Adding a Pre processing Phase to ASMOV for Improving the Alignment Result

Bahareh Behkamal[1*], Mahmoud Naghibzadeh[2] and Reza Askari Moghadam[3]

[1]*Department of Software Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran*

[2]*Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*

[3]*Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran*

**Correspondent Author: behkamal@mshdiau.ac.ir*

Abstract

Ontology matching is one of the most important challenges in the semantic web. It is a technique which is used to find the semantic correspondences between entities which are modeled in different ontologies. Despite many research efforts in ontology matching, the matching process still suffers from severe problems with respect to the quality of the matching results. Furthermore, finding the correspondences, for the most part, is very time-consuming. In this paper, we examine how a pre-processing phase can improve the results of the matching process. A pre-processing phase is added to the matchers for the analysis of input ontologies so as to detect some inappropriate patterns which are modeled by various developers. Then, refactoring operations are utilized on detected patterns for achieving assimilated ontologies. Finally, our proposed approach is tested by ASMOV (Automated Semantic Matching of Ontologies with Verification), which is one of the top matchers in OAEI (Ontology Alignment Evaluation Initiative). Experimental results show that ontologies achieved from this proposed approach are more efficient than the original unrepaired ones, with respect to standard evaluation measures.

Keywords: *Ontology matching, Pattern detection, Refactoring, ASMOV*

1. Introduction

The semantic web is a web technology that enables semantic interoperability between web sources and web users. One approach for modeling the knowledge in the semantic web is ontology. Ontologies play an essential role in semantic interoperability and they are the main vehicle for the development of the semantic web. Ontologies are developed by different developers and by various methods; therefore, heterogeneity problems in ontologies are encountered. For this reason, an approach is needed to overcome this problem. One approach is the ontology matching technique. Ontology matching is a significant operation in traditional applications, such as ontology integration, schema integration, data warehouses, e-commerce, query mediation, distributed expert system, and so on. Typically, these applications are characterized by heterogeneous structural models that are analyzed and matched either manually or semi-automatically at design time [1].

The aim of the matching process is to discover the interchangeable and synonymous concepts and relations of the ontological content.

Thus far, many diverse matchers have been proposed which use various methods to find the associations between ontological components. Despite serious research efforts to addresses matching problems, matchers still suffer from severe difficulties associated with the quality of matching results.

In this paper, we propose an approach for obtaining better results from the matching process. In this approach, a pre-processing phase is applied to the input ontologies of ASMOV. The reason for selecting this matcher, among other matchers participating in the OAEI contest, is that ASMOV was one of the top level matchers in OAEI 2009 and OAEI 2010.

The rest of this paper is organized as follows: Section 2 presents some introductory definitions of the concepts relevant to this literature. Section 3 provides an overview of the related works and Section 4 describes the problem definition. In Section 5, the proposed approach is elaborated on. Section 6 presents the experimental results. Finally, in Section 7, the conclusion and future work are given.

2. Definition of Basic Concept

The following are some preliminary definitions and terms which are relevant to this literature and used throughout this paper.

2.1. Ontology

An ontology O contains a set of entities related to a number of relations. Entities of ontology can be divided into components including: classes (C), individuals (I), literals (L), and data types (T), object properties (OP), data type properties (DP). There are four relations: equivalence, subsumption, disjointness, and membership [2].

2.2. Matching process

A matching process can be seen as a function f that takes two ontologies o and o', a set of parameters p and a set of oracles and resources r and returns an alignment A between o and o' [1].

2.3. Correspondence

A correspondence between an entity e belonging to ontology o and an entity e' belonging to ontology o' is a 5-tuple < id, e, e', R, conf> where: id is a unique identifier of the correspondence; e and e' are the entities (e.g. properties, classes, individuals) of o and o'; R is a relation and Conf is a confidence measure (typically in the [0, 1] range) holding for the correspondence between the entities e and e' [3].

2.4. Alignment

The alignment of ontologies o and o' is a set of correspondences between two or more (in the case of multiple matchings) ontologies. The alignment is the output of the matching process between the entities of o and o' [1].

2.5. Refactoring

Refactoring is recognized as a change made to the internal structure of the software in order to make it easier to understand and to modify without changing its observable behavior [4].

3. Related Works

This section discusses the state-of-the-art ontology matching techniques and describes previous work on pattern detection and refactoring in the field of ontology matching.

