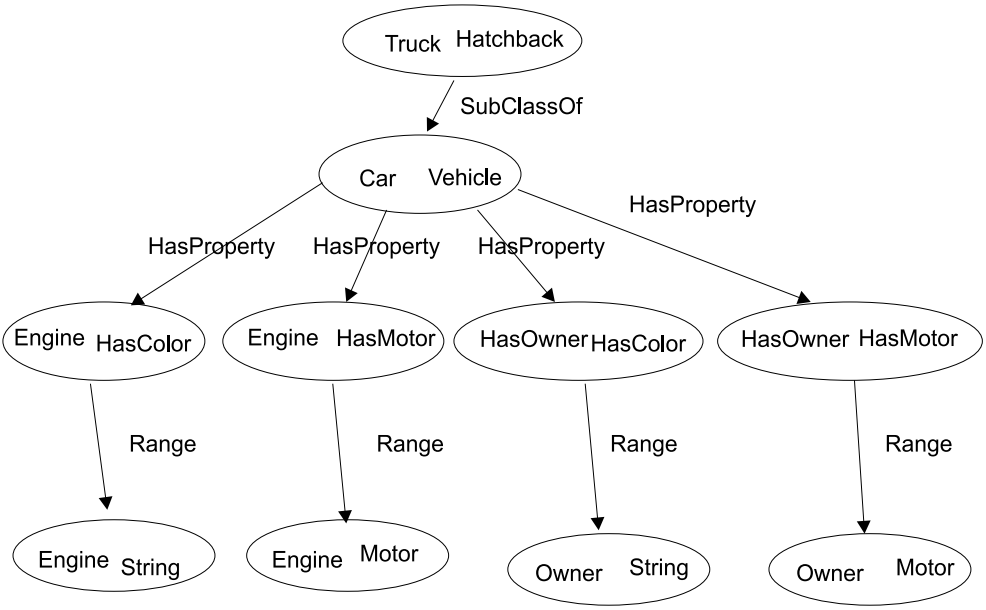

Fig. 5. PCG graph from ontologies in Figure 3

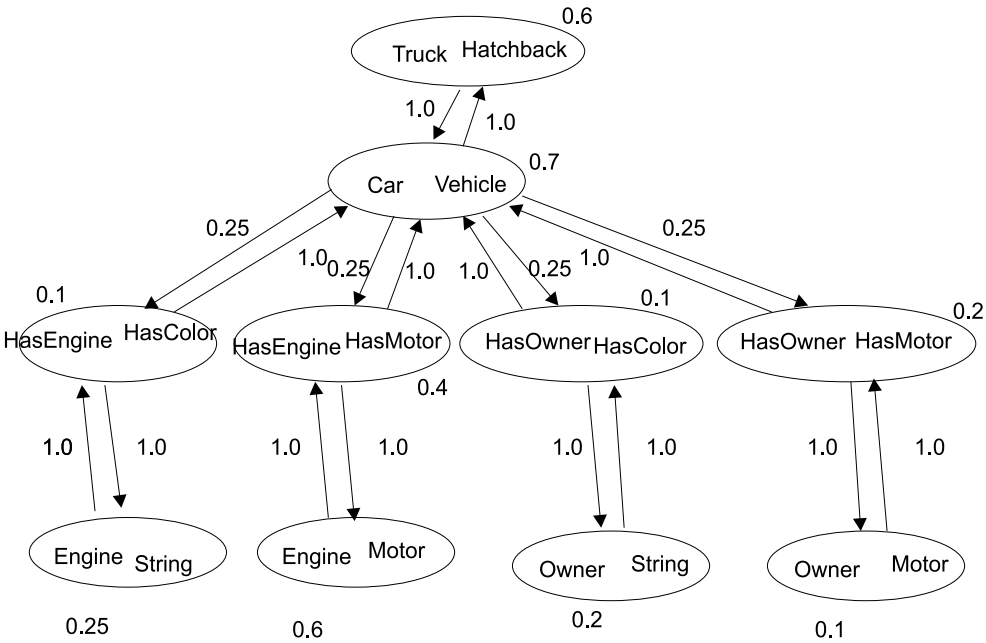

Fig. 6. IPG created from PCG graph

The initial values of pair similarity in PCG nodes are set by text-based algorithm or if text similarity is low and text strategy is not run start values of PCG nodes are set by computing $g\_sim_2(e_1,e_2)$ factor.

