IEEE- Fourth International Conference on Advanced Computing, ICoAC 2012
MIT, Anna University, Chennai. December 13-15, 2012

Semantic Enrichment in Ontology Mapping using Concept Similarity Computing

Shunmughavel. V[#1],
[#1] Professor and Head,
Department of Computer Science and Engineering,
K.L.N. College of Information Technology,
Pottapalayam-630611.
Email: shuns@rediffmail.com

Jaganathan. P [#2]
[#2] Professor and Head,
Department of Computer Applications,
P.S.N.A. College of Engineering & Technology,
Dindigul-624222
Email:jaganathodc@gmail.com

***Abstract*-**In semantic web ontology heterogeneity is a big bottleneck of ontology application, and ontology mapping is the base for integration of heterogeneous ontology. The ontology mapping model contains several aspects, and concept similarity computing is the most important part. This paper presents a concept similarity computing algorithm combining lexical matching to achieve semantic enrichment and high accuracy results. It has been proved that the evaluation of concept similarity between ontologies is more accurate by considering both semantic similarity and semantic relativity.

Keywords: Ontology mapping, Semantic web, Lexical matching, Semantic heterogeneity

I. INTRODUCTION

The vision of the Semantic Web [13] provides many new perspectives and technologies to overcome the limitation of the World Wide Web. Ontologies are a key component to solve the problem of semantic heterogeneity, and thus enable semantic interoperability between different web applications and services. The exploration of ontologies could be an efficient and powerful way for organizations or communities of practice to share knowledge [11]. Semantic interoperability is required in order to represent domain knowledge in a generic way and provide a better understanding of that domain.

Ontology is a formal, explicit specification of a shared conceptualization [14] that provides precise notations and explicit meanings of data (i.e. semantics) in a domain [15]. The task of finding semantic correspondences between similar elements of different ontologies is known as ontology mapping or ontology matching. In this area, different authors use different words to refer to similar concepts, and, vice versa, sometimes different concepts are referred to by the same name" [3]. For example, ontology mapping is defined as the process of "mapping concepts in the various ontologies to each other". Ontology matching is also defined as "finding semantic mappings between ontologies" [2].

Presently, semantic web technologies are bringing about a revolutionary change on the Internet where the World Wide Web will not just be based on text based pages but those with meaning (semantics) attached to them. Facilitating the implementation of the semantic web is the concept of ontologies, i.e. formal specification of concepts [6]. It is common for different experts to want to use their own ontology to support particular semantic web applications. However, it is also common that there exist different ontologies for similar domains. This leads to the problem of ontology heterogeneity [6] which in turn leads to problems in knowledge sharing. The efforts to improve system interoperability will thus rely on the reunion of different ontologies used in different systems. The reunion is often done by manual or semi-automated integration of ontologies. The technical issue here is to help resolving ontology mismatches that clearly appear in semantic integration.

One of the fundamental elements of the ontology integration process is to establish mappings between ontologies. Sets of mapping assertion are the main output of mapping processes. [19] These mapping assertions can be used in a translational component directly, which translates statements that are formulated by different ontologies. Instead, a follow-up integration process can use the mappings to detect merging points. So, interoperability among applications in heterogeneous systems depends vitally on the ability to map between their corresponding ontologies.

Ontology heterogeneity can be addressed by carrying out a mapping process on the multiple ontologies concerned. The methods mentioned so far have many constraints and limitations such as:

- While expressing the same thing, people may use synonyms and even different languages. Therefore, it is necessary to use lexicons to match ontologies. Existing tools do not use lexicons.

- Existing tools are not good enough in dealing with large-scale ontologies. They also need a large amount of memory and a long time for finding the alignments of large ontologies.

While there are current efforts in ontology mapping, these can be further optimized by using hybrid approaches to achieve a higher accuracy of mapping results. A hybrid strategy framework for ontology mapping to address the ontology heterogeneity problem is proposed in this paper. This framework employs a concept based ontology mapping coupled with lexical matching to produce results with high accuracy and semantic enrichment.

The rest of the paper is organized as follows: Section 2 deals with detailed literature survey, section 3 focuses on the ontology mapping framework, section 4 describes system architecture, and section 5 concentrates about implementation and section 6 on the conclusions with further scope for improvement.

II. RELATED WORK

Increasing usage of the Internet has enabled us to focus on ontology mapping. Several studies have already been proposed in this area. Some important research works are presented here.

