IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-6, Issue-1 E-ISSN: 2347-2693

Comparative Study on Ontology Management Approaches in Semantic Web

Ranjna Jain [1*], Neelam Duhan [2], A.K.Sharma [3]

[1*]Dept. of Computer Engineering, YMCAUST, Faridabad, India
[2]Dept. of Computer Science and Engineering, BSAITM, Faridabad, India
[3]Dept. of Computer Science and Engineering, BSAITM, Faridabad, India

Corresponding Author: ranjna.gupta@gmail.com

Available online at: www.ijcseonline.org

Received:19/Dec/2017, Revised: 30/Dec/2017, Accepted: 20/Jan/2018, Published: 31/Jan/2018

Abstract**—** Ontology defines a common vocabulary which includes machine interpretable definitions of basic concepts in the domain and relation among them for researchers who need to share information in a domain. But in actual, in place of reusing existing ontologies of required domain, domain experts create their own ontology leading in formation of multiple ontologies of the same domain containing incomplete concepts and relations. This causes ontology heterogeneity and inconsistency problem. For better and precise results managing, these heterogeneous ontologies are necessary. A large number of work has been done in the recent past for managing the existing ontologies so that they can be reused for data integration, information integration, data warehousing and other fields. This paper provides different approaches that have been used in the recent years for managing these ontologies.

Keywords**—** Ontology, Ontology Merging, Ontology Alignment, Ontology Mapping, Ontology Matching

I. INTRODUCTION

Initially, ontologies were developed in artificial intelligence to smoothen the process of knowledge sharing and reuse. After some time, it became a popular topic in knowledge engineering, natural language processing, and knowledge representation and information integration. The reason of popularity of ontologies is the shared and common understanding of some domain that can be communicated among people and application systems. For example, suppose several different websites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different web sites. The agent can use this aggregated information to answer user-queries or as output data to other applications. But in actual, this rarely happens. For example, as there exists more than one web directory (YAHOO, ODP etc.), more than one shopping site, in the same way there exists more than one ontology for the same domain on the semantic web with their own data vocabulary. Therefore, integration of ontologies is a major challenge and research issue in semantic web.

A number of challenges such as finding similarities and difference among ontologies in automatic and semi-automatic way, defining mapping between ontologies, composing mappings across different ontologies has to be faced during managing these diverse ontologies. Ontology management is possible through interoperability of semantic data sources. Ontology management includes operations such as ontology integration, ontology merging and ontology alignment. *Ontology merging* is the process of generating a single coherent ontology from two or more existing and different ontologies related to the same subject. *Ontology alignment* is the task of creating links between two original ontologies. *Ontology integration* is the process of generating a single ontology in one subject from two or more existing and different ontologies in different subjects. The different subjects of different ontologies may be related.

This paper presents a survey on the state of the art of ontology alignment, merging and integration. It includes recent approaches specifically addressing the concept matching methods employed by these methods for ontology management.

The rest of the paper is organized as follows: Section 2 describes the survey done on various ontology management methods; section 3 dictates the comparison study performed on the discussed ontology management methods and finally, Section 4 concludes the paper.

II. ONTOLOGY MANAGEMENT METHODS

This section elaborately discusses about the existing methods, tools and frameworks based on ontology management operations. BLOOMS+, ASMOV, RiMOM, COMA 3.0, YAM++, SIMTSS, MAPSSS, SEM+, and MEDLEY come under ontology alignment operations. Chimaera, Prompt, HCONE, SAMBO, ATOM come under ontology merging operations and falls in ontology integration operations. This survey gives the overview of the working of the underlying system, what inputs are required, what similarity measures are used for concept matching and the outputs generated thereof.

2.1 Ontology Alignment Methods

Ontology alignment is the process of determining correspondences between concepts in ontologies. A set of correspondences is also called an alignment. There are three main dimensions for similarity- syntactic, semantic and structural, on the basis of which it finds correspondence between two concepts or relations of two different ontologies. For example, one concept says ‘worker’ from one ontology O1 and another concept ‘employee’ from other ontology O2. Syntactically, they are not similar but semantically they are same as they are synonym to each other. So, with these similarity methods, alignment approach finds the correspondence between the concepts and relations of two different ontologies. Some of the prevalent ontology alignment methods have been discussed as below.

