Hybrid approach for efficient Multiple ontology mapping

A.Kavidha(1), Dr.A.Saradha(2)

1Department of CSE, Institute of Road & Transport Technology, Erode, Tamil Nadu, India
2Department of CSE, Institute of Road & Transport Technology, Erode, Tamil Nadu, India

Abstract
Enormous information available on the Web creates a demand for automatic ways of processing and analyzing data. Semantic Web technologies are used to provide automatic ways of web data processing by integrating several heterogeneous knowledge bases available on the web. Information integration of a domain includes the process of mapping and merging of ontologies. In the semantic web applications, ontologies are used as knowledge representation models. Automatic integration of ontologies is a difficult task as they are developed for different contexts by different developers. Ontologies from same domain may differ in their syntax, semantics and also in their structure. Information processing across ontologies in a semantic web is possible only when they are semantically mapped to each other.

In this paper, we have proposed a hybrid automatic ontology mapping algorithm for mapping OWL based ontologies. The proposed mapping algorithm uses syntax semantics and structural mappings of entities present in the contributing ontologies. The mapping algorithm is tested for mapping university ontologies downloaded from web. The result shows that our approach outperforms existing mapping techniques in terms of the evaluation measures precision and F-measure.

Keywords: Interoperable knowledgebase, Integration of knowledge bases, Ontology integration, Ontology mapping, Semantic web application.

1. Introduction
Today, most of the organizations around the globe are sharing, exchanging and sometimes integrating the information in a distributed fashion. In order to guarantee semantic accuracy they must be modeled using semantic web. The semantic web proposed by Tim Berners-Lee provides a solution to the problem of knowledge sharing between enterprise applications over web. In distributed environment, semantic web applications inevitably use the knowledge bases described by multiple ontologies. Different people define the ontologies differently based on time, region, and applications.

Semantic web application is developed by integrating data from various heterogeneous sources. As integration of distributed information sources has been made more automated, the ambiguity in concept interpretation, also known as semantic heterogeneity, has become one of the main obstacles to this process said Avigdor Gal and PavelShvaiko [1].

In order to achieve semantic interoperability in a heterogeneous information system, it is essential to discover and map related concepts from the information sources. The semantic web technologies enable proper integration of knowledge in ontology based applications. Ontologies are formal specification of a shared conceptualization. Ontologies are normally developed for some specific purpose and hence they differ in their terminological use and in their abstraction in ontology modeling. Various ontologies developed by multiple authors may contain overlapping, conflicting and incoherent domain knowledge. In [2], ontologies merged in DL-Lite considering only T-Box for query answering. Semantic conflicts occur whenever two contexts do not use the same interpretation of the information. Shared understanding is necessary to overcome differences in terminology. Syntactically similar entities normally possess the same meaning but entities having different syntax refer different meaning in usual practice.

Sometimes, terms assume meaning based on the context, as illustrated below:

i. Entities are syntactically similar, but semantically different

E.g : The term ‘course’ may refer ‘degree’ or ‘subject’ in an educational domain
E.g: The term ‘plane’ may refer plane in a geometry or aircraft in different domain

ii. Entities are syntactically different but semantically same
E.g: ‘location’ and ‘area’ refers to ‘region’
E.g: ‘gender’ and ‘sex’ refers to ‘masculinity’

Interpretation of information of an entity in ontologies not only depends on thesaurus semantics but also depends on several factors like structural position of entities in ontology, attributes of concepts, relationship of entity with other entity, constraints assigned by axioms, domain and range values of properties. Hence, semantics of the entities in ontology follows from all these factors. Human can interpret the meaning based on the context. But, automatic mapping of ontologies requires agents to guide the system to avoid incorrect mapping. However, most previous research efforts on ontology integration have not resolved many issues of integration as they have considered ontologies with only concept name mapping in taxonomical relations. So, existing approaches often result in incorrect matches. Recently, ontologies are developed using OWL DL technology. They provide much expressive power in defining relations, restrictions and rules. Mapping or alignment of ontologies results in new relations like the equivalence, subsumption and incompatibility among the contributing entities. This paper proposes a mapping algorithm for automatic mapping of domain ontologies using the hybrid approach. Hybrid approach combines syntax, semantics and structural similarities to overcome the issues in mapping OWL-DL ontologies. This paper is organized as follows: section 2 discusses related work; section 3 discusses proposed algorithm; section 4 is devoted to implementation; section 5 discusses results and discussions for comparison of result with the existing ontology mapping approaches and section 5 concludes the paper.