Research in ontology matching has been burgeoning since the early 2000's. So far, most articles on the ontology matching field have focused on the method of matching processes and have introduced some matchers with diverse algorithms. Authors in [1] present a comprehensive review of current approaches in the ontology matching field. They classify them along three main dimensions: input interpretation, kind of input, and granularity. The input interpretation dimension is divided into syntactic, external, and semantic. The kind of input dimension categorizes techniques working on textual strings, structural, extensional, and semantic. The granularity dimension distinguishes between element-level and structure-level techniques [2].

Here, we introduce some matchers that have participated in the ontology alignment evaluation initiative (OAEI). ASMOV [2] used lexical and structural characteristics of two ontologies to iteratively calculate a similarity measure between them, derive an alignment, and then verify it to ensure that it does not contain semantic inconsistencies. Semantic verification phase for this matcher leads to have a high performance in contrast other matchers. Therefore, in this paper for evaluating our proposed approach, this matcher was used. RiMOM [5] is a dynamic multi-strategy ontology alignment framework that combines multiple strategies to improve matching efficiency. The key intuition in this framework is that similarity characteristics between ontologies may vary widely. This approach considered both the textual and structural characteristics of ontologies. RiMOM is a framework based on risk minimization of the Bayesian decision. It employs multiple ontology alignment strategies and sets the combination weight. Another system is Falcon-AO [6], a practical ontology matching system with good performance that acts based on a number of remarkable features. It is an automatic ontology matching system that uses multiple elementary matchers (V-Doc, GMO and PBM), coordination rules and the similarity combination strategy. S-Match [7] is a deductive technique for semantic ontology matching which employs a number of element-level matchers to express ontologies as logical formulas and then a propositional satisfiability (SAT) solver to check the validity of these formulas. GLUE [8] discovers mappings through multiple learners which analyses the taxonomy and the information within concept instances of ontologies. PROMPT [9] algorithm consists of an interactive ontology merging tool and a graph-based mapping called Anchor-PROMPT. Anchor-PROMPT [10] uses linguistic "anchors" as a starting point and analyses these anchors in terms of the structure of the ontologies.

Previous works on ontology patterns in the field of ontology matching are limited [11-14]. Ontology patterns have been used in some fields, e.g. ontology engineering, but they have not been applied to the field of ontology matching. Ontology patterns are mainly inspired by software engineering and knowledge engineering [15]. Here, we describe some previous works in the field of ontology matching by considering ontology patterns. The survey in [11] consists of testing the impact of ontology refactoring on the results of three matcher systems. For accomplishing this, some modeling errors via name structure analysis were found, to which three refactoring operations were applied. By considering semantic structures, authors in [12] analyzed collections of OWL ontologies in order to determine the number of occurrences of several combined name and graph patterns. These structures ranged from simple subsumption to more complex constructions. The goal of this paper is to assist automatic alignment among different models by finding such patterns in some ontologies. In [13], the authors concentrate on the detection and mutual matching of semantic structures in ontologies. To this end, the authors use the equivalence relation, as well as analyzing homogeneous correspondence. Research in [14] presents a simple method of tracking name patterns over OWL ontology taxonomies. This method helps to detect several probable taxonomic errors and modeling inconsistencies with respect to their set-theoretic interpretations.

Although ontology refactoring is employed in many different areas [16-19], the impact of ontology refactoring on the ontology matching field is rarely seen [11, 13, 20]. In [20], a semi-automatic process for lifting metamodels into ontologies is proposed, which allows for the creation of the semantic integration of modeling languages. In so doing, implicit concepts in the metamodel are changed to explicit concepts in the ontology. Thus, the implementation of a certain modeling language for the explicit reification of concepts is covered by this language. The application of refactoring patterns on the resulting ontologies could improve the automation support for semantic integration tasks.

4. Problem Definition

Ontology matching is one of the hottest topics in many fields of semantic web research. The matching of ontologies (or schemas) is a critical operation in many application domains, such as the semantic web, ontology integration, data warehouses, e-commerce, query mediation, and so on. All these applications can benefit from ontology alignment, that is, a set of correspondences between the entities of two ontologies. Each correspondence represents a kind of relationship between entities or, more generally, between whole semantic structures.

