Classification and Comparison of Ontology Matching Systems

1Saruladha, K. and 2Arthi, J.

1Assistant Professor, Pondicherry Engineering College, Puducherry, India.
2Assistant Professor, National Institute of Technology Puduchery, Karaikal, India.

ARTICLE INFO

Article history:
Received 12 October 2014
Received in revised form 26 December 2014
Accepted 1 January 2015
Available online 25 February 2015

Keywords:
Ontology Matching
Classification
Computational Complexity
Heterogeneity

ABSTRACT

The widespread usage of ontologies has led to development of efficient techniques for matching ontologies. The matching of ontologies gains importance as it finds applications in query rewriting, information retrieval etc. Ontology matching determines the correspondences or equivalences between the concepts that are associated in distinct ontologies developed for the same domain. It aims at representing and accessing the associations between similar elements of different ontologies to create a sharable semantic space. As the ontologies developed are larger in size, finding an efficient technique to match ontologies still remains a challenge. Basically the major challenges in ontology matching techniques are heterogeneity and scalability. Apart from these, the reduction of computational complexity occupies a significant space in the matching of large ontologies. This paper provides an insight of the various techniques that are employed for matching ontologies and the associated issues. From the literature survey made the existing ontology matching techniques are classified into four different categories viz., scalability addressing techniques, machine learning techniques, search space reduction techniques and semantic web based techniques.

© 2015 AENSI Publisher All rights reserved.


INTRODUCTION

The existence of large and heterogeneous amount of data that exist in various domains has led to the need for the efficient representation, management and accessing of data. Ontologies support the task of capturing, processing, reusing and communicating knowledge. They describe individuals, classes, attributes, relations, events within a domain and their relationships. Tom Gruber defines ontology as “a formal specification of a shared conceptualization”. For the specification of a conceptualization, axioms that coerce the possible elucidations for the defined concepts becomes essential apart from reasoning about the properties of that domain. Ontologies are used so that distributed applications and systems avoid the various heterogeneity problems enabling the exchange of knowledge for the completion of more complicated tasks. In order to enable communication, ontologies must be shared. The increased usage of ontologies, has led to the development of a domain ontology in various perspectives by different designers which brought in the need for various operations with the ontologies. For the effective utilization of ontologies in a particular domain, merging of ontologies by linking with the existing ontologies may be done. Ontology integration determines the same part in two different ontologies and creates a new ontology which facilitate interoperability among systems. The main goal of ontology integration is to inherit all the available concepts, relations, entities that exist among the ontologies that are considered to facilitate concept reuse. Though there are various operations in ontologies, check for consistency among concepts has to be performed as the same concept may be portrayed in a different way by different designers. The determination of correspondences among the various conceptual structures is inherent to systems that conjoin various knowledge sources. As integration of distributed information sources is becoming automated, the need for mapping among the various information sources is increasing. Ontologies being the structural framework for organizing information requires the explicit matches among concepts in order to enable the creation of a sharable semantic space. Ontology matching is the process of finding the relations between semantically related entities of different ontologies and “alignment” termed as the result of this process expressing these relations declaratively. These correspondences may stand for equivalence, subsumption, or disjointedness, between ontology entities. The matching process also plays a major role in approaches that
depend on ontologies to solve the semantic heterogeneity problem between information systems. Basically semantic heterogeneity may categorized into the following types:

- Syntactic heterogeneity (Sources that use different knowledge representation languages)
- Terminological heterogeneity (Sources that use different vocabularies)
- Notion of modelling the identical vocabularies may be unique.

Apart from the heterogeneity problem, several other problems exist in ontology matching which include handling scalability, optimization of search space for the selection of concepts to match, reduction of computational complexity, determination of appropriate matchers and matching techniques, aggregation of similarity measures leading to uncertainty etc.,

Several techniques were proposed to handle the above mentioned problems and to create a sharable semantic space. These techniques illustrate the various methods with which the challenges in ontology matching can be handled. Due to the increased use of ontologies, there is evolution of ontologies and emergence of large ontologies which had several issues to be dealt with. Among those issues, computational complexity in ontology matching has to be addressed as the matching process has to consider all the concepts in the complex ontologies that are considered for matching. There are quite a number of reasons for the increase in computational complexity among which the reduction of search space for matching occupies a considerable space. Numerous works have focussed on achieving accurate results and high performance rather than reducing the computational complexity. A contemporary classification of the various approaches that intend to reduce the computational complexity as well as achieving better precision and recall results is portrayed.