2.1.1 BLOOMS+ [1]: Jain P. et al. developed BLOOMS+ in 2011 is an ontology alignment system based on bootstrapping information already present on LOD cloud. It utilizes the Wikipedia category hierarchy for aligning ontologies. BLOOMS constructs a forest (i.e. a set of trees) T_C (BLOOM forest for concept C) for each matching candidate class name C_i. It tokenizes the name of C and removes stop-words from the name and then it gives resulting terms as a search string to retrieve relevant Wikipedia pages using Wikipedia search web service. BLOOMS+ treats each page as a possible sense S_i of C and constructs a category hierarchy tree. It then compares each class C’s forest T_C in the source ontology with each class D’s in forest T_D in the target ontology to determine their similarity. Once the class similarity has been determined, it then computes contextual similarity. It uses superclass of C and D to determine if they are contextually same. Using class similarity and context similarity, BLOOMS+ finally determines whether or not C & D should be aligned.

2.1.2 ASMOV [2]: Mary Y. et al. proposed *Automated Semantic Matching of Ontologies with Verification (ASMOV)* in 2010, which is a novel algorithm that uses lexical and structural characteristics of two ontologies to iteratively calculate a similarity measure between them. It derives an alignment and then verifies to ensure that it does not contain semantic inconsistencies. It retrieves as input two ontologies to be matched. ASMOV process is an iterative process and is divided into two components: similarity calculation and similarity verification.

1. The *similarity calculation* process computes a similarity value between all possible pairs of entities, one from each of the two ontologies using four similarity measures: lexical similarity, structural similarity, restriction similarity and extensional similarity. This process results in a similarity matrix containing the calculated similarity values for every pair of entities. From those similarity matrices, a pre-alignment is extracted by selecting the maximum similarity value for each entity.
2. This pre-alignment is passed through a process of *semantic verification* which eliminates correspondences that cannot be verified by the assertions in the ontologies. Semantic verification process uses multiple entity correspondence, crisscross correspondence, disjointness subsumption, contradiction subsumption, equivalence incompleteness and domain range incompleteness for verification.

2.1.3 CIDER [3]: Gracia J. et al. proposed *Context and Inference baseD alignER (CIDSER)* in 2011 which is an ontology alignment system that extracts the ontological context of the compared terms by using synonyms, hyponyms, domains, etc. and then enriches such context by means of some lightweight inference rules. It performs similarity by first extracting the ontological context of each ontology term up to a certain depth (using synonym, hypernym, hyponym, textual description, properties, domains, roles, associated concepts etc.) using lightweight inference mechanism to add more semantic information that is not explicit in the asserted ontologies. Then, it uses linguistic (using Levenhstein method) and structural similarity (using vector space model) to find the similarity between each pair of terms. After this, the different similarities are combined within an Artificial Neural Network (ANN) to provide a final similarity degree. ANNs constitute an adaptive type of systems composed of interconnected artificial neurons, which change the structure based on external or internal information that flows through the network during a learning phase. CIDER uses two different neural networks for computing similarities between classes and properties, respectively. Finally, a matrix *M* with all similarities is obtained. The final alignment *A* is then extracted from this matrix *M*, finding the highest rated one-to-one relationships among terms, and filtering out the ones that are below the given threshold.

2.1.4 RiMoM [4]: Tang J. et al. proposed a multi-strategy ontology alignment framework in 2009 which aims at finding the optimal alignment by combining different strategies. It uses five strategies- edit distance based strategy, statistical learning based strategy for linguistic matching and three similarity propagation based strategies (including concept to concept propagation strategy, property propagation strategy and concept to property propagation strategy) for structural matching. If two ontologies have high structure similarity factors, then RiMoM employs an algorithm called similarity propagation to refine the discovered alignments.

