

A Better Approach to Ontology Integration using Clustering Through Global Similarity Measure

Ashwin Makwana and Amit Ganatra

Department of Computer Engineering,

Charotar University of Science and Technology, CHARUSAT Campus, Changa, Gujarat, India

Article history

Received: 08-03-2018

Revised: 26-05-2018

Accepted: 20-06-2018

Corresponding Author:

Ashwin Makwana

Department of Computer
Engineering, Charotar

University of Science and
Technology, CHARUSAT

Campus, Changa, Gujarat, India
Email:

ashwinmakwana.ce@charusat.ac.in

Abstract: Knowledge representation is a crucial area of work in the intelligent system, especially in query answering system development. Ontology is used to represent shared knowledge of a particular domain for query answering system. Domain-specific ontology can be designed and developed by many groups and researchers, because of which there is heterogeneity in the knowledgebase. Ontology integration or merging is necessary in order to solve this problem of mixed knowledge. Finding similarity between two ontologies is crucial to achieve integration or merging of ontology. In this study, we present a method to generate a cluster of ontologies using global similarity measure of two ontologies. Ontology matching tools are used to find matched classes between two ontologies. Output of ontology matching tool is mapping between two ontologies and is used for generating clusters of ontology. We use Jaccard Similarity Index as a global similarity measure for clustering. Based on this measure, the popular k-means clustering algorithm is used to perform clustering of ontologies. Bins of ontologies are generated from each cluster. From each bin, all ontologies are finally merged into a single ontology, which helps us in reducing search effort in querying knowledge in query processing. The outcome of this research paper to provide better solution for merging ontology. Here, we use agriculture domain ontology corpus from the standard dataset for experimentation.

Keywords: Ontology, Knowledge Reuse, Knowledge Merging, Ontology Matching, Semantic Web, Clustering

Introduction

For developing semantic web-based applications (Hitzler *et al.*, 2011; Fensel *et al.*, 2005) ontologies are widely accepted as a means for providing a shared understanding of common domains. For such semantic web applications, there is a requirement of reasoner and query answering machines to process complex queries of the user on a particular domain.

Ontologies (Gruber, 1993) capture fixed domain knowledge and give an ordinary meaning of a domain, which can be reused and shared across systems and groups. Ontology (Chandrasekaran *et al.*, 1999) shares conceptualization of knowledge representation in different forms and formats and can be represented by many people or groups, either for the same domain or different domains. Searching or querying this knowledge from one or more ontologies requires a query answering system.

Ontology (Fensel *et al.*, 2005; Chandrasekaran *et al.*, 1999) provides a vocabulary of a specific domain and for a particular domain, many people and groups have generated their ontologies. Ontologies are designed and developed by different people or groups with different context and usage. This creates heterogeneity in knowledge represented by ontologies about the same domain. Also, many concepts are common to different domains, which creates heterogeneity in concept presentation. Thus, there is a demand for ontology matching (Euzenat and Shvaiko, 2007) to generate a total number of matching concepts among ontologies.

Multiple ontologies need to be merged on a standard platform. Random merging of ontologies poses a challenge to any query answering system. Systematic and logical merging of ontologies is required for solving problem of random merging followed by creating clusters of those ontologies, which results in a standard to

represent knowledge. Merging of ontologies applying ontology matching is a challenging area of semantic web research. In ontology matching, two different ontologies are matched based on several similarity measures (Choi *et al.*, 2010; Lesot *et al.*, 2009). Defining and quantifying these measures is a crucial problem of ontology matching where many research groups have contributed to the domain of semantic web in the past. This ontology matching and alignment can be very useful in merging knowledge presented in the ontology. Global similarity computation (Euzenat and Shvaiko, 2007) can be used to find similarities between two ontologies. Clusters of ontologies can be generated using this measure. The ontology is considered as a whole, though the final similarity value depends on all the entities of an ontology.

Before merging ontologies, they need to be matched with each other using standard state of the art ontology matching techniques (Shvaiko and Euzenat, 2013). There are different approaches to matching ontologies; the primary ontology matching system makes different lexicons by using lexical matchers. Different tokens are separated from the whole ontology and these separated tokens are matched with each other to get the similarity value. As per (Euzenat and Shvaiko, 2007; Shvaiko and Euzenat, 2013; 2008), this matching may be word based, string-based or structure based matching.

The problem of ontology mapping (Choi *et al.*, 2006) automatically among different ontologies is known as ontology matching. The primary purpose of ontology matching (Shvaiko and Euzenat, 2008) is to reduce heterogeneous information and knowledge presented by these ontologies. By ontology matching, syntactic, terminological and conceptual heterogeneity present between multiple ontologies can be reduced. Two ontologies expressed in different ontology language represent syntactic heterogeneity. When same entities referred in two different ontologies generate variation in names, it is called as a terminological heterogeneity. It is possible to reduce a few types of heterogeneity by matching ontologies.

In literature (Euzenat and Shvaiko, 2007; Hitzler *et al.*, 2011, Shvaiko and Euzenat, 2013; 2008) there are various application areas of ontology matching; alignments are discussed and elaborated. According to a survey, these applications are in the field of “ontology engineering, information integration, peer-to-peer information sharing, web service composition and query answering on the web.”

Out of the applications and usage mentioned above, there is a scope for improvement in the allied area of ontology engineering and information integration. Where specifically in ontology engineering: ontology editing, ontology import, ontology evolution and ontology versioning are application areas. While in information

integration: Schema integration, data integration, data warehousing and catalog integration are areas of focus.

These applications motivate to contribute to the field of ontology integration and query answering, especially in searching and querying on multiple ontologies of the same domain. Challenges to perform heterogeneous knowledge integration and merging motivate in developing a technique for ontology clustering based on ontology matching results. Local ontologies store information and data in their respective formats. To solve the problem of fetching knowledge from multiple local ontologies, a global ontology that can provide uniform query interface is required. Knowledge merging using ontology reuse - ontology reuse is research problem in the ontology field - can be categorized into two different processes (Pinto *et al.*, 1999) - merging and integration.

Domain-specific knowledge is present in multiple ontologies and it is required to merge various heterogeneous ontologies into one in order to obtain complete knowledge. For this purpose, use of ontology clustering, merging and matching is essential. Since it is not advisable to make a random selection of ontology from the corpus for merging, the ontologies should be merged based on similarity. Hence, clustering should be done from a pool of ontologies present in the corpus. Clusters should be based on similarities between ontologies, which can be found through element level similarity of two ontologies calculated using ontology matching. Ontology matching can be performed using different approaches, like lexical, string, word, semantic and structure (Euzenat and Shvaiko, 2007; Otero-Cerdeira *et al.*, 2015a). Out of the many approaches, it is advisable to select and use open source tools and techniques that give better precision, recall and F-measure. Out of available standard techniques and tools, Agreement Marker Light (AML) tool (Faria *et al.*, 2013a; 2013b) is promising. This tool can be used to find global similarities of two ontologies, which is then used for creating clusters of ontology.

Researchers for ontology matching and alignment have developed many techniques to merge knowledge present in ontology. Using these techniques, relevant ontologies to merge with similar or nearer ontology present in the corpus can be identified. For this purpose, there is need to develop techniques which can create clusters of similar ontology from a corpus of ontologies and merge them into single ontology, cluster wise.

The work contribution of the paper is divided into two parts. The first contribution of this paper is to develop a technique to determine the degree of global similarity between two ontologies. Here, similarity measure considers ontology as a whole instead of element level similarities. The second contribution is a technique to merge real-world ontologies; clustering algorithm is used to create bins of ontologies and

ontologies in each bin are subsequently merged into single global ontology.

