Ontology Alignment Technique for Improving Semantic Integration

Mohammad Mustafa Taye
Faculty of Information Technology
Philadelphia University, Jordan
mtaye@philadelphia.edu.jo

Nasser Alalwan
Software Technology Research Laboratory (STRL),
De Montfort University
Leicester, UK line
nasser@dmu.ac.uk

***Abstract*— A new technique for ontology alignment has been built by integrating important features of matching to achieve high quality results when searching and exchanging information between ontologies. The system is semi-automatic and enables syntactical and semantic interoperability among ontologies. Moreover, it is a multi-strategy algorithm which can deal with and solve more than one critical problem. Therefore, it is likely to be more conveniently applicable in different domains. Also, we improve a semantic matcher based on combining lexical matcher with several rules and facts. Moreover, our technique illustrates the solving of the key issues related to heterogeneous ontologies, which uses combination-matching strategies to execute the ontology-matching task. Therefore, it can be used to discover the matching between ontologies. The main aim of the work is to introduce a method for finding semantic correspondences among heterogeneous ontologies, with the intention of supporting interoperability over given domains. Our goal is to achieve the highest number of accurate matches.**

Keywords-Ontology; Semantic Interoperability; Heterogeneous; Ontology Alignment.

I. INTRODUCTION

Ontology [1] has been developed to offer a commonly agreed understanding of a domain that is required for knowledge representation, knowledge exchange and reuse across domains. Therefore, ontology organizes information into taxonomies of terms (i.e., concepts, attributes) and shows the relationships between them. In fact, it is considered to be helpful in reducing conceptual confusion for users who need to share applications of different kinds, so it is widely used to capture and organize knowledge in a given domain.

Although ontologies are considered to provide a solution to data heterogeneity, from another point of view, the available ontologies could themselves introduce heterogeneity problems.

In order to deal with these problems, ontologies must be available for sharing or reusing; therefore, semantic heterogeneity and structural differences need to be resolved among ontologies. This can be done, in some cases, by aligning or matching heterogeneous ontologies. Thus, establishing the relationships between terms in the different ontologies is needed throughout ontology alignment [4, 5, 7, 14].

Semantic interoperability can be established in ontology reconciliation. The original problem is called the "ontology alignment". The alignment of ontologies is concerned with the identification of the semantic relationships (subsumption, equivalence, etc.) that hold between the constituent entities (which can be classes, properties, etc.) of two ontologies.

In this paper, an ontology alignment technique has been developed in order to facilitate communication and build a bridge between ontologies. An efficient mechanism has been developed in order to align entities from ontologies in different description languages (e.g., OWL, RDF) or in the same language. This approach tries to use all the features of ontologies (concept, attributes, relations, structure, etc.) in order to obtain efficiency and high quality results. For this purpose, several matching techniques have been used such as string, structure, heuristic and linguistic matching techniques with thesaurus support, as well as human intervention in certain cases, to obtain high quality results. This paper is organized as follows: section II over view about our system; Section III describes our system in details. Section IV and Section V shows the related work and the evaluation process. Finally Section VI concludes the paper.

II. SYSTEM ARCHITECTURE

A framework relies on a well-established measure for comparing the entities of two ontologies which are combined in a homogeneous way. The Figure 1 shows the system components.

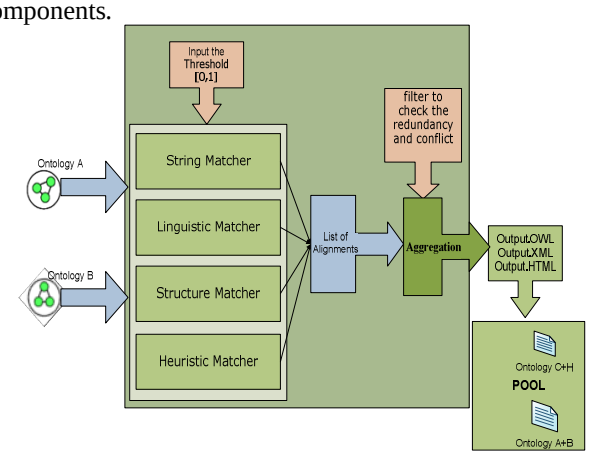

FIGURE 1: THE SYSTEM FRAMEWORK.

