International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 08 | Aug -2017 www.irjet.net p-ISSN: 2395-0072

SEMI AUTOMATIC TO IMPROVE ONTOLOGY MAPPING PROCESS IN SEMANTIC WEB DATA ANALYSIS

[*1]Ms. Suganya E., [*2] Ms. Abarna N

*[*1]M.Phil Research Scholar, PG & Research Department of Computer ScienceArcot Sri Mahalakshmi Women's College , Vilapakkam, Tamil Nadu, India*
*[*2].Assistant Professor, PG & Research Department of Computer Science Arcot Sri Mahalakshmi Women's College, Vilapakkam, Tamil Nadu, India*

Abstract: *Ontology Mapping is an important issue, widely recognized in a research community of Semantic Web. Large number of ontologies, developed across the World Wide Web in a distributed manner demands for automatic or at least semi-automatic ontology mapping system to integrate information from different ontologies and to make the vision of Semantic Web reality. The integrated approach here means two things: (1) it integrates several techniques from different computational area such as Computational Linguistic, Information Retrieval, and Machine Learning in order to provide semi-automatic ontology mapping process; and (2) It takes care of ontology mapping process right from the creation of the ontologies. The algorithm performs these steps in iterative and interleaved manner depending on its execution configuration. The language processing activities such as Tokenization, Lemmatization, Abbreviation Expansion, Spelling Correction, Elimination of Stop words, etc. are performed whenever required. It uses domain specific thesaurus for abbreviation expansion and synonym. It also uses Word Net, an online lexical database, to strengthen the Linguistic Matcher. The AI - ATOM has been evaluated using the measures precision, recall, and F-measure on two small real world data sets. The system is tested with different algorithm configuration to decide the best possible default configuration for the domain under consideration. The preliminary case studies show encouraging result.*

Keywords: Ontology Mapping, ATOM, wordnet, Tokenization, Semantic Web, F-Measure Algorithm, XML/ RDF Framework.

I. INTRODUCTION

1.1 Semantic Web

The present WWW has huge collection of pages, but majority of them are in human readable format only. As a consequences software agent cannot understand and process this information, and much of the potential of the Web has so far remain untapped. To overcome this problem, delivered at a meeting of the World Wide Web Consortium (W3C), Tim Berners-Lee presented his vision for a new Web, Semantic Web, where machines (software agents) would be able to automatically find and process knowledge[8]. The Semantic Web, as the name implies, is a Web focused on the conveyance of meaning. It facilitates modeling, sharing and reasoning with knowledge available on the Web through the formal representation of knowledge domains with ontologies (roughly, agreed terms)[8].The present HTML based WWW defines Web pages syntactically and were intended only for human consumption. The HTML defines how content of the Web pages should be displayed on browser, but does not tell anything about the subject and nature of the content. Hence, these Web pages cannot be read and processed by machines without human intervention to derive any meaning out of it. With the Semantic Web, information on the Web can be defined semantically in such a way that it can be used by machines, not only for display purposes, but also for interoperability and integration [4].

1.2 Semantic Web Layered Architecture

The Semantic Web is consisting of a philosophy (idea), a set of design principles, collaborative working groups, and a variety of enabling technologies. Some elements of the Semantic Web are expressed as prospective future possibilities that have yet to be implemented or realized. Other elements of the Semantic Web are expressed in formal specifications [16]. Some of these formal specifications include Resource Description Framework (RDF), a variety of data interchange formats (e.g., RDF/XML, N3, Turtle, and N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain. Key technologies in Semantic Web include explicit meta-data, ontologies, logic and inference, and intelligent agents [7].