Ontology matchers use different methods to find correspondences and there are many matchers to do so [2, 5-10]. However the quality of the delivered alignments is not very high. Higher quality could be achieved by manually designing alignments, but this is very demanding. We can obtain a better quality of matchers by using refactored ontologies for inputs, instead of the original unrepaired ones. For accomplishing this, we utilize a thorough survey on naming which is dedicated to entities and a taxonomy which is used in the ontological structure. Taxonomy is a classification of entities that is usually hierarchical. It performs two objectives: giving exact names for all items dealt with (i.e. domain) and showing which items are parts of others (this is sometimes called parent-child relationships and broader-narrower).

Different naming conventions and taxonomic structures for declaring the ontology entities lead to several problems for matchers. Some can be overcome by using ontology design patterns (ODP) [21]. ODPs identify ontological design structures, terms, larger expressions, and semantic contexts [22]. An ontology design pattern is a successful reusable solution to a recurrent modeling Problem. The ODP contains ontology patterns (OPs) that assist in the design of ontologies. OPs are of different types and offer help in a variety of ways. In this paper, we concentrate on Naming OP and Structural OP types. Naming OP is one of the pattern types of ODPs that act as a convention on how to create names for namespaces, files, and ontology elements in general (classes, properties, etc.). By supporting homogeneity in naming procedures, Naming OPs are recommended practices for improving ontology readability and understanding by humans and matchers. Assimilating the name of ontology entities, based on naming conventions and establishing uniform ontologies, renders the ontology easier to understand by both users and matchers. Structural ODPs include logical ODPs and architectural ODPs, which are related to the hierarchical relations in ontologies [23]. Based on a common rule in hierarchy and granularity for defining the entities of ontologies, these patterns are proposed to facilitate the design of ontologies. In this paper, for solving problems caused by ontology diversity, we use these pattern techniques and ontology refactoring in the pre-processing phase of ASMOV. This approach makes ontologies easier to understand by ASMOV and also avoids some common errors in the alignment results.

5. Overview of Our Approach

In this section, an approach is proposed for improving the quality of the matching results. The aim of this approach is to improve the alignment result by adding a pre-processing phase to ASMOV, whose workflow is illustrated in Figure 1. In the pre-processing phase, at first, a comprehensive survey on some ontologies is performed for identifying some inconsistencies in the ontologies. Then various lexical and structural patterns, which have been modeled by different developers, are detected. Afterward, some refactoring operations are applied on those patterns for repairing the incompatible ontologies. Finally, these repaired ontologies are used as inputs of the matching process with ASMOV.