 ISBN: 978-1-61208-000-0

III. DETAILED SYSTEM

The system starts by loading two ontologies and extracts useful features such as class names, property names and subsumption relationships from them. In case ontology does not exist, we use our algorithm in [22] to transform relational database to OWL ontology.

A. String Matching

In general, the name of a class (i.e., label) is presented as a chain of characters without space characters. It is used to provide a human-readable description of class. Therefore, a name of class may be a word, or a combination of words. In fact, the name of each class should be unique in the ontology.

Terminological methods compare strings. Hence, these methods can be applied to the name, the label or the comments concerning entities to discover those which are similar. In general, it can be used for comparing class names and/or URIs.

A string matcher [2, 3, 7] usually takes as input the names of two concepts, then calculates the distance between them by distance functions that map a pair of strings to a real number. Consequently, the output will be a numeric value c $\in$[0, 1] to represent the confidence of the similarity. The main reason for using such measures is the fact that similar entities have similar names and descriptions across different ontologies.

String similarities are based on the assumption that the names of concepts and properties representing semantic similarity will have similar syntactic features. Thus, a string matcher usually first normalizes the input string of names and/or descriptions via stemming and tokenization. In the simplest form, the equality of tokens will be obtained and combined to give a score of the equality for the whole string.

In general, two properties are used to identify terms: the label and the name. The label is a string usually expressed in natural language to describe the purpose of the field to humans, while the name can be any string that is constrained by some name rules. These techniques are usually applied to names, labels, comments concerning entities and the URI. The scaled range is [0, 1] for comparing strings. To achieve high quality results and based on many experiments, the system disregards similarities that are smaller than a threshold of 0.65, and matches similarities greater than 0.65 to the full range [0, 1].

B. Linguistic Matching

The terminology used for naming and labeling concepts and properties is an important aspect of ontologies and provides information on the similarity between the ontology elements. However, linguistic features are also important for deriving an initial set of alignments to be refined by exploiting other kinds of matching. In fact, names of classes or properties are considered to provide one of the most important clues as to whether two terms are equal or not; therefore, we try to find relations between terms from different ontologies based on the details of their names. Such linguistic matching relies on algorithms and the use of external lexicon-based resources such as dictionaries, which are typically used to find close relationships such as synonymy between two terms and to compute the semantic distance between them in order to decide if a relationship holds.

This process is based on linguistic analysis [10,16]. There are two general techniques for label matching, the first of which employs linguistic analysis steps, such as abbreviations, avoiding recurrence and particle-ending. The other is matching the labels to determine the relationship between them.

In general, the linguistic similarity between terms is computed by considering labels and descriptions. Knowledge-based matchers take as input two concept (or synset) identifiers defined in WordNet [12] and produce semantic relations by exploiting their structural properties. They are often based on either similarity or relatedness measures. If the value of the measure exceeds the given threshold, a certain semantic relation is produced. Otherwise, "Idk" (I don't know) is returned. This technique is implemented by using thesauri and WordNet, following an approach which is essentially the structural congruence between labels based on the hidden meanings of the words that they represent. WordNet, which takes two concept (synset) identifiers as input and returns the semantic relation holding between them, is considered not only to provide synonyms, hypernyms and hyponyms, but also to exploit additional structure to detect relationships between concepts (dinner$\rightarrow$meal). For example, it considers synonyms as equivalent and hyponyms as subsumed, finding Match and Alignment to be similar classes (car$\rightarrow$automobile).

In using a WordNet-based matcher we have to translate the (lexical) relations, which are provided by WordNet to logical relations [12], based on the following rules:

- A $\subseteq$ B, if A is a hyponym or meronym of B. For example, author is a hyponym of creator, therefore deducing that author $\subseteq$ creator.
- A $\supseteq$ B, if A is a hypernym or holonym of B. For example, Asia is a holonym of Jordan, therefore deducing that Asia $\supseteq$ Jordan.
- A = B, if A and B are connected by a synonymous relation or they belong to one synset. For example, quantity and amount are synonyms, therefore deducing that quantity = amount.
- A $\perp$ B, if A and B are connected by antonymy relations or are siblings in a part of hierarchy. For example, Jordan and Syria are siblings in the WordNet part of hierarchy, therefore deducing that Jordan $\perp$ Saudi Arabia.