4.4. Prototype of ontology alignment framework

Figure 7 describes our ontology alignment framework. The two ontologies are input to the system. Framework reads them using OWL-API. Then preprocessing is done: text and structure similarity metrics are computed and if is needed WordNet operations to get synonyms, hypernyms etc.

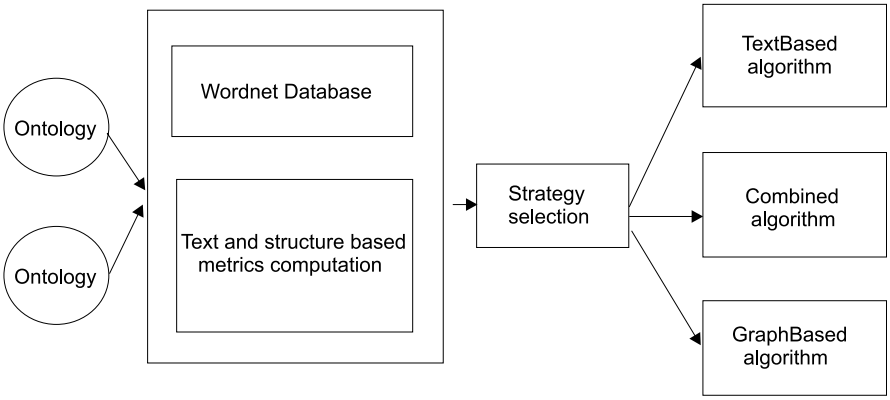

Fig. 7. Architecture of framework

After that, strategy to align is chosen. The strategy selection is most important part of the system. It decides which type of algorithm suite to alignment task. If text-similarity metric is lower then user set *text_sim_treshold* and graph-similarity metrics is higher then *struct_sim_treshold* then only graph-based algorithm is run in opposite situation text-based algorithm is chosen. If both metrics are higher than set treshold values then combiantion of algorithms is run and similarity is equal:

$$sim(e_1,e_2) = text\_factor \times text\_sim(e_1,e_2) + struct\_factor \times struct\_sim(e_1,e_2) \quad (12)$$

where:

$text\_factor$ – is text metric similarity

$struct\_factor$ – is structural metric similarity

In each strategy it is instance matching undertaken. The algorithm checks if instances exist in source ontologies. If answer is positive it compares them. It compares each – to – each instance from two matched ontologies. If values of instances are similar (similar means that matching is higher than set treshold parameter) then all properties are compared. If compared values of instances or properties are similar then similarity between two instances

types or properties are increased. In the end pairs with highest values (similarities) are chosen as matched entities. Figure 8 and Table 2 describe accurately how matched entities are extract from instance definitions.

```
<Journal rdf:about="#a70981683">
    <rdfs:label>Cybernetics and Control Theory</rdfs:label>
    <name rdf:datatype="&xsd;string">Cybernetics and Control Theory</name>
</Journal>
<Publisher rdf:about="#a85849488">
    <rdfs:label>The MIT Press</rdfs:label>
    <name rdf:datatype="&xsd;string">The MIT Press</name>
    <address>
      <Address>
        <city rdf:datatype="&xsd;string">Cambridge</city>
        <state rdf:datatype="&xsd;string">MA</state>
        <country rdf:datatype="&xsd;string">US</country>
      </Address>
    </address>
</Publisher>
<Journal rdf:about="#a362042121">
    <rdfs:label>International journal of intelligent system</rdfs:label>
    <name rdf:datatype="&xsd;string">International journal of intelligent system</name>
    <shortName rdf:datatype="&xsd;string">IJIS</shortName>
</Journal>
```