Background and Related Work

It is often difficult to find single ontology consisting of all the relevant information and knowledge for a particular domain, especially for query answering; many people (The OBO Foundry, NCBO BioPortal and AgroPortal) have developed different ontologies in same domain or sub-domain. It is also possible that one person or group has developed different ontologies for the same domain or sub-domain. Searching information and knowledge in these multiple ontologies is a strenuous and time-consuming task for query answering. To solve this problem different research have contributed in past using various techniques of knowledge clustering, ontology alignment, ontology matching, ontology mapping and ontology merging. These areas motivate us to contribute in field of knowledge or ontology clustering for better query answering system. Here, in this section we briefly describe background and related work done in field.

XML Document Clustering

XML Document Clustering method, XClust (Lee *et al.*, 2002) presents works for clustering of XML Schemas for integration (Guerrini *et al.*, 2006), in which an approach to find similarities between Document Type Definition (DTD) and generate clusters of DTDs is presented. DTDs are grouped into clusters by using DTD similarity matrix in hierarchical clustering. Clusters are formed at different cut-off values that tend to be together from the same application domain. Clustering facilitates integration and merging process to produce newly integrated schema. Union of all the elements in DTDs is done in integration, which avoids loss of information. The more compactly the DTD is integrated, better is the result of integration process. Integration is used to retain only the common DTD elements in the integrated schema. Related work on similarity measures for clustering of XML documents is discussed in (Torres *et al.*, 2009) which represents various similarity measures of XML documents containing the annotation to provide similar resources on the web.

Clustering Algorithms

The process of dividing data points into similar classes or clusters is known as clustering. The objective of clustering is to determine the intrinsic group in a set of unlabeled data.

A cluster is a collection of objects with similarity between clusters and dissimilarity between objects into another cluster. An algorithm for clustering analyses natural groups of data based on similarity. There are several clustering algorithms like Repeated Bisection,

Direct, Agglomerative, SOM, Graph- gCLUTO tool and K-means, K-medoids- Matlab fuzzy clustering and data analysis toolbox (Bennett and Christiane, 2006).

Ontology Alignment and Ontology Mapping

According to surveys and research done by (Shvaiko and Euzenat, 2008) "Ontology matching is a solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities of ontologies." An ontology alignment (Bennett and Christiane, 2006) is the expression of relations between different ontologies. The set of mappings between two ontologies is called an alignment.

Different matching algorithms are used for ontology matching described in (Euzenat and Shvaiko, 2007). They are called as matchers. They assign a numerical value to each mapping. This value represents the similarity between terms. These matchers also include element level and structural level.

Ontology Matching

Numerous effective matching systems have been developed in the past decade; some of the famous matchers are described here. Ontology matching is used to solve the problem of semantic heterogeneity and is often achieved either manually or by using semiautomatic tools.

Several research groups have developed many systems and matchers for ontology matching: Graphical user interface supported matching systems like SAMBO, DSSim and Agreement Maker; generic matchers like Falcon, RiMOM, Anchor-Flood; and application domain specific matchers like SAMBO, ASMOV. Matching systems like Falcon, DSSim or Anchor-Flood are developed by strategies like ontology partitioning and anchor-based strategies. We studied few of ontology matching system in detail viz-a-viz Agreement Maker, Agreement Maker Light, LogMap, AROMA, CIDER, Lily, RiMOM, TaxoMap, YAM++.

Agreement Maker, developed by research group (Faria *et al.*, 2013b), has semiautomatic matching with good GUI, flexible architecture and user involvement in the matching process. Agreement Maker Light (AML), an enhanced version of Agreement Maker, is an automated ontology matching system that is extensive and efficient. AML has been participating in all OAEI tracks over the past few years, including the year 2017 and has been proving its performance as one of the best ontology matching systems in almost all tracks and tasks. Ruiz and Grau have developed LogMap, which uses reasoning and logic based semantics for better alignments. LogMap is a scalable and logic-based ontology matching system, which has been participating in OAEI since the past seven years in all tracks, delivering top performance. However, since the system

uses similarities between vocabularies for ontology matching, it performs poorly if ontology is lexically disparate or missing lexical information.

An extensive survey of the current state of the art ontology matching approaches and the application of such approaches in real-life has been conducted recently (Otero-Cerdeira *et al.*, 2015b). The results of the survey suggest that though majority of researchers who develop ontology matching approaches have done theoretical work, very few practical, real-life applications have been developed. In a survey (Daskalaki *et al.*, 2016) to present the benchmarking techniques, for instance, matching for Linked Data by discussing its principles, dimensions, characteristics and providing a survey of benchmarks and generator of the benchmark. They consider the presented benchmarks from the standpoint of the systems to identify the appropriate benchmark for a given setting. The system for matching of heterogeneous ontologies proposed by (Essayeh and Abed, 2015) is an automatic one that uses different techniques to find similarity between entities of ontology. The Similarity Flooding algorithm is adopted to study the internal structures of ontology, the result obtained from which is used as the global matrix. Hungarian algorithm is used to select alignment that is most appropriate and best using all the measures. Mecca *et al.* (2015) have done work on mapping process of ontology. They have developed an algorithm that translates and automatically rewrites a mapping from the source schema to the target ontology and also from the source ontology to the target databases using equivalence mapping. They have used non-recursive Datalog rules with negation. The issue of mapping information within sight of ontology-based depictions of the source and target information sources is considered in this study. A paper by (Forsati and Shamsfard, 2016), proposes a productive technique of ontology mapping, named as Harmony Search based Ontology Mapping (HSOMap), that successfully finds a close ideal mapping for two information ontologies. This approach uses various rating functions, defined as base matchers to find the similarity of ontology entities. Each base matcher catches the closeness between substances from an alternate point of view and can use the accessible side data about the entities successfully. HSOMap algorithm performance is compared with other methods using benchmark datasets. Another paper (Xue and Liu, 2017) presents a technology called collaborative ontology matching that enables multiple users to collaborate with each other to help the automatic tool for high-quality matching quickly and efficiently. This paper proposes a Compact Interactive Memetic Algorithm (CIMA) based collaborative ontology matching technology to solve challenges of shared ontology matching. It introduces a CIMA based community ontology matching innovation, which can

reduce the user's workload in matching process and increases the validation value of the user. This proposal can reduce user's workload by adaptively determining the time of getting users involved and the limited candidate correspondences presented for users. The work carried out by (Cerón-Figueroa *et al.*, 2017) portrays another model of classification of patterns to adjust instances from various ontologies, as an e-learning educative substance in an education domain. The first model introduced has been approved through trials, using OAEI- 2014 initiative. The second model that is presented is for ontology matching more than two educative substance archives to enhance the homogenous assets of e-learning consequently.

Recent developments in the field of the ontology matching system have made a significant impact on performance. Saruladha and Ranjini (2016) a reasoning based ontology matching system named (COGOM) has been presented, which is based on concepts that combine the structural similarity degree, attribute similarity degree and semantic conception degree. This system is adaptive as it is a reasoning based expression of knowledge. OAEI 2015 Datasets are used and ontology matching system is evaluated through the use of precision and recall metrics, thus improving its overall effectiveness. A group of researchers working on YAM++ (Ngo and Bellahsene, 2016) offered a better elementary matcher and framework. YAM++ version presented here is scalable and provides large-scale ontology matching. The technique of YAM++ has been proposed based on Graph Matching, Machine Learning and Information Retrieval. The latest version of YAM++ obtained great matching results in comparison to OAEI datasets. YAM++ is a matcher producing a good result that uses several algorithms for matching, consolidating the algorithms to match ontologies. This matcher provides self-configurable and flexible user preferences by the customized matching approach. YAM++ has recently been extended as YAM-BIO dedicatedly used for biomedical ontology matching using existing mappings as background knowledge.

