System for Parallel Heterogeneity Resolution (SPHeRe) results for OAEI 2013

Wajahat Ali Khan, Muhammad Bilal Amin, Asad Masood Khattak, Maqbool Hussain, and Sungyoung Lee

Department of Computer Engineering
Kyung Hee University
Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, Republic of Korea, 446-701
{wajahat.alikhan, mbilalamin, asad.masood, maqbool.hussain, sylee}@oslab.khu.ac.kr

Abstract. SPHeRe is an ontology matching system that utilizes cloud infrastructure for matching large scale ontologies and focus on alignment representation to be stored in the Mediation Bridge Ontology (MBO). MBO is the mediation ontology that stores all the alignments generated between the matched ontologies and represents it in a manner that provides maximum metadata information. SPHeRe is a new initiative therefore it only participates in the large biomedical ontologies track of the OAEI 2013 campaign. The objectives of SPHeRe system participation in OAEI is to shift focus of ontology matching community towards areas such as cloud utilization, effective mapping representation, and flexible and extendable design of the matching system.

1 Presentation of the system

Ontology mappings enables accessibility of information by aligning the resources in ontologies belonging to diverse organizations [3]. These also resolves semantic heterogeneities among data sources. Mainly two steps are required to overcome semantic heterogeneity: Matching resources to determine alignments and interpreting those alignments according to application requirements [5].

We have started developing SPHeRe system in 2013 and its an ongoing project. The objectives of SPHeRe system are performance [2], accuracy, mapping representation, and flexible and extendible design of the system.

1.1 State, purpose, general statement

SPHeRe system target a complete package of a system with main objectives as accuracy, mapping representation, and flexible and extendible system. Its precision is on the higher side in large biomedical ontologies track, that shows its potential of improving the accuracy. It is based on different algorithms such as String Matching Bridge, Synonym Bridge, Child Based Structural Bridge (CBSB), Property Based Structural Bridge (PBSB), and Label Bridge. We plan to include further bridge algorithms in next version of the proposed system by incorporating new matching techniques.

Parallelism has been overlooked by ontology matching systems. SPHeRe avails this opportunity and provides a solution by: (i) creating and caching serialized subsets of candidate ontologies with single-step parallel loading; (ii) lightweight matcher-based and redundancy-free subsets result in smaller memory footprints and faster load time; and (iii) implementing data parallelism based distribution over subsets of candidate ontologies by exploiting the multicore distributed hardware of cloud platform for parallel ontology matching and execution [2].

Mapping representation is another aspect of SPHeRe system which is not covered in this paper. We have followed OAEI alignment representation format, but we consider mapping representation as an important dimension to be worked by ontology matching research community. The more expressive the alignments should be, the easy its expert verification and the more will be confidence level in transformation process.

1.2 Specific techniques used

SPHeRe system is based on bridge algorithms run in the parallel execution environment to generate alignments to be stored in the MBO as shown in Fig. 1. Matcher Library components stores all the bridge algorithms to be run on the parallel execution environment represented by Parallel Matching Framework. Communication between these two components is regulated by SPHeRe Execution Control module that behaves as a controller. The alignments are stored in the MBO; generated by the bridge algorithms stored in Matcher Library that are run by the Parallel Matching Framework.

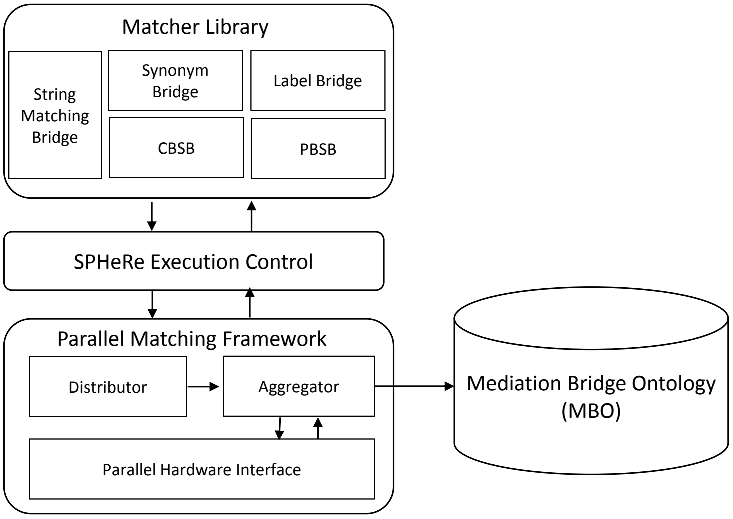


Fig. 1. SPHeRe System Working Model

String Matching Bridge provides matching results by finding similar concepts based on string matching techniques in the matching ontologies. Mainly the algorithm is based on applying edit distance technique [4] of string matching. For any two concepts C_i and

C_j of the ontologies O_i and O_j respectively, edit distance is applied to find matching value, $SimScore \longleftarrow C_i$. EditDistance ($C_j$). A threshold $Threshold$ value of n is set for matching in String Matching Bridge algorithm to limit the number of impure mappings.