Wei Hu et al., [18] have designed a system, Falcon-AO for aligning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is an automatic ontology matching system which helps to enable interoperability between (Semantic) Web applications using different but related ontologies. It consists of five components: the Repository to temporarily store the data during the matching process; the Model Pool to manipulate ontologies and to construct different models for different matchers; the Alignment Set to generate and evaluate exported alignments; the Matcher Library to manage a set of elementary matchers; the Central Controller to configure matching strategies and execute matching operations.

Yves R. Jean-Mary et al., [20] have developed ASMOV, an automatic ontology matching tool which has been designed in order to facilitate the integration of heterogeneous systems, using their data source ontologies. The current ASMOV implementation produces mappings between concepts, properties, and individuals, including mappings from the object to data type properties and vice versa. The ASMOV algorithm iteratively calculates the similarity between entities for a pair of ontologies by analyzing four features: lexical description (id, label, and comment), external structure (parents and children), and internal structure (property restrictions for concepts; types, domains, and ranges for properties; data values for individuals), and individual similarity. The measures obtained by comparing these four features are combined into a single value.

Juanzi Li., [7] proposed a framework RiMOM for ontology matching. The RiMOM consists of six major steps. The input ontologies are loaded into the memory and the ontology graph is constructed in Ontology Preprocessing and Feature Factors Estimation. During Single strategy execution, selected strategies are obtained to find the alignment independently. Each strategy outputs an alignment result. In Alignment combination phase RiMOM combines the alignment results obtained by the selected strategies. If the two ontologies have high structure similarity factor, RiMOM employs a similarity propagation process to refine the found alignment and to find new alignment according to the structural information. Alignment refinement refines the alignment results from the previous steps.

Giuseppe Pirró, Domenico Talia have developed UFOme [5] which assists users during the various phases of a mapping task execution by providing a user friendly ontology mapping environment. This paper presents an ontology mapping software framework that has been designed and implemented to help users (both expert and non-expert) in designing and/or exploiting comprehensive mapping systems. It is based on a library of mapping modules implementing functions such as discovering mappings or evaluating mapping strategies.

Language-based [3] methods rely on using natural language processing techniques to help extract the meaningful terms from a text. Terms are phrases that identify concepts; they are often used for labeling concepts in ontologies. Comparing these terms and their relations should help to assess the similarity of the ontology entities they name and comment upon. These are based on linguistic knowledge. Indeed, there are two general techniques, one relying on algorithms and the other using external lexicon-based resources such as dictionaries. As a consequence, ontology matching can take great advantage of recognizing and identifying them in strings. In other words, it uses the internal linguistic properties of the instances, such as their syntactic properties. In general, extrinsic linguistic methods [3] are used for finding extra similarities between terms; external linguistic resources such as dictionaries and lexicons increase the chances of finding matching terms. Lexical information (e.g. names, definitions and distance between strings) can help reclassification of matching results. Word Net [9, 20] is an electronic thesaurus database for the English which distinguishes clearly between word senses by grouping words into synsets (concepts or senses of groups of terms) or sets of synonyms.

Concept similarity is defined first, Sim: W1* W2* O1 *O2 -> [1, 0] which means the similarity value is between 0 and 1. *Sim* (A, B) stands for the similarity between A and B. W1 and W2 are term sets on which the two ontologies base, O1 and O2 are the two ontologies that are to be mapped. *Sim* (e, f) =1 stands for the two concepts e and f which are the same; *Sim* (e, f) =0 stands for the two concepts e and f which are totally different [16, 17].

The PROMPT suite [12] contains of a set of tools that had an important impact in the area of merging, aligning and versioning of ontologies. The suite includes an ontology merging tool (iPROMPT, formerly known as PROMPT), an ontology tool for finding additional points of similarity between ontologies like iPROMPT (Anchor PROMPT), an ontology versioning tool (PROMPT Diff) and a tool for factoring out semantically complete subontologies

(PROMPTFactor). PROMPT takes two ontologies as input and guides the user in the creation of a merged ontology as output.

Amjad Farooq in his work has stated that [1] the semantic web is mainly based on ontologies whereas ontologies themselves suffer from heterogeneity when simultaneously used in some integrating processes such as merging, aligning and mapping. Those ontologies may contain some lexically similar concepts belonging to different context and likewise some contextually similar concepts may have different roles or granularities in their respective ontologies. When such ontologies are required to reuse simultaneously in some operations for sharing and acquiring of information, the heterogeneity usually arises and then it is required to find the similarity between their concepts to handle the situation.