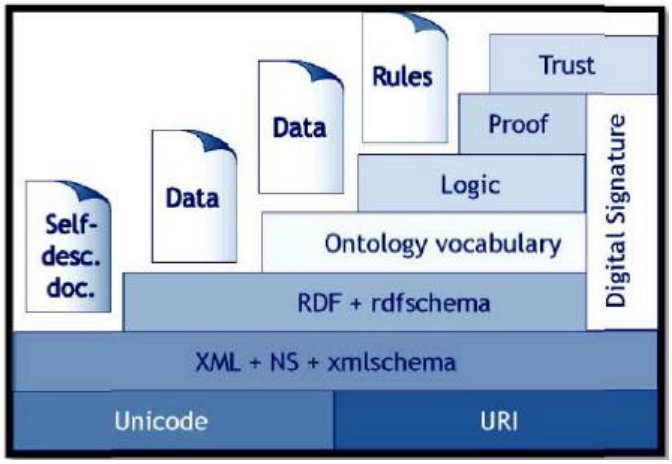


Fig.1.1 Semantic Web Layered Architecture

This diagram shows set of technologies, tools, and standards; organized into a certain structure that is an expression of their interrelationships; which form the basic building blocks of an infrastructure to support the vision of the Web associated with meaning[6].

3-step methodology for knowledge integration, which consists in:

- Expressing ontological content in a linguistically motivated fashion, as a necessary part of the development of ontologies
- Automatically discovering linguistic and semantic evidences to suggest conceptual similarities during automatic ontology alignment.
- Supporting users in the process of producing assessed ontology mapping documents, offering reliable knowledge for providing semantic links across different information sources

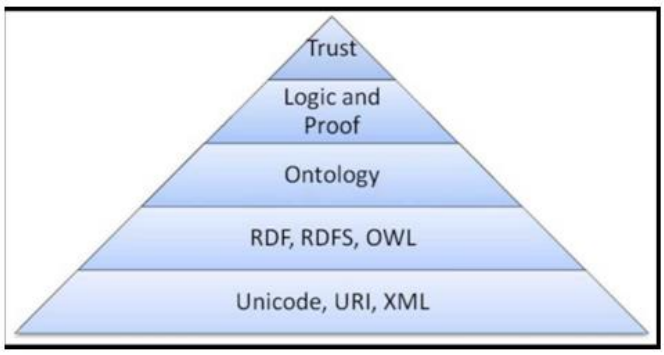


Fig.1.2 Modified Semantic Web Layered Architecture

This diagram represent slightly modified diagram for Semantic Web Layered Architecture. There are two reasons considered for this modification: first, OWL, a new recommendation from W3C for writing ontology, is missing in original diagram; and second, I believe that Trust Layer include the concept of digital signature along with other means such as certificate and trusted agencies as similar to the one, already implemented in present Web.

Linguistic element level matching: It uses the information such as names, data types, and domains of element. It also uses tokenization for compound element names before computing element level score. For name matching it uses morphological normalization, categorization, string based techniques such as common prefix and suffix; and a thesauri lookup[12].

Structural matching: This component transforms input schemas into trees that enrich the structure by augmenting referential constraints. The linguistic similarity of node and similarity of their leaf nodes are combined to decide similarity between two elements. Finally, it calculates weighted mean of linguistic and structural similarity.

Mapping identification: This step depends on the application in which the algorithm is used and decides the mappings based on weighted mean calculated in previous phase [2][1].

Cupid represents the input schemas internally as trees. These trees are processed by name matchers assisted with auxiliary thesaurus to get linguistic similarity at element level and then by structural matchers that uses name and data type information of leaves in an iterative manner with knowledge propagation until the desired threshold is exceeded. It combines linguistic and structural similarities using a weighted sum. Finally, it generates the alignments based on this weighted similarity crossing some threshold values[7][8].

II. RELATED WORK

2.1 ONTOLOGY MAPPIN SYSTEM

COMA

COMA represents the schema internally as a directed acyclic graph where elements are the paths. This is done to capture the contexts in which elements occur. It allows the user interaction to improve the accuracy of matching based on approval of suggested matches or mismatches. It may also be used to evaluate the different combinations of matchers. The main components of COMA are: the Repositoryto persistently store all match-related data, the Modeland Mapping Poolsto manage schemas, ontologies and mappings in memory, the Match Customizerto configure matchers and match strategies, and the Execution Engineto perform match operations. The latest version of COMA, COMA++, improves the algorithm and provides a GUI. The Figure 16 shows the overall architecture of COMA++[8].