2. Related Work
David Sánchez et al.,[3] have used edge-counting measures by considering the shortest path between concept pairs. But, when they are applied to detailed ontologies that incorporate multiple taxonomical inheritances they result in several taxonomical paths which are not taken into account. Other features also influencing the concept semantics, such as the number and distribution of common and non-common taxonomical ancestors and are not considered. As a result, by taking only the minimum path between concepts, many of the taxonomical knowledge explicitly modeled in the ontology is omitted. Feature-based methods try to overcome the limitations of path-based measures by considering different kinds of ontological features. In their approach, feature-based method for semantic assessment exploits the taxonomical and non-taxonomical knowledge available in ontology. In Petrakis et al., [4] a feature-based method called X-similarity considers synsets and a concept’s glosses extracted from WordNet for matching. Both the methods compare only concepts in a single ontology. Li et al.,[5] combine structural characteristics such as path length, depth and local density and assign weights to balance the contribution of each component to the final similarity value. Compare to edge-counting measures, their accuracy is higher but the method depends on tuning of weights for the ontologies input terms.

In [6], considered matching of formal attributes that include a reference to the process and the objects that are produced by the process. The feature-based similarities investigated in this paper are: the Tversky index, the Dice’s coefficient, the Jaccard’s coefficient, the Overlap coefficient, the all confidence similarity, and the Cosine similarity. They found Tversky index and Jaccard’s coefficient performs better. MAFRA (Mapping FRAmework) is another ontology mapping methodology that prescribes “all phases of the ontology mapping process, including analysis, specification, representation, execution and evolution” said ErMaedche et.al in [7]. It uses the declarative representation approach in ontology mapping by creating a Semantic Bridging Ontology (SBO) that contains all concept mappings and associated transformation rule information. In this model, given two ontologies (source and target), it requires domain experts to examine and analyze the class definitions, properties, relations and attributes to determine the corresponding mapping and transformation method. Then, all accumulated information will be encoded into concepts in SBO. Therefore, SBO serves as an upper ontology to govern the mapping and transformation.
between two ontologies. Each concept in SBO consists of five dimensions: they are Entity, Cardinality, Structural, Constraint and Transformation. During the process of ontology mapping, software agent will inspect the values from two given ontologies under these dimensions and execute the transformation process when constraints are satisfied.

The authors Andrew Choi and Marek Hatala [8] followed the approach WordNet syntet usage for finding semantic similarity. As oppose to the classic or traditional keywords-based representation, semantic-based indexing with WordNet senses can include more lexicon information than simple syntactic approach. In [9], authors developed suite of tools for managing multiple ontologies. They support semi-automatic merging, it is called iPROMPT. ANCHORPROMPT uses a graph structure of ontologies to find correlation between concepts and to provide additional information for iPROMPT but it depends on the user suggestions for mapping. They experimented with two ontologies CMU and UMD. iPROMPT manually checks the inconsistencies and potential problems caused in the merged ontologies that require human attention. To evaluate the quality of iPROMPT suggestions, they used precision and recall measure. Precision is the fraction of the tools suggestions that the users decided to follow and recall is the fraction of the operations that the users performed that were suggested by the tool. The average precision was 96.9% and the average recall was 88.6%.

The algorithm Anchor-Prompt discussed about ontology mapping uses anchors (or related concepts) and are used to establish a link between common terms in the source ontologies. The user can input the set of anchors for partial alignment. The central observation behind Anchor-Prompt is that if two pairs of terms from the source ontologies are similar and there are paths connecting the terms, then the elements in those paths are often similar as well.

In [10] the structural similarity between two concepts is computed from the average similarity of their respective super and sub-concepts. The super and sub concepts of two concepts being compared are fetched into separate sets and then the resultant similarity is computed from the similarities of concepts in those sets. If both the super and sub concepts similarity are undefined, then the concepts being compared are declared as dissimilar otherwise they are declared similar concepts.