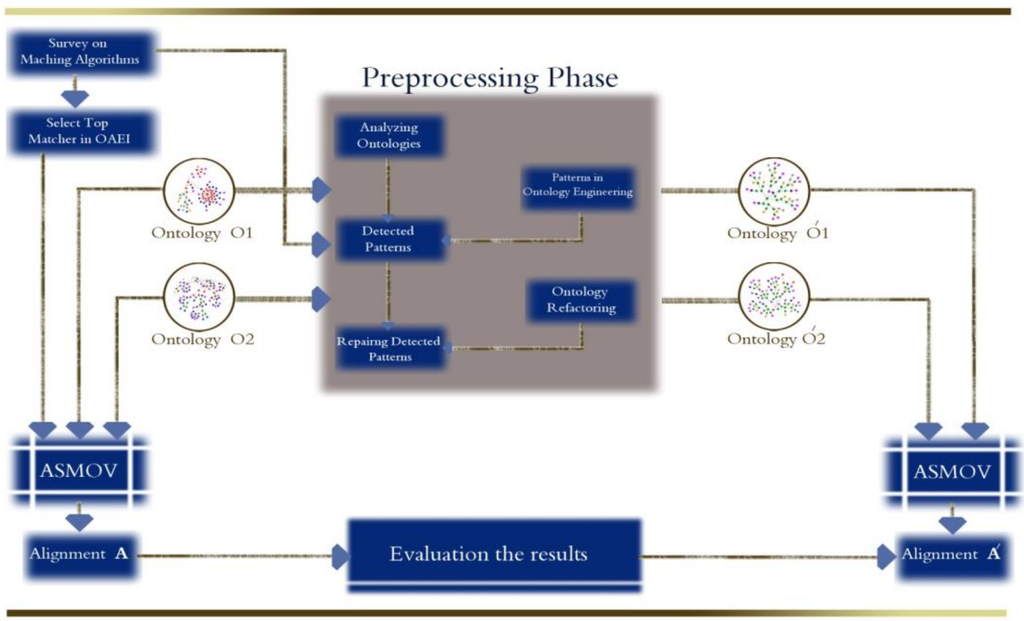


Figure 1. Proposed Approach Diagram

5.1. Pattern Detection Phase

Patterns in this study were detected based on our preliminary analysis of some ontologies from the conference track. The patterns considered in this study are lexical and structural ones. For detecting lexical patterns, we analyzed the name of entities, especially classes in OWL ontologies. Furthermore, it was found that ASMOV uses the Lin method [24] for calculating lexical similarity. The lexical feature consists of all information readable by humans as provided in the ontology. In ASMOV, three lexical features in OWL ontologies are considered: id, label, and comment. One issue is that various ontologies use different methods for defining the names of homogeneous concepts, especially for compound words. In OWL ontologies, different styles in concept naming lead to many obstacles for calculating lexical similarities in ASMOV. For example, in two ontologies of a conference track, namely Conference and Ekaw we discovered two different class namings for a similar concept, that is <Conference#conference-www> ~ <Ekaw#website> and also <Conference#rejected-contribution> ~ <Ekaw#rejected-paper >, both of which are not found in ASMOV. Therefore,

for solving these kinds of problems, some lexical patterns can be detected based on naming Ops and ontology design patterns [21] for the purpose of equally naming for these different styles of naming. To accomplish this, we used one refactoring operation called RN, which is described in Section 4.2, for assimilating the lexical features of OWL ontologies. By doing this, calculating the lexical similarity with ASMOV, which is used Lin method, can do better than before. Therefore, better results can be obtained from the matching process by ASMOV.
Another pattern is based on the fact that the taxonomic structure of ontologies is often varied and confusing. One reason for this is that different developers have dissimilar viewpoints for developing ontologies. Therefore, different hierarchies and granularities for defining the entities of ontologies are utilized in the same domain. For example, in two ontologies of the conference track namely Conference and Ekaw, we realized there were two different granularities in concept naming for the similar concept author. In Conference, we found three levels of granularity for author, including: contribution_regular-author, contribution_co-author, and Conference _1th-author. However, in Ekaw, there was only one level of granularity for author, namely Paper_ author.

Furthermore, some problems for calculating the relational similarity by ASMOV have been recognized. The relational or hierarchical similarity in ASMOV is computed by combining the similarities between the parents and children of entities that are to be compared. For detecting structural patterns, the structural and hierarchical relations of ontologies are analysed. By considering the problems mentioned above and ASMOV's work, we realized that different taxonomic structures and different granularities in peer ontologies cause many problems in the matching process. For solving this, another refactoring operation, called RS, is employed for assimilating the structural features of OWL ontologies. In this way, calculating the relational similarity with ASMOV is performed better than before, thus producing better results from the matching process by ASMOV.

Our results show that, in most ontologies, there are a significant number of occurrences of the aforementioned patterns. For example, Figure 2 illustrates different styles in class naming and different taxonomic structures for defining the same concepts in a part of two ontologies, namely Conference and ConfOf.

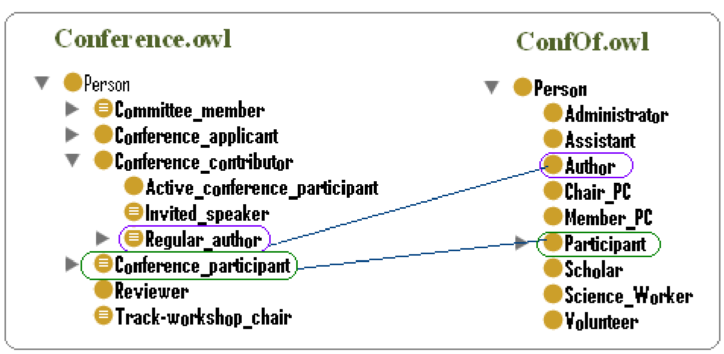


Figure 2. Different Styles in Class Naming and Different Taxonomic Structures in Two Ontologies (Conference.owl, ConfOf.owl)

Figure 3 illustrates the correspondences found by ASMOV after applying refactoring operations (RN and RS) on ConfOf and Sigkdd. The lines in Figure 3 show that correspondences exist in the reference alignment and they could be identified by ASMOV after the refactoring operations.