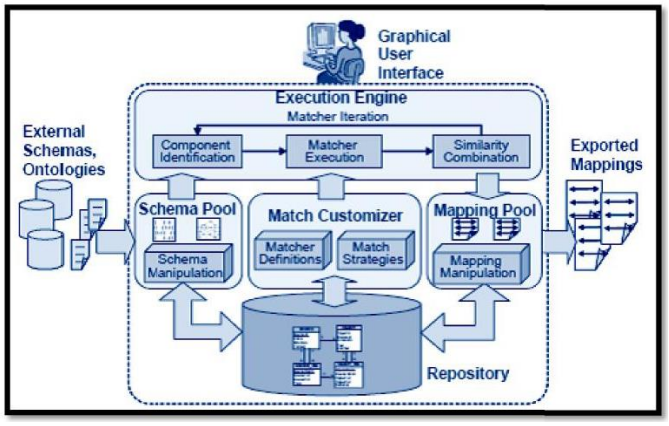


Fig2.1 Architecture of COMA++

2.2 Cupid

Cupid (Microsoft Research) implements a hybrid matching algorithm combining linguistic and structural schema matching techniques. It calculates the normalized similarity with the help of external thesaurus. It is generic in the sense that it can be applied to XML as well as relational schemas.Cupid represents the input schemas internally as trees[2]. These trees are processed by name matchers assisted with auxiliary thesaurus to get linguistic similarity at element level and then by structural matchers that uses name and data type information of leaves in an iterative manner with knowledge propagation until the desired threshold is exceeded. It combines linguistic and structural similarities using a weighted sum. Finally, it generates the alignments based on this weighted similarity crossing some threshold value[11].

2.3 GLUE

GLUE uses machine learning technique to find mappings. It employs multiple learning strategies and uses different types of knowledge to improve the process. GLUE finds the most similar concepts between two ontologies and calculates the joint probability distribution of the concept using a multi-strategy learning approach for similarity measurement. It uses machine learning techniques to compute for every pair of concepts their joint probability distribution. For this purpose it implements two base learners, viz., Content Learner and Name Learner; and third learner called Meta Learner that combines the two base learners' prediction. But, it involves huge manual effort to train these algorithms before they can be used effectively. It also uses a technique called Relaxation Labeling that assigns labels to nodes of a graph, given a set of constraints. It consists of three modules, viz.,Distribution Estimator, Similarity Estimator, and Relaxation Labeler. The Figure 19 shows the overall architecture of GLUE[16].