Gulić *et al.* (2016) CroMatcher is an ontology matching framework that conveys different developments to the automated weight estimation process which is used here. They displayed another technique that can create the last balanced arrangement of ontology structures and is a vital change over other non-iterative strategies. In this study, we analyze the arrangement delivered by matchers and underline the matchers whose arrangement is particular and one of a kind.

Ontology Merging

Ontology integration process and methodology are described by (Pinto and Martins, 2001) as "a direct consequence of its generality. One of the advantages of

this integration methodology is the fact that it can be used with different methodologies to build ontologies from scratch.”

A new approach show for merging ontologies by research group (Mahfoudh *et al.*, 2014) utilizing typed graph grammars. Simple Push Out (SPO) and another ontology merging algorithm Graph Rewriting for Ontologies Merge (GROM) are the techniques utilized as a part of this approach. GROM is another apparatus that is actualized here and created a worldwide ontology from given two ontologies and their mapping in a planned manner. A paper by (Mahfoudh *et al.*, 2013) introduced the utilization of the graph grammars to approve and apply the ontology changes. The Algebraic Graph Grammar (AGG) apparatus is utilized to show system made out of various diagram-reworking rules and to systematize the forward and backward procedure of change of the ontologies to diagrams. They produced two projects - *OWLToGraph* and *GraphToOWL*.

Research for FCA (Fu, 2016) gives a proper and semi-automated approach for ontology development in light of Formal Concept Analysis (FCA). Its motivation is to incorporate information that shows inferred and uncertain data. The technique depicted in this study can help consolidate information from different sources and bolster the advancement of ontologies that backs the fundamental learning structure of the area. They did a contextual investigation on a few datasets and their outcome demonstrates that this strategy offers a viable component to coordinate information and address the requirements of the business. By using semantic, name and measurable based strategies (Maree *et al.*, 2015) have introduced a completely automated system for merging domain-specific ontologies. They grouped this methodology into three classes further: Single-technique based methodologies, different procedure based methodologies and methodologies that exploit external semantic assets. The semantics-based procedure to blend heterogeneous ontologies by finding semantic relations is connected by a framework. They additionally utilize a coupled measurable and semantic strategy to build up other semantic relations between missing ideas and ideas in the combined ontology. They had to utilize a few publically accessible datasets to achieve this.

Various ontology merging approaches have been proposed to address utilized procedures in recognizing semantic correspondences. Fahad *et al.* (2011) have exhibited a system of naturally recognizing semantic irregularities in the early stages of ontology merging. In this way, the ontology is free from 'common class/occurrence between disjoint classes mistake,' 'excess of disjoint relations' 'repetition of subclass/sub property relations,' 'circulatory error in class/property

progressive system,' and different kinds of 'semantic irregularity' errors. The procedure of DKP-AOM framework is introduced here utilizing similar word, phonetic and axiomatic coordinating. It fortifies a bigger pool of knowledge and data to be consolidated to ease new dependable correspondence and faculties, piece and conceptualization errors between heterogeneous ontologies. Researchers (Raunich and Rahm, 2014) proposed the ATOM approach for automatically merging a source taxonomy into a target taxonomy. The approach is target-driven, i.e., they consolidate a source scientific classification into the objective scientific classification and protect the objective ontology, as much as could reasonably be expected. The proposed calculations have straight intricacy for various leveled scientific categorizations. The ATOM approach could be efficiently connected to substantial genuine scientific categorizations from various areas.

The Proposed System for Knowledge Integration

In this section, we propose a new approach to perform merging and integration of ontology knowledge using global similarity measure derived using ontology matching processes.

The Process Flow for Knowledge Integration in Ontologies

Figure 1 is a schematic flow diagram depicting the main steps of the proposed system. For knowledge integration, a step-by-step process integrating knowledge presented in different ontologies is required. For this, following steps are carried out practically in an experiment on a specific domain of ontology cluster.

The first step is to identify global similarity measure for matching two ontologies as a whole. The second step is selecting an ontology matching system to evaluate the matching process and an appropriate open source tool to improve any one method. The third step is modifying this tool for finding ontology class, properties and individuals in source and target ontologies. The fourth step is to identify a domain-specific ontology corpus and apply ontology matching tool to match ontologies with each other. It is necessary to write an automated script to run the tool and find global similarity measure between this pair of ontologies. The fifth step is creating a cluster of ontologies by applying any standard clustering algorithm. The sixth step is generating bins of ontology from these clusters. Final step is to integrate ontologies inside these bins and merge this knowledge in a single ontology. Check accuracy and efficiency of merge ontology Vs. corpus of a different ontology using the benchmark of query answering in SPARQL (SPARQL Query Language).