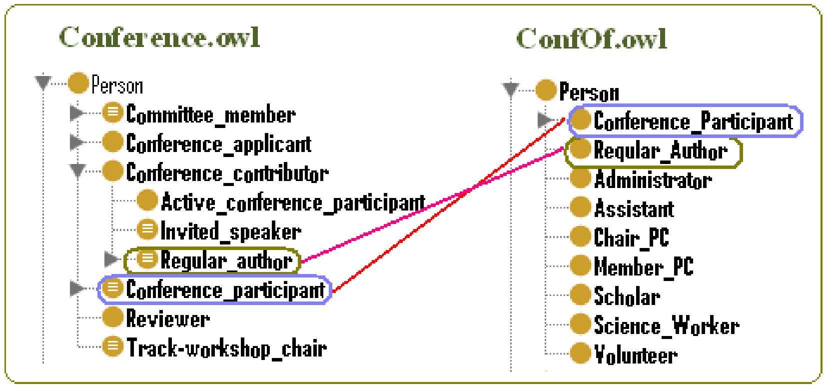


Figure 3. Correspondences Identified by ASMOV After Refactoring

5.2. Refactoring Phase

All cases of the modeling errors detected via some of the patterns mentioned earlier can be repaired by two refactoring operations. The detection of these patterns is the starting point for a refactoring. Generally, refactoring is a process for performing some changes in the internal structure of software in order to make it easier to understand and to modify without changing its discernible behavior. In this literature, we use the refactoring process in an ontology matching area. Thus, some changes were made in ontologies by a semi-automatic process for achieving new more understandable versions of ontologies for users and matchers. These versions of ontologies can be utilized more effectively by different ontology matching tools. There are three general refactoring operations: adding operation (ADD), restructuring operation (RS), and renaming operation (RN). These operations consist of different steps depending on the detected situation [11]. In this paper, RN and RS are used for lexical patterns and structural patterns.

More desirable results in lexical similarity of ASMOV can be obtained by applying rename operations (RN) for the name of classes in ontology. This is done by considering the name of these classes in a peer ontology that has the same taxonomic structure. Furthermore, by considering the parent-child relations and various granularities used in peer ontologies, restructuring operations (RS) are applied for assimilating the structural features of OWL ontologies. Experimental results show that, better results can be achieved from the structural similarity phase of ASMOV by transforming part of ontology into another one.

We carried out experiments on seven pairs of ontologies from the conference track. The reason for choosing these seven pairs among other ontologies is described in Subsection 4.2. The percentage of RN and RS operations applied on these seven pairs of ontologies is illustrated in Figure 4. As can be seen in Figure 4, in four pairs of ontologies, <Cmt-ConfOf>, <Cmt-Ekaw>, <Conference-Ekaw>, and <Edas-Ekaw>, RN operations are applied more than RS operations, because of the many different lexical patterns found in these pairs. Besides, in other ontology pairs, <Cmt-Sigkdd>, <Conference-ConfOf>, and <ConfOf-Sigkdd>, RS operations are utilized more than RN operations, because these pairs of ontologies have different hierarchical structures and RS operations is used for assimilating the taxonomies.

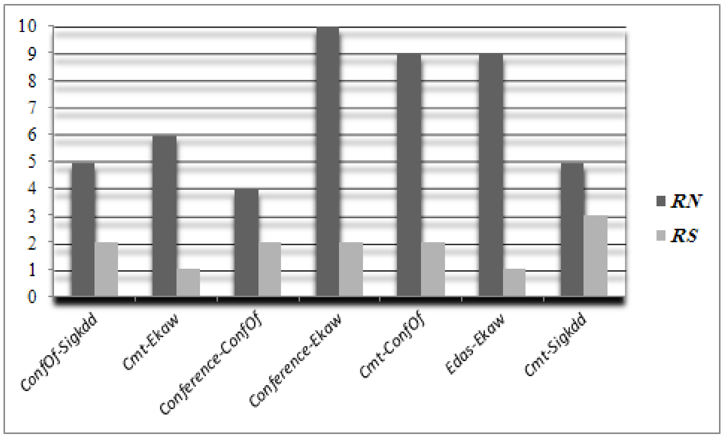


Figure 4. Comparison the Number of Refactoring Operations on Different Pairs

6. Experimental Results

6.1. Implementation

Implementation is based on the employment of Java language with Jena API in Net Beans IDE. Furthermore, we used protégé and the Ontology Pre-Processor Language (OPPL) for manipulating ontologies written in OWL. OPPL is a domain-specific macro language, based on the Manchester OWL Syntax. OPPL instructions can add/remove entities and add/remove axioms to/from entities in OWL ontology. The OPPL instruction manager is a Java library that processes OPPL instructions to make changes in OWL ontology. This language is also suitable for defining independent modeling macros (e.g. Ontology Design Patterns) that can be applied across ontologies [25].

6.2. Data sets

Ontologies were used in our experiments are part of OAEI. The OAEI offers several tracks and subtracts concentrated in different types of matching problems. Our approach was tested on the conference track. The conference dataset can be seen as a much harder test case compared to other ontologies of OAEI like benchmark dataset, because it is more heterogeneous and has been extensively studied over the past years. The conference tests consist of 15 ontologies where each pair of ontologies constitutes a matching problem. These ontologies have been developed as part of the OntoFarm project and dealing with conference organization. They are described in OWL-DL and serialized in the RDF/XML format [26]. Six out of sixteen ontologies of the conference track were used in our experiment. These ontologies are Cmt, ConfOf, Ekaw, Conference, Edas and Sigkdd. The reason for selecting these six ontologies among others is that reference mapping are available for all possible combinations of these selected ontologies. To evaluate exactness of matching process, it is necessary to determine both the number of correctly found correspondences and the number of incorrectly found correspondences. This is done by using a reference alignment between two ontologies which have been previously extracted by human experts.

6.3. Match Quality Measures

To provide a basis for evaluating the quality of automatic matching techniques, the matching result should first be manually obtained. The obtained real match result can be used as the "gold standard" to assess the quality of the matching process. Comparing the result of the automatically derived matches with those of the real matches should be performed to determine the quality measures for schema matching.

In particular, the set of automatically derived correspondences is composed of true positives and false positives. False negatives are correspondences needed but not automatically identified, while false positives are correspondences falsely found by the automatic match operation. True negatives are false correspondences which have also been correctly discarded by the automatic match operation. Instinctively, both false negatives and false positives reduce the match quality.

Based on the cardinality of these sets, two common measures can be computed, these are precision and recall, which are actually derived from the information retrieval field. Precision reflects the share of real correspondences among all found ones and recall determines the share of real correspondences that have been found. Ideally, when no false negatives and false positives are returned, we have Precision=Recall=1. However, neither precision nor recall alone can accurately assess the match quality. Hence, it is necessary to consider both measures and obtain a combined measure, namely F-Measure [27]. In this paper, three measures were utilized for evaluating.

6.4. Comparing Results using Original Ontologies and Repaired Ones

Experiments were performed on seven pairs of ontologies from the conference track. We automatically generated alignments by ASMOV for these pairs of ontologies, namely Cmt-Ekaw, Cmt-ConfOf, Cmt-Sigkdd, Edas-Ekaw, Conference-ConfOf, Conference-Ekaw, and ConfOf-sigkdd. Alignment from ASMOV is generated for those pairs of ontologies before and after implementing the proposed approach. The results were illustrated in Figure 5. The results of our experiments show that refactored ontologies improve the matching results with respect to standard evaluation measures.

Test Ontologies	Precision		Recall		F-measure	
	Before	*After*	*Before*	*After*	*Before*	*After*
ConfOf-Sigkdd	0.3	0.4	0.57	0.85	0.39	0.54
Cmt-Ekaw	0.4	0.5	0.54	0.72	0.45	0.59
Conference-ConfOf	0.2	0.4	0.2	0.66	0.2	0.5
Conference-Ekaw	0.44	0.63	0.48	0.76	0.45	0.68
Cmt-ConfOf	0.44	0.52	0.25	0.56	0.31	0.53
Edas-Ekaw	0.34	0.44	0.52	0.69	0.4	0.53
Cmt-Sigkdd	0.41	0.47	0.58	0.66	0.47	0.55