C. Structure Matching

The aim of structural matching is to link an element of source taxonomy with an element of target taxonomy. The mappings generated are mainly specialization matches, based on calculations of the similarity of the source element with all those under the target taxonomy. Indeed, this kind of matching depends on what are considered the most important features of ontology nodes (e.g., class: super-classes and

 ISBN: 978-1-61208-000-0

Sub-class, property: super properties and sub properties). Therefore, similarity is based on the nodes of graphs.

The similarities between two concepts can be obtained from the language and from real attributes; and not only the similarities between the descriptions of their components, but also from similarities between the structures of the graphs representing them. The internal structure of similarities can be obtained by calculating the number of similar attributes divided by the attributes of a class.

Taxonomy is generally represented by an acyclic graph whose nodes are concepts and arcs corresponding to linked subclasses. Class inheritance analysis (is-a) considers the hierarchical connection between classes in order to identify "is-a" relationships.

Taxonomy (C, HC) includes a set of concepts C and a hierarchy subsumption between concepts HC. A concept is defined by its label and subclass relationships, which connect to other concepts. The label is a name (string) which describes entities in natural language and can be an expression composed of several words. Subclass relations establish links between concepts.

The intuition behind the algorithm is that if two concepts lie in similar positions with respect to is-a or part-of hierarchies relative to concepts already aligned in the two ontologies, then they are likely to be similar as well. For each pair of concepts (C1, C2) in the original list of alignment relationships, the structural matcher augments the original similarity value for pairs of concepts (C'1, C'2), such that C'1 and C'2 are equivalent to, are in an is-a relationship with or participate in a part-of relationship with C1 and C2 respectively. The augmentation depends on both the relationship and the distance between the concepts in the is-a and part-of hierarchies. It is important to mention here two important rules that help to ensure correct matching. First, if the super-concepts of two classes are the same, then these two concepts may be similar to each other. The second rule is that if the sub-concepts of two classes are the same, we can say that the concepts are also similar.

Structural analysis identifies identical classes by looking at their attributes and related (linked) classes. The main idea is that two classes of two ontologies are similar or identical if they have the same attributes and the same neighbor classes. Hence, matching concepts are based on structural similarity with regard to class hierarchy.

D. Heuristic-based Strategies

This phase of our system uses several features of ontologies (i.e., their structure, definitions of concepts and instances of classes) in order to find matches. Based on the idea that labeling is important and helps to align most of the entities, the structure also provides valuable help in identifying alignments. We have developed this step based on these two elements.

It considers the entities and structure of an ontology, i.e., class (C), property (P), relation (R), instance (I) and super-class (S). The distances between the input structures are then expressed in a set of equations. As described on Figures (2, 3).

$$Tot_{Sim(c,c')} = w * Sim_C(c,c') + w * Sim_P(c,c') + w * Sim_R(c,c') + w * Sim_I(c,c') + w * Sim_S(c,c')$$

where

- Sim_C (c, c') is the similarity between labels of classes,
- Sim_P (c, c') is the similarity between properties of classes,
- Sim_R (c, c') is the similarity between relations of classes,
- Sim_I (c, c') is the similarity between instances of classes,
- Sim_S (c, c') is the similarity between super-classes of classes,
- w is the weight, which is set at $1/5$ in order to obtain results in the range $[0,1]$,
- $Tot_{Sim(c,c')}$ is the average of all of these similarities, in the range $[0,1]$