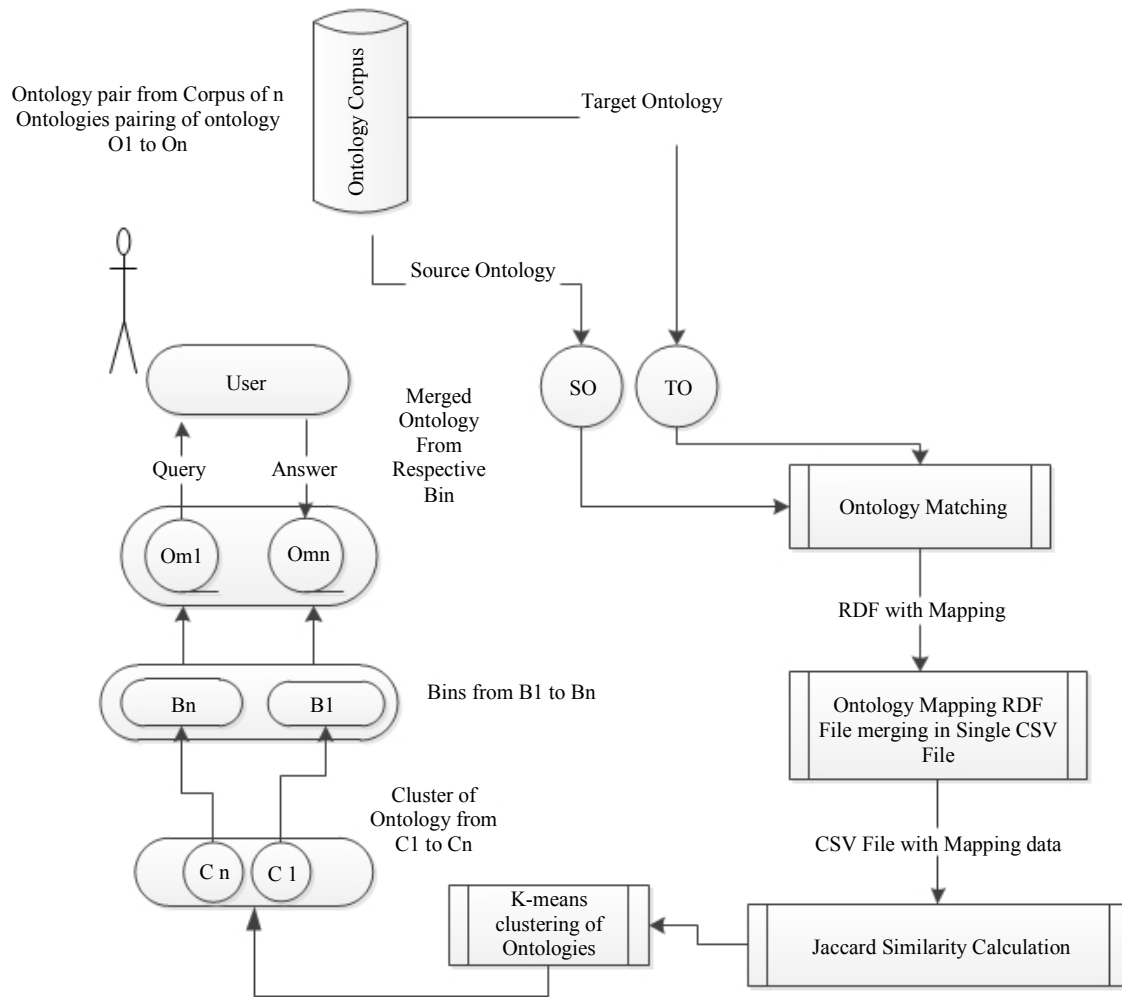


Fig. 1: System flow diagram

Global Similarity Measure- Jaccard Similarity Index

For matching two ontologies, the global similarity index like Jaccard similarity index (Choi *et al.*, 2010; Lesot *et al.*, 2009) is used, which is calculated by (number of similar objects) divided by (the total number of objects minus number of similar objects):

$$JS(X, Y) = \frac{(x * y)}{|x| + |y| - x * y} \quad (1)$$

To calculate Jaccard similarity index between two ontologies, we used number of classes, properties and individual of source ontology and target ontology. As shown in equation -1, X and Y will become O_1 and O_2 , which are the source and target ontologies respectively. Furthermore, $x * y$ becomes a number of common (similar) classes, properties and individuals between O_1

and O_2 , which are calculated using standard ontology matching techniques. We present similar (matched) mapping using $(o_1 * o_2)$.

$|x|$ and $|y|$ is the total number of classes, properties and individuals in O_1 and O_2 respectively, which can be identified here with $|o_1|$ and $|o_2|$ respectively. Hence, Equation 1 can be rewritten in ontology context as given in Equation 2:

$$JS(O_1, O_2) = \frac{(o_1 * o_2)}{|o_1| + |o_2| - (o_1 * o_2)} \quad (2)$$

Pseudo Code of Proposed System

Merge_Knowledge (OC [O_1 --- O_N])

Input: Ontology Corpus- OC (domain specific ontology corpus of O_1, \dots, O_N and Ontology features (#class, #properties, #individuals) of unique pair source and target ontologies.

Output: Integrated Ontologies- O_{IM}

1. For each i & $j=1$ to N
 $OM = AML_OM(<O_i, O_j>)$
2. $OM = \{Cs, PsIs, Ct, Pt, It, Cm, Pm, Im\}$
3. For each $k=1$ to P
4.
$$JS_k(O_i, O_j) = \frac{(Cm + Pm + Im)}{(Cs + Ps + Is + Ct + Pt + It) - (Cm + Pm + Im)}$$
5. $C_m = \text{Generate_cluster_Kmeans}(<O_i, O_j, JS_k>)$
6. End For
7. End For
8. For each $x = 1$ to m
9. $B[x] = C[x]$
 $B_m = \{O_{im} | (O_{iml}, O_{jmr}) \text{ is unique pair, } l = 1 \text{ to } L, r = 1 \text{ to } R\}$ Where, $|B_m| \leq L+R < N$
10. Return $O_{IM} = \text{Integrate_onto}(O_{im})$
11. End For

$AML_OM(<O_i, O_j>)$

Input: O_i, O_j

Output: $OM = \{Cs, PsIs, Ct, Pt, It, Cm, Pm, Im\}$

1. $S = O_i, T = O_j$
2. $\{Cs, PsIs\} = \text{Parse}(S)$
3. $\{Ct, Pt, It\} = \text{Parse}(T)$
4. $\text{Mapping} = \text{Call_Lexical_Matcher}(S, T)$
5. $\text{Mapping} = \text{Call_Mediating_Matcher}(S, T)$
6. $\text{Mapping} = \text{Call_Word_Matcher}(S, T)$
7. $\text{Mapping} = \{Cm, Pm, Im\}$
8. Return $OM = \{Cs, PsIs, Ct, Pt, It, Cm, Pm, Im\}$

Description of Pseudo Code

Generating clusters of ontologies requires ontology corpus, which is a pool of ontologies of a different domain; we defined this as the input of this pseudo code as $OC [O_1, O_N]$. In the initial stage, we collected various domain-specific ontologies used by OAEI portal and also from another source of ontology location as a dataset.

For finding Jaccard Similarity Index using Equation 2, the existing ontology alignment tool Agreement Marker Light (AML) was applied. From AML tool, we got mappings between any two ontologies from the corpus. The work carried out on a number of ontologies (O_1 to O_n) was taken from corpus from the same domain in one particular set C to match each ontology O_i from C , with all other ontologies from $C-O_i$. For this maximum number, unique matching required to execute an iteration of ontology matching tool using combination formula is $P = {}_N C_2$, as we were required to do P number of the pairing of ontology at a time from given Corpus C .

For finding Jaccard similarity index, a few numerical values were to be found from source ontology: Number of classes of source ontology- C_s , number of properties of source ontology- P_s and number

of individuals of source ontology- I_s . We were required to find the summation of all these three numbers of source ontology, i.e., $C_s + P_s + I_s$. Also, the numbers considered from target ontology are classes of target ontology- C_t , number of properties of target ontology- P_t and number of individual of target ontology- I_t . We were required to find the summation of all these three numbers of target ontology, i.e., $C_t + P_t + I_t$.

From the ontology matching tool, we found the total number of mappings, that is M , which is equal to the summation of the number of class map C_m , number of properties map P_m and number of individual map I_m -between source ontology and target ontology.

From this, we can rewrite Equation 2 as below:

$$JS_k(O_i, O_j) = \frac{(Cm + Pm + Im)}{(Cs + Ps + Is + Ct + Pt + It) - (Cm + Pm + Im)} \quad (3)$$

Where $0 < JS_k < 1$; $i \neq j (i, j = 1, 2, \dots, n)$, $k = 1, 2, \dots, P$

Equation 3 was used to find Jaccard similarity index between two ontologies: Source ontology O_1 and target ontology O_2 . The ideal value of JS_k is between 0 and 1. If there is no mapping between two ontologies, the value of JS_k is 0. If all class, properties and individual are similar and mapped perfectly, the value of JS_k is 1. For rest of the cases, JS_k is between 0 and 1 depending upon the number of mappings between two ontologies O_1 and O_2 . Above Equation 3 gives triplets $<O_i, O_j, S_k>$ where S_k represents k^{th} Jaccard Similarity of particular ontology O_i and O_j .

The Lloyd's algorithm (k-means algorithm), is used to solve the k-means clustering problem. K-means uses unsupervised learning methods and is the most straightforward and easy to implement algorithm and works well with large datasets. Its results are easy to interpret for clustering. K-means algorithm is fast and efficient regarding computational cost for one-dimensional data. The complexity of k-means is $O(n*k*i)$. Using Jaccard similarity index, we created a cluster of an ontology using K-means algorithm. Using orange tool, we provided a dataset of our ontology Jaccard similarity index values of various Ontology pairs from the corpus.

We generated clusters of ontologies based on Jaccard similarity index field using K-means algorithm. After generating M number of clusters, we identified corresponding SO and TO pairs from which we selected a corpus of ontology. Finally, we identified bins of ontologies created from these pairs of SO and TO from respective clusters.