Ontology 1

```
<qsdsquj rdf:about="#a70981683">
    <dszabdza rdf:datatype="&xsd;string „>Cybernetics and Control Theory</dszabdza>
</qsdsquj>
<zauio rdf:about="#a85849488">
    <dszabdza rdf:datatype="&xsd;string „>The MIT Press</dszabdza>
    <qzd>
     <qzddj>
        <zdzndh rdf:datatype="&xsd;string">Cambridge</zdzndh>
        <zdnzadh rdf:datatype="&xsd;string „>MA</zdnzadh>
        <zadszabnds rdf:datatype="&xsd;string „>US</zadszabnds>
     </qzddj>
    </qzd>
</zauio>
<qsdsquj rdf:about="#a362042121">
    <dszabdza rdf:datatype="&xsd;string">International journal of intelligent system</dsz-
abdza>
    <dsza rdf:datatype="&xsd;string „>IJIS</dsza>
</qsdsquj>
<qsdsquj rdf:about="#a362042121">
    <dszabdza rdf:datatype="&xsd;string">International journal of intelligent
system</dszabdza>
    <dsza rdf:datatype="&xsd;string „>IJIS</dsza>
</qsdsquj>
```

Ontology 2

Fig. 8. Two matched ontologies with instances definitions

Table 2
Instance matching of ontologies from Figure 8

Entity_O1	Entity_O2	No. of matchings
Journal	qsdsquj	2
name	dszabdza	3
Publisher	zauio	1
address	qzd	1
Address	qzddj	1
city	zdzndh	1
state	zdnzadh	1
country	zadszabnds	1
shortName	dsza	1

As was said earlier framework generates 1:1 alignment. Therefore after all computations it must extract best 1:1 set of alignments. In our framework it is done by stable marriage algorithm [10, 16].

5. Results

The prototype of our system was tested on simple OWL DL ontologies to validate metrics and algorithms. Then we run benchmark ontologies from OAEI (Ontology Alignment Initiative) [22].

Table 3
OAEI benchmarks for testing ontology alignment

Name of test sets	Test sets	Ontology characteristics	Number of ontologies
D1	101-104	similar in both label description and hierarchy structure	4
D2	201-210	Similar in hierarchy structure	10
D3	221-247	Similar in label description	18
D4	248-266	different in both label description and hierarchy structure	15
D5	301-304	real world ontologies	4

The benchmarks of testing quality of ontologies alignment can be divide to five main groups (Tab. 3):

- similar in label description and hierarchy structure;
- similar in hierarchy structure, label description are random, in different languages etc.;
- similar in label description, different hierarchy and structure;
- different in both label description and hierarchy structure;
- real word ontologies.

Precision and Recall are two parameters to evaluate alignment results:

- precision (P) – percentage of correctly discovered alignments in all discovered alignments
- recall (R)– percentage of correctly discovered alignments in all correct alignments

$$P = \frac{|m_a \cap m_m|}{|m_a|} \tag{13}$$

$$R = \frac{|m_m \cap m_a|}{|m_m|} \tag{14}$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{15}$$

where:

m_a – alignments discovered by framework,
m_m – alignments assigned manually before algorithm execution.

In our first experiments we run framework on all sets apart from D5 (real world ontologies). The results are shown in Table 4.

Table 4
Experimental results of algorithm

Test set	$F1$
D1	100.0
D2	89.5
D3	99.6
D4	48.6

In D2 set where hierarchy is similar in all ontologies except 201, 205, 206 and 207 only structure algorithm is run. In 201 meta_sim metric is high (names are random) and has main influence on linguistic similarity. In 205 high density of synonyms is detected by syns_sim metric. The ontologies 206 and 207 are created in different language (french),

lbl_sim is higher than treshold and linguistic similarity is calculated in final equation. In D3 most cases are run with text algorithm because low structure simialrity. Our text algorithm deals with all cases. The last set D4 is run by both algorithms text and structure. Metrics in most cases are between 0.25 and 0.75.

6. Conclusions and future work

As was described the ontology alignment is quite complex process. Metrics computations and combined strategy makes alignment more effective beacuse of eliminating not necessary computations. Metrics were divided to text-based metrics and structure-based metrics. The text based metrics concentrate on linguistic similarities between entities from aligned ontologies, structure-based focus on taxonomy similarities. After these computations it is possible to choose combination of algorithms to match ontologies. It is worth to say that this approach can increase quality of alignment and very often reduce complexity of this process.