FIGURE 2: HEURISTIC MATCHER EQUATION

The following is the function of heuristic match:

```
Function heuristicMatch (Ontology o1, Ontology o2) {
  for (All concept pairs (c, c') where c ϵ o1 and c' ϵ o2) {
Sim_C = ComputeNameSimilarity (c, c')
Sim_P = (W* findCommonAttributes (c, c')) + (W* matchDataTypes (c, c')) +
(W * matchDataInstance (c, c'))
Sim_R = (W * findRelationship (c, c')) + (W* matchNameRelationship(c, c'))
Sim_I = (W* findCommonInstance (c, c') + (W* matchInctance (c, c'))
Sim_S=W* ComputeNameSuperClass (c, c')
    //compute overall similarity
Tot_Sim(c,c') = w * Sim_C(c,c') + w * Sim_P(c,c') + w * Sim_R(c,c') + w
                * Sim_I(c,c') + w * Sim_S(c,c')
    }//end for
        }//function heuristicMatch
```

FIGURE 3: HEURISTIC MATCHER FUNCTION

The output is one-to-one or one-to-many correspondences. This strategy is based on string similarity (i.e., Monge-Elkan metric [3]) structure and instances.

Monge-Elkan distance is recursive matching scheme for comparing two long strings s and t. By assuming that the strings s and t are broken into substrings (tokens), i.e., s = s1 . . . sK and t = t1 . . . tL. The intuition behind this measure is the assumption that si in s corresponds to a tj with which it has highest similarity. The similarity between s and t equals the mean of these maximum scores.

$$Monge - Elkan(A,B) = \frac{1}{|A|} * \sum_{i=1}^{|A|} \max_{J=1}^{|B|} match\ (A_i, B_j)$$

In heuristic matching, iteration is one of the most important steps in ontology alignment, which takes into account the structure of the input ontologies. It enables the whole process to be repeated several times, by propagating and updating the assessed similarities.

The sigmoid strategy combines multiple results using a sigmoid function, which is a smoothed threshold function,

showing the importance of retaining high individual prediction values and removing low ones.

This technique starts after the application of the normalization process on the input elements, then compares class and property names in terms of editing distance and substring distance between entity names. The algorithms next create a distance matrix in order to determine the alignment group from the distance.

This algorithm is used in order to cover most possible features of ontologies (i.e., terminological, structural and extensional); on the other hand, we explicate recursive relationships and try to find the best matches through iteration. In general, this method faces problems when the viewpoints of two ontologies are highly different; thus, in order to achieve a high quality result, several of the above criteria should be combined, so that the rules which can be applied here are:

Any two concepts are probably the same if their labels are the same.

Any two concepts are equal if their properties are equal, even if their labels are different.

Two concepts that have the same instances are the same.

E. Aggregation

The results discussed here have been calculated using string, linguistic, structure and heuristic matchers. Indeed, with several matching strategies/ algorithms, there are several similarity values for a candidate matching (e1; e2). Therefore, combining them is an effective way to achieve high accuracy for a larger variety of ontologies, so this step extracts the combined matching result from the individual strategy results stored in the similarity cube. For each combination of ontology entities, the strategy-specific similarity values are aggregated into a combined similarity value, e.g., by taking the average or maximum value.

The easiest way to proceed consists of selecting correspondences above a particular threshold. Such threshold-based filtering allows us to retain only the most similar entity pairs. For the combination of the match results, the average value has been computed and a selection has been made using a threshold, for example: $\text{Semantic Distance}(C, C') \leq \text{Threshold}$

IV. RELATED WORK

The following literature offers several approaches to the alignment of ontologies, based on measures of similarity.

A. The Naive Ontology Mapping

This approach [17] is simple, constituting a straightforward baseline for later comparisons. It comprises six steps. Feature Engineering demands that the ontologies be represented in RDF. Search Step Selection compares all entities of the first ontology with all entities of the second. Similarity Computation computes the similarity between entities in different ontologies, using a wide range of similarity functions. In Similarity Aggregation, NOM highlights individually significant similarities by weighting individual similarity results and aggregating them. This, however, neglects individual similarities that are of less significance. Interpretation uses the individual or aggregated similarity values to derive mappings between entities. Finally, Iteration repeats the previous step several times. This gives the capacity to access the already computed pairs and use more sophisticated structural similarity measures, whereas neglecting this step provides only a comparison based on labels and string similarity. A new version has more features and heuristic combinations, such as Quick Ontology Mapping (QOM) [18].