These bins, B_1 to B_m correspond to clusters C_1 to C_m respectively. In these bins, we inserted SO and TO from respective clusters. $<O_{im}, O_{jm}, S_{km}>$ denotes which connect SO and TO in m^{th} Cluster $C_m (m = 1 \text{ to } M)$.

$$B_m = \left\{ O_{im} \mid (O_{iml}, O_{jmr}) - \text{unique pair}, \right. \\ \left. l = 1 \text{ to } L, r = 1 \text{ to } R \right\} \quad (4)$$

Where, $|B_m| \leq L + R < N$

L = Number of source ontologies and R = Number of target ontologies in above triplets. Ontologies presented in bins are likely to be similar and hence can be used for any research on query answering through ontology integration.

Performance Analysis of Ontology Clustering

Finally, a bin of similar ontologies are merged into a single ontology using Protégé tool. After merging two or more ontologies, we checked for ontology pairs having Jaccard Index of zero or less; the number of axioms, classes and elements in individual ontology is equal to merged new ontology. In case the Jaccard index is more than zero or higher than the number of axioms, classes and elements in individual ontology, total is higher than merged new ontology:

$$CO_1 + CO_2 = CO_m \text{ (If Jaccard Index} = 0) \\ CO_1 + CO_2 > CO_m \text{ (If Jaccard Index} > 0)$$

Here, CO_1 , CO_2 are the number of classes in ontology from the unique pair and CO_m is the number of classes in merged ontology from CO_1 and CO_2 .

The outcome of this analysis we proposed that the number of axioms and classes will reduced in integrated ontology compare one individual ontology. This will reduce search space and time for querying ontology for required knowledge, as loading time and response time will be reduce through this approach.

Experimentation and Results

In this section, we describe experimentation setup, implementation and results based on our approach.

Experimentation Setup

Agreement Maker Light (AML) is an open source tool for ontology matching. The Machine Configuration used to implement and test the proposed design is a 2.50 GHz Intel Core i5 processor, 16 GB RAM and Windows 10 OS. On this machine, we installed JVM 1.8, JDK 1.8 to use programming language JAVA 8 using IDE NetBeans-8.01, which is used to modify open source tool AML. We performed the experimentation of K-means clustering using ORANGE 3.4.1, in which we gave a CSV file as the input and got excel file as an output of Cluster details. We used Notepad++ v 6.9, Protégé tool for ontology editing and visualization. We also

used Java to create a script for executing AML tool and generating CSV file as an output.

Implementation Details

The Agreement Maker Light (AML) ontology matching system is an open source code available on GitHub (Agreement Light) developed by SOMER project. It can be implemented with NetBeans IDE tool and Java8. AML is one of the leading and best performing ontology matching tools used in ontology alignment contest track (Faria *et al.*, 2013b). Ontology Alignment Evaluation Initiative (OAEI) is the benchmark for ranking all ontology matching tools. AML is capable of handling large ontology and contains different matching algorithms that match in a customized or an automated manner. There are other matchers including lexical matcher, structural matcher, string matcher, word matcher, background matcher and property matcher. AML also has filters for obsolete, cardinality and coherence filtering.

Steps for Clustering of Ontologies

For the implementation of our technique, experimentation by some of the tools on a specific domain is required. Following steps are necessary to create clusters of ontologies:

1. Loading source and target ontologies from corpus
2. Matching ontology through custom setting
3. The mapping between two ontology using alignment
4. Clustering of ontologies using k-means

Figure 2 shows step 1 for selecting source and target ontology pair from corpus directory with owl, obo or rdf format. Figure 3 shows selecting ontology matcher and algorithm from various options. Figure 4 shows step 3 for ontology matching result i.e., alignment or mapping between ontology pairs. Figure 5 shows step 4 for k-means clustering process using ORANGE tool.

Dataset

OAEI has defined standard dataset for benchmarking ontology matching and alignment tools; a few domain-specific ontology corpora were selected from this dataset. We selected agriculture and bio domain initially. Experiments were implemented using a different corpus of ontology belonging to different domains. These ontologies downloaded from OBO-Foundry, Bio-Portal and Agro-Portal have ontologies belonging to different domains. Different sets of ontologies belonging to different domains were given as input to the widespread implementation of AML tool. The AML tool takes a pair of ontologies as input into the system and gives alignment values as the output.