In [11], Ontology was constructed by selecting keywords for tourism domain site and keyword mapping of user query with keywords of ontology was carried out. Ontology has no relationships. The weight was assigned for the related keywords using hits. The sub_graph with more weights are selected and answered for the query. The paper did not possess any details regarding improvement of efficiency.

In [12], authors have used only similarity matching for schema integration. In [13], given two ontologies, for each concept in one ontology GLUE finds the most similar concept in the other ontology. This project utilizes machine learning to find semantic mappings across heterogeneous ontologies in the semantic web. Details on the ontology languages as well as how this approach caters for syntactic interoperability are not provided.

3. Hybrid Approach Owl Ontology Mapping

Ontologies are developed based on the conceptualization of the developers which may vary from person to person and it is mostly based on the application. Before the development of OWL, the ontologies were having taxonomical hierarchy. Hence most of the earlier papers on ontology mapping were considered taxonomical relations. OWL based Ontology mapping proposed in this paper take into account taxonomical as well as non-taxonomical entities for their syntactic, semantic and structural mapping to make most effective correlation among the contributing OWL-ontologies. Our technique considers the properties of ontologies and the relationship of entities with other entities for the process of alignment. Syntactic mapping alone will not result in correct mapping as ‘human’ and ‘person’ are syntactically different but semantically equal. Even if they are semantically aligned, the attributes used to define the entities creates the difference as depicted in the figures Fig.1 and Fig.2.

In the figures Fig-1 and Fig-2, if the concepts ‘Human’ and ‘Person’ are aligned, it would not perform any useful action as they are from different contexts. These concepts from different ontology may not be aligned as they differ in their attributes. In earlier papers, the
semantic similarity of any entities is mostly mapped using WordNet but from the example it is certain that it also depends on the attributes used to define the concept as the context of the entity is decided using attributes. Structural measure in our proposed paper considers taxonomical and non-taxonomical relationship of entities with other entities in ontology.

In figure, the concept ‘Human’ may be related to the concepts ‘degree’, ‘project_aid’ and hence it is associated with the domain of research. But, the concept ‘Person’ having relationship with concept ‘Bank’ and defines the context of bank application. Hence, the proposed algorithm improves the ontology mapping by its hybrid approach while aligning with other ontology.

3.1 Ontology Mapping Techniques

OWL ontology may include descriptions of classes, properties and their instances. Given such ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single or multiple distributed ontologies. Semantic web applications are developed by integrating the knowledge of various OWL ontologies for query answering and information retrieval. In [14] authors have said that traditionally semantic mapping was undertaken by human domain experts and only recently have approaches been developed to automate this process. The resolution of semantic interoperability is achieved through semantic processing. Semantic applications in query answering use heterogeneous ontologies and hence those ontologies are required to be mapped to get best response. Integration is applied in the development of semantic applications for the interaction of user with semantic agent as well as for the interoperability among semantic agents.

Ontology mapping is possible only if the contributing ontologies possess certain common vocabularies from the same domain, hence the proposed mapping process assumes the contributing ontologies are from same domain knowledge. Multiple ontologies may be mapped by the algorithm but for simplicity, mapping is done with two ontologies.

Proposed algorithm for ontology mapping identifies syntactic and semantic correspondence of the entities in different source ontologies and it also considers different
granularity of contributing ontologies. The relationship of one entity with other also contributes to the semantics of the entity. When ontologies are used for interoperability of heterogeneous systems, ontology mappings can resolve the mismatches between the systems, thereby realizing semantic integration.

Ontology alignment methods are classified using various techniques. They can be classified as in [15]. Various methods are Terminological (T) comparing the labels of the entities; string based (TS) does the terminological matching through string structure dissimilarity (e.g., edit distance); terminological with lexicons (TL) does the terminological matching modulo the relationships found in a lexicon (i.e., considering synonyms as equivalent and hyponyms as subsumed); internal structure comparison (I) comparing the internal structure of entities (e.g., the value range or cardinality of their attributes); external structure comparison (S) comparing the relations of the entities with other entities; taxonomical structure (ST) comparing the position of the entities within a taxonomy; external structure comparison with cycles (SC) an external structure comparison robust to cycles; extensional comparison (E) comparing the known extension of entities, i.e. the set of other entities that are attached to them (in general instances of classes); semantic comparison (M) comparing the interpretations (or more exactly the models) of the entities. In [16], the algorithm proposed ontology mapping by considering taxonomic and nontaxonomic relations of ontologies. But, it failed to prove the algorithm correctness.