Advantage and Disadvantage: this approach applies string matching, structure matching and an instance matching, but it doesn't use auxiliary information. The means of defining the ontology is based on concepts, properties, and instances. The input-ontologies for this approach are in RDF format only. Moreover, it does not use a normalisation process. The way of selecting matching elements is threshold based.

B. PROMPT

Prompt [21] is a tool for merging ontologies, developed by Stanford University Knowledge Systems Laboratory. The knowledge model underlying PROMPT is frame-based and is compatible with Open Knowledge Base Connectivity. In general, this tool provides a semi-automatic approach to merging two ontologies; it is based initially on alignment relations, which should be held before providing output as a coherent ontology. More specifically, PROMPT performs some tasks automatically: it takes two ontologies as input and creates an initial list of matches based on class names. This list will be a coherent ontology. The following cycle then occurs: (1) the user triggers an operation by either selecting one of PROMPT's suggestions from the list or by using an ontology-editing environment to specify the desired operation directly; and (2) PROMPT performs the operation, automatically executes additional changes based on the type of the operation, generates a list of suggestions for the user, based on the structure of the ontology around the arguments of the last operation, and determines conflicts that the last operation introduced in the ontology, finding possible solutions for them. PROMPT then guides the user in performing other tasks for which his intervention is required. Its top level contains Classes (collections of objects arranged into hierarchies), Slots (binary relations), Facets (ternary relations) and Instances (individual members of classes).

Advantage and Disadvantage of PROMPT:

It applies string matching and semantic matching but it does not provide instance or structure matching. The input-ontologies for this approach are in different format like RDF(s), OWL-Lite, and OWL-DL. The output is merged ontology. The way of defining ontology is based on concepts, properties and instances. It does not deal with normalisation process. The way of selecting matching elements is based on highest value. This approach provides interactive suggestions to the users. It solves mismatches at terminological and scope of concept level, and it helps alignment by providing possible edit points and it supports repeatability. But it is not automatic which means every step requires user interaction.

 ISBN: 978-1-61208-000-0

C. Chimaera

Chimaera [19, 20] is a semi-automatic or interactive tool for merging ontologies. The engineer is in charge of making decisions that will affect the merging process. This tools starts by analysing the ontologies to be merged. It automatically finds linguistic match merges, and if it cannot find any matching terms, it gives the user control over any further action. In fact, it is similar to PROMPT, as both are embedded in ontology editing environments and offer the user interactive suggestions.

Advantage and Disadvantage of Chimaera:

It uses string matching, semantic matching and structure matching but it does not provide instance matching. The input-ontologies for this approach are OKBC ontologies and the output is a merged ontology. This approach analyses ontologies to be merged; if linguistic matches are found then the merge is processed automatically; otherwise, it uses subclass and super class relationship. In fact, this approach solves mismatches at the terminological level in a very light way, and provides interactive suggestions to the users. It solves mismatches at terminological and scope of concept level, and it helps alignment by providing possible edit points and it is not repeatability. But it is not automatic which means everything requires user interaction. (It is very similar to PROMPT).

V. EVALUATION

It can be argued that the most significant and crucial issue when suggesting a new approach is its evaluation. Therefore, this section presents many test cases which are used to evaluate the performance of our system in different scenarios, followed by the experimental methodology, test data sets and results.

The Ontology Alignment Evaluation Initiative **(**OAEI) is a coordinated international initiative to establish agreement for evaluating and improving the available ontology alignment techniques. The OAEI ontology matching campaign is a contest organised annually since 2004, comprising several kinds of tests, processes and measures for assessing the results.

The benchmark data tests were divided into five groups, as shown in Table 1.

TABLE 1: DESCRIPTION OF BENCHMARK DATA SET

Test Sets	Ontology Description	Num of Ontologies
101-104	Similar in both label description and hierarchy structure	4
201-210	Similar hierarchy structure	10
221-247	Similar in label description	18
248-266	Different in both label description and hierarchy structure	15
301-304	Real-world ontologies provided by different institutions	4

In order to assess the different approaches or evaluate the degree of compliance of the results of matching algorithms, standard information retrieval metrics are used, presenting four values which are widely used to estimate the quality of the alignment process and its results: precision, recall, overall and F-measure.