III. ONTOLOGY MAPPING FRAMEWORK

The ontology mapping framework proposed in this paper comprises of three modules namely pre-processing module, mapping strategies module and post-processing module. This framework is represented in Figure1.

A. Pre Processing

This module entails obtaining useful information from the ontologies that are to be matched. This begins by loading two ontologies and extracting useful ontological features such as class names and properties. Then normalization is carried out on these elements, by removing stop words.

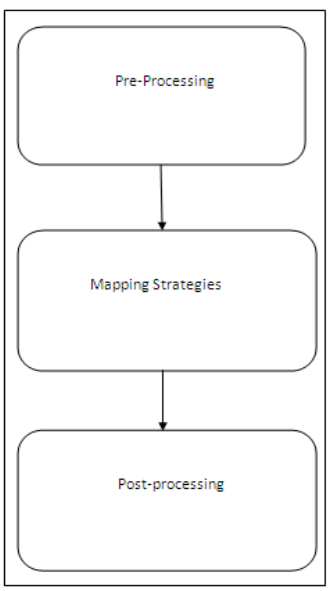

Figure 1 Ontology Mapping Framework

B. Mapping Strategy

In general, the similarity between entities needs to be calculated in order to find the correspondence between ontology entities. For that reason, the concept based ontology mapping strategy along with lexical matching is used for achieving similarity between entities.

a. Concept based

In our approach, in order to produce mapping-pairs and implement ontology mapping and knowledge sharing, a concept similarity algorithm combining semantic similarity with semantic relativity is adopted to estimate the similarity between concepts [16, 17].

B.Lexical Matching

It is certain that the use of lexicons may definitely bring benefits for ontology mapping. Hence, in our framework, lexicon matching technique is added to all the above mentioned approaches for better results. While expressing the same thing, people may use synonyms and even different languages. Therefore, it is necessary to use lexicons to match ontologies. [10] A lexicon, or dictionary, is a set of words together with a natural language definition of these words. Of course, for a particular word, e.g. 'article', there can be several such definitions. The Word Net lexical database is an example of a resource for this purpose. In our framework, Word Net acts as the intermediary between the ontologies to be mapped as it will store words with similar meanings. In essence, Word Net helps to form links between two concepts. These links will be used to facilitate ontology mapping.

C. Post–Processing

a. Similarity Aggregator

Similarity aggregator aggregates strategy wise similarity values before and after lexical matching.

b. Similarity Evaluator.

Currently, there are many ontology matching systems that have been developed based on different strategies for various purposes. In order to assess the different approaches or evaluate the degree of compliance of the results of matching algorithms, standard information retrieval metrics precision, recall and f-measure are used. These values are widely used to estimate the quality of the alignment process and its results.

Precision measures the number of correct mappings found against the total number of the set of alignments obtained by a certain method. Recall measures the number of correct mappings found against the total number of existing mappings. F-measure combines measures of precision and recall as single efficiency measure.

IV. SYSTEM ARCHITECTURE

The ontology Mapping Architecture is represented in Figure.2.