3.2 Proposed Mapping Algorithm

Proposed ontology mapping algorithm is constituted with the following stages.

1. Concepts in both the ontologies are checked for lexical equivalence followed by semantic equivalence forming the resultant set of concepts. All the pair of similar concepts from the resulting set is subjected to structural comparison. The super class of the concepts in the pair is tested for equivalence. If they are equal, then those concepts are subjected to attribute test.

2. Attributes of the equivalent concepts from stage-1 are compared for lexical and semantic similarity. Concepts in source ontologies are equivalent if they satisfy stage 1 and 2.

3. Object properties of contributing ontologies are tested for their correspondence in the respective ontologies. Object properties are relations of entity with other entity in ontology. The domain and range of the properties in both ontologies are checked to see if they are from equivalent category. Further, object properties of the concepts are tested to find whether they have equal in the constraints like inverse, disjoint properties. Then those concepts are called as strongly equivalent.

4. The entities which are not strongly equivalent but are equivalent in some aspects are considered for mapping manually. The above steps are executed in the sequence to find the correspondence among the contributing ontologies automatically. Mapping logic finds all strongly equivalent concepts of contributing ontologies and maps them with “OWL: equivalent” automatically. It automatically finds any inferred relation among entities. These inferred relations of entities are given to the user for final mapping. To avoid incorrect mapping, mapping logic is used for mapping only strongly equivalent concepts automatically. The entities which are not strongly equivalent but equivalent are considered for manual mapping.

4 Implementation of Hybrid algorithm

4.1 Syntax similarity: The syntactic similarity is found out using Dice similarity method[17]. In syntactic comparison, the entities of contributing ontologies are compared for their syntactic equivalence. The set of substrings of length 2 are found out for entity A from ontology1 and entity B from ontology 2 and the resulting sets are called as C and D respectively. Finally, number of similar strings from sets C and D are calculated. The dice score is calculated as in equation-1.

\[
\text{Dice Score} = \frac{2 \times (\text{Number of similar strings} \ / \ \text{Total number of sub strings})}{---} \quad (1)
\]
The procedure is repeated for each and every entity of both ontologies. Threshold for syntax similarity is fixed after analyzing the F-measure and any two concept names having dice score equal or above the threshold are similar in syntax wise. In the algorithm, syntax similarity of any entity (concept, attribute, object property) is found out using this technique.

4.2 Semantic similarity
Semantic Similarity bridges the heterogeneous gap of ontologies by identifying related concepts. The semantics similarities of concepts are found out in stages. At stage-1, they are verified for their similarity using WordNet. Here, the synonyms, hypernyms and hyponyms of the concept terms are checked for their equivalence. Single word term could be mapped to possible different word senses through WordNet. Each word sense is represented in synset which may have multiple synonymous terms.

WordNet is an online lexical reference system. It is based on a massive semantic network containing over 90,000 word senses. Word Net API edu.mit.jwi_2.2.3 is used for checking semantic similarity of concepts of the contributing ontologies. The entities of reference ontologies are compared for semantics using synonymy, hypernyms and hyponyms in Word Net. The Anchor-Prompt algorithm[18] produces good results only if ontology developers link the concepts in a similar fashion even though different names are assigned to them. They claim 75% of results are correct. But, it does not produce the same result for OWL ontologies.

The attributes or data type properties of the concepts actually define context of entities. Hence, the equivalence of concepts are determined only after testing the similarity of their attributes at stage-2 of algorithm.

4.3 Structural Similarity
The concepts which are equal in semantics are checked whether their super class relations are correct. If Entity A is semantically equivalent to Entity B, then they are structurally equal ifSuper Class (A) = Super Class (B). In [19], it is recommended that structural equality is not sufficient for measuring the alignment.

Technique proposed in [20] computes the similarity between concept Ci of Ontology A and concept Cj of ontology B, based on the criteria: (i) At least one of the super-concepts of Ci and Cj must be similar; (ii) Similarity between subconcepts of Ci and Cj must be similar; (iii) Similarity between siblings of Ci and Cj is optional; (iv) Similarity between non-taxonomic relations of both concepts in their ontologies is also optional.