Currently, there are many ontology matching systems that have been developed based on different strategies for various purposes. In order to evaluate their performance and their qualities, we will focus on OAEI evaluation which employs a systematic approach to evaluate ontology matching algorithms and identify their strengths and weaknesses. After that we chose the following tests to show the evaluation:

A. Tests 221 to 247

In the third test set, the names, labels and comments had no special features that might confuse the alignment, but the structures of these ontologies were manipulated and some instances or/and properties were added. Therefore, in these ontologies our algorithm performed very well on string-, linguistic- and heuristic-based strategies in computing the similarity between features. This was due to the fact that the terms in these test cases had high string similarity; moreover, the heuristic matcher performed very well in these tests. On the other hand, where specific terms did not have similar names or comments, our algorithm was able to apply structural or semantic features of ontologies in order to derive the remaining alignments.

The most important issues affecting each of these are briefly stated here. Ontologies 221 to 247 featured no specialization (221), a flattened hierarchy test (222), an expanded hierarchy test (223), no instances (224), no restrictions (225), no datatypes (226), unit differences (227), no properties (228), class vs. instances (229) and flattened classes (230); all of these were matched with a very high recall and precision rate. As a conclusion, on this group of tests our algorithm performed well, which can be attributed to the fact that we carried out both syntactic and semantic similarity assessments.

TABLE 2: RESULT OF TESTS 221-247

Test ID	Precision	Recall
221	1.00	1.00
222	1.00	1.00
223	1.00	1.00
224	1.00	1.00
225	1.00	1.00
228	1.00	1.00
230	1.00	1.00
231	1.00	1.00
232	1.00	1.00
233	1.00	1.00
236	1.00	1.00
237	1.00	1.00
238	1.00	1.00
239	1.00	0.99
240	1.00	0.99
241	1.00	1.00
246	1.00	1.00
247	1.00	1.00
Average	1.00	0.999

Although the structures of the candidate ontologies were changed, our algorithm found most correct alignments by using strings (label similarity, comment similarity), the

 ISBN: 978-1-61208-000-0

linguistic perspective and heuristic matching. Therefore, both precision and recall were excellent.

While tests 221-247 shared the same names and comments, their structures differed. Instances were similar, but some ontologies did not contain them. The information given was sufficient to reach very good results. For most of these tests the structures were modified, which means that structural similarity was low, but the label similarity was very high. Because of this low structural similarity, the structure matcher did not work well for some tests; for example, tests 221, 232, 233 and 241 had high label and structural similarity factors, so both linguistic and structure-based strategies were employed, although the structure matcher made little contribution. Table 2 shows the results. Table 2 shows the results which appeared from tests 221-247.our results are very high and are nearly equal to 1. Our algorithms are heavily using linguistic and string matching algorithms.

B. Comparison with other existing approaches

In order to evaluate our system, a comparison of the system results was made against the published results from the 2007 Ontology Alignment Evaluation Initiative.

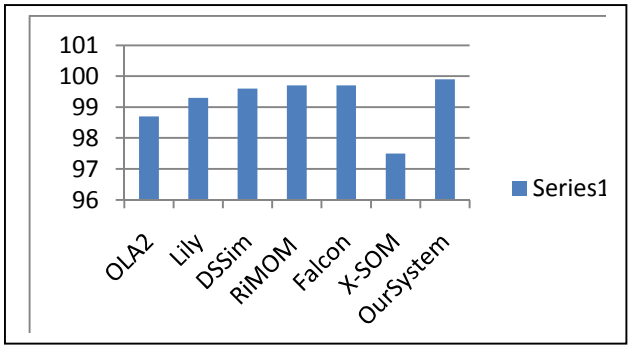

FIGURE 4: RESULTS OF TESTS 221-247

For most of tests 221-247, the structures of ontologies were manipulated, so that structural similarity was low; however, names, labels and comments in these ontologies had no special features, so linguistic similarity was very high. The information given was sufficient to yield very good results. In this set of tests, where the ontologies had high similarity with the reference ontology on linguistic information, our system performed very well and was the best, with precision, recall and F-measure scores of 1.00, 0.999 and 0.999 respectively. Other systems, including Falcon, DSSim and RiMOM also performed very well, with results on the F-measure of 0.997, 0.996 and 0.997 respectively.