2.1.5 COMA 3.0 [5]: Massmann S. proposed *COmmon MAtcher (COMA)* in 2011 which is a schema and ontology matching tool. It is divided in four modules where the three modules storage, match execution and mapping processing follow the input-processing-output pattern and the user connection module provides different ways to access the program. The *storage* consists of the importers that load schemas, ontologies, existing mappings and auxiliary information in the repository. From repository, these files can be directly used to carry out matching task. The *match execution* is the core of COMA. It gets two schema or ontologies as input, runs several matching algorithms on those ontologies and calculates the match result. In this module, the execution engine determines the relevant schema components for matching, applies multiple strategies and finally combines the partial results to the final match result. The obtained mappings are further used as input in the next iteration for further refinement. The match library is a large bundle of schema matching strategies that can be combined to extensive workflows. The *mapping* processing module allows automatically enriching mapping, merging module or transforming data. The user connection module consists of full-fledged GUI to provide convenient way to use COMA.

2.1.6 YAM++ [6]: Ngo D. et al. proposed a semi-automatic mapping tool in 2012 which maps two ontologies at three levels. At the first level which is known as elementary level, it uses machine learning based combination methods such as decision tree, SVM, Naive Bayes etc. for this, it takes training data either from the user or from knowledge base. After this, at the second level named as structural level, input ontologies are parsed and transformed into graph data structure. For this, YAM++ takes elementary level mapping results as input and runs a similarity flooding algorithm to run a similarity propagation process. Finally, at the third level it performs semantic checking where it uses global constraint optimization. The resulting mapping of the match process is displayed at the GUI and then user judges if the mapping is correct or not according to his/her knowledge.

2.1.7 SIMTSS [7]: Essayeha A., Abeda M. proposed a research process in 2015 for the alignment between ontologies written in different languages such as RDF, SKOS, turtle etc. including heterogeneous information. The result is new data stored as an XML file stored in inference phases (query answering and integrating data). The system is divided into five layers. The first layer called Resource layer contains a collection of ontologies written in different languages. The system integrates all the ontologies in the matching process by mapping only the entities (concept, instances, and properties). In the pre-processing layer, ontologies written in different languages are standardized to OWL and then are normalized (lemmatization, lower case conversion, stop words and delete links). After this process, these ontologies are moved to the matching process layer. It aims to find first the relationship between their entities and degree of similarity by calculating the similarity measure. It measures the similarity at three levels: terminological, structural and semantic. Different methods are used at each level for similarity measurement and correspondingly generate measures in matrix format. This matrix is given as the input to the extracting alignment layer where an algorithm, *Hungarian algorithm,* is applied which highlights the most correct matches and eliminates less relevant once. The obtained alignments are stored as an XML file containing the two entities matching similarity relationship and similarity values between them. At last, this file is passed to the expert and configuration layer where expert confirms and suggests another alignment; and finally configures the output by using available tools.

2.1.8 MAPSSS [8]: Hitlzer C. proposed an ontology alignment system in 2013 that uses syntactic, structural and semantic metrics. This paper has evaluated wide range of string similarity metrics along with string preprocessing strategies on different type of ontologies. It mainly concentrates on following points:

1. Which effective string similarity metric for ontology alignment to choose if the primary concern is precision, recall and f-measure.
2. How to automatically select which string similarity metric and pre-processing strategies are best without any training data available.

It has grouped string metrics along three major axes: Global versus local, set versus whole string and perfect sequence versus imperfect sequence. *Global versus local* refers to the amount of information the metric needs in order to classify a pair of strings as match or a non-match. Global metrics must compute some information over all of the strings in one or both ontologies before it can match any strings whereas for local metrics it only requires only input string. *Perfect sequence metrics* require characters to occur in the same position in both strings in order to be considered a match. *Imperfect sequence metrics* equate matching characters as long as their positions in the string differ by less than some threshold. A *set based string metric* works by finding the degree of overlap between the words contained in two strings. *Word based set metrics* are generally perform well on long strings. For preprocessing, it has divided the categories in two major categories: syntactic and semantic. Syntactic pre-processing methods are based on the characters in the strings such as tokenization, normalization, stemming, stop-word removal. Semantic methods relate to the meaning of the string.

