Ontology Matching Algorithms: Comparative Analysis

MARGO HANNA, EL-SAYED M. EL-HORBART, ABDEL-BADEEH M. SALEM
Department of Computer Science
Faculty of Computer and Information Sciences
Ain Shams University, Cairo, Egypt
margofouad@yahoo.com sayed.horbaty@yahoo.com absalem@cis.shams.edu.eg

Abstract: The recent growing of the Semantic Web requires the need to cope with highly semantic heterogeneities among available ontologies . The Semantic Web envisions a metadata-rich Web where presently human-readable content will have machine-understandable semantics. Ontology matching techniques aim to tackle heterogeneity by establishing correspondences between ontologies' elements. An intricate obstacle faces the ontology matching problem is its scalability against large number and large-scale ontologies . This paper is comparative analysis among the different matching algorithms that tackle heterogeneity and recommend the algorithms that would fit for highly scalable systems.

Keywords:- Ontology, Matching algorithms, Semantic web, Heterogeneity, Scalability, Distribution, Performance

1 Introduction

Recently, matching algorithms have been introduced to focus on matching large-scale and large number schemas and ontologies, i.e. considering the efficiency aspect of matching algorithms, such as COMA++ , QOM, Bellflower, and PORSCHE [2]. Most of these systems rely heavily on either rule-based approaches or learner-based approaches. In the rule-based systems, schemas to be matched are represented as schema trees or schema graphs which in turn requires traversing these trees (or graphs) many times. On the other hand, learning-based systems need much pre-effort to train its learners [10]. As a consequence, especially in large-scale schemas and dynamic environments, matching performance declines radically.

Ontology matching is the task of identifying correspondences among elements of two or more ontologies. It is the process of finding correspondences between entities belonging to different ontologies.
The Web Ontology Language (OWL) from W3C is an expressive formalism for modelers to define various logical concepts and relations [10]. OWL ontologies come in three species: Lite, DL, and Full, ordered in increasing expressivity. Every Lite ontology is also DL ontology, and every DL ontology is also a Full ontology. OWL Lite and OWL DL are the species that use only the OWL language features in the way that complete and sound reasoning procedures exist. OWL Full, on the other hand, is undesirable. While OWL recently became a W3C recommendation in 2004, people have been working with it a few years, and many interesting ontologies already exist on the Web [10].

The paper aims at:

1. Presenting the different matching algorithms that could be used within the domain of HR Ontologies to tackle highly semantic heterogeneity among available ontologies.
2. Discussing efficiencies of the different algorithms
3. Providing a comparative analysis among the different presented algorithms
4. Recommending the algorithm/combination of algorithms to use for future work

The paper is organised as follows; section 2 presents the various ontology matching algorithms explaining advantages and disadvantages as well as some experiments details. While Section 3 demonstrates comparative analysis among the different presented algorithms with a recommendation to possibly merge some of the algorithms to enhance the matching efficiency, enhance scalability and overall system performance. Finally section 4 introduces some concluding remarks.

2.1 A Sequence-Based Ontology Matching Approach

This work proposes a novel approach for matching XML schemas. In particular, it developed and implemented the XPr¨uM system, which consists mainly of two parts —schema preparation and schema matching. Schemas to be matched are first parsed and represented internally using rooted ordered labeled trees, called schema trees. Then, it constructed a Pr¨ufer sequence for each schema tree. Pr¨ufer sequences construct a one-to-one correspondence between schema trees and sequences. The paper captured schema tree semantic information in Label Pr¨ufer Sequences (LPS) and schema tree structural information in Number Pr¨ufer Sequences (NPS). LPS is exploited by a linguistic matcher to compute terminological similarities between schemas elements [2].

Linguistic matching techniques may provide false positive matches. Structural matching is used to correct such matches based on structural contexts of schema elements. Structural matching relies on the notion of node2 context. This work distinguishes between three types of node contexts depending on its location in the schema tree. These types are child context, leaf context and ancestor context. It exploits the number sequence representation of the schema tree to extract node contexts for each tree node in an efficient way.

Then, for each node context, it applies its associated algorithm. For example, the leaf context similarity between two nodes is measured by extracting leaf context for each node as a gap vector. Then, the cosine measure is applied between two gap node vectors. Other context similarity measures are determined similarly. By representing schema trees as Pr¨ufer Sequences these trees needs to be traversed only once to construct these sequences. Then, a novel structural matching algorithm is developed to capture semantic information existing in label sequences and structural information embedded in number sequences [2].