Further research will especially be focused on finding accurate impact of computed metrics on quality of alignment. The work should also concentrate on testing framework on representative sets of ontologies and adding testing module to it. The next challenge is to improve performance of the framework and reduce complexity of matching algorithm as it is possible.

References

[1] Aumueller D., Massmann S., Rahm E., *Schema and ontology matching with COMA++*. Proc. ACM SIGMOD, 2007, 1–12.

[2] Bernstein P.A., Melnik S., *Model Management 2.0:Manipulating Richer Matchings*. Proc. ACM SIGMOD, 2007, 1–12.

[3] Ding P.A., Finin S., Joshi A., Pan R., Cost R.S., Peng Y., *Swoogle: A Search and Metadata Engine for the Semantic Web*. Proc. International Conference Information and Knowledge Management (CIKM'04), 2004, 652–659.

[4] Do H.H., Rahm E., *Coma, a System for Flexible Combination of Schema Matching Approaches.* Proc. 28th International Conference Very Large Databases (VLDB'02), 2002, 610–621.

[5] Ehrig M., Staab S., Sure Y., *QOM:Quick Ontology Mapping.* Proc. International Conference Semantic Web (ISWC'04), 2004, 683–697.

[6] Ehrig M., Sure Y., *Foam-Framework for Ontology Alignment and Alignment Results of the Ontology Alignment Initiative.* Proc. Workshop Integrating Ontologies, vol. 156, 2005, 72–76.

[7] Euzenat J., Valtchev Y., *Similarity-Based Ontology Alignment in OWL-Lite.* Proc. European Conference Artificial Intelligence (ECAI'04), 2004, 333–337.

[8] Euzenat J., *State of the Art on Ontology Alignment.* http://www.inrialpes.fr/exmo/cooperation/kweb/heterogenity/deli/, 2008.

[9] Euzenat J., Shvaiko P., *Ontology Matching.* Springer, 2007.

[10] Gruber T., *A Translation Approach to Portable Ontologies*. Knowledge Acquisition, vol. 5, No. 2, 1993, 199–220.

[11] Gusfield D.,Irving R.W., *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, 1989, 54.
[12] Issac A., Van der Meij L., Schlobach S., Wang S., *An Empirical Study of Instance-Based Ontology Matching*. Proc. International Semantic Web Conference and Asian Semantic Web Conference, ISWC and ASWC, 2007, 253–266.
[13] Li Y., Li J., Zhang D., Tang J., *Toward Strategy Selection for Ontology Alignment*. Proc. International European Semantic Web Conference (ESWC '07), 2007.
[14] Madhavan J., Bernstein P., Rahm E., *Generic Schema Matching with Cupid*. Proc. International Conference Very Large Database (VLDB '01), 2001, 48–51.
[15] Maedche A., Moltik B., Silva N., Volz R., *MAFRA-An Ontology Alignment Framework for Distributed Ontologies*. Proc. International Conference Knowledge Engineering and Knowlegde Management (EKAW'02), 2002, 235–250.
[16] Melnik S., Molina H.G., Rahm E., *Similarity Flooding: A Versatile Graph Matching Algorithm.* Proc. 18th International Conference Data Engineering (ICDE'02), 2002, 117–128.
[17] Melnik S., Rahm E., Bernstein P.A., *Rondo: A Programming Platform for Model Management.* Proc. ACM SIGMOD'03, 2003, 193–204.
[18] Noy N.F., *Semantic Integration: A Survey of Ontology Based Approaches.* SIGMOD Record, special section on semantic integration, vol. 33, No. 4, 2004, 65–70.
[19] OWL, Ontology Web Language, http://www.w3c.org/TR/owl-features/, 2008.
[20] Rahm E., Bernstein P.A., *A Survey of Aproaches to Automatic Schema Matching.* VLDB J., vol. 10, 2001, 334–350.
[21] Shvaiko P., Euzenat J., *A Survey of Schema-Based Matching Approaches.* J. Data Semantics, vol .4, 2005, 146–171.
[22] OAEI, Ontology Alignment Initiative, *http://oaei.ontologymatching.org.*