Figure 5. Experimental Results of Proposed Approach

7. Conclusions and Future Work

In this paper, a pre-processing phase is added to ASMOV in order to analyse input ontologies for detecting various lexical and structural patterns. Detecting and fixing these kinds of patterns could solve problems which are created by ontology diversity. Afterward, renaming operations (RN) and restructuring operations (RS) can be applied on detected patterns to establish uniform ontologies for improving the matching results. This approach renders ontologies easier to understand by both humans and matchers and it avoids some common mistakes in the alignment results of matching tasks. The experiments were carried out on some ontologies of the conference track. The results show that refactored ontologies improve the results of ASMOV with respect to standard evaluation measurements, i.e. precision, recall, and F-Measure. For future work, we suggest testing this approach for other matching tools, especially those participants in the OAEI contest with high ranks. Besides, more patterns can be enlarged for recognizing anomalies and errors which have been modelled by different developers and novel refactoring operations. Furthermore, this proposed approach can be used in the reasoning of ontologies.

The final suggestion is to find a way for the post-processing of the matching process to recognize anomaly patterns in the alignment result during the free time of the matching process.

Acknowledgment

I would like to thank Ondˇrej ˇSv´ab-Zamazal from the University of Economics Prague for the helpful suggestion provided during this research.

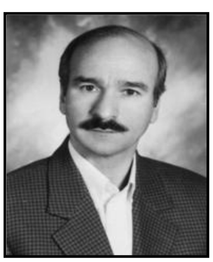
References

[1] J. Euzenat and P. Shvaiko, "Ontology matching", Berlin: Springer, **(2007)**, pp. 51-54.
[2] Y. R. Jean-Marya, E. P. Shironoshitaa and M. R. Kabuka, "Ontology matching with semantic verification", Journal of Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, **(2009)**, pp. 235-251.
[3] V. Mascardi, A. Locoro and P. Rosso, "Automatic ontology matching via upper ontologies: A systematic evaluation", IEEE Transactions on Knowledge and Data Engineering, vol. 22, **(2009)**, pp. 609-623.
[4] D. A. Ostrowski, "Ontology Refactoring", Proceedings of the IEEE International Conference on Semantic Computing, **(2008)**, pp. 476-479.
[5] J. Li, J. Tang, Y. Li and Q. Luo, "RiMOM: A Dynamic Multistrategy Ontology Alignment Framework", IEEE Transactions on Knowledge and Data Engineering, vol. 21, **(2009)**, pp. 1218-1232.
[6] W. Hu and Y. Qu, "Falcon-AO: A practical ontology matching system", Journal of Web Semantics: Science, Services and Agents on the World Wide Web, vol. 6, **(2008)**, pp. 237-239.
[7] F. Giunchiglia, P. Shvaiko and M. Yatskevich, "S-Match: an algorithm and implementation of semantic matching", Proceedings of the first European Semantic Web Symposium (ESWS), **(2004)**, pp. 61-75.
[8] A. Doan, J. Madhavan, P. Domingos and A. Halevy, "Ontology matching: a machine learning approach", Handbook on Ontologies,Berlin: Springer-Verlag, **(2004)**, pp. 385-404.
[9] N. F. Noy and M. A. Musen, "PROMPT: algorithm and tool for automated ontology merging and alignment", Proceedings of the 17th International Conference on Artificial Intelligence (AAAI), **(2000)**, pp. 450-455.
[10] N. F. Noy and M. A. Musen, "Anchor-PROMPT: using non-local context for semantic matching", In Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI), **(2001)**, pp. 63-70.
[11] O. Sváb-Zamazal, V. Svátek, C. Meilicke and H. Stuckenschmidt, "Testing the impact of pattern-based ontology refactoring on ontology matching results", Proceedings of the ISWC 2008 Workshop on Ontology Matchin, **(2008)**, pp. 229-233.
[12] O. Svab-Zamazal and V. Svatek, "Analysing Ontological Structures through Name Pattern Tracking", Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management, Springer LNCS, **(2008)**, pp. 213-228.
[13] O. Svab-Zamazal and V. Svatek, "Towards Ontology Matching via Pattern-Based Detection of Semantic Structures in OWL Ontologies", Proceedings of the Znalosti Czecho-Slovak Knowledge Technology conference, **(2009)**.
[14] O. Svab-Zamazal and V. Svatek, "Tracking Name Patterns in OWL Ontologies", Proceedings of the 5th International EON Workshop Located at the 6th International Semantic Web Conference (ISWC), **(2007)**, pp. 61-70.
[15] P. Clark, J. Thompson and B. W. Porter, "Knowledge patterns", Proceedings of the international Conference on Principles of Knowledge Representation and Reasoning (KR2000), **(2000)**, pp. 591-600.