2.2 Ontology Matching Using an Artificial Neural Network to learn Weights

Ontologies form a semantic foundation for many domains, such as Web services, Ecommerce, and the Semantic Web, where applications can mutually understand and share information with each other. However, because ontologies reflect their designers' conceptual views of part of the world, heterogeneity is an inherent characteristic for ontologies. During (semi)automated matching among ontologies, different semantic aspects, i.e., concept names, concept properties, and concept relationships contribute in different degrees to the matching result. Therefore, a vector of weights is needed to be assigned to these aspects. It is not trivial to determine what those weights should be, and current research work depends a lot on human heuristics. This work focus on an artificial neural network approach to learning and adjusting the above weights, and thereby support a new ontology matching algorithm, with the purpose to avoid some of the disadvantages in both rule-based and learning-based ontology matching approaches.

This work presents a new approach to match ontologies that combines both rule-based and learning-based algorithms [5]. It integrated an artificial neural network (ANN) technique in the algorithm, such that the weights mentioned above can be learned instead of being specified in advance by a human [5]. Moreover, this learning technique is carried out based on the ontology schema information alone, which distinguishes it from most other learning-based algorithms.

2.2.1 Proposed solution:

Based on the insight of the advantages and drawbacks of these Rule-Based and Learning-Based Ontology matching, this work presents a new matching algorithm, Super concept Formation System (SFS), which combines rule-based and learning-based solutions. This integrates machine learning techniques, such that the weights of a concept's semantic aspects can be learned from training examples, instead of being ad-hoc predefined ones.

In addition, in order to avoid the problem of lacking instance data (either quality or quantity), which is common for real-world ontologies, the weight learning technique is carried out at the schema level, instead of the instance level. The main idea is, given a pair of ontologies being matched, although it is true that many design diversities might exist, it is still reasonable to assume that the contributions of different semantic aspects to ontology understanding should be independent of specific concepts. In fact, different contributions, which are the foundation for different weights, are characteristics of ontologies viewed as a whole. That is, during ontology matching, weights are features with regard to ontologies, rather than individual concepts. Therefore, it is possible to learn these weights for all concepts by training examples from a subset of concepts.

The following assumptions are made in this work for two ontologies being matched.

- They are from the same or similar general domain, and they have “Thing” as a built-in common root.
- Each is correct or reasonable in itself.

Ontology matching consists of many mapping tasks, SFS concentrates on finding pairs of equivalent concepts as the first step. In addition, after the successful discovery of equivalent concept pairs, it is not difficult to design an algorithm to merge/align corresponding ontologies. There are many different kinds of relationships in ontologies, e.g., superClassOf, subClassOf, partOf, contains, etc. In this paper, only the super/subClassOf relationships are considered for they are the most common ones in most real world ontologies [5].

2.3 Genetic Algorithm based Ontology Matching

This is a genetic algorithm-based optimization procedure for ontology matching problem which is presented as a feature-matching process. First, the problem of ontology matching is modelled as an optimization problem of a mapping between two compared ontologies and every ontology has its associated feature sets. Second, as a powerful heuristic search strategy, genetic algorithm is employed for the ontology matching problem. Given a certain mapping as optimizing object for GA, fitness function is defined as a global similarity measure function between two ontologies based on feature sets. Finally, a set of experiments are conducted to analysis and evaluate the performance of GA in solving ontology matching problem.

As a powerful search strategy based on natural selection and population genetics, genetic algorithm [4] outperforms conventional optimization methods such as the gradient ascent and simulated annealing. The implementation of GA in application of the ontology matching problem incorporates three basic steps so that the algorithm is formulated for the specific application: the presentation of individual, i.e. the encoding mechanism of problem, and the formulation of the fitness function that gives to each individual a measure of performance [6].

2.3.1 Solution Strings

Let n1 be the number of concepts in ontology O1, and n2 be the number of concepts in ontology O2. GA is applied to find the optimal mapping function between concepts from these two ontologies, that is the solution string is corresponds to a mapping function

M defined in Definition6, indicating a mapping from each concept in O1 to a corresponding concept in O2. Each solution string or individual in the population would be a one-dimensional array with n1 integer elements that may take values between 1 and n2 [6].