Our technique uses semantic equivalence of super concepts of any concept and hence automatically covers the general category of the concepts. So, it is relevant to compare only immediate super concepts. Further when considering OWL ontologies, attributes of concepts must be checked to see the concepts similarities but most of the earlier work considers only concept name matching as in[21]. In [22], structural aligning is based on direct super concepts.

4.4 Attribute Similarity
The equivalent concepts between contributing ontologies are further compared for their attribute matching. Attribute of concepts influences semantics of any concept. Stage-2 of the algorithm does attribute matching of equivalent concepts resulted in stage-1. Most of the earlier works considered only concepts terms for mapping. But, equivalent relationship between concepts depends on syntax, semantic and structural similarity of concept names and in addition, the concepts meaning is also based on their attributes. Hence, it is important to check for attribute mapping for semantic equivalence of concepts. Application with the facility of query-answering involves data from various heterogeneous data models. For example, buying a cellphone in an online shop compares attributes like ‘make’, ‘model’, ‘price’ of query against the attributes of various data models.

Attributes or data properties of equivalent concepts at stage-1 are verified for their semantic equivalence at first using WordNet. Then Concept correspondence is analysed using Jaccord Similarity measure which uses the semantic similarity of attributes.

Jaccard similarity [23] coefficient is a statistical measure to define the extent of overlap between two vectors. It is a popular similarity analysis measure of term-term similarity and it has retrieval effectiveness. Also in [24], statistical relatedness approach for semantic similarity for mapping document corpus in the civil industry and
found that Jaccard similarity shows the highest precision. The matching process proposed in [25] introduces syntactic and semantic matching to perform matchmaking for agent advertisements and requests in Internet.

Two concepts are considered semantically equivalent if there is a high extent of overlap between the two sets of attribute values. To illustrate, let ‘C’ be the number of attributes common in both concepts C1 and C2. Let ‘A’ be the number of attributes belong to concept C1 alone and not mapped to any attributes of C2, B be the number of attributes of concept C2 alone and not mapped to any attributes of concept C1. The Jaccard similarity between two entities is then computed as per equation-2.

\[
\text{Sim}(C_1, C_2) = \frac{C}{(C + A + B)} \quad (2)
\]

This measure defines the correlation between concepts based on their attributes. The threshold for accepting the concepts equivalence is set as 50%. The similarity value of equal or above the threshold is considered as similar concepts.

### 4.5 Object Property Correspondence

In OWL ontologies, relationships among entities are defined using object properties. The concepts are labeled “equivalent” after passing through two stages of comparisons like syntax, semantics, structure and attribute. At stage-3 of the algorithm, concepts are compared if their object properties are equal for labeling them as ‘strongly equivalent’. The object properties of equivalent concepts are checked if domain (P1) = domain(P2) & Range(P1) = Range(P2) Where P1 and P2 are object properties of equivalent concepts A and B respectively. If they are equal, then they are checked for equivalence of axioms like inverse property and disjoint properties if any, as these properties also contribute to the semantics of concepts. The results of concepts after passing through first three stages are considered ‘Strongly Equivalent’ and mapped automatically. Object properties are forming the relationship of one entity with other entity. To maximize the efficiency of mapping precision, our techniques also consider this non-taxonomic relation. Similarities of non-taxonomic relations of concepts are omitted in most of the earlier works.

### 5 Results and Discussion

#### 5.1 Results of Comparison

OWL ontology mapping takes places at stages. At stage-1, concepts are mapped after passing through syntax, semantic and structural similarity measure tests. At stage-2, attribute similarity measures are carried out for semantic similarity of concepts. At stage-3, similarity of object properties and their axioms are measured for the concept correspondence of contributing ontologies. Since the proposed algorithm considers various features of OWL ontology for ontology mapping, it enhances the efficiency of alignment. In [26], when the features defining each concept are different then obtained similarity is asymmetric. Jena framework is used for the implementation of the algorithm. Jena is a Java framework that provides an API to work with RDF, RDFS (i.e. RDF schema), OWL ontologies and SPARQL (i.e. Query language for RDF). It also includes a rule-based inference engine.

The proposed algorithm for ontology mapping is evaluated using two reference ontologies namely The University of Maryland (UMD) ontology and Carnegie Mellon University (CMU) ontology. CMU ontology has 24 classes namely Government, Employee, Scientist, Student, Email, Organization, Teaching, Education, Address, Management, Office, Sex, Person, Employment, Staff, Publication, Research, Project, Director, Academic, Post doc, Industry, Faculty, Work. UMD ontology has 21 classes namely Student, Activity, Artificial, Address, Web, Sexuality, Agent, Location, Employee, Event, Graduate student, Commercial Organization, Government Organization, Temporary, Person, Non-Profit Organization, Undergraduate student, Education Organization, Organization, Schedule and Experience.

#### Step 1 – Syntax, Semantic and Structural Similarity Test
Based on the proposed algorithm, Syntax similarity test for concepts of two reference ontologies UMD and CMU were conducted. Dice Co-efficient method is adopted for checking the syntax similarity and the performance was evaluated with varying threshold co-efficient as in Table-1 by comparing it with manual syntax similarity. Manual mapping of concepts are used to check the performance measure at every stage of the algorithm. It is found in the graph that the threshold value of 0.7 and above gives precision value as1 and recall value as .83 and hence threshold is set as 0.7. The equivalence vector of concepts for CMU and UMD is\{Student=Student, Address=Address, Person=Person, Organisation=Organization, Employee=Employee\}. Following Table-1and Graph in Fig-3 shows the Precision and Recall measure for various threshold values of during the Dice Syntax Similarity.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>.625</td>
<td>.83</td>
</tr>
<tr>
<td>0.7</td>
<td>1</td>
<td>.83</td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>.83</td>
</tr>
</tbody>
</table>

Table-1 Threshold Test for Syntax Similarity

![Syntax Similarity](image)

The concepts at column 2 and column 3 of the table are tested for their equivalence using WordNet. The resultant mapping vector is \{Employee, Address, Person, Sex/Sexuality\}. Concepts ‘Student’ and Organization have different super classes and hence they are not part of structural similarity map though these concepts find similarity in syntax and semantic tests. The concepts fail in structural match would lead to wrong interpretations for the query in the semantic web applications. Hence it is necessary to test for structural match after syntax and semantic test. Finally, at the end of stage-1 of the algorithm, the resultant equivalent mapping vector is \{Employee, Address, Person, Sex/Sexuality\}.

### Table-2 Super Class of Ontology Classes

<table>
<thead>
<tr>
<th>Concept Name</th>
<th>Super Class - CMU</th>
<th>Super Class - UMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employee</td>
<td>Person</td>
<td>Person</td>
</tr>
<tr>
<td>Student</td>
<td>Research</td>
<td>Person</td>
</tr>
<tr>
<td>Address</td>
<td>Resource</td>
<td>Resource</td>
</tr>
<tr>
<td>Person</td>
<td>Resource</td>
<td>Resource</td>
</tr>
<tr>
<td>Organisation</td>
<td>Resource</td>
<td>Agent</td>
</tr>
<tr>
<td>Sex/Sexuality</td>
<td>Resource</td>
<td>Resource</td>
</tr>
</tbody>
</table>

At stage1, only dissimilar items resulted from syntax similarity of both concept vectors are considered for semantic similarity. Here semantic similarity is done using WordNet and it is found that the concept ‘Sex’ in ontology CMU is equivalent to the concept ‘Sexuality’ in UMD. Hence the equivalence vector \{Student=Student, Address=Address, Person=Person, Organisation=Organization, Employee=Employee, Sex=Sexuality\}. Structural similarity test is applied to all the above concepts in the vector. Super Class of the concepts in the vector are found out and it is tabulated in Table-2.

The concepts at column 2 and column 3 of the table are considered as attributes of concepts. The reference ontologies do not have data type properties and hence only object properties are considered for similarity of
one concept with another. Similarity of object properties makes the mapping strong as it decides the individual mapping or ‘ABOX’ mapping of ontologies. The concepts from the equivalent vector found in stage 2 are considered for comparison of their object properties. The concept ‘Employee’ in CMU and UMD ontology have properties vector [position, name] and [sex, job, position, name] respectively. The resultant object similarity vector is [position, name] after WordNet match. Jaccord Co-efficient is calculated for finding the correlation of the concepts and it is $2/(2+0+2)=.5$ as per equation-2 and it gives 50% correlation and hence the concept ‘Employee’ is acceptable as the threshold value is equal or above 0.5. The concept ‘Address’ in both the ontologies have object property vector [city, street] and hence Jaccord co-efficient results in $2/(2+0+0)=1$ and the concept ‘Address’ is having 100% correlation.