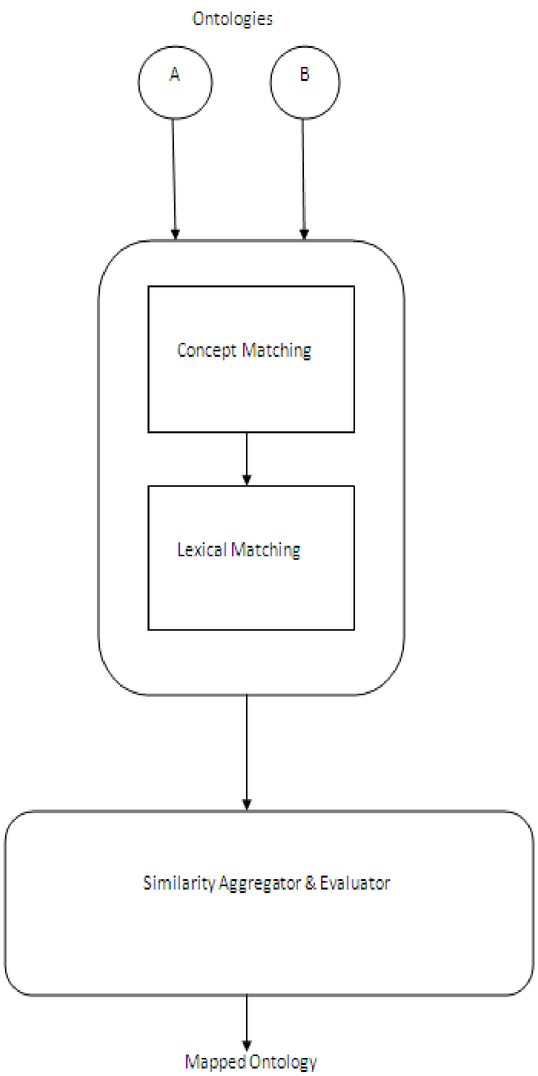

Figure.2 Ontology Mapping Architecture

A. Concept Matching

Concept similarity is defined as follows: $Sim\ (A, B): W_1 \times W_2 \times O_1 \times O_2 \rightarrow |$ which means the similarity value is between 0 and 1. *Sim* (A, B) stands for the similarity between A and B. W1 and W2 are term sets on which the two ontologies base, O1 and O2 are the two ontologies that to be mapped. *Sim* (e, f) =1 stands for the two concepts e and f are the same; *Sim* (e, f) =0 stands for the two concepts e and f are totally different.

At present, each pair of concepts in the ontology is considered when calculating the concept similarity of two ontologies, so huge calculations are needed. Now two methods are commonly used, making use of the concept instances and illumination rules, but neither is perfect.

Having summarized the existent calculation methods of concept similarity, this paper proposed an improved method combining semantic similarity with semantic correlativity. It calculates concept similarity via combining semantic similarity with semantic correlativity to establish ontology mapping rules.

Calculating semantic similarity is mainly comparing the concept similarity by the concept's semanteme itself, while calculating semantic correlativity is mainly comparing the concept similarity by the concept's attributes and relations. Finally, concept similarity is gained by measuring the semantic similarity based on concept's semantic distance and semantic correlativity based on concept's attributes and relations together. Entity's mapping pairs between ontologies are gained by iteration of the method mentioned above.

In the Concept Similarity computing module [16] both Concept Semantic Similarity and Concept Semantic Relativity are employed. Concept semantic similarity evaluates the similarity between concept's meaning and concept semantic relativity evaluates the similarity between concept's attributes or the relations.

The algorithm proposed in this paper is calculating the semantic similarity grounded on concept semantic distance firstly, and then calculating semantic correlativity founded on concept's relations and attributes, finally integrating the two.

Concept semantic similarity matching between ontologies has been considered by concept semantic distance. Concept semantic relativity matching between ontologies has been considered by concept's attributes and the relations between concept's attributes. Concept similarity in ontology base can be obtained through linear combining of the semantic similarity and semantic relativity

B. Lexical Matching

While expressing the same thing, people may use synonyms and even different languages. Therefore, it is necessary to use lexicons to match ontologies. Hence, lexical matching is combined with all the above mentioned strategies to ensure semantic interoperability.

Linguistic similarities are computed by examining the similarities between the local descriptions of the classes. As a basic group of matching techniques, they are usually the initial step to suggest a set of raw mappings with which other matchers can work. This uses additional linguistic resources such as lexicons and thesauri in order to identify synonyms.

The semantic measure determines the meaning of the terms, which includes information such as synonyms and hyponyms. In this stage, an attempt is made to find a common element in the synsets of two names. This phase also generates a synset for each element that includes the synonym set retrieved from Word Net. The two terms may be similar, even if they are spelt differently. The retrieval of the synonym set (e.g. car → automobile) is an example of the use of synonyms. In general, the names of two nodes having a related sense are expected to be somehow related.

The lexical semantics similarity measure explores the semantic meanings of the word constituents. This can be achieved by using external resources, such as user-defined lexica and/or dictionaries to help identify synonyms in matching. Semantic interoperability is about ensuring that the precise meaning of the information exchanged can be understood by other systems, especially if they are not tailored for this specific information exchange.

C. Similarity Aggregator & Evaluator

The similarity values of the concept matcher module are aggregated. Then the standard information retrieval metric precision, recall and f-measure are calculated. After lexical matching again the similarity values are aggregated and performance metrics are calculated.