2.3.2 Fitness Function

In order to determine the quality or performance of each encoded solution string in the population, the GA associates a fitness measure with each solution string.A global similarity measure function is defined between two compared ontologies with respect to the mapping as fitness function in GA. Since an ontology is represented as a collection of features, similarity between two compared ontologies will be described as a feature-matching process. Based on the Tversky’s similarity model and the set theory, a global similarity measure function between two ontologies 1 O and 2 O is defined, then Intentional Rule, and Extensional Rule [6].

2.4 Effective Ontology Matching in High Performance Computing Environments

The main motivation behind this work is that the procedure of ontology matching requires a robust and scalable solution that ensures the maximal efficiency of matching operations. That is especially important when thinking of matching large scale data among several ontologies, where the performance and scalability of performing the matching algorithms is settled to the point. In this paper, the proposed approach for distributed ontology matching improves the matching’s efficiency and scalability due to the distribution and parallelization of implemented algorithms. This enables applications performing ontology matching to get benefit of running in high-performance computing environments and ensures that the full potential of computing resources is enabled for the matching process [3].

Ontology matching is a solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities of ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, etc. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate. Thus, this work presents an approach for matching ontologies in an effective and robust way.

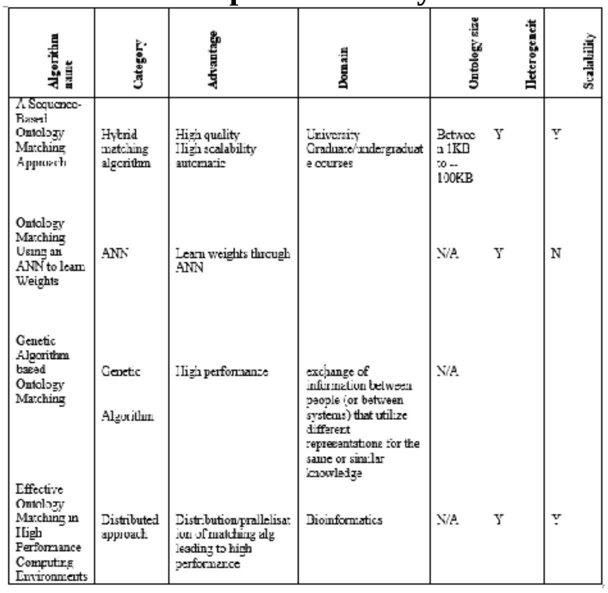
One of the objectives of this work is to provide a user with one priority ontology which includes the knowledge structures of a given set of large scale ontologies from the field of bioinformatics. The benefit for such an ontology is the possibility to receive required information (e.g. medical datasets) very fast by considering only one data source, the extended priority ontology. Through this, learning and teaching in the field of semantic web are provided. The presented approach supports receiving required datasets in order to extend already available knowledge structures [3].

3.1 Approaches for Ontology Matching

Currently approaches for ontology matching are used in order to merge ontologies together. Therefore the selected ontologies are matched by an adequate algorithm in order to ensure a proper merging. However, these approaches require a high amount of computing resources in order to meet the requirements of the matching and merging methods. Hence, there are several issues which have to be solved for ensuring a scalable matching solution, e.g. identification of the most beneficial matching approach, ensure scalability and robustness, sequence for matching the ontologies, identification of beneficial ontology repositories.

At present approaches consider a division of selected ontologies with the aim to execute the matching algorithms independently from other parts of the ontology. This one ontology is divided into several parts, Falcon-AO, an automatic ontology matching system and part of the Falcon1 infrastructure, is related to this issue. Falcon-AO supports the division of ontologies into several parts by the PBM2 with the aim to match selected ontologies together. However, it is still a challenge to provide the matching with the required computing resources.

Table1: Ontology Matching Algorithms' Comparative Analysis



Algorithm name	Category	Advantage	Domain	Ontology size	Heterogeneit	Scalability
A Sequence-Based Ontology Matching Approach	Hybrid matching algorithm	High quality High scalability automatic	University Graduate/undergraduate courses	Between 1KB to -- 100KB	Y	Y
Ontology Matching Using an ANN to learn Weights	ANN	Learn weights through ANN		N/A	Y	N
Genetic Algorithm based Ontology Matching	Genetic Algorithm	High performance	exchange of information between people (or between systems) that utilize different representations for the same or similar knowledge	N/A		
Effective Ontology Matching in High Performance Computing Environments	Distributed approach	Distribution/prallelisation of matching alg leading to high performance	Bioinformatics	N/A	Y	Y

The above table shows the strength of the surveyed algorithms in their own and highlight the fact of achieving higher levels of performance via combining some of these algorithms together to maximise the benefit, for example SFS which combines rule-based and learning-based solutions. It's difficult to have clear cut classification of algorithms; however this paper attempted to provide such classification based on granularity an input interpretation on the one side and the kind of input on the other side.