An overall classification of the various techniques that intend to reduce computational complexity is described followed by a brief description about the approaches that aim to handle scalability problems in ontology matching, the methods that use machine learning techniques, techniques that uses semantic web for the determination of matches and the techniques that aim to reduce the search space using various other approaches. Finally, a comparison of the discussed approaches based on the reduction of the complexity is analysed and conclusion is presented.

1. **Ontology Matching Classification:**

The reduction of search space is one of the most attempted approach for large match tasks to bring down the computational complexity in order to improve efficiency which focusses on the accuracy and effectiveness that includes time and space complexity.

![Classification of ontology matching approaches focusing on reduction of computational complexity.](image)

Several authors have aimed to classify the process of ontology matching based on various factors depending on the level of classification as in (Shvaiko and Euzenat, 2005) which focuses on the element and structural level classification. (Nelson K. Y. Leung, Seung Hwan Kang, Sim Kim Lau, and Joshua Fan, 2009) proposed a three tier classification framework in which the identification of matching technique and the approaches for designing was explained. There are several other classifications that focus on one particular method as
in (Saruladha et al., 2011) which is based on techniques that deal with large scale ontology matching, based on the approach or algorithm that is used in the alignment tool as in (Ivanova and Valentina Terzieva, 2010) according to the type of mapped ontology elements as in (Tatyana Ivanova, 2010). In this paper, we aim to propose a classification that focuses on approaches that aimed to reduce the computational complexity and to achieve high precision results as presented. Fig. 1 portrays the overall classification of the matching problem which is classified into four broad categories which include scalability focussed approaches, machine learning techniques, search space reduction techniques and semantic web based approaches.

The need for handling ontology matching in scalability perspective arises as there are updations that are done to ontologies periodically due to their increased utilization. Scalability techniques basically aim in handling the ontology matching problem for large and evolving ontologies. As we consider large ontologies the processing demands more importance. Partitioning based approaches not only contributes to handling scalability but it in turn optimizes the search space by refining the concepts that are to be considered for matching. The distributed approaches in ontology matching occupy a significant space as there are many ontologies that are available for a single domain. The processing of these ontologies is being efficiently handled by these approaches.

Machine learning techniques exploits the various approaches like the neural network approaches to determine possible matches among the two ontologies. Each and every concept has to be considered in these approaches which intend to increase the complexity of the approaches. Very few works have been proposed to handle uncertainty in ontology matching but they do not focus much on the complexity that is involved. Quite a number of approaches have been proposed to reduce the search space in case of the matching problem. These techniques may initially determine a set of matches or based on the determined matches further matches may be found. This intends to increase the precision and recall results as well as reducing the complexity of the approach by optimizing the search space. The alignment results may be better refined by using semantic web as this may be a source of knowledge to obtain information regarding the concepts that are used in the ontologies. A brief analysis of the techniques is presented in the following sections.

3. Scalability Handling Techniques:

The major challenge to the field of ontology mapping is handling large ontologies. This is due to the updations that are taking place periodically. Hence scalability becomes an important factor that has to be focussed in ontology matching. Several approaches aimed to handle the scalability problem but very few methods were able to successfully handle this. Scalability in turn has to focus on handling the computational complexity as well this is because large ontologies will have to be considered in this approach. The determination of candidate matches can very much reduce the computational complexity as this reduces the search space. Scalability problems can be handled using two basic approaches that include partition based approaches and distributed approaches. Partitioning based approaches includes considering the structural proximities of the ontologies and determining the candidate concepts for matching. This has been an effective means of determining all possible matching candidates while distributed approaches mainly focus on reducing the complexity of the method.

3.1 Partitioning Based Approaches:

There are several methods that use partition based methods to handle large ontologies. These approaches tend to partition the ontologies that are to be matched and determine the matched entities. The consideration of partitions is very effective in case of large ontology matching problems as it becomes a challenge to determine the concepts that are to be matched. Anchor PROMPT (Noy, N. and M. Musen 2001) takes a set of anchors as input terms a set of related terms, anchors that are obtained as a result of lexical matching. The path in the sub graph that is determined by the anchors is considered and the classes that appear in the similar positions are treated as semantically matched entities. Falcon AO (Ningsheng Jian, W.H., Gong Cheng and Yuzhong Qu, 2005) system has higher performance when compared to other systems. It parses the ontology into models and based on the set of elementary matches that are available with which alignments having better precision and recall results are achieved. Anchor-Flood (Hanif MS and Aono M, 2009) algorithm preprocesses ontologies to normalize the textual contents of entities. The accurately matched strings are taken as anchors. Based on these anchors, pair of concepts are selected to delive into. The neighboring concepts are being gathered from the selected concepts from the ontologies that are considered and they are being aligned. Iteration of the aforesaid process is being done until either all the gathered concepts are explored, or no new aligned pair is found.

