AN ANALYSIS OF ONTOLOGY MATCHING CHALLENGES AND ITS SYSTEMS

K. SARULADHA, B. SATHIYA

Computer Science Department, Pondicherry Engineering College, Puducherry, India.
(charusanthaprasad@yahoo.com, (*Communication Author: sathiyal balloon gmail.com.)

Abstract: This paper aims at analyzing the key challenges of the ontology matching field and the system which worked on these challenges. There has been an enormous effort to provide a range of solutions to the matching problem by developing a variety of tools, yet it does not appear to be a finest one that is a standard for developing other matching tools. In this paper we first provide the basics of ontology matching. Then, we present various challenges of ontology matching and system, thereby highlighting areas where this field can be strengthened.

Keywords: Ontology, Ontology matching, Challenges.

1. INTRODUCTION

Ontology matching is a solution to the semantic heterogeneity between ontologies. It aims at finding correspondences between semantically related entities of different ontologies. These correspondences may stand for equivalence, subsumption, or disjointness, between ontology entities. The matching technique matches the two ontologies which are first imported into a format suitable for preprocessing. For a faster computation the ontology may be preprocessed. The preprocessor prepares neighbor list for each entity to speed up the structural similarity calculation, removes redundant information from ontology, normalizes the textual contents of entities by the process of tokenization, stopword removing, word-stemming and text standardization such as person names, dates in a uniform format. The similarity value is calculated for all possible entity pairs one from each of two preprocessed ontologies using a match workflow consisting of set of matchers (e.g., lexical, structural, instance and semantic matcher). The output of the matcher workflow is the semantic similarity value cube. The semantic value ranges from 0 to 1, i.e., value 1 indicates that the two entities are similar and value 0 indicates that the two entities are dissimilar. The matcher workflow can be sequential, parallel, iterative or mixed fashion. The similarity value obtained from the match workflow can be aggregated to obtain the final alignment set. Each alignment in the alignment set is a 5-tuple: < id, e_i, e'_i, R, SM> where id is an unique identifier, e_i is an entity in first ontology O, e'_i is an entity in second ontology O', R is an equivalence (=), subsumption (⊆ or ⊇) or disjointness (⊥) relationship holding between e_i and e'_i, and SM is a similarity between e_i and e'_i in the [0,1] range.

The remainder of this paper is organized as follows. Section 2 discusses in detail various challenges of ontology matching systems and accompanied for each of these with an overview of the recent advances in the field. Section 3 concludes the paper.

2. ONTOLOGY MATCHING CHALLENGES AND SYSTEMS

Although numerous ontology matching systems are proposed, there are still open issues that impose new challenges for researchers. These challenges should be rooted out to build a strong basis for ontology matching. This section discuss in detail about these challenges faced by the ontology matching system and the system which tries to handle these challenges. The challenges of the ontology matching system are as follows:
2.1. Large Scale Evaluation
According to OAEI’10 [12] the international Ontology Alignment Evaluation Initiative contest, only less than half of the ontology matching systems are capable of matching the large ontologies within one hour. Most of the systems run out of memory or time due to the huge computations required to match large ontologies. To match two ontologies of size \( n \), the system needs to process \( n^2 \) entity pairs. If \( m \) matchers are used then the system should process \( m \times n^2 \) entity pairs. For ontologies with thousands of entities the time required to process these exponential pairs takes a long time, even days. Moreover the space required for these computations is also high. The current ontology matching system tries to handle the scalability issue by various techniques like early pruning of ontologies [4,13], partitioning of ontologies [7,8,9,3], parallelization [6], self tuning [11,9] and match reuse [3]. Even though systems use these scalability techniques there is a trade-off between the efficiency and effectiveness of the system which should be considered by the future systems.

2.2. Dynamic and Self Configurable Matcher
Different ontologies have different dominant features or properties. The ontologies which have well defined descriptions and labels should be processed by linguistic matcher. The ontologies which have well defined structure properties by rich set of relations should be processed by structural matcher. The ontologies which have good number of instances of entity can be processed by instance matcher. The input ontology should be preprocessed to find which kind of matcher suits them. Even though preprocessing leads to extra time, it will be paid off later if the quality of the match result increases. The matchers should be chosen based on dominant features of input ontology rather than a static choice of matcher. In [11] a dynamic matcher RiMOM has been proposed which chose the matcher based on the input ontology. Beside matcher selection, the combination of the match results should also be self tuned i.e. the matcher should able to judge the relative importance of the matcher and combine it accordingly. The FALCON [9] heuristically combines the linguistic and structural matcher based on the input ontologies.

2.3. Effectiveness and Efficiency
The effectiveness of the matching system is determined by the quality of the match result. The parameters used to measure effectiveness are Precision and Recall. The efficiency of the matching system is determined by the time and space complexity of the system. The parameters used to measure efficiency are execution time and memory requirement. There is always a trade-off between the effectiveness and efficiency. The system with good effectiveness like ASMOV [10] lacks in efficiency. The system with good efficiency like FALCON [9] has less efficiency comparatively. This is due to the fact that good effectiveness is achieved by more computation which will automatically lead to less efficiency. Similarly when efficiency have to be increased certain operation have to be eliminated which will lead to reduced system effectiveness. The most common case is the maximization of the effectiveness or efficiency and the other parameter is compromised.