Fig. 2: Loading source and target ontologies

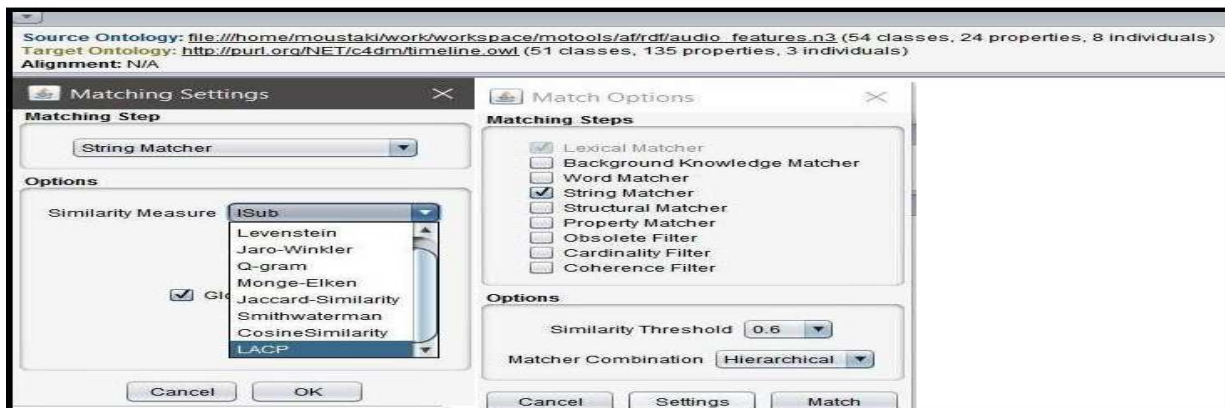


Fig. 3: Ontology matching selection

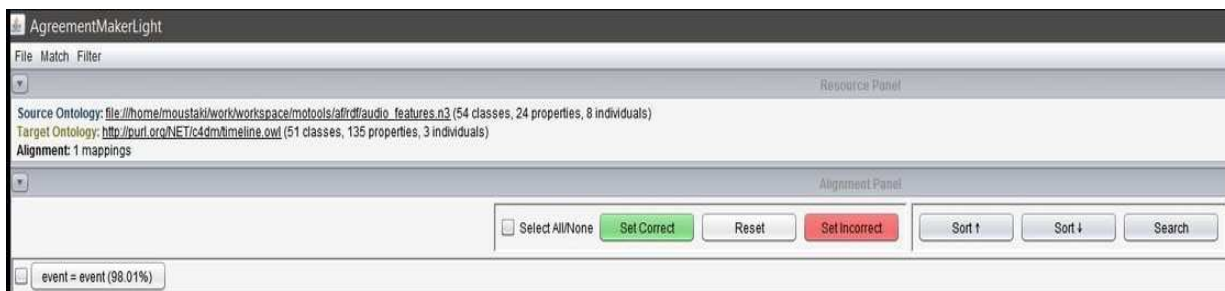


Fig. 4: Ontology Alignment Results

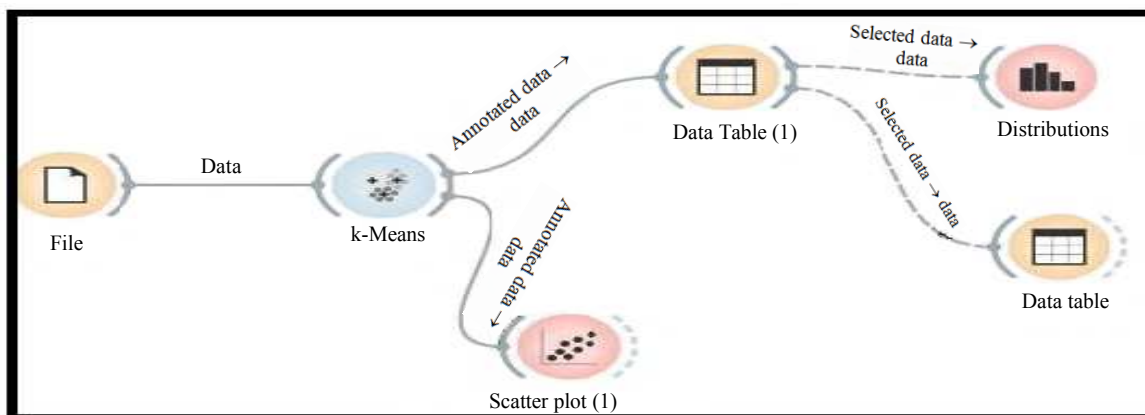


Fig. 5: Clustering using k-means

Results and Discussion

In this section, we discuss the result of our approach of clustering. Figure 6 is a screenshot of Orange tool graph for cluster created for ontology pair using k-means algorithm base on Jaccard similarity index. Here we have done fine tuning in cluster parameter of ORANGE tool for k-means algorithm. We can observe that similar type of ontology pair present in each cluster. For this experiments we have generated C1 to C6 number of clusters, which generate bins B1 to B6. From individual cluster we can form bins of unique ontologies.

Table 1 shows sample data for clusters of ontology. It is based on Jaccard similarity index field using K-means algorithm. The data presented in this table is a sample for experimentation performed. Table 1 describes 12 different columns, which represent consolidated data about work carried out. The second column, Source-onto and Target-onto are the source and target ontologies respectively taken in order pair from ontology corpus of agriculture domain. We took $N = 20$ sample ontologies from the corpus, here $P = {}_N C_2 = 380$. We can observe from this table that in one particular cluster of ontology there are similar type of ontology present. There is direct correlation of number of mapping with Jaccard Similarity index.

Table 2 is used to calculate Jaccard Similarity index between two specific ontologies from the corpus. Jaccard Similarity index works as a global similarity measure between two ontologies. Based on this, cluster of ontologies is created using the k-means algorithm. We generated bin for one individual cluster. Table 2 describes the names of all ontologies in a particular bin, number of ontologies pair per bin and number of different ontologies in the bin. From this table, we can notice density of each bin. We can observe that from number of ontology pair reduced to unique ontology which shrink and minimize search space for query.

Table 3 is used to represent the result of ontology integration statistics. Table 3 describes data for Ontology 1 # of Classes, Ontology 2 # of Classes, Common Classes in both ontologies, Total Classes in both ontologies, Jaccard Similarity Index and Integrated Ontology #of Classes. From this table, we can observe that number of similar classes is proportional to Jaccard similarity index. We can conclude from this experimental result that number of unique knowledge present in each bin is reducing search space for querying knowledge compare to individual ontology. By experimentation from approach we can observe that reducing search effort in querying knowledge in query processing by 68% and merging ontology by 30%.

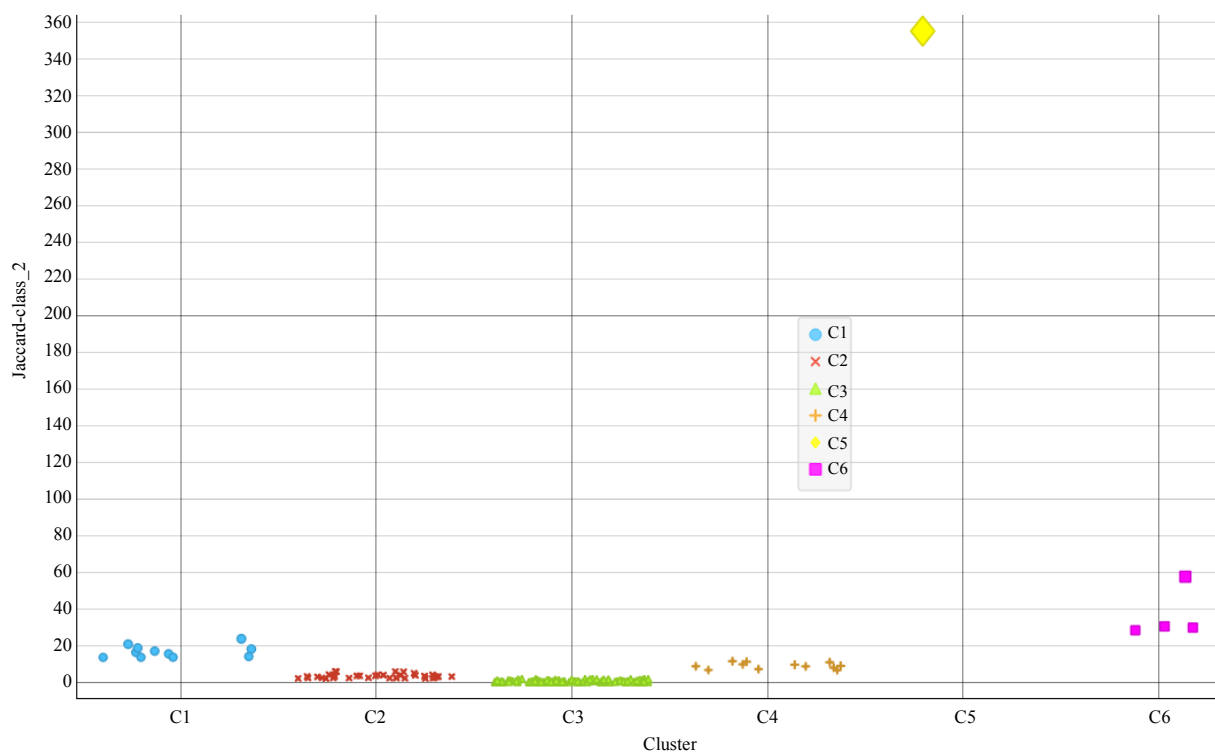


Fig. 6: Clustering Graphs with Jaccard Similarity Index

Table 1: Sample Experimentation data for Calculating Jaccard Similarity Index and Clustering