2.1.9 SEM+ [9]: Zheng J. G. proposed similarity based entity matching in 2014, which implements a novel semantic computation model called the information entropy and weighted similarity model to suggest similarity measures between concepts from different ontologies and vocabularies. Based on the similarity measures, SEM+ creates “same as” links among those concepts. SEM+ also implements a new prefix based blocking algorithm, which groups possible matching pairs into one block. This blocking algorithm reduces the number of concepts pairs that are needed for similarity computation, which is useful when it is required to perform mapping between two large domain ontologies. The prefix blocking groups concepts that are likely to be similar to each other into one block and dissimilar concepts into different blocks based on literal description of the concepts such as rdfs:label, rdfs:comment. SEM+ builds an indexer of these literals and computes the concept frequency of words appears in the literal description and then compares only the prefix of concepts. Similar concepts come in one block and thus prefix of that block get associated with the block. With this approach, similar concepts come in one block which reduces the similarity computation between each concept. For concept matching, it uses information entropy and weighted similarity model.

2.1.10 MEDLEY [10]: Hassen W. proposed an ontology alignment system in 2012 that uses lexical and structural methods to compute the alignment between classes, properties and instances. It also uses an external dictionary to tackle the problem of having concepts expressed in different natural languages. In the primary step, each entity in the first ontology is aligned with each entity in the second. In lexical metrics, it uses q-gram and levenshtein measure to calculate the similarity measure between nodes and then structural treatment is applied. For this, if an entity belonging to a given ontology has a neighbor that is always a part of alignment set then the node, that neighbor is aligned to, must be a neighbor of any prospective match for this entity.

2.1.11 RiMOM-IM [11]: S.Chao et al. proposed a novel iterative framework for instance matching in 2016. The main idea behind the framework is to maximize the utilization of distinctive and available matching information to handle large scale instance matching tasks in an iterative way. It has proposed a new blocking method which uses predicate and their distinctive object features to select candidate instance pairs and unique instance set which effectively reduces the running time. For each candidate set, similarities over all aligned predicates with similarity over predicates and then through aggregation, final matching score of two instances is computed. For unique instance sets, it iteratively uses unique subject matching and one left object matching to generate aligned set until no new matching pairs are generated.

In the next section, the Ontology Merging methods proposed in the recent past have been reviewed.

2.2 Ontology merging methods

The process of creation of a new ontology from two or more existing ontologies belonging to same domain is known as ontology merging. For instance, say one ontology say O1 contains the information of cars in the context of brand and another ontology say O2 also explains information of car but in the context of price. By merging these two ontologies O1 and O2, coverage area of car information can be extended and can be further used for annotation.

A number of ontology merging methods have been proposed by various researchers out of which some of the prevalent ontology merging methods is discussed as below.

2.2.1 Chimaera [12]: McGuinnessis D.L. et al. developed *Chimaera* at Knowledge Systems Laboratory at Stanford University in 2000 to provide assistance to users for browsing, editing, merging and diagnosing of ontologies. It is built on top of the *Ontolingua* Distributed Collaborative Ontology Environment. The project started with keeping the goal is to develop a tool that can give substantial assistance for the task of merging knowledge bases produced by different users for different purposes with different assumptions and different vocabulary. Later, the goals of supporting testing and diagnosing ontologies arose as well. Chimaera merges two semantically identical terms from different ontologies so that they can be referred to by the same name in the resulting ontology. It identifies terms that are related via is-a, disjointness or instance relationships and provide support for introducing those relationships. Chimaera also supports the identification of the locations for editing and performing the edits. To assist the user, Chimaera generates name resolution lists that suggest terms that are candidates to be merged or to have taxonomic relationships not yet included in the merged ontology. Chimaera also generates a taxonomy resolution list where it suggests taxonomy areas that are candidates for reorganization. On the basis of these lists, user decides what should be done.

2.2.2 ATOM [13]: *ATOM* is an asymmetric merge approach that gives preference to the target taxonomy. In preliminary phases, it takes two taxonomies O_s, O_t and a match mapping between them, provided by the set of concept correspondence and attribute correspondence. Its goal is the generation of an integrated concept graph. The main contribution of this work is new target-driven algorithm that automatically integrates taxonomies. The base algorithm takes as input two taxonomies and an equivalence matching between concepts. The algorithm generates taxonomies that preserve all instances of the input taxonomies as well as the structure of the target taxonomy. In contrast to previous work of ATOM, it does not necessarily preserve all source concepts but aim at limiting the semantic overlap in the merged taxonomy for improved understandability. This is achieved by utilizing the input mapping and giving preference to the target taxonomy when the same concepts are differently organized in source and target.

2.2.3 SAMBO [14]: Lambrix P. and Tan H. proposed a system for Aligning and Merging Biomedical Ontologies in 2006. It is an alignment method for defining the relationship between terms in different ontologies and creating a new ontology containing the knowledge included in the source ontologies. The framework of SAMBO consists of two parts. The first part computes alignment suggestion. The second part interacts with the user to decide on the final alignments. The alignment algorithm receives as input two source ontologies. Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers. The suggestions are then presented to the user who accepts or rejects them. SAMBO contains five basic matchers: two terminological matchers, a structure-based matcher, a matcher based on domain knowledge and a learning matcher for terminological matching. It uses n-gram and edit distance and linguistic algorithm. Structural matchers are based on *is-a* and *part-of* hierarchies of ontologies. This algorithm checks if two concepts lies in the similar position with respect to is-a or part-of hierarchies relative to already aligned concepts in the two ontologies, then they are likely to be similar as well. Next strategy is to use domain knowledge. SAMBO matcher uses UMLSK search that uses the meta-thesaurus in the Unified Medical Language System. The similarity of two terms in the source ontologies is determined by their relationship in UMLs. The fifth matcher is learner matcher. It is based on the intuition that a similarity measure between concepts in different ontologies can be defined on the probability that documents about one concept are also about the other concept and vice-versa. SAMBO has used Naive Bayes classification algorithm. The user has given the choice to employ one or several matchers during the alignment process. The similarity values for pair of concepts can then be determined based on similarity values computed by one matcher or as a weighted sum of the similarity values computed by different matchers.

2.2.4 HCONE [15]: Vouros G. A., Kotis K. proposed Human-centered ontology engineering based method for merging heterogeneous ontologies in 2004. The goal of the approach is to validate the mapping and to find the minimum set of axioms for the new merged ontology. This approach is based on-

(a) capturing the intended informal interpretation of concepts by mapping them to wordnet senses using lexical semantic indexing and

(b) exploiting the formal semantics of concepts definition by means of description.

In HCONE, ontology concepts are being mapped to wordnet senses. Using this mapping, HCONE merge constructs from the intermediate ontology that includes- a *vocabulary* with the lexicalization of the specific senses of wordnet synsets corresponding to the ontologies concepts and *axioms* that are translated axioms of the original ontologies. Having specified the mappings to the hidden intermediate ontology, the translated ontologies are merged following some merge actions such as rename, merge and classify.

2.2.5 PROMPT [16]: This method was proposed by Noy N. and Musen M. in 2000 which is an ontology-merging and ontology-alignment algorithm. It takes two ontologies as input and guides the user to generate a merged ontology as an output. It creates an initial list of matches based on class names and then the user triggers an operation by either selecting one of PROMPT's suggestions from the list or by using an ontology-editing environment to specify the desired operation directly.

2.2.6 Automatic Ontology Merging by hierarchical clustering and inference mechanism[17]: This method is based on the combination of statistical aspects represented by hierarchical clustering techniques and the inference mechanism. It generates global ontology automatically by four steps:

1. It builds class of equivalent entities of different categories (concepts, properties, instance) by applying a hierarchical clustering algorithm.
2. It makes an inference on detected classes to find new axioms and solves synonymy and homonymy conflicts. It also generates a set of concept pairs from ontology hierarchies.
3. It merges different sets together and uses classes of synonyms and sets of concept pairs to solve semantic conflicts in the global set of concept pairs.
4. Finally, it transforms this set to a new hierarchy which represents the global ontology.

In the next section, some of the popular Ontology Integration approaches have been discussed.

2.3 Ontology integration tools

The process of creation of new ontology by combining existing ontologies belonging to different domains is known as ontology integration. For example, combining ontology A of music domain and ontology B of singer domain and forming ontology C will hold the knowledge of songs along with their singers information thereby expanding the coverage area by using existing knowledge available on the web. Below are presented some prevalent methods proposed by researchers in this area.

2.3.1 Generating an urban domain ontology through the merging of cross-domain lexical ontologies [18]:

Lacatsa et al. proposed a method to integrate multi-lingual thesaurus (AGROVOC, EUROVOC, GEMET, UNESCO, URBISOC thesaurus) in order to build a first draft of a domain ontology in urbanism. The goal is to extract concepts and semantic relations from terms and linguistic relations. This method merges the knowledge from different domains to obtain a better definition of the urban domain. The process is composed of several steps:

1. Initially, system takes as input a set of thesaurus of different knowledge area and transforms them in the same

format to avoid from format related issues that may arise during the merging process.

2. Once, thesauri get transformed in the common format, the next objective is to extract the concepts related to urbanism from the analyzed thesauri. For this, it uses linguistic similarity between the concepts for mapping. In the mapping process, every concept of every thesaurus is compared with every concept of the other treasures to find equivalence. Each set of mapped concepts is grouped into a cluster which is identified with the one of the URI of the original concepts.
3. The clusters generated in the previous step describe the urban terminology used in different knowledge area. Now, the next task is to build a relation between these clusters to generate a network of urban concepts that can be seen as an urban ontology.
4. For this, relations of the concepts contained in each cluster are used as a basis for the generation of the relations between clusters.
5. Finally, in order to facilitate the visualization and reusability of the generated output, it is transformed into XTM and OWL formats.

2.3.2 Using semantic web services to integrate data and processes from different web portals [19]:

In this paper, authors propose an integration system which combines domain ontologies and semantic web services to provide an integrated access to the information provided by different web portals. In order to provide this functionality, it provides a user interface that allows users to express their query using an ontology guided tool which assist user's to express their goals. The domain ontology is loaded through the protégé OWL API and its main concepts are used to form a simple menu where the user can choose the type of the objects they are looking for.

Through the query component, the system searches and selects the most appropriate web services by accessing their semantic description. The query component returns a list with the services that attends the user's need. The invoker in turn uses the grounding of the web services to invoke each service selected by the query component. It returns the result of each service execution to the core component. This component integrates the results and finally presents them to the user.

Relevant details should be given including experimental design and the technique (s) used along with appropriate statistical methods used clearly along with the year of experimentation (field and laboratory).

III. Discussion

The comparison of various ontology management methods is performed on various parameters like operation, input, output, knowledge source, concept matching methods and user interaction etc. The detailed comparison study is outlined in Table 1shown in Appendix-1.

IV. Conclusion and Future Scope

The ontology interoperability is a prominent issue in many application domains such as semantic query processing, data integration and data-warehousing. A number of methods such as ontology alignment, ontology mapping, ontology merging, ontology integration etc. have come to deal with the issues of heterogeneity and inconsistency among the ontologies of same or similar domains. Because of the wide usage of ontology interoperability techniques, there is a need to consolidate different techniques and tools have been proposed to handle ontology Alignment, ontology mapping and merging processes. In this paper, we have surveyed the literature of these techniques and described the different criteria and approaches adopted by these techniques. In this survey, several approaches to ontology mapping, ontology matching, ontology merging and ontology integration are also compared to identify the gaps and have better future research directions.

References

[1] P. Jain, P.Z. Yeh, K. Verma, R.G. Vasquez, M. Damova, P. Hitzler, Amit P. Sheth, *"Contextual Ontology Alignment of LOD with an Upper Ontology: A Case Study with Proton",* In Proceedings of ESWC, 2011.

[2] Y.R. Jean-Marya , E.P. Shironoshita and M.R. Kabuka, *"Ontology matching with semantic verification, Journal of Web Semantics: Science, Services and Agents on the World Wide Web"*, 7(3) (2009) 235-251.

[3] J Gracia, J Bernad, E Mena, *"Ontology Matching with CIDER: evaluation report for OAEI 2011"*,Proceeding of Ontology Matching Workshop (OM'11), at 10th International Semantic Web Conference (ISWC'11), Bonn (Germany)

[4] J. Li, J. Tang, Y. Li, Q. Luo, *" RiMOM: A Dynamic Multistrategy Ontology Alignment Framework"*, IEEE Transactions on Knowledge and Data Engineering 2009

[5] Sabine Massmann , Salvatore Raunich , David Aumüller , Patrick Arnold , Erhard Rahm, *"Evolution of the COMA match system"*, Proceedings of the 6th International Conference on Ontology Matching, p.49-60, October 24, 2011, Bonn, Germany.

[6] D. Ngo, Z. Bellahsene, *"YAM++: A multi-strategy based approach for Ontology matching task"*, In Proceedings of EKAW, 2012.

[7] Aroua Essayeha ,Mourad Abeda, *"Towards ontology matching based system through terminological, structural and semantic level"*, 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, Procedia Computer Science 60 (2015) 403 – 412

[8] Cheatham, M., Hitzler, P., *"String similarity metrics for ontology alignment".* In ISWC 2013, Part II. LNCS, vol. 8219, pp. 294–309. Springer, Heidelberg (2013)

[9] J. Guang Zheng, L. Fu, X. Ma, P. Fox, *"SEM+: tool for discovering concept mapping in Earth science related domain*" Earth Science Information, 2015.

[10] Walid Hassen, *"Medley results for OAEI 2012"*, Proceedings of the 7th International Conference on Ontology Matching, p.168-172, November 11, 2012, Boston, MA

[11] C. Shao, L. Hu, J. Li Z. Wang, T. Chung, J. Xia, ***"****RiMOM-IM: A Novel Iterative Framework for Instance Matching"*, Journal of computer science and technology, 2016.

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

[13] S. Raunich, E. Rahm, **"***Target-driven merging of taxonomies with ATOM* , Information Systems, 2014.

[14] P. Lambrix, H. Tan, *"SAMBO - A System for Aligning and Merging Biomedical Ontologies"*, Journal of Web Semantics, 2006.

[15] K. Kotis, G. A. Vouros, K. Stergiou, **"***Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach,* Journal of Web Semantics," 2006.

[16] N. Noy and M. Musen, *"The PROMPT Suite: Interactive tools for ontology merging and mapping"*, International Journal of Human-Computer Studies, 2003.

[17] Nora Maiz, Muhammad Fahad ,Omar Boussaid, Fadlia Bentayeb, *"Automatic Ontology Merging by Hierarchical Clustering and Inference Mechanisms"* , proceeding of IKNOW (2010),pp.81-93

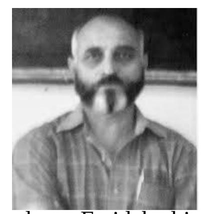
[18] Lacasta, J., Nogueras-Iso, J., Zarazaga-Soria, F.J., Muro-Medrano, P. *"Generating an urban domain ontology through the merging of cross-domain lexical ontologies"*, in Proceedings of Second Towntology Workshop "Ontologies for Urban Development: Conceptual Models for Practitioners," 17, 18 Oct 2007. Castello del Valentino, Turin, Italy (2007)

[19] Marília T. de Mello, Mara Abel Francisco García-Sánchez, *"Using Semantic Web Services to Integrate Data and Processes from Different Web Portals"*, International Workshop on Intelligent Web Based Tools (IWBT-07) in conjunction with 19th IEEE ICTAI-07

Authors Profile

Ms. Ranjna Jain received B.E. degree in Computer Science & Engineering with Hons.in 2005 and M.Tech. in 2010 Computers from Maharshi Dayanand University . Presently, she is working as Assistant Professor in Computer Engineering department in B.S.A. Institute of Technology & Management, Faridabad. She is also persuing Ph.D in Computer Engineering and her areas of interests are Semantic Web, Search Engines, Web Mining.