V. IMPLEMENTATION

A. Preprocess

The main aim of this process is obtaining useful information from the ontologies that are to be matched, beginning by loading two ontologies and extracting useful ontological features such as class names, property, etc. The process starts with two ontologies (RDF/OWL) as input; these need to be aligned with one another. All entities of the first ontology are compared with all entities of the second ontology. Before the comparison of entities (concepts, relations and instances) can be processed, it is necessary to choose an entity from each of the ontologies. In order to compare them, the two entities need to be extracted from extensional and intentional ontology definitions. In general, the obvious technique is to compare all entities of the first ontology with all entities of the second ontology. This process is based on most of the ontology features. In fact, a top-down strategy is adopted. In order to make an efficient mapping algorithm, several measures are used in the processing. The system starts by loading two ontologies and extracts useful features such as class names, property names and subsumption relationships from them.

B. Concept Matching

Concept semantic similarity is calculated using the concept semantic distance in the two ontologies to be mapped. The semantic distance is defined as the sum of the each concept's depths in the ontology conceptualization hierarchy. The specific process can be represented as [16]: *Dis* (*C*1, *C*2) = the super class list's length of *C*1 + the super-class list's length of *the C2*.

Given *C*1 and *C*2 are two concepts of two ontologies, *SemSim* (*C*1, *C*2) stands for the similarity between the two terms, then

$$SemSim(C1,C2) = \frac{\sum_{i=1}^{n} \theta i(C1,C2)}{Dis(C1,C2)} + \frac{\sum_{j=1}^{m} \theta i(C1,C2)}{Len(C1,C2)} \tag{1}$$

In the formula above, n stands for the largest value of concept model's depth in Ontology, m stands for the largest number of sub-classes in the ontology concept model, *Dis* (*C*1, *C*2) stands for the concept semantic distance between C1 and C2. The value of θi (*C*1, *C*2) is defined as follows:

When the former i parent-classes of *C*1 and *C*2 are identical, θi (*C*1, *C*2) =1, and otherwise θi (*C*1, *C*2) = 0. The value of θi (*C*1, *C*2) is defined as follows: When the last j parent-classes of *C*1 and *C*2 are identical, θj (*C*1, *C*2) =1, and otherwise θj (*C*1, *C*2) = 0.

The calculation of concept similarity based on concept semantic distance just compares concepts in ontology on structure. The valuation of semantic relativity mainly evaluates the semantic similarity of concept in two ontologies. The similarity between the concepts of two ontologies is largely related to the concept's attributes. So the similarity between the two concepts' attributes *Pi* and *Pj* must be evaluated before the two concepts' semantic relativity. Given *Pi* and *Pj* are attributes belonging to two different ontologies separately, *PSim* (*Pi*, *Pj*) stands for the similarity of attributes between the two terms, then

$$PSim(p_i, p_j) = aPD(p_i, p_j) + bSM(p_i, p_j) + cPR(p_i, p_j) \tag{2}$$

PD (*Pi*, *Pj*) stands for grammatical similarity between Pi and Pj in the domain concept, PR (*Pi*, *Pj*) stands for grammatical similarity between *Pi* and *Pj* in the range concept, SM (*Pi*, *Pj*) stands for grammatical similarity of attributes between *Pi* and *Pj*. a, b, c are the weight of the three similarities mentioned above separately, and satisfy a + b + c = 1. If the ranges of the two attributes are the same, then the value is 1. Here, only a single attribute is considered, and then the matching value between the concept's attributes is obtained.

Grammatical similarity SM (*Pi*, *Pj*) evaluates attribute similarity with the name of the attribute. This paper utilizes "edit distance" to calculate the grammatical similarity between attributes and instances. Here the similarity of strings is weighed by the editing operation times of transforming one string to another. The operation of transforming includes character's modification, insertion, deletion, and exchange. Then,

$$SM(p_i, p_j) = \max(0, \frac{\min(|L(p_i)|, |L(p_j)|) - d(p_i, p_j)}{\min(|L(p_i)|, |L(p_j)|)}) \tag{3}$$

In the formula above, *d* (*Pi*, *Pj*) stands in editing distance, for instance, *SM* ("seat", "heat") =3/4. The calculation method of *PD* (*Pi*, *Pj*) is similar as *SM* (*Pi*, *Pj*), and the only difference that is to be calculated is the editing distance of concepts. After calculating the attribute sets' similarity between two concepts, its semantic correlativity is defined as follows:

$$Sem\ Rel(C_1, C_2) = \frac{set(p_i) \cap set(p_j)}{\sum_{i=1}^{n} \sum_{j=1}^{m} PSim(p_i, p_j)} \tag{4}$$

Concept semantic similarity matching between ontologies has been considered by concept semantic distance, and concept semantic relativity matching between ontologies has been considered by concept's attributes and the relations between concept's attributes. Concept similarity in ontology base can be got through linear combining of the semantic similarity and semantic relativity as

$$Sim\,(C_{1,}C_2) = hSemSim\ \ (C_{1,}C_2) + (1-h)Sem\ \mathrm{Re}\ l(C_{1,}C_2) \qquad (5)$$

In the formula above, h stands for weight, which satisfies the sum of the semantic similarity's weight and the semantic relativity is equal to 1.