Cluster	Source-Onto	Target-Onto	Cs	Ps	Is	Ct	Pt	It	Mapping	Jaccard similarity index
C1	aeo	cdao	56	36	0	143	78	8	3	0.009433962
C1	aeo	sio	56	36	0	1522	208	0	21	0.011660189
C1	atol_ontology	eol_ontology	2096	0	4192	624	0	1248	48	0.005917160
C1	envo	foodon	6444	125	0	10014	29	0	332	0.020393120
C1	envo	sio	6444	125	0	1522	208	0	109	0.013308913
C1	foodon	hSC9z#	10014	29	0	1173	5	1401	257	0.020784472
C2	atol_ontology	cl	2096	0	4192	6538	74	0	31	0.002408890
C2	atol_ontology	doid	2096	0	4192	37973	27	0	86	0.001945613
C2	atol_ontology	envo	2096	0	4192	6444	125	0	19	0.001479981
C2	atol_ontology	sio	2096	0	4192	1522	208	0	7	0.000873799
C2	atol_ontology	to	2096	0	4192	4455	48	0	21	0.001949861
C2	cl	foodon	6538	74	0	10014	29	0	48	0.002890347
C2	cl	hSC9z#	6538	74	0	1173	5	1401	23	0.002508726
C2	cl	pato	6538	74	0	2603	23	0	27	0.002931278
C2	cl	po	6538	74	0	1972	13	0	31	0.003618959
C2	cl	sio	6538	74	0	1522	208	0	46	0.005544841
C2	cl	to	6538	74	0	4455	48	0	43	0.003883671
C2	doid	pato	37973	27	0	2603	23	0	88	0.002170803
C2	Durum_Wheat#	sio	127	4	137	1522	208	0	5	0.002508781
C2	envo	eo	6444	125	0	562	1	0	33	0.004648542
C2	envo	eol_ontology	6444	125	0	624	0	1248	21	0.002494062
C2	envo	pato	6444	125	0	2603	23	0	35	0.003820961
C3	cl	doid	6538	74	0	37973	27	0	26	0.000583143
C3	cl	Durum_Wheat#	6538	74	0	127	4	137	1	0.000145370
C3	cl	envo	6538	74	0	6444	125	0	18	0.001367469
C3	cl	vario	6538	74	0	433	21	0	7	0.000991642
C3	doid	Durum_Wheat#	37973	27	0	127	4	137	1	2.61E-05
C3	doid	envo	37973	27	0	6444	125	0	49	0.001100629
C3	doid	eo	37973	27	0	562	1	0	18	0.000466987
C3	doid	eol_ontology	37973	27	0	624	0	1248	2	5.02E-05
C3	doid	foodon	37973	27	0	10014	29	0	45	0.000937539
C3	doid	hSC9z#	37973	27	0	1173	5	1401	24	0.000591789
C4	atol_ontology	pato	2096	0	4192	2603	23	0	30	0.003376857
C4	bfo	sio	35	0	0	1522	208	0	12	0.006845408
C4	cdao	vario	143	78	8	433	21	0	5	0.007374631
C4	envo	hSC9z#	6444	125	0	1173	5	1401	53	0.005827378
C4	eol_ontology	eo	624	0	1248	562	1	0	10	0.004123711
C4	ms2o	sio	38	60	0	1522	208	0	17	0.009387079
C4	SLN_Ontology	bfo	83	236	657	35	0	0	1	0.000990099
C4	SLN_Ontology	Durum_Wheat#	83	236	657	127	4	137	2	0.001610306
C4	SLN_Ontology	eo	83	236	657	562	1	0	6	0.003913894
C4	SLN_Ontology	sio	83	236	657	1522	208	0	18	0.006696429
C5	ms2o	ontofp	38	60	0	46	91	13	22	0.097345133
C6	aeo	ontofp	56	36	0	46	91	13	3	0.012552301
C6	aeo	SLN_Ontology	56	36	0	83	236	657	4	0.003759398
C6	bfo	ms2o	35	0	0	38	60	0	2	0.015267176
C6	SLN_Ontology	ontofp	83	236	657	46	91	13	7	0.006255585

Table 2: Summary of Bins generated from Clusters

Bin	Cluster	Ontologies	Number of ontology pairs	Unique ontologies in Bin
B1	C1	aeo,cdao,atol_ontology,eol_ontology,ontofp,sio,Durum_Wheat#,ms2o,envo,foodon,hSC9z#,SLN_Ontology	11	12
B2	C2	aeo,Durum_Wheat#,pato,to,vario,atol_ontology,cl,atol_ontology,doid,sio,bfo,cdao,cdao,ms2o,foodon,po,doid,eo,po,envo,so,eol_ontology,eo,hSC9z#,SLN_Ontology	38	24
B3	C3	cl,doid,Durum_Wheat#,envo,vario,eo,eol_ontology,foodon,hSC9z#	109	13
B4	C4	aeo,eo,ms2o,atol_ontology,pato,cdao,vario,envo,hSC9z#,eol_ontology,SLN_Ontology,bfo,Durum_Wheat#,sio	12	14
B5	C5	ms2o,ontofp	2	2
B6	C6	aeo,ontofp,SLN_Ontology,bfo,ms2o	4	5

Table 3: Sample Integrated Ontology results

Cluster	Source-Onto	Target-Onto	Ontology 1	Ontology 2	Common Class	Total Class	Jaccard similarity index	Integrated ontology # of Classes
			# of Classes Cs	# of Classes Ct				
C1	aeo	cdao	56	143	3	199	0.01530612	196
C1	aeo	sio	56	1522	21	1578	0.01348748	1557
C1	envo	foodon	6444	10014	332	16458	0.02058787	16126
C2	atol_ontology	cl	2096	6538	31	8634	0.00360339	8603
C2	cl	sio	6538	1522	46	8060	0.00573996	8014
C2	cl	to	6538	4455	43	10993	0.00392694	10950
C2	doid	pato	37973	2603	88	40576	0.00217348	40488
C2	envo	eo	6444	562	33	7006	0.00473254	6973
C3	cl	doid	6538	37973	26	44511	0.00058447	44485
C3	doid	envo	37973	6444	49	44417	0.0011044	44368
C4	atol_ontology	pato	2096	2603	30	4699	0.00642536	4669
C4	bfo	sio	35	1522	12	1557	0.00776699	1545
C4	cdao	vario	143	433	5	576	0.00875657	571
C4	envo	hSC9z#	6444	1173	53	7617	0.00700688	7564
C4	SLN_Ontology	sio	83	1522	18	1605	0.01134216	1587
C5	ms2o	ontofp	38	46	22	84	0.35483871	62
C6	aeo	ontofp	56	46	3	102	0.03030303	99
C6	aeo	SLN_Ontology	56	83	4	139	0.02962963	135

Conclusion

In this study, we presented a better approach for ontology integration. Several different ontologies can be merged into one from the same or different domains. Ontology integration from a corpus of ontologies of a particular domain can be done one by one, for which matching similarities between two ontologies is essential. We performed ontology clustering using ontology matching tools and merged knowledge shared by different ontologies using ontology integration. For clustering, as a similarity measure, we used global similarity measure Jaccard similarity index. Results illustrate how ontology clustering is performed and relevant similar ontologies are integrated into merged knowledge. This approach is help us in reducing search effort in querying knowledge in query processing by 68%. The outcome of this research paper to provide better solution for merged ontology by 30%. Here, we use agriculture domain ontology corpus from the standard dataset for experimentation.

Future Work

For future work, we propose an application of ontology clustering approach for multiple ontologies merging. To improve querying multi ontology search of SPARQL query for query answering system. We can validate this approach for multiple domains and compare with benchmarking parameters of ontology merging.

Acknowledgment

The Authors would like to thank management and staff of Charotar University of Science and Technology,

CHARUSAT, Changa, India for supporting this research and providing resources for same.

Author's Contributions

Ashwin Makwana: Participated in Literature survey and problem formulation, Design solution and algorithm, Data collection and analysis, Participated in all experiments, Contributed to the writing of the manuscript.

Amit Ganatra: Analysis of Problem area, Designed the research plan, Contribute in experimentation setup of orange tool and clustering algorithm, Contribute in verification and compiling manuscript.

Ethics

This article is an original research paper. There are no ethical issues that may arise after the publication of this manuscript.