Dr. Neelam Duhan received B.Tech degree in Computer Science Engineering with Hons. from Kurukhetra University and M.Tech degree with hons. in computers from Maharshi Dayanand University in 2002 and 2005 respectively. Presently, she is working as Lecturer in Computer Engineering Department in YMCA University of Science & Technology, Faridabad. She obtained her Ph.D in Computer Engineering in 2011 and her areas of interests are Databases and Web Mining.



Prof. A. K. Sharma received his M.Tech. (Computer Science & Technology) with Hons. from University of Roorkee in the year 1989 and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi in the year 2000. From July 1992 to April 2002, he served as Assistant Professor and became Professor in Computer Engg. at YMCA University of Science & Technology, Faridabad in April 2002. He obtained his second Ph.D. in IT from IIIT & M, Gwalior in the year 2004. His research interests include Fuzzy Systems, Object Oriented Programming, Knowledge representation and Internet Technologies.

S.No.	Ontology Management Methods	Operation	Input	Output	Knowledge source	Concept matching methods	Language	User Interaction
1	BLOOMS+	Ontology Alignment	two ontologies	alignment between those ontologies	Wikipedia pages	Retrieves synset of each concept from wikipedia pages and uses them as context of that concept.	OWL	No
2	ASMOV	Ontology Matching	two ontologies	alignment between those ontologies	set of input alignment containing a set of predetermined correspondence.	Lexical similarity, structural similarity, restriction similarity and extensional similarity	OWL	No
3	CIDER	Ontology Alignment	two ontologies	alignment between those ontologies	wordnet	Uses artificial neural network to generate final similarity measure by combining semantic, lexical and structural similarity.	OWL	No
4	RiMOM	Ontology Alignment	two ontologies	alignment between those ontologies	None	Lexical and structural similarity.	OWL	No
5	COMA 3.0	Ontology Matching	two ontologies	alignment between those ontologies	None		OWL	No
6	YAM++	Ontology mapping	two ontologies	mapping between two ontologies	training data at elementry level	Machine learning based method at elementary level then structural and at last semantic matching.	OWL	Yes
7	SIMTSS	Ontology Alignment	two ontologies	XML file	None	terminological, structural and semantic	RDF, SKOS, turtle	Yes
8	MAPSSS	Ontology Alignment				syntactic, structural and semantic metrics		
9	SEM+	Ontology Alignment	two ontologies	alignment between those ontologies	None	the information entropy and weighted similarity model	OWL	No
10	MEDLEY	Ontology Alignment	two ontologies	alignment between those ontologies	external dictionary	lexical and structural methods	OWL	No

S.No.	**Ontology Management Methods**	**Operation**	**Input**	**Output**	**Knowledge source**	**Concept matching methods**	**Language**	**User Interaction**
11	RiMOM-IM	instance matching	two ontologies	alignment between those ontologies	None	Checks similarity over all aligned predicates for instance set, uses unique subject matching and one left object matching iteratively to generate aligned set.	OWL	No
12	Chimarea	Ontology merging	initially two knowledge bases, later on two ontologies	merged ontology	None	Identifies similarity via is-a, disjointness or instance relationships between two terms.	initially knowledge bases ,later on OWL	Yes
13	ATOM	Ontology merging			None			No
14	SAMBO	Ontology alignment and merging	two ontologies	merged ontology	None	A structure-based matcher, a matcher based on domain knowledge and a learning matcher for terminological matching	OWL	Yes
15	HCONE	Ontology merging	two ontologies	merged ontology	wordnet	Semantic matching	OWL	Yes
16	PROMPT	Ontology merging	two ontologies	merged ontology	None	Concept name string matching	OWL	Yes
17	Automatic ontology merging by hierarchical clustering and inference mechanism	Ontology merging	two ontologies	merged ontology	wordnet	Terminological and structural matching	OWL	No
18	Generating an urban domain ontology through the merging of cross domain lexical ontologies	Ontology Integration	Thesaurus of different knowledge area	merged ontology	None	Linguistic matching	different knowledge bases of different formats	No