Journal of Engineering and Applied Sciences 12 (Special Issue 10): 8813-8818, 2017
ISSN: 1816-949X

Critical Review on Graph-Based Ontology Matching Model

Muhammad Mahmud, Azman Yasin and Mazni Omar
Human-Centered Computing Research Lab., School of Computing,
Universiti Utara Malaysia (UUM), 06100 Sintok, Malaysia

Abstract: A deep review has been done in analysing graph-based ontology matching. Several researches have been selected for reviews which are extended Falcon-AO, YAM++, Shiva++, iMatch and Affinity-Preserving Random Walks Model (APRWM). The selection criteria are based on state of the art graph-based ontology matching and able to cater for large scale ontologies. A comparison table has been presented at the end of this study.

Key words: Ontology matching, graph-based, affinity-preserving random walks model, extended Falcon-AO, YAM++, Shiva++, iMatch

INTRODUCTION

In designing ontologies, heterogeneity issue arises due to different interpretation from the ontology engineers in term of terms, depth or area of coverage (Jauro, 2014). The significant issue within semantic heterogeneity is semantic ambiguity in which refers to numerous intended meanings are associated with the same word (Gracia and Mena, 2012). Semantic ambiguity is addressed through ontology matching by discovering similarities between terms from different ontologies.

One of the techniques implemented is graph-based ontology matching. This technique is focusing on the structure-level of the ontology rather than the element. Compared with element-level matchers, structure-level matchers seek mapping by analysing how entities are represented structurally while element-matchers analyse the similarities individually without considering the relationships with other entities.

Graph matching ontology is a technique where a set of ontology input is processed as a labelled graph. The ontology file which is commonly in RDF/OWL naturally can be constructed in graph format (Guizzardi *et al*., 2004). In this way, the correspondence of the ontology can be analyse in a structural manner.

Once these ontologies have been constructed in a graph form, structurally it will consist of terms and their inter-relationships. The matching process will compute the pair of nodes similarities by referring the positions within the graphs. The basis is that if both nodes share the same similarity, the next neighbours also has the possibility to be the same.

Over the years there have been many development and improvement on graph-based ontology matching technique. This study will explore and analyse the state of the art on recent models for this technique. The selected models are extended Falcon-AO, YAM++, Shiva++, iMatch and affinity-preserving random walks model.

Literature review: In general, based on observation and review done the process of ontology matching technique consists of three steps which are parsing, matching and output alignment. Based on these main steps different configuration on each of them are being made depending on the variation of the algorithm designed.

Parsing module involves extracting ontologie's representation such as the concepts, sub-concepts, properties and instances (Mathur *et al*., 2014). Most of the parser can be easily adopted from existing model such as Falcon-AO which is using Jena API to parse the input ontologies.

The most crucial part for ontology matching technique is the matching module. In here, all of the matching process will be executed in order to seek the best similarities between both ontologies. Each ontology matching will have variant in the matching process. As an example Yet Another Matcher (YAM) is providing multiple machine learning based combination such as decision tree, support vector machine and Naive Bayes for user to select during the element-level matching. In addition, YAM is applying similarity propagation method for structural-level matching process (Ngo *et al*., 2012).

Finally, based on the matching result obtained, the alignment between both ontologies will be made. Normally, the quality of an ontology matching technique will be evaluated in term of precision, recall and F-measure.

Corresponding Author: Muhammad Mahmud, Human-Centered Computing Research Lab., School of Computing, Universiti Utara Malaysia (UUM), 06100 Sintok, Malaysia

MATERIALS AND METHODS

Graph-based ontology matching models: Several graph-based ontology matching model have been selected for review in this study which are extended Falcon-AO, YAM++, Shiva++, iMatch and affinity-preserving random walks model.

Extended Falcon-AO: Falcon-AO is a fully automated ontology matching model which combines linguistic and structural matching techniques. Linguistic matcher seeks similarities in term of the names, labels, comments and other additional descriptions within the ontology. As explained before structural alignment is established through analysing the occurrences of built-in properties which are subject, predicate and object of the involved statements.

This model provides the capability to cater for large scale ontologies with its own novel matcher libraries. Figure 1 shows the matcher libraries consists of linguistic matcher V-Doc and I-SUB while the structural matcher is GMO and PGM.