References

- Bennett, B. and F. Christiane, 2006. Proceedings of the 4th international conference on Formal ontology in information systems.
- Cerón-Figueroa, S., I. López-Yáñez, W. Alhalabi, O. Camacho-Nieto and Y. Villuendas-Rey *et al.*, 2017. Instance-based ontology matching for e-learning material using an associative pattern classifier. *Comput. Human Behav.*, 69: 218-225.
 DOI: 10.1016/j.chb.2016.12.039
- Chandrasekaran, B., J.R. Josephson and V.R. Benjamins, 1999. What are ontologies and why do we need them? *IEEE Intell. Syst. Applic.*, 14: 20-26.
 DOI: 10.1109/5254.747902

- Choi, N., I. Song and H. Han, 2006. A survey on ontology mapping. *ACM Sigmod Record*, 35: 34-41.
DOI: 10.1145/1168092.1168097
- Choi, S., S. Cha and C.C. Tappert, 2010. A survey of binary similarity and distance measures. *J. Syst. Cybernet. Inform.*, 8: 43-48.
- Daskalaki, E., G. Flouris, I. Fundulaki and T. Saveta, 2016. Instance matching benchmarks in the era of linked data. *Web Semant.: Sci. Services Agents World Wide Web*, 39: 1-14.
DOI: 10.1016/j.websem.2016.06.002
- Essayeh, A. and M. Abed, 2015. Towards ontology matching based system through terminological, structural and semantic level. *Proc. Comput. Sci.*, 60: 403-412. DOI: 10.1016/j.procs.2015.08.154
- Euzenat, J. and P. Shvaiko, 2007. *Ontology matching*. Springer, Heidelberg.
- Fahad, M., N. Moalla and A. Bouras, 2011. Towards ensuring satisfiability of merged ontology. *Proc. Comput. Sci.*, 4: 2216-2225.
DOI: 10.1016/j.procs.2011.04.242
- Faria, D., C. Pesquita, E. Santos, M. Palmonari and I.F. Cruz *et al.*, 2013a. Agreement maker light results for OAEI 2013. *Proceedings of the 8th International Conference on Ontology Matching, (COM' 13)*, CEUR-WS.org, Sydney, Australia, pp: 101-108.
- Faria, D., C. Pesquita, E. Santos, M. Palmonari and I.F. Cruz *et al.*, 2013b. The agreement maker light ontology matching system. *Proceedings of the Confederated International Conferences on the Move to Meaningful Internet Systems*, Sept. 9-13, Springer, Graz, Austria, pp: 527-541.
DOI: 10.1007/978-3-642-41030-7_38
- Fensel, D., A.J. Hendler, H. Lieberman, W. Wahlster and Berners-Lee, 2005. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. 1st Edn., The MIT Press, ISBN-10: 026256212X, pp: 272.
- Forsati, R. and M. Shamsfard, 2016. Symbiosis of evolutionary and combinatorial ontology mapping approaches. *Inform. Sci.*, 342: 53-80.
DOI: 10.1016/j.ins.2016.01.025
- Fu, G., 2016. FCA based ontology development for data integration. *Inform. Process. Manage.*, 52: 765-782.
DOI: 10.1016/j.ipm.2016.02.003
- Gruber, T., 1993. What is an ontology.
- Guerrini, G., M. Mesiti and I. Sanz, 2006. An overview of similarity measures for clustering XML documents. *Web Data Management Practices: Emerging Techniques and Technologies*, Vakali, A. and G. Pallis (Eds.), Idea Group Publishing, Hershey, London, Melbourne, pp: 56-78.
- Gulić, M., B. Vrdoljak and M. Banek, 2016. CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment. *Web Semant.: Sci. Services Agents World Wide Web*, 41: 50-71.
DOI: 10.1016/j.websem.2016.09.001
- Hitzler, P., M. Krotzsch and S. Rudolph, 2011. *Foundations of semantic web technologies*. CRC Press.
- Lee, M.L., L.H. Yang, W. Hsu and X. Yang, 2002. XClust: Clustering XML schemas for effective integration. *Proceedings of the 11th International Conference on Information and Knowledge Management*, Nov. 04-09, ACM, McLean, Virginia, USA, pp: 292-299. DOI: 10.1145/584792.584841
- Lesot, M.J., M. Rifqi and H. Benhadda, 2009. Similarity measures for binary and numerical data: A survey. *Int. J. Knowl. Eng. Soft Data Paradigms*, 1: 63-84. DOI: 10.1504/IJKESDP.2009.021985
- Mahfoudh, M., G. Forestier, L. Thiry and M. Hassenforder, 2013. Consistent ontologies evolution using graph grammars. *Proceedings of the International Conference on Knowledge Science, Engineering and Management*, Aug. 10-12, Springer, Dalian, China, pp: 64-75.
DOI: 10.1007/978-3-642-39787-5_6
- Mahfoudh, M., L. Thiry, G. Forestier and M. Hassenforder, 2014. Algebraic graph transformations for merging ontologies. *Proceedings of the International Conference on Model and Data Engineering, (MDE' 14)*, Springer, Cham.
- Maree, M. and M. Belkhatir, 2015. Addressing semantic heterogeneity through multiple knowledge bases assisted merging of domain-specific ontologies. *Knowledge-Based Syst.*, 73: 199-211.
- Mecca, G., G. Rull, D. Santoro and E. Teniente, 2015. Ontology-based mappings. *Data Knowl. Eng.*, 98: 8-29. DOI: 10.1016/j.datak.2015.07.003
- Ngo, D.H. and Z. Bellahsene, 2016. Overview of YAM++—(not) yet another matcher for ontology alignment task. *Web Semant.: Sci. Services Agents World Wide Web*, 41: 30-49.
DOI: 10.1016/j.websem.2016.09.002
- Otero-Cerdeira, L., F.J. Rodriguez-Martinez and A. Gómez-Rodríguez, 2015a. Ontology matching: Current trends among practitioners. *Expert Syst. Applic.*, 42: 949-971.
- Otero-Cerdeira, L., F.J. Rodríguez-Martínez and A. Gómez-Rodríguez, 2015b. Ontology matching: A literature review. *Expert Syst. Applic.*, 42: 949-971.
DOI: 10.1016/j.eswa.2014.08.032
- Pinto, H.S. and J.P. Martins, 2001. A methodology for ontology integration. *Proceedings of the 1st International Conference on Knowledge Capture*, Oct. 22-23, ACM, Victoria, British Columbia, Canada, pp: 131-138. DOI: 10.1145/500737.500759

- Pinto, H.S., A. Gómez-Pérez and J.P. Martins, 1999. Some issues on ontology integration. Proceedings of the 16th International Joint Conference on Artificial Intelligence, Aug. 02-02, Stockholm, Sweden.
- Raunich, S. and E. Rahm, 2014. Target-driven merging of taxonomies with the atom. Inform. Syst., 42: 1-14. DOI: 10.1016/j.is.2013.11.001
- Saruladha, K. and S. Ranjini, 2016. COGOM: Cognitive theory-based ontology matching system. Proc. Comput. Sci., 85: 301-308. DOI: 10.1016/j.procs.2016.05.237
- Shvaiko, P. and J. Euzenat, 2008. Ten challenges for ontology matching. Proceedings of the Confederated International Conferences on the Move to Meaningful Internet Systems, Nov. 09-17, Springer, Monterrey, Mexico, pp: 1164-1182. DOI: 10.1007/978-3-540-88873-4_18
- Shvaiko, P. and J. Euzenat, 2013. Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng., 25: 158-176. DOI: 10.1109/TKDE.2011.253
- Torres, G.J., R.B. Basnet, A.H. Sung, S. Mukkamala and B.M. Ribeiro, 2009. A similarity measure for clustering and its applications. Int. J. Electrical Comput. Syst. Eng., 3: 164-170.
- Xue, X. and J. Liu, 2017. Collaborative ontology matching based on a compact, interactive evolutionary algorithm. Knowledge-Based Syst., 137: 94-103. DOI: 10.1016/j.knosys.2017.09.017