4 Conclusion and Future work

As presented researchers have applied different matching algorithms, each has got its strengths and drawbacks. This paper aims at analysing the different presented algorithms to gain in-depth understanding and be in position to combine and merge standard matching algorithms as well as distributed/parallel ones to improve efficiency deploying multiple resources in the process and being able to maximize the benefit, speed up the process which will be more beneficial and of higher add value to the people who will be seeking jobs using the proposed job seekers ontology.

For future work, I'd propose implementing a hybrid algorithm where we could merge few of the matching algorithms that have been discussed previously in order to maximize and optimise the efficiency of the Web Semantic matching process. Also my next research would focus on scientifically investigating whether to merge ontologies or align ontologies. Two approaches are possible: (1) merging the ontologies to create a single coherent ontology, or (2) aligning the ontologies by establishing links between them and allowing them to reuse information from one another.

References

[1] Ahmed Alasoud, Volker Haarslev, Nematollaah Shiri, “An Effective Ontology Matching Technique”, ISMIS'08 Proceedings of the 17th international conference on Foundations of intelligent systems , Springer-Verlag Berlin, Heidelberg ©2008, pp1-7

[2] Alsayed Algergawy, Eike Schallehn and Gunter Saake1, “A Sequence-based Ontology Matching Approach”, http://www.google.co.uk/url?sa=t&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fwwwiti.cs.unimagdeburg.de%2F~alshahat%2Fonto.pdf&ei=bF49TZKUHJ6qhAes3eDeCg&usg=AFQjCNHsQYs6A1B_vWu_12w5Rk3qcTv5HA, last updated June 2009, pp1-5

[3] Axel Tenschert1, Alexey Cheptsov1, “Effective Ontology Matching in High-Performance Computing Environments”, 1st International Workshop on Semantic Web Applications for Learning and Teaching Support in Higher Education (SemHE'09), 2009, ECTEL'09, pp1-5

[4] Axel Tenschert1, Matthias Assel, Alexey Cheptsov, Georgina Gallizo, Emanuele Della Valle, Irene Celino,” Parallelization and Distribution Techniques for Ontology Matching in Urban Computing Environments”, Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, 2009, pp1-4

[5] Jingshan Huang1, Jiangbo Dang2, Jos´e M. Vidal1, and Michael N. Huhns, “Ontology Matching Using an Artificial Neural Network to Learn Weights”, Computer Science and Engineering Department, University of South Carolina, Siemens Corporate Research, http://www.csd.abdn.ac.uk/ cmckenzi/playpen/rdf/, pp1-6

[6] Junli Wang1, Zhijun Ding1, 2, Changjun Jiang1, GAOM: “Genetic Algorithm based Ontology Matching”, Proceedings of IEEE Asia-Pacific Conference on Services Computing, 2006, pp1-4

[7] Malgorzata Mochol, “Metadata-Based Matching Framework for Ontologies”, DEXA '06 Proceedings of the 17th International Conference on Database and Expert Systems Applications, IEEE Computer Society Washington, DC, USA ©2006, pp1-5

[8] Prasenjit Mitra, Peng Liu, Chi-Chun Pan, “Privacy-preserving Ontology Matching”, American Association for Artificial Intelligence, 2005, pp1-4

[9] Viviana Mascardi, Angela Locoro, Paolo Rosso, “Automatic Ontology Matching Via Upper Ontologies: A Systematic Evaluation”, IEEE Transactions and Knowledge Engineering, May 2010 (vol. 22 no. 5) , pp. 609-623

[10] Taowei David Wang1, Bijan Parsia2, James Hendler1, “A Survey of the Web Ontology Landscape”, 1 Department of Computer Science, University of Maryland, College Park, MD 20742, USA, {tw7, hendler}@cs.umd.edu , 2 The University of Manchester, UK, bparsia@cs.man.ac.uk, pp1-6