Label Bridge uses the labels of the source and target concepts for matching. Initially, concept labels are normalized e.g. using stop word elimination, then list of the source concept labels are matched with list of the target concept labels. The source and target concepts label list $LabelList_i$ and $LabelList_j$ are matched using (($LabelList_i$ $\cap$ $LabelList_j$) $\neq \phi$). If any label in the lists matches, the source and target concepts are stored in the MBO as mappings.

Synonym Bridge is based on finding the similarity between concepts using wordnet [1]. The relationship is identified based on matching the synonyms of the concepts accessed using wordnet. Initially synonyms of source $List_l$:= C_i.GetSynonymWordnetList() and target $List_m$:= C_j.GetSynonymWordnetList() concepts are extracted using wordnet; where C_i and C_j are the source and target concepts respectively. The number of common synonyms $MatchedItems$ is found for calculating the matching value $SimScore$. If its value is less than the threshold then this alignment is discarded, otherwise stored in the MBO.

Child Based Structural Bridge (CBSB) bridge generates mappings between source and target ontologies based on matching children of the concepts. Initially, children of source C_i and target C_j concepts are accessed as lists $ChildList_i$ and $ChildList_j$ respectively. The number of common children in the lists is identified as $MatchChildren$. Finally the matching value $SimScore$ is calculated and compared with the threshold $Threshold$ that is assigned value n. The matching value is calculated using $SimScore$ $\longleftarrow$ $MatchedChildren$ / Average($ChildList_i$, $ChildList_j$). Property Based Structural Bridge (PBSB) uses String Matching Bridge techniques to match properties of source and target concepts for finding similar properties. This information is utilized as in CBSB for matching the source and target ontologies concepts based on their properties. These bridge algorithms are run on a parallel execution environment for better performance of the system.

Multiphase design of SPHeRe system is represented in Fig. 2(taken from [2]) that describes the parallelism inclusion in ontology matching process for better performance. The first phase of the system is ontology loading and management, in which the source and target ontologies are loaded in parallel by multithreaded ontology load interface (OLI). The main tasks of OLI includes; parallel loading of source and target ontologies, parsing for object model creation, and finally ontology model serialization and de-serialization. This is an important phase for data parallelism over multi-threaded execution into the second phase of distribution and matching [2].

Serialized subsets of source and target ontologies are loaded in parallel by multithreaded ontology distribution interface (ODI). ODI is responsible for task distribution of ontology matching over parallel threads (Matcher Threads). ODI currently implements size-based distribution scheme to assign partitions of candidate ontologies to be matched by matcher threads. In a single node, matcher threads correspond to the number of available cores for the running instance. In multi-nodes, each node performs its own parallel loading and internode control messages which are used to communicate

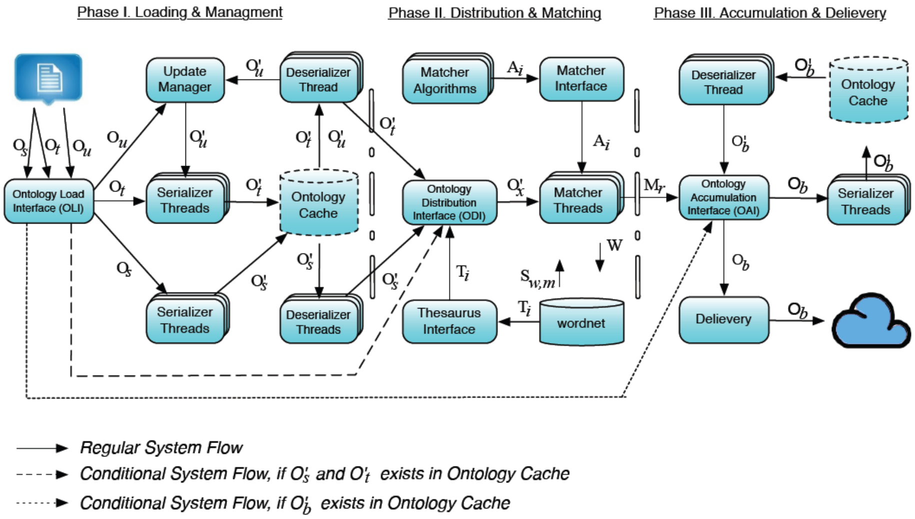


Fig. 2. Performance of SPHeRe System [2]

regarding the ontology distribution and matching algorithms. Matched results provided by matcher threads are submitted to accumulation and delivery phase for the MBO creation and delivery [2].