3.2 Distributed Approaches:

LOM (Li, J. 2004) takes as input two concepts from each ontology and performs whole term matching, word constituent matching, synset matching and type matching. The computational complexity is comparatively higher as all the concepts are considered for determining the matches. Naïve Ontology Mapping (NOM) (Marc Ehrig and York Sure, 2004) approach highlights individual similarity results by weighting individual similarity
results and aggregating them. Though NOM was automatic when compared to the previous approaches, it had a high computational complexity. A new version of this known as Quick Ontology Mapping (QOM) (Ehrig Staab,2004) emphasizes speed over accuracy of results. The loss in accuracy is quite marginal, yet the gain in efficiency is tremendous where often the ontology schemas can get quite large. Though QOM may be argued to be faster, the effectiveness is marginally lower. Linguistic ontology mapping (Willem Robert van Hage, Sophia Katrenko, and Guus Schreiber, 2005) chooses matching candidates by creating hypothetical relations between pairs of concepts from the datasets that were being considered. The matching candidates are chosen based on the hits to queries that are constructed based on certain patterns. IAOM (Jiangning Wu and Yonggui Wang,2007) approach integrates the linguistic structure and instance based matching and the appropriate matching candidates are chosen based on the respective weights and sigmoid function.

4. Machine Learning Techniques:

Machine learning techniques may be applied at different levels to ontology matching problems. It may be in determining the candidate mappings, in computing the similarity, in integrating the similarity measures etc.. In case of computing similarity, ontologies are considered as taxonomies similarity matches among the concepts in the nodes are computed. In case of determining the candidate mappings, identification of concepts and the dependencies among is determined which are then utilized for matching process. It is found to be very effective in case of ontology integration processes.

4.1 Techniques handling uncertainty:

Uncertainty in ontology matching refers to the concepts that cannot be termed as similar or dissimilar. The aggregation of similarity matching techniques gives rise to the problem of uncertainty. Very few works have focused on handling uncertainty in ontology matching. DSSim(Nagy et al,2006) is an approach in handling uncertainty by making use of the question answering technique which eliminates the inconsistencies and inefficient information that is present. (Y. Wang, W. Liu, and D. Bell,2007) uses the aggregated similarity matches and applied Dempster Shafer theory of evidence (DS theory) and Possibility theory to merge the results. The utilization of these theories managed to enhance the matching results. (N. Laamari and B. Ben Yaghlane,2007) uses similarity metrics to determine matches among ontologies. Uncertainty is handled by using belief functions which gives better matching results.

4.2 Other Approaches:

The GLUE system(Doan et al.,2002) exploits the various machine learning methodologies produce semantic mappings. The ontologies are considered as taxonomies and correspondences among the node concepts in taxonomies are obtained by determining the most similar components that are present in the given two taxonomies. The joint probability distribution among concepts are computed which determines if an instance belongs to a particular node concept or not. Since the machine learning techniques are being used, the complexity is being considered for the training set of samples and thus computation of the joint probability distribution contributes to the complexity of O (n²). In Manifold algorithm(Mehta and Hofmann, 2007), the machine learning techniques are applied to find correspondences among the profiles that are existing in one system with that of the other with the constraint that the profiles are to be sharable. To determine the correspondences and to perform matching, data sets are taken as points in space and manifold learning techniques are applied. It is based on non-linear dimensionality reduction which is considered as one of the fast and efficient supervised learning technique. The complexity is based on the data set that is being considered. SemInt(Li et al.,2000) is a tool that is used for finding attribute correspondences among the entities in a database. The schema design information and data contents are taken as metadata and is being considered to characterize an attribute in a particular domain. A procedure that automatically incorporates the domain knowledge from metadata is proposed which will discover matching among the data elements. Neural network approach managed to yield high precision and recall results when compared to computing the similarity using functions.

(Marc Ehrig and York Sure,2004) proposed a method in which different similarity measures are combined using intelligent approaches to get better mapping results. The mapping candidates are chosen based on rules but additional actions could not be incorporated into machine learning techniques which lead to lower results. (Dennis Hooijmaijers and Markus Stumpner,2008) extended the RIPOSTE framework that provides methods for determining matches using subjective logic (an extension of the probabilistic logic for determining ontologies).