Then standard information retrieval metrics precision and recall are calculated which are widely used to estimate the quality of the alignment process and its results.

The unmatched pairs from the Concept similarity module are fed into the lexical matching module. After lexical matching, once again the information retrieval metrics are calculated.

C. Lexical matching

Lexical matching takes two strings as input. In this case, its class names. Stemming is performed for the class names to obtain the root value r1 and r2. The root value of the class names is compared using String Compare method. If a match is found the function returns a Boolean value as true or false. Semantic Analysis takes two string values.

It uses Word Net as a database to identify the synonyms of the class names. In this module, class name cn1 is searched in the WorldNet. This gives all synonyms of cn1, which is stored in str [], an array of strings. Those synonyms and cn2 are passed for Lexical Analysis for comparison. If a match is found, the Boolean true value is returned or false value is returned. Similarity Check takes class cn1, cn2 and an array of string p3 which is used to store the properties of the merged class. In this module the properties of the cn1 are stored in w1 and c2 in w2. The maximum of the number of properties in w1 and in w2 is assigned to an integer variable 'a'. Every property in w1 is compared with every other property in w2 using Lexical and Semantic Analysis.

If a match is found the counter variable ctr is incremented and the property is stored in w3.This module returns the value p which is the counter value and it is divided by a. This value defines the similarity between the two classes.

D. Pseudo code

The ontology mapping procedure is illustrated by the pseudo code shown in Figure.3 and has been implemented as a Java application. The pseudo code comprises of three modules namely the preprocessing module, the mapping strategy module and the aggregator-evaluator module.

```
Load ontology O1, O2

---preprocessing
        Do Preprocess ()
---mapping strategy

        Do Strategy1 ()
        Do Evaluator ()
        Do Strategy2 ()
        Do Evaluator ()

---post processing
        Do Aggregator & Evaluator
O/P mapped ontology
Preprocess ()
Strategy 1 ()
        Calculate Semantic similarity (SES)
        Calculate Semantic relativity (SR)
        Concept Similarity CS = SES + SR
Strategy II ()
        Lexical matching using WorldNet
Evaluator ()
        Calculate precision
        Calculate recall
Aggregator ()
        Aggregate performance metrics

---Strategy1 : Concept Matching
---Strategy II: Lexical matching
        Finding similarity using WorldNet
```

Figure 3 Pseudo code for Hybrid strategy ontology mapping

In the mapping strategy module concept matching strategy is adopted. For semantic enrichment the lexical matching strategy is coupled with concept matching. The performance metrics are calculated. Unmatched pairs from all the above three strategies are fed into the lexical matching module. After lexical matching once again the performance metrics precision, recall and the f-measure are calculated.

VI. RESULTS AND DISCUSSIONS

A. Dataset and Evaluation Metrics

The sample ontologies related to medical domain are created using Protégé tool. Two ontologies such as MeSH and Word Net are used in the mapping process. Medical Subject Headings (MeSH) is the National Library of Medicine's vocabulary thesaurus, which contains a collection of words representing descriptors in a hierarchical structure and the type of relationship between nodes in each sub-tree is IS-A relationship, and more. The proposed system is implemented using Java (jdk 1.6). We use precision and recall and F-measure to evaluate mapping results. Precision (P) is the percentage of correctly discovered alignments in all discovered alignments. Recall (R) is the percentage of correctly discovered alignments in all correct alignments. F-measure (F) combines measures of precision and recall as single efficiency measure.

$$\text{Precision (P)} = \frac{\text{no of correct found mapping}}{\text{no of retrieved mapping}} \qquad (6)$$

$$\text{Recall (R)} = \frac{\text{no of correct found mapping}}{\text{no of existing mapping}} \quad (7)$$

$$\text{F-Measure (F)} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

B .Concept Matching Strategy analysis

In Table 1, before lexical matching, the performance metrics for the four pairs of ontologies are tabulated for the threshold values 0.6 to 1.0. After lexical matching , the performance metrics for the four pairs of ontologies are tabulated in Table 2. In Figure.6, the precision and recall values after lexical matching show an increase of 6 percent and 9 percent respectively.