Ontology Aggregation Interface (OAI) accumulates matched results provided by matcher threads. OAI is responsible for MBO creation by combining matched results as mappings and delivering MBO via cloud storage platform. OAI provides a thread-safe mechanism for all matcher threads to submit their matched results. After the completion of all matched threads, OAI invokes MBO creation process which accumulates all the matched results in a single MBO instance [2]. In case of multi-node distribution, OAI also accumulates results from remote nodes after completion of their local matcher threads. This is a summary version of the performance oriented ontology matching process of SPHeRe system, extracted from [2], that provides a detailed version of the overall process.

1.3 Link to the system and parameters file

https://sites.google.com/a/bilalamin.com/sphere/results

1.4 Link to the set of provided alignments (in align format)

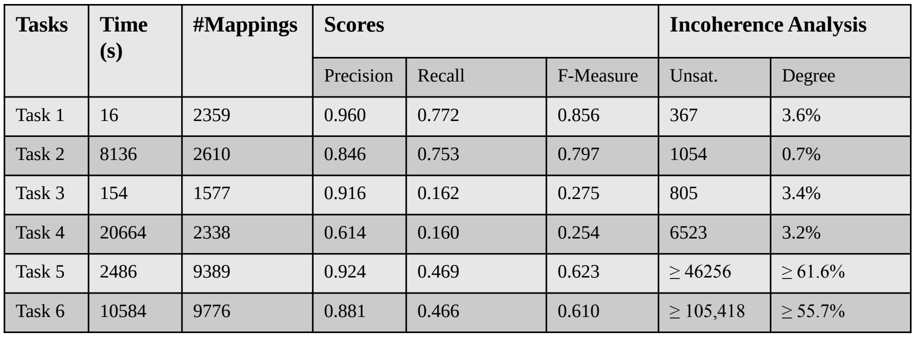
https://sites.google.com/a/bilalamin.com/sphere/results

2 Results

SPHeRe is deployed in multi-node configuration on virtual instances (VMs) over a tri-node private cloud equipped with commodity hardware. Each node is equipped with Intel(R) Core i7(R) CPU, 8GB memory with Xen Hypervisor. Jena API is utilized for its inferencing capabilities. As SPHeRe is using cloud infrastructure therefore initially we have only targeted large biomedical ontologies track. The results are as follows:

2.1 Large biomedical ontologies

SPHeRe is a cloud based ontology matching system that provides the facility to user for matching large scale ontologies without changing their hardware specifications. Figure 3 shows the results of our proposed system in large biomedical ontologies track. It has shown better precision values in almost all the tracks except task 4, while the recall of the system needs to be improved.



Tasks	Time (s)	#Mappings	Scores			Incoherence Analysis	
			Precision	Recall	F-Measure	Unsat.	Degree
Task 1	16	2359	0.960	0.772	0.856	367	3.6%
Task 2	8136	2610	0.846	0.753	0.797	1054	0.7%
Task 3	154	1577	0.916	0.162	0.275	805	3.4%
Task 4	20664	2338	0.614	0.160	0.254	6523	3.2%
Task 5	2486	9389	0.924	0.469	0.623	≥ 46256	≥ 61.6%
Task 6	10584	9776	0.881	0.466	0.610	≥ 105,418	≥ 55.7%

Fig. 3. SPHeRe Large Biomedical Ontologies Track Results

3 General comments

3.1 Comments on the results

Performance and precision are the strengths of our system. The design of proposed system also adds to its strength as it is a extendible and reusbale system. Recall is the main weakness of our system, but with the addition of new matching techniques as bridge algorithms can improve this aspect and therefore accuracy can be improved. Extendibility allows adoption of new bridge algorithms easily into the proposed system.

3.2 Discussions on the way to improve the proposed system

New bridge algorithms incorporating new matching techniques is the next line of plan for the proposed system. Object oriented and ontology alignment design patterns are to be implemented for matching different tracks of OAEI campaign. We also tend to include instance based matching, and incorporate change management techniques in the system.

4 Conclusion

SPHeRe system is a new initiative that relies on parallel execution of matcher bridge algorithms for achieving better performance and accuracy. The system is still working on improving the accuracy by incorporating more matcher bridge algorithms to increase the recall value of the system. Performance of the proposed system is better as compare to other system due to running large biomedical ontologies on a single system in appropriate time.

References

1. Wordnet a lexical database for english. http://wordnet.princeton.edu/, last visited in October 2013
2. Amin, M.B., Batool, R., Khan, W.A., Huh, E.N., Lee, S.: Sphere: A performance initiative towards ontology matching by implementing parallelism over cloud platform. In: Journal of Supercomputing. Springer, in Press
3. Li, L., Yang, Y.: Agent-based ontology mapping and integration towards interoperability. Expert Systems 25(3), 197–220 (2008)
4. Navarro, G.: A guided tour to approximate string matching. ACM computing surveys (CSUR) 33(1), 31–88 (2001)
5. Pavel, S., Euzenat, J.: Ontology matching: state of the art and future challenges. Knowledge and Data Engineering, IEEE Transactions on (25), 158–176 (2013)