[16] J. Baumeister and D. Seipel, “Verification and Refactoring of Ontologies with Rules”, Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW), **(2006)**, pp. 82-95.
[17] J. Reutelshoefer, J. Baumeister and F. Puppe, “A Data Structure for the Refactoring of Multimodal Knowledge”, Proceedings of the 5th Workshop on Knowledge Engineering and Software Engineering, **(2009)**, pp. 33-34.
[18] J. C. Caralt, “Ontology-Driven Information systems: Pruning and Refactoring of Ontologies”, Proceedings of the Doctoral Syposium of 7 th International Conference on the Unified Modeling Language, **(2004)**.
[19] J. Baumeister, J. Reutelshoefer, F. Haupt and K. Nadrowski, “Capture and Refactoring in KnowledgeWikis”, Proceedings of the 2nd Workshop on Scientific Communities of Practice, **(2008)**.
[20] E. Kapsammer, H. Kargl, G. Kramler, T. Reiter, W. Retschitzegger and M. Wimmer, “Lifting Metamodels to Ontologies - A Step to the Semantic Integration of Modeling Languages”, Proceedings of the ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems, **(2006)**, pp. 528 - 542.
[21] N. Noy and D. L. McGuinness, “Ontology development 101: A guide to creating your first ontology”, in Technical report, Stanford Knowledge Systems Laboratory, **(2001)**.
[22] J. R. Reich, “Ontological design patterns for the integration of molecular biological information”, **(1999)**.
[23] A. Gangemi and V. Presutti, “Ontology design patterns”, Handbook on Ontologies, **(2009)**, pp. 221-243.
[24] D. Lin, “An information-theoretic definition of similarity”, in presented at Proceedings of the 15th International Conference on Machine Learning (ICML), **(1998)**, Morgan Kaufmann, San Francisco.
[25] M. Egaña, R. Stevens and E. Antezana, “Transforming the Axiomisation of Ontologies: The Ontology Pre-Processor Language”, Proceedings of the 4th International Workshop, OWL: Experiences and Directions (OWLED), **(2008)**.
[26] O. Svab, V. Svatek and H. Stuckenschmidt, “A study in empirical and `casuistic' analysis of ontology mapping results”, in In Proceedings of the 4th European conference on The Semantic Web (ESWC-07), **(2007)**, Berlin, Heidelberg: Springer-Verlag.
[27] H. -H. Do, S. Melnik and E. Rahm, “Comparison of schema matching evaluations”, In Proceedings of the 2nd Workshop on Web, Web-Services, and Database Systems, **(2002)**, pp. 221–237, Erfurt, Germany.

Authors

Bahareh Behkamal is working as a lecturer in Azad University. She received her B.S. in 2005, from the Azad University of Mashhad, Iran and her M.S in 2011 from PNU University of Tehran, Iran. Her research interests include Semantic Web and Ontology Matching.



Mahmoud Naghibzadeh received his BS degree in statistics and computer science from Ferdowsi University of Mashhad, Iran and his MS and PhD degrees in computer science and Computer Engineering, respectively, from the University of Southern California, USA. He is currently a full professor at the Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.

His research interests include process scheduling and knowledge engineering. He has published numerous papers in international journals and conference proceedings as well as eight books in the field of computer science and engineering. He has been the general chair and/or technical chair of many international computer conferences and He is technical committee member of two international journal as well as the reviewer of many journals and a member of many computer societies

including IEEE. He is the recipient of many awards including MS and PhD study scholarship and outstanding professor award.

Dr. Reza Askari Moghadam received his M.S. degree in Control Engineering and his Ph.D. degree in Electronic Engineering in 1999 and 2006 respectively. His research interests include MEMS Devices, Design of Digital Circuits & Systems, Growth & Characterization of Semiconductor Materials, Neural Networks & Fuzzy Systems, and he has published several research papers in these areas.