VI. CONCLUSION

We develop new ontology alignment technique by using different matching strategies. This new ontology alignment approach utilizes both linguistic and structural information from ontologies in order to solve ontology alignment problems. The system is applying different matching algorithms, which includes: String matching, Linguistic-based strategies Structural matching, and Heuristic-based Strategies.

REFERENCES

[1] Berners-Lee T., Hendler J., and Lassila O., "The Semantic Web", Scientific Am, May 2001, pp. 34–43.

[2] Bunke H., Csirik J., "Parametric String Edit Distance and Its Application to Pattern Recognition", IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, pp. 202-206, 1995.

[3] Cohen W.W., Ravikumar P., and Fienberg S.E., "A Comparison of String Distance Metrics for Name-Matching Tasks", In Proceedings of II Web, 2003, pp.73-78.

[4] Ehrig M., "Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond)", New York, Springer, 2006.

[5] Ehrig M., Euzenat J., "State of the Art on Ontology Alignment", Knowledge Web Deliverable D2.2.3, University of Karlsruhe, 2004.

[6] Euzenat J. , Shvaiko P., "Ontology Matching", Springer-Verlag, Heidelberg (DE), 2007.

[7] Euzenat J., Valtchev P., "Similarity-Based Ontology Alignment in OWL-Lite", In Proceedings of ECAI, 2004, pp.333-337.

[8] Euzenat J., Loup D., Touzani M., and Valtchev P., "Ontology Alignment with OLA", In 3rd EON Workshop, 3rd Int. Semantic Web Conference, 2004, pp. 333–337.

[9] Fensel D., "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce", Springer, 2001.

[10] Giunchiglia F., Yatskevich M., "Element Level Semantic Matching", In Proceedings of the Meaning Coordination and Negotiation workshop at ISWC, (2004).

[11] Lambrix P. , Tan H., "A Tool for Evaluating Ontology Alignment Strategies", Presented at Journal Data Semantics, 2007, pp.182-202.

[12] Leacock C., Chodorow M., "Combining Local Context and Wordnet Similarity for Word Sense Identification", In WordNet: An Electronic Lexical Database, Christiane Fellbaum, MIT Press, 1998, pp. 265–283.

[13] Mao M., Peng Y., "The PRIOR+: Results for OAEI Campaign 2007", In Proceedings of OM, 2007.

[14] Nagy M., Vargas-Vera M., and Motta E., "DSSim - Managing Uncertainty on the Semantic Web", In Proceedings of OM, 2007.

[15] Taye M., " Ontology Alignment Mechanisms for Improving Web-based Searching", Ph.D. Thesis, De Montfort University, United Kingdom, England, 2009.

[16] Schorlemmer M., and Kalfoglou Y., "Progressive Ontology Alignment for Meaning Coordination: An Information-theoretic Foundation", In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, 2005, pp. 737–744.

[17] Ehrig M., Staab S., "Efficiency of Ontology Mapping Approaches", International Workshop on Semantic Intelligent Middleware for the Web and the Grid at ECAI 04, Valencia, Spain, August 2004.

[18] Ehrig M., and Staab S., "QOM - Quick Ontology Mapping", In Proceedings of International Semantic Web Conference, 2004, pp.683-697. .

[19] McGuinness D.L., Fikes R., Rice J., and Wilder S., "The Chimaera Ontology Environment", In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), 2000.

[20] McGuinness D.L., Fikes R., Rice J., and Wilder S., "An Environment for Merging and Testing Large Ontologies", In Proceedings of KR2000, 2000, pp. 483-493.

[21] Noy N.F., and Musen M.A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment", In Proceedings of AAAI/IAAI, 2000, pp.450-455.

[22] Alalwan N. ,Zedan H.,and Siewe F., "Generating OWL Ontology for Database Intgeration",In proceedings of Third International Conference on Advance in Semantic Processing 2009,pp.22-31.

 ISBN: 978-1-61208-000-0