S-match(F. Giunchiglia et al.,2010) is a framework that is used for transforming data structures like business catalogues, web directories etc., into lightweight ontologies and then finding correspondences among them. Classifier constructs the concepts in the form of a tree for determining correspondences but achieves very less precision and recall results. Augmentation techniques make use of an agent to encapsulate various matching techniques so that they may be shared and used. FOAM(M. Ehrig and Y. Sure,2005) was able to choose the
candidates for alignment using intelligent and rule based approaches. The involvement of user at each and every step was able to achieve highly efficient results. (Trojahn et al., 2010) presented a comparison of the various augmentation techniques that were available. To enhance the matching results, FOAM+ (AbdulHameed Haddad and Akram Selah, 2011), an extension of FOAM made use of the genetic algorithm to dynamically assign weights using the fitness function to the concepts so that better precision results may be achieved. YAM++ (DuyHoa Ngo and Zohra Bellahsene, 2012) is a technique that is based on the similarity flooding method which makes use of the gold standard for ontology alignment as the training dataset and aims to find matches in ontologies.

The machine learning techniques that were proposed, aimed to achieve better results rather than their aim to determine new matches or reduce the search space hence these approaches were not able to reduce the computational complexity to a much greater extent and achieved good precision results.

5. Semantic Web Based Techniques:

Ontology matching has gained much importance as several ontologies are being developed for the same domain due to its increased usage. Ontology matching becomes a much more challenging task when we intend to find alignments among ontologies of the same domain. The effective usage of semantic web as said in (Euzenat et al., 2007) tends to bring in better recall results. The approaches that use semantic web to determine effective matches is discussed in this section.

(M. Sabou, M. d’Aquin, and E. Motta, 2008), (R. Vazquez and N. Swoboda, 2007), (L. Po and S. Bergamaschi, 2010) proposed techniques that makes use of the semantic web in order to determine new matches based on the already existing matches. By using semantic web as the background, online knowledge sources may be utilized to derive mappings. These techniques managed to achieve promising results in terms of precision and recall. iMapper(Xu et al., 2004) technique enriches concepts with its description so that it can be termed semantically rich so that accurate conceptualization may be achieved. The concept enrichment is being done with the help of a classifier which results in semantically enriched ontologies which is being mapped with the help of vector space model. SILAS - Simple Instance-based Library-thesaurus Alignment System (Roelant Ossewaarde, 2007) is based on instance based matching technique in which the terms that are to be matched are converted into search terms. Using automatic spiders, they are being tagged to the other ontology with which the correspondences are determined. (B.A.C. Schopman, S. Wang, and S. Schlobach, 2008) exploited the instance based methods by using a lexical search engine to map instances from different ontologies. The main advantage of this method is that it does not depend on the concept labels as it may not be efficient in case of multi lingual ontologies and does not depend on the structure which may not always lead to accurate results. ServOMap(M. Ba and G. Diallo, 2013) is a technique that is designed for facilitating real time interoperability between different applications. The ontologies involved are considered as documents and inverted indices are created. The dynamic description process that is involved, considers the concept and performs the normalization process and computes the lexical and conceptual similarity which are further refined. (T. Groza, A. Zankl, and J. Hunter, 2012) focus on the semantic annotation process and mapping among the entities involved. The lexicon is being created and optimized with which the similarity matrix is being computed and appropriate matches are determined.

6. Search Space Reduction Techniques:

Search space optimization is a major problem that has to be handled in case of the ontology matching as this indirectly contributes to the reduction of complexity in the process of ontology matching. There are various approaches that focus on the selection of matching candidates based on initial matches and approaches that intend to find newer matches based on the matches that are determined. The approaches that focus on the reduction of search space using single and multiple matches are discussed.