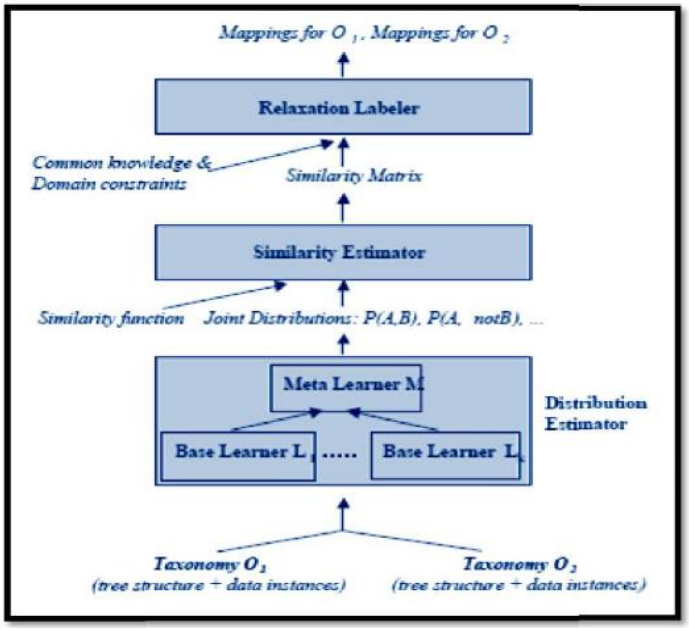


Fig2.2: GLUE Architecture

III. PROPOSED ANALYSIS

3.1 Terminology

Ontology: Ontology is a 6-tuple denoted by O = {C, CH, P, PH, I, A}, where C is the set of concepts (Classes), CH defines Class Hierarchy (the hierarchical relationships among the concepts), P is the set of properties (Object Property defining the relationship between concepts and Data Property defining the attributes of the concepts), PH defines Property Hierarchy (the hierarchical relationships among the properties), A is the set of axioms, I is the set of instances of concepts and properties. The present work considers only C and CH components of ontology.

Ontology Mapping*:* Ontology mapping finds correspondences between semantically related entities of ontologies covering the same domain in such a way that intended interpretations in shared domain are preserved[3][9].

Mapping Element*:* A mapping element (also known as correspondence) is a 6-tuple: ME=(Id, e1, e2, n, r, uid), where Id is a unique identifier of the given correspondence; e1 and e2 are entities (e.g., Class, Object Property, Data Property, or Instance) of the first and the second ontology, respectively; n is a confidence measure (typically in the [0, 1] range) holding for the correspondence between e1 and e2; and r is a relation holding between e1 and e2, which could be, for example, equal (=),more general (>), less general (<), or disjoints (!); and uid is ID of user who approved this mapping

element. The present work considers only equality relation.

Concept's Local Label*:* Concept's Local Label is the class name given by user to describe the concept.

Concept's Path Label: Concept's Path Label is Label derived based on path from root (or grandparent or specified super class) node to class node. It augments labels of all parent class nodes with label of class node.

The basic objective of the proposed system is to increase the automation in ontology mapping process by using all possible source of information from ontology and various techniques in an integrated manner[12].

A GUI based system AI-ATOM is developed as a prototype for the proposed integrated approach. The system consists of five horizontal components, viz., Ontology Management, Ontology Project Management, User Management, System Configuration and Ontology Mapping Engine; and six vertical components, viz., Language Processing, VSM Engine, Label Matcher, Linguistic Matcher, Structure Matcher, and User Interaction[5]. The brief description of each component is given below. It allows the user to create new ontology, maintain an existing ontology, delete existing ontology, and to import ontology from file containing parenthesized tree. While user creates new ontology, the system suggests the concept labels along with their meaning from the user specified context dictionary (Domain Specific Labels), based on its usage frequency, that are best matching with partially entered labels by user. The context dictionary is maintained with the help oftable. This helps the user from typing and thinking different label from intention of the community of users in the same domain. This helps a lot when two suchontologies need to be mapped later. Thus, systemproposed here tries to improve ontology mapping process right from its creation.

Fig3.1: Architecture of AI-ATOM

While user creates new ontology, the system suggests the concept labels along with their meaning from the user specified context dictionary (Domain Specific Labels), based on its usage frequency, that are best matching with partially entered labels by user. The context dictionary is maintained with the help of following table. Domain Specific Label (DSL ID, Label, Meaning, Frequency of Use).

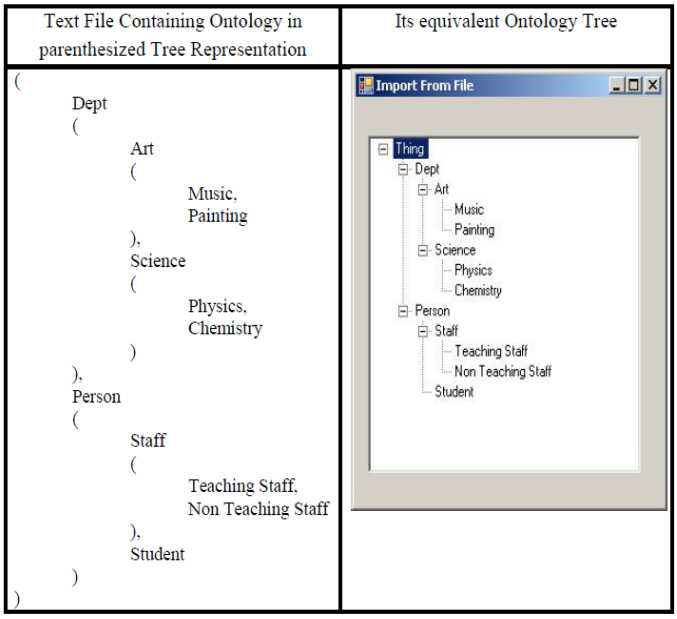


Fig3.2: General Tree Representation using parenthesis for Ontology

This helps the user from typing and thinking different label from intention of the community of users in the same domain. This helps a lot when two such ontologies need to be mapped later. Thus, system proposed here tries to improve ontology mapping process right from its creation.

3.2 Data Structure

The Data Structure used by Algorithm is listed below:

- User (User ID, User Name, Password)
- Ontology (Ontology ID, Ontology Name, Description, Owner User ID, Date of Creation)
- Class (Class ID, Class Name, Description, Ontology ID
- Class Has Sub Class (CHSC ID, Class ID, Sub Class ID)
- Ontology Mapping Project (OMP ID, OMP Name, Description, From Ontology ID, To Ontology ID, Created By User ID, Start Date, Due Date, Completed Date)
- Ontology Mapping Project User (OMPU ID, OMP ID, User ID)

- Domain Specific Label (DSL ID, Label, Meaning, Frequency of Use)
- Domain Specific Abbreviation (DSA ID, Abbreviation, Expanded Text)
- Domain Specific Synonym (DSS ID, Synonym, Synonym Group ID)
- Stop word (Stop word ID, Stop word)
- Algorithm Configuration (AC ID, AC Name, Description, For OMP ID, Precision, Recall, F Measure)
- Parameter (Parameter ID, Parameter Name, Description, Default Value Text, Default Value Number)
- Parameter Possible Value (PPV ID, Parameter ID, Possible Value Text, Possible Value Number)
- Algorithm Parameter Value (APV ID, For AC ID, Parameter ID, User PPV ID)
- Mapping Element (ME ID, OMP ID, From Class ID, From Class Name, To Class ID, To Class Name, Status, Relation, Confidence, Explanation, Processed By System YN, Processed By User YN, User ID)
- Status=Candidate/Potential/System Generated/Accepted/Rejected
- Accepted Rejected Mapping Group (ARMG ID, Class Name, Relation, Group ID, Accepted or Rejected).

3.3. Ontology Mapping Project Management

It allows the users to create, maintain, and delete an ontology mapping project. It also allows assigning users to mapping project, and to review completed and pending mapping pairs which are yet to be processed for its acceptance or rejection [3]. Thus, this system supports a new approach of multi-user, multi-session ontology project management in order to reduce burden on singledomain expert and to provide the time flexibility to users.

Mapping Algorithm

Linguistic Matcher Similarity=0.0
Structure Matcher Similarity=0.0
System Similarity=0.0
IF (Label Matcher Should Be Used) Then

//Compute Edit Distance
IF (Edit Distance Should Be Used) Then
Edit Distance Similarity=get Edit Distance Score (*Ci, Cj*)
If (Edit Distance Threshold Should Be Used) Then
If (Edit Distance Similarity >= Edit Distance Threshold) Then
SELECT ME(*Ci,Cj*)
Process Next ME (*Cp, Cq*)
End If
End If
End If
Label Matcher Similarity=Max (Label Matcher Similarity, Edit Distance Similarity)
If (Label Matcher Threshold Should Be Used) Then
If (Label Matcher Similarity >= Label Matcher Threshold) Then
SELECT ME(Ci,Cj)
Process Next ME (*Cp, Cq*)
End If
End If
End If

//Compute N-Gram
IF (NGram Should Be Used) Then
NGram Similarity=get NGramScore(Ci, Cj)
If (NGram Threshold Should Be Used) Then
If (NGram Similarity >= NGram Threshold) Then
SELECT ME(Ci,Cj)
Process Next ME (Cp, Cq)
End If
End If
Label Matcher Similarity=Max (Label Matcher Similarity, NGram Similarity)
If (Label Matcher Threshold Should Be Used) Then
If (Label Matcher Similarity >= Label Matcher Threshold) Then
SELECT ME(Ci,Cj)
Process Next ME (Cp, Cq)
End If
End If
End If
/*

Similarly, perform following Label sub-matchers.

i. N-Gram
ii. Prefix and Suffix
iii. Soundex
iv. Heuristic Rules-1 to N

If (Label Matcher Threshold Should Be Used) Then
If (Label Matcher Similarity >= Label Matcher Threshold) Then
SELECT ME (Ci,Cj)
Process Next ME (Cp, Cq)
End If
System Similarity = Max (System Similarity, Label Matcher Similarity)
If (System Similarity Threshold Should Be Used)
If (System Similarity >= System Similarity Threshold) Then
SELECT ME(Ci, Cj)
Process Next ME(Cp, Cq)
End If
End If
End If

If (Linguistic Matcher Should Be Used) Then
If (Domain Synonym Should Be Used) Then
If (isDomainSynonym (Ci, Cj)) Then
SELECT ME(Ci, Cj)
Process Next ME(Cp, Cq)
End If
End If
If (WordNet Synonym Should Be Used) Then
If (isWordNetSynonym(Ci, Cj)) Then
SELECT ME(Ci, Cj)
Process Next ME(Cp, Cq)
End If
End If
If (WordNet Gloss Should Be Used) Then
WordNet Gloss Similarity = getWordNetGlossScore(Ci, Cj)
System Similarity=System Similarity + (1-System Similarity) * WordNet
Gloss Similarity
End if
End If

IV. EVALUATION RESULT

The effectiveness of the system in processing the mapping elements is measured by looking at Precision and Recall [67] [24]. The ontology mapping systems are evaluated with respect to the notion of correctness perception – a judgment by a human that a mapping element found by ontology mapping algorithm is correct or not. A system's ability to retrieve correct mapping elements is assessed with a Recall measure that is defined as below:
Recall = | Relevant and Retrieved | / | Relevant |

A system can achieve 100% recall by simply returning all the possible mapping elements between two ontologies.A system's accuracy is based on how many of the mapping elements generated by system areactually correct as per user's decision, which can beassessed by a Precision metric and is defined below.
Precision = | Relevant and Retrieved | / | Retrieved |

Ideally both should be 100%. But, most of the system scarifies one for the other. Hence, to measure the optimum balance between these two measures, F-Measure is used which is defined as below:
F-Measure = 2 * (Precision * Recall) / (Precision + Recall)

The Precision and Recall can be understood from the Figure 47, Where:

- A = False Positives
- B = True Positives
- C = False Negatives
- D = True Negatives

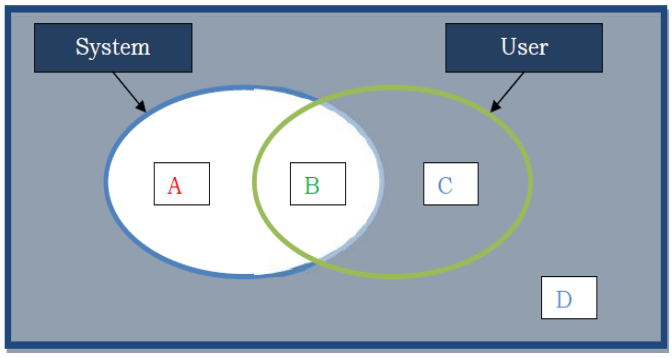
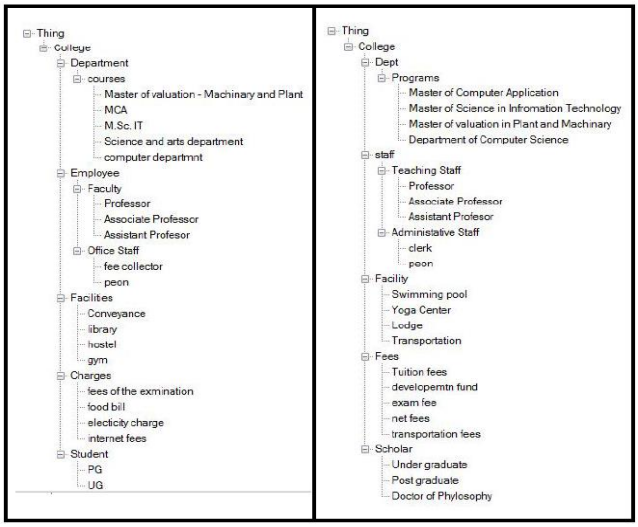


Fig3.3 : Precision, Recall, and F-Measure

Using these notions; Precision, Recall, and F-Measure can be defined as following.

- Recall = | Relevant and Retrieved | / | Relevant | = B / (B+C).
- Precision = | Relevant and Retrieved | / | Retrieved |= B / (A+B).
- F-Measure= 2 * (Precision * Recall) / (Precision + Recall) = 2 * B/ ((A+B) + (B+C)).

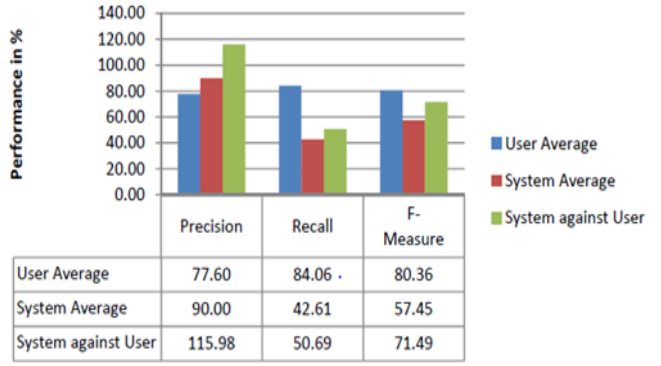
The first data set represents sample ontologies represented by two different academic institutes, whereas second data set represents snapshot of database schema from two academic institutes. Both the data sets are selected from the academic domain as one of the objectives of study is to analyze and to present the significance of domain knowledge in the automated ontology mapping process. In experimental setup, these data sets are given to few users having varying knowledge regarding the ontology mapping.



Data Set4.1: Sample Ontologies from two different Academic Institutes

	Precision	Recall	F-Measure
Actual	100.00	100.00	100.00
System	100.00	43.48	60.61

Fig4.2: Overall System Performance for Data set System Performance against User



	Precision	Recall	F-Measure
User Average	77.60	84.06	80.36
System Average	90.00	42.61	57.45
System against User	115.98	50.69	71.49

Fig4.3: System Performance against User for Data set

V.CONCLUSION

An Integrated approach is proposed that takes care of ontology mapping process from the very first step of ontology creation by allowing the user to set and use domain specific context dictionary and allowing the user to set other domain specific thesaurus such as abbreviation and synonym to provide context information to ontology mapping process that improves automation of ontology mapping process. It effectively combines the power of vector Space Model, used in information Retrieval, to generate potential candidate mapping elements to be selected for further processing. Thus it expedites the process by eliminating large number of weak candidate mapping elements. It proposes the novel approach of using previously rejected mappings to speed-up the process in addition to re-use of accepted mappings.

It supports extensive configuration of algorithm. User can select matchers and sub-matchers to be included in algorithm and can decide their execution order. Allows the users to work in multi-user and multi-session environment. A set of heuristic rules with high degree of feasibility is used in Label Matcher and in Structure Matcher to support automation. The algorithm includes a basic learning component which takes advantage users' feedback to improve ontology mapping process in future.

FUTURE WORK

The VSM is used in algorithm for filtering weak mapping elements, but it may be improved and used as a matcher to find mapping elements whose VSM similarity (cosines score) crosses specified threshold. The advantage of algorithm configurability with different inclusion of matchers, their order of execution, and with different threshold values needs to be assessed. The evaluation of the algorithm for different combinations of matchers and system parameters, a gigantic job, is yet to be performed.The algorithm can be improved using knowledge of all components of ontology such as instances and complex relationship.

REFERENCES

1. Abels, S., Haak, L., & Hahn, A. (2005). Identification of Common Methods Used for Ontology Integration Tasks. IHIS'05. Bremen, Germany.
2. Ahmed, A., Volker, H., & Nematollaah, S. (2008). An Empirical Comparison of Ontology Matching Techniques. Journal of Information Science, 1-20.
3. AMIA2003 Tutorial.ppt. (n.d.). Retrieved 11 2, 12, fromprotege.stanford.edu/amia2003/AMIA2003Tutorial.ppt.
4. Antoniou, G., &Harmelen, F. v. (2008). A Semantic Web Primer (Vol. 2e). Massachusetts London, England: The MIT Press Cambridge.
5. Beckett, D., &Broekstra, J. (2008). SPARQL Query Results XML Format. W3C Recommendation. World Wide Web Consortium.
6. Borst, W. N. (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Enschede, The Netherlands:University of Tweenty.
7. Choi, N., Song, I.-Y., & Han, H. (2006, September). A Survey on Ontology Mapping. SIGMOD Record , 35 (3).
8. Clark, K. G., Feigenbaum, L., & Torres, E. (2008). SPARQL Protocol for RDF. W3C Recommendation. World Wide Web Consortium.
9. def.txt. (n.d.). Retrieved 2004, from http://www.tartarus.org/:http://www.tartarus.org/~martin/PorterStemmer/def.txt
10. DERI-TR-2003-10-29.pdf. (n.d.). Retrieved 11 2, 12, from www.deri.ie/: www.deri.ie/fileadmin/documents/DERI-TR-2004-06-30.pdf
11. Do, H.-H., & Rahm, E. (2002). COMA - A System for Flexible Combination of Schema Matching

Approaches. 28th Intl.Conference on Very Large Databases (VLDB). Hong Kong, China.

12. Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2003). Learning to Map between Ontologies onthe Semantic Web. VLDBJournal, Special Issue on the Semantic Web .
13. Dublin Core Metadata Initiative (DCMI). (n.d.). Retrieved February 28, 2013, from Dublin Core: http://dublincore.org/
14. Ehrig, M., &Staab, S. (2004). QOM – Quick Ontology Mapping JA. ISWC 2004, LNCS 3298, (pp. 683-693).
15. Ehrig, M., &Staab, S. (2004). QOM— Quick Ontology Mapping Report. LNCS, Institute AIFB, University of Karlsruhe
16. Euzenat, J., Isaac, A., Meilicke, C. S., Stuckenschmidt, H., Svab, O., Svatek, V., et al. (2007). First results of the Ontology Alignment Evaluation Initiative 2007. Second Intrnational Workshop on Ontology Mapping. IEEE Xplore.
17. Euzenat, J., Loup, D., Touzani, M., &Valtchev, P. (2004). Ontology alignment with OLA. Proceedings of the 3rd EON Workshop, 3rd International Semantic Web Conference.