2.4. Automation
The ontology matching system can range from fully automated system to user designed workflow system. The fully automated systems have fixed set of matcher and its workflow is predetermined. Mostly the matcher suite consists of linguistic, structural and instance based matcher to handle all kind of ontologies. This will lead to unnecessary computation sometimes since all ontology need not have linguistic, structural and instance property. However the fully automated system can be used by naive
users easily compared to the user defined workflow systems. In user defined workflow the user can design the match system. The set of matchers can be chosen from a matcher suite and its workflow such as sequentially or parallel or mixed fashion can also be chosen. One such system is proposed in [13]. The drawback of this system is that the end user should have a prior knowledge about the matchers. Semi automated system also exist where certain parameters can be fixed by users.

2.5. Background Knowledge
The ontologies are constructed with certain background knowledge and in some circumstance the detail about the background knowledge can’t be a part of ontology specification and thus are not accessible to the ontology matching systems. Hence, the lack of background knowledge increases the complexity of the ontology matching process. A variety of approaches have been used to tackle the problem of the lack of background knowledge. One such method is to use one or more larger and detailed ontology of the same domain as background knowledge. The use of domain-related background knowledge in the ontology matching process has good impact on the recall of the match result, but the execution time of the matching process increases and hence not seems to scale well with large ontologies. The challenge in using this is to develop an approach to use background knowledge without compromising on the ontology matching performance. System like QOM [4], COMA++ [3], RiMOM [11], Agreementmaker [2], ASMOV [10] used external dictionary for matching process.

2.6. Ontology Matching Visualization
Semi-automated and user defined workflow system needs a good user interface to interact with the system. The human intervention is still needed by the ontology matching systems to design the match workflow and to ensure quality of the match result. Understanding an entity of one ontology in the context of other ontology is a cognitively difficult task since it requires the understanding of semantic relations among entities of different ontologies. Hence to make better understanding and human intervention possible, the visualization becomes mandatory. The visualization helps to graphically display data and process of ontology matching to understand and control the process in a better manner.

2.7. Distributed Ontology Matching
Distributed ontology matching uses the advantage of the existing network to divide the matching process between the peers. Rather than each system performing a matching process for large ontology, a set of system can share and do the matching process. This comes from three features:

(i) Amount of work done by each system is very small.
(ii) Improve the work done by other system.
(iii) Reduction in errors.

The work in [1, 5] used a distributed infrastructure to match and share ontologies.

Distributed ontology matching heavily depends on the infrastructures of matching system which should allow for sharing of data like alignments, annotation and sub-ontology. The alignment sharing can considerably reduce efforts required to match ontologies as alignment can be reused. Post processing of the alignments may also be done to avoid incomplete and contradictory alignments shared by systems. Determining the level of annotation support needed for description of the data is the current challenge in distributed ontology matching.

2.8. Match Results (Alignments)
Various challenges exist in finding, presenting and storing the alignments. To find alignments several matchers process the input ontologies and the output of each matcher are aggregated which leads to uncertainty. If aggregation function such as maximum is chosen only the alignment with high similarity value is considered and other alignments are discarded which may eliminate some correct alignments too. The alignment set formed may have contradicting alignments also. Thus post processing of alignment is required to eliminate these uncertainties. In [10], the system ASMOV used 5 post processing rules to eliminate the contradicting alignments. But post processing
leads to more execution time which should be taken care by the future systems.

The ontology matching system can provide justification along with the matching results. This is because the matching results produced may not be obvious to understand and therefore need justification. The important issue here is to present the justification in a clear and simple way for user to understand which will eventually allow them to edit and give feedback about the matching results.

Ample tools and standards are required to store and retrieve the alignments. The storage and retrieval of alignments can have the following features.

(i) Persistent storage is used to store the alignment and hence facilitate alignment reuse.
(ii) Proper indexing structure should be maintained for efficient retrieval of the alignment set for the given two ontologies.
(iii) Proper indexing structure should be maintained for efficient retrieval of the alignment within the alignment set given the alignment identifier.
(iv) The alignment set can be edited either by adding correspondences or discarding correspondences.
(v) Prune alignments based on a threshold.
(vi) Rewrite a message based on the alignments.

One of the major challenges in storage and retrieval of alignment is to provide a proper infrastructure that can support alignment storage, retrieval and reuse at the web scale level.

We are working with the ontology matching system to overcome issues like large scale evaluation and efficiency vs. effectiveness. We have proposed an ontology matching system which can handle large ontologies with increased efficiency and without compromising on effectiveness.

3. CONCLUSION

Even though major advancements are made in the ontology matching research field, the current status still possesses more important open issues which are major challenge to achieve efficient and effective ontology matching tools. This paper presents challenges for ontology matching systems and accompanied for each of these challenges with an overview of the recent advancements in the field. The challenges outlined are on the vital design part, hence, resolving them should lead to better ontology matching system.

References