The object property vector [position, name] is mapped for the concept ‘Person’ of both ontologies and achieves 100% correlation. The concept ‘Sex’ of CMU has object property vector [male, female] but the concept ‘Sexuality’ concept in UMD has no object vector and hence these concepts are not tagged as ‘Strongly equivalent’. At this stage, some concepts are not mapped to each other though those concepts are tagged as ‘equivalent’ in earlier stages of algorithm. The resultant vector with Jaccard co-efficient is {((Employee, .5), (Address,1), (Person,1), (Sex,0))}.

Hence, the concepts in vector {‘Employee’, ‘Address’ “Person”} are mapped automatically and the concept map of ‘Sex/Sexuality’ is suggested for manual mapping as they are not matching with the attributes used to define them. The ontologies may differ in their size and the relationships among concepts. But the algorithm tries to find out only the concepts which could be mapped as ‘equivalent’ so that the individual of those concepts are aligned for application interoperability. Some of the earlier mapping techniques considered individuals for mapping which will not allow for dynamic and autonomous mapping possibilities for a dynamic growth of a distributed knowledge base.

Fig-4 CMU and UMD ontology mapping using PROMPT
5.2 Evaluation of the Approach

Precision and recall are commonly used as the metrics to evaluate the accuracy of ontology mapping approach. Precision measures the fraction of predicted matches that are correct, i.e., the number of true positives over the number of pairs predicted as matched. Recall measures the fraction of correct matches that are predicted, i.e., the number of true positives over the number of pairs that are actually matched.

They are computed as

\[
\text{Precision} = \frac{|\text{True Matches} \cap \text{Predicted Matches}|}{|\text{Predicted Matches}|}
\]

\[
\text{Recall} = \frac{|\text{True Matches} \cap \text{Predicted Matches}|}{|\text{True Matches}|}
\]

The result of ontology alignment after having passed all stages of the algorithm are the concepts in the vector {Employee, Address, Person} and the concept ‘Sex’ is suggested for manual mapping. Hence, precision = \( \frac{3}{4} \) and recall = \( \frac{3}{3} \). F-measure is weighed harmonic mean of precision and recall and it is the weighed reciprocal of the arithmetic mean of the reciprocals of precision and recall. It is given as

\[
\left[ 2 \times \frac{(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \right]
\]

Prompt estimates F-Measure value to be 0.67 and proposed hybrid approach calculates it as 0.86.

5.3 Comparison with popular algorithm

Results are further compared for its evaluation with popular ontology matching algorithm “Prompt” using the same ontologies. The Fig-4 shows the results of “Prompt” where it finds a matching among 5 concepts namely {Employee,Employee}, (Student,Student), (Sex,Sexuality), (Publication, Location), (Person,Person). But the predicted Match is {Employee,Person,Address,Sex}.

The result of evaluation is Precision = \( \frac{3}{4} \) and Recall = \( \frac{3}{5} \) using Prompt algorithm for mapping. Prompt mapping is semi-automatic and proposed algorithm maps the ontologies automatically for any number of input ontologies. Comparison of performance of proposed algorithm with PROMPT is given in Fig.5.

6. Conclusion

The proposed ontology alignment algorithm combined the approaches of syntax, semantics, structural matching for automatic mapping of ontology for any given OWL ontologies. The algorithm makes the ontology mapping stronger as it executes the stages in the order mentioned. Automatic alignment of OWL ontologies are guaranteed by the software developed using JENA technology. The reference university ontologies CMU and UMD are downloaded from web and tested for their alignment. The ontology mapping is evaluated for precision and recall measures which are 75% and 100% respectively. The result of the algorithm is also compared with the popular ontology matching algorithm “Prompt” which gives the precision 75% and recall is 60% for the same reference ontologies.

The incorrect mapping was (Publication, Location) found in PROMPT test and which may lead to erratic result in the semantic applications. Finally, our algorithm finds any complex mappings between OWL ontologies.

REFERENCES