	Concept Matching				
	Pair-1				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.63	0.74	0.93	0.83	0.93
Recall	0.57	0.68	0.81	0.71	0.77
F-Measure	0.60	0.71	0.87	0.77	0.84
	Pair-2				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.59	0.71	0.94	0.84	0.93
Recall	0.51	0.61	0.83	0.73	0.81
F-Measure	0.55	0.66	0.88	0.78	0.87
	Pair-3				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.53	0.63	0.91	0.78	0.88
Recall	0.48	0.53	0.81	0.65	0.76
F-Measure	0.50	0.58	0.86	0.71	0.82
	Pair-4				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.57	0.69	0.89	0.91	0.91
Recall	0.45	0.61	0.73	0.76	0.88
F-Measure	0.50	0.65	0.80	0.83	0.89
	Average				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.58	0.69	0.92	0.84	0.91
Recall	0.50	0.61	0.80	0.71	0.81
F-Measure	0.54	0.65	0.86	0.77	0.86

Table 1: Performance metric values before Lexical matching

	Concept Matching with Lexical Analysis				
	Pair-1				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.69	0.81	0.98	0.91	0.98
Recall	0.64	0.76	0.91	0.79	0.83
F-Measure	0.66	0.78	0.94	0.85	0.90
	Pair-2				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.65	0.77	1.00	0.91	0.99
Recall	0.58	0.68	0.92	0.82	0.87
F-Measure	0.61	0.72	0.96	0.87	0.92
	Pair-3				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.58	0.69	0.98	0.85	0.95
Recall	0.54	0.61	0.91	0.72	0.85
F-Measure	0.56	0.65	0.94	0.78	0.90
	Pair-4				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.62	0.75	0.96	0.97	0.97
Recall	0.51	0.68	0.82	0.84	0.94
F-Measure	0.56	0.71	0.88	0.90	0.95
	Average				
Threshold	t=0.6	t=0.7	t=0.8	t=0.9	t=1.0
Precision	0.64	0.76	0.98	0.91	0.97
Recall	0.57	0.68	0.89	0.79	0.87
F-Measure	0.60	0.72	0.93	0.85	0.92

Table 2: Performance metric values after Lexical matching

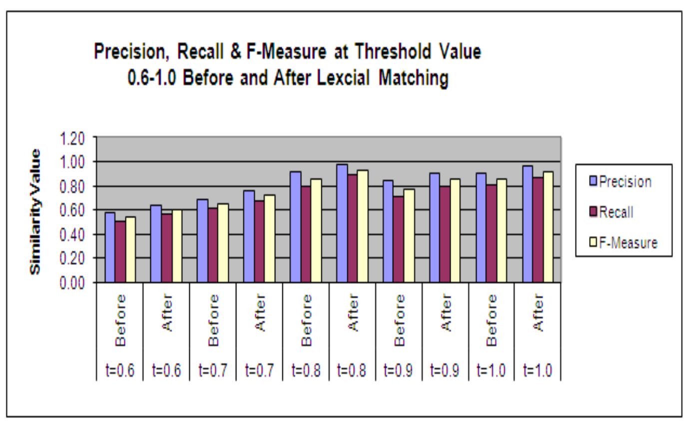

Figure.4 Performance metrics analysis

In all the three strategies - Strategy 1 (Table1 and Table2), Strategy 2 (Table3 and Table4) and Strategy 3 (Table 5 and Table 6) - after lexicon matching the performance evaluation metrics Precision and Recall have significantly increased.

The average mapping result for all four pairs (Figure.6, Figure.7 and Figure.8) shows that the threshold value 0.8 generates the best precision and recall. It contributes towards improvement in the F-measure and matching accuracy in mapping. Therefore, threshold value 0.8 will be adopted to automate the ontology mapping process. However, we desire to perform more experiments to justify the current threshold value.

VII. CONCLUSION

In this paper, a hybrid strategy framework is proposed to automatically and dynamically execute ontology mapping tasks. This framework integrates conceptual and semantic characteristics of ontologies to achieve semantic enrichment and high quality results. Experimental results show that the proposed approach can significantly outperform combined methods.

Despite major advances in the ontology matching research topic have been presented so far, our position is that there are still important open issues to be tackled towards a more efficient and effective use and integration of ontology matching tools with current applications.

From the experiments, some lessons have been learned to make improvements in the later versions. The following three improvements should be taken into account. Scalability i.e. capability handling large ontologies may be addressed in future. In future, Mapping Multilingual content using translator component may be incorporated. Future directions should be towards a fine tuning of all parameters such as the overall performance of an ontology mapping algorithm and the development of linguistic, structure and concept based multi strategy mapping algorithm.

REFERENCES

[1] Amjad Farooq, M. Junaid Arshad and Abad Shah.: A Layered approach for Similarity Measurement between Ontologies. Journal of American Science. ;6(12):69-(ISSN: 1545-1003),2010.

[2] Doan, A., Madhavan, J., Domingos, P. and Halevy, A.: Learning to map between ontologies on the semantic web. Proceeding of the 11th international conference on World Wide Web,2002

[3] Euzenat, J. and Shvaiko, P.: Ontology Matching. Springer-Verlag, Heidelberg (DE), 2007.

[4] Feiyu, L.: State of the Art: Automatic Ontology Matching. School of Engineering, Jönköping Universiy, Gjuterigatan 5, 551 11 Jönköping, Sweden. ISSN 1404-0018 Research Report November 2007.

[5] Giuseppe Pirró, Domenico Talia.: UFOme: An ontology mapping system with strategy prediction capabilities. ELSEVIER,Data & Knowledge Engineering 69 (2010) 444–471

[6] Jingshan, H., Jiangbo, D., Jose, M.V., and Michael, N.H.: Ontology Matching Using an Artificial Neural Network to Learn Weights. Proceedings IJCAI Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa-07), Hyderabad, India, January 2007.

[7] Juanzi Li, Jie Tang, Member, IEEE, Yi Li, and Qiong Luo.: RiMOM: A Dynamic Multi Strategy Ontology Alignment Framework. IEEE Transactions on knowledge and data Engineering VOL 21 No.8, Aug 2009

[8] Melnik, S.,Molina,H.G. and Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm. Proc. 18th Int'l Conf. Data Eng. (ICDE '02), pp. 117-128, 2002.

[9] Miller, G.A. and Hristea, F. 2006.: WordNet Nouns:Classes and Instances. Presented at Computational Linguistics, (2006), pp.1-3

[10] Mohammad Mustafa Taye.: State-of-the-Art: Ontology matching Techniques and Ontology Mapping Systems.The International Journal of ACM Jordan (ISSN 2078-7952), Vol. 1, No. 3, September 2010

[11] Monika Lanzenberger, Jennifer J. Sampson, Markus Rester, Yannick Naudet and Thibaud Latour.: Visual ontology alignment for knowledge sharing and reuse. JOURNAL OF KNOWLEDGE MANAGEMENT,VOL. 12 NO. 6 2008, pp. 102-120, *Q* Emerald Group Publishing Limited, ISSN 1367-3270

[12] Natalya F. Noy*, Mark A. Musen.: The PROMPT suite: interactive tools for ontology merging and mapping. Elsevier Int. J. Human-Computer Studies 59 (2003) 983–1024

[13] Tim Berners-Lee, James Hendler, and Ora Lassila.: The semantic web: Scientific american. *Scientific American*, May 2001.

[14] Tom Gruber.: A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 1993.

[15] Uschold, M., and Gruninger, M.: Ontologies and semantics for seamless connectivity. *SIGMOD Record*, 33(4):58–64, 2004.

[16] Wenjie Li, Yan Zhao, Nan Shen.: Concept Similarity Calculation in Ontology Mapping. 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery 978-0-7695-3735-1/09 $25.00 © 2009 IEEE DOI 10.1109/FSKD.2009.351214

[17] Wei Hu, Yuzhong Qu.: Falcon-AO: A practical Ontology Matching System. Journal of Web Semantics, Elsevier, p: 237-239, 2008.

[18] Wen-jie Li, Qiu-xiang Xia.: Study of Semantic-based Ontology Mapping Technology.978-1-4244-5874-5/10/$26.00 ©2010 IEEE

[19] Xiaomeng Su, Jon Atle Gulla.: An information retrieval approach to ontology mapping. Data & Knowledge Engineering 58 (2006) 47–69 ,Elsevier

[20] Yves R. Jean-Mary, E. Patrick Shironoshita and Mansur R. Kabuka.: Ontology matching with semantic verification. Journal of Web Semantics, Article in Press, Elsevier, 2009.