6.1 Search Space Optimization Using Single Matches:

Chimera (DLMcGuinness et al., 2000) is a browser based editing, merging and diagnosis tool which solves mismatches at the terminological level in a very light way. Chimera finds linguistic matches and it takes subclass and superclass relationship into account for determining alignments by means of edit points. It is based on PROMPT (Noy and Musen, 2003) approach which takes two ontologies as input creating an initial list of matches which forms a coherent ontology. There is computation of the string matches in which choice is given to user to determine the matches. (Siddharth Kaza and Hsinchun Chen, 2008) proposed the MIM mapper technique which determines matches in two steps which include the lexicon driven and data driven approach. MIM mapper managed to achieve an f-measure 0.84 on compared to several other techniques. (P. Ceravolo et al., 2008) proposed the ODDI supports various kinds of matching operations that are based on relationships involved as they define the semantics of the matching operators. The relationship operators are allocated with weights based on which the matching candidates are chosen and matches are being determined. (Z. Eidoon, N. Yazdani, and F. Oroumchian, 2008) modelled the ontologies in a vector space and the similarity comparisons
were based on the concept vectors. The properties are considered in order to represent the domain and range achieving promising results in terms of precision and recall. RiMOM(Li et al.,2009) technique utilizes certain ontology features for which candidate matches are determined based on a set of strategies. The strategies compute the weights and executed independently. (H. Liu et al.,2010) consider a vector of values that are transformed into a point sequence curve for which the similarity computation across datasets are done to determine mappings. The employment of clustering technique to group the set of similar concepts in the datasets tend to increase the efficiency of the results obtained. (B. Fu, R. Brennan, and D. O’Sullivan,2011) extended the cross lingual ontology mapping technique in order to eliminate the mismatches that may arise due to inefficient translations. The pseudo feedback method, based on the relevance feedback mechanism checks the correctness of the determined matches resulting from the AOLT process and presents the most effective translation media.

Table 1: Comparison of various approaches.

<table>
<thead>
<tr>
<th>Feature Selection</th>
<th>Selection of Candidate Matches</th>
<th>Selection of New Matches</th>
<th>Similarity Computation</th>
<th>Similarity Aggregation</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scalability Handling Techniques</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Partition based approaches</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anchor PROMPT</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Falcon AO</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fast and accurate alignment</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Distributed approaches</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOM</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>LOM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>QOM</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>IAOM</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Machine Learning Techniques</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Techniques handling uncertainty</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OWL-CM</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DS&amp;PT</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Other approaches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLUE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Augmentation Framework</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FOAM</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>FOAM++</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Manifold</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RIPOSTE</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SemInt</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>S-Match</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>YAM++</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Search Space Reduction Techniques</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search space optimization using single matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linguistic matching</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>PROPMPT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>MIM Mapper</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>ODDI</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pseudo feedback</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chimera</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Similarity Flooding</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ontology based mining of brainwaves</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VBOM</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SOCOM</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Search space optimization using multiple matches</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLMA+</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scarlet matcher</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SSC&amp;DSI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Semantic Web Based Techniques</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SILAS</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ServoMap</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Instance Mapping</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>iMapper</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
6.2 Search Space Optimization Using Multiple Matches:

OMEN(Mitra et al.,2005), Ontology Mapping Enhancer is a tool that utilizes a set of meta-rules which captures the ontology structure and semantics of relations and matches the nodes that are neighbors of the already matched nodes. The matching is done by inferring the matches from the already available matches. W. (Sunna and I.F. Cruz,2007) proposed a method in which ontology alignment takes single concept as input at a time, several concepts or the data associated with the concepts as inputs. The structure of the ontologies that are considered are taken for alignment purpose. (A. Alasoud, V. Haarslev, and N. Shiri,2008) proposed the MLMA+ technique which uses multilevel matching technique utilizing neighbour based search to find newer correspondences among entities in the ontology. Initially a set of entities are chosen to which neighbours are determined and matching process is applied to it and it achieves high quality results. (M. Kalaev, V. Bafna, and R. Sharan,2008) proposed an alignment based framework used heuristic methods to merge graphs for finding alignments. This method managed to have reduced run time and memory requirements.

7. Comparison:

(Marc Ehrig and Steffen Staab,2004) analysed the factors that greatly influence the computational complexity of ontology matching techniques. The factors have been extended and a comparison for various approaches has been presented. The below comparison is based on the various factors that contribute to the computational complexity in the discussed approaches. As the search space optimization occupies a considerable space, it is being considered as one of the factor that contributes to the computational complexity measure.

8. Conclusion:

The matching technique in ontology plays a very important role for handling interoperability problems. The increased use of ontologies has led to large ontologies in various domains and so computational complexity is one of the major issues that has to be addressed. Various methods can be used to reduce the computational complexity in regard to the numerous problems that exist with the ontology matching. A classification based on the reduction of computational complexity as well as those that yield high precision and recall results have been presented. A comparison based on the factors that contribute to the increase in complexity is analysed for the existing approaches is portrayed. It is found that uncertainty is one of the problems that are less focussed in case of the ontology matching. It is observed that adapting methods to handle uncertainty may yield better and accurate matching results. We intend to handle this issue in our future works.

REFERENCES