Based on observation and reviews, Falcon-AO has gone through two stages of improvement from the initial version. Jauro (2014) from her research has added user restrictions module in order to involve the contribution of a domain expert in the matching process. Even though this increased the result quality it makes the matching process semi-automated due to expert reliance step (Jauro, 2014).

Currently, Falcon-AO has been extended with lexical database (Alhassan *et al*., 2015). Compared with Jauro (2014), it does not include the user restrictions module but provide her own synonym search module.

In this module, Falcon-AO will search for synonymous terms between ontologies by using WordNet which is a lexical knowledge source. Even though it impacted in low execution time overhead, the results display more precision rate compared with the original model.

Falcon-AO contains its own novel graph matching technique in which it constructs bipartite graph from the normal graph method. Bipartite graph has a proper algorithm for human data visualization (Hayes and Gutierrez, 2004). In this way, the relationships between entities in ontology can be depicted in a better way compared with normal graph representation. Therefore, better result is produced for the ontology matching process.

Yet Another Matcher (YAM++): Yet, Another Matcher (YAM) is an ontology matching model which exploits machine learning technique to combine multiple similarity metrics 10. This model has been extended with latest called YAM++ which provides matching process without using machine learning in case learning data are not available. To achieve this, ontologies intrinsic textual features are used to find the alignment. In addition, from the structural and semantic information, similarity propagation method and semantic verification module are developed to search and refine the alignments 6. Figure 2 shows depict the model's architecture.

Compared with Falcon-AO Model, YAM offers semantic matching on top of element and structural

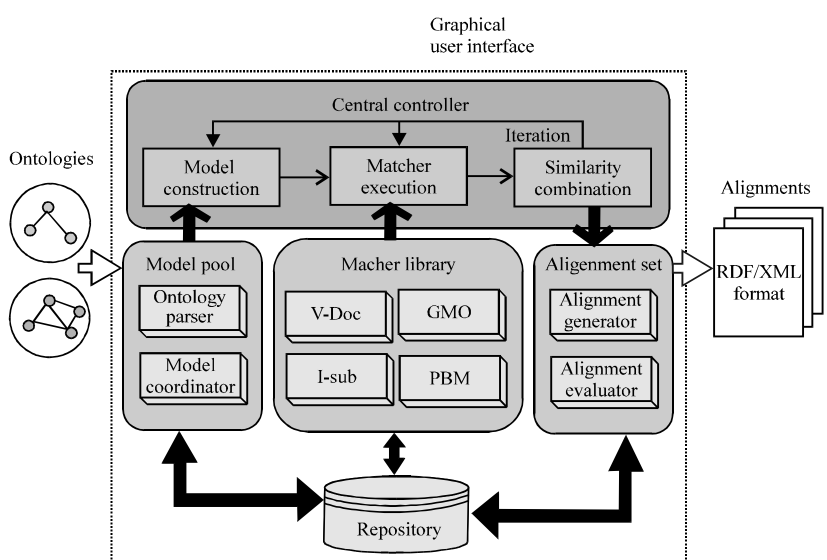

Fig. 1: Falcon-AO system architecture

matching. In this way, any inconsistent mappings from both element and structural level are revised and removed. Other than that for graph matching process, it is applying similarity propagation method compared to Falcon-AO which is using bipartite graph algorithm. Alternatively, YAM provides a flexibility to ignore machine learning technique when related benchmark data sets are not available by utilizing information retrieval techniques. In addition, YAM has the capability for multi-lingual ontology matching other than English. Basically, the language is determined by reading annotations of entities and later all labels are translated by using microsoft bing translator tool.

Shiva++: Similar with YAM++, Shiva++ is an ontology matching model that performs matching on the element, structure and semantic-level in order to obtain better result. This model still applying bipartite graph matching technique same with Falcon-AO described before. This model has been enhanced from previous model Shiva which only focusing on string or element-level matching (Mathur *et al*., 2014).

Shiva++ provides the flexibility to select four string matching algorithms which are Levensthein edit distance, Qgrams, Smith Waterman and Jaccards's coefficient. Any exceptional elements that are not matched by the algorithm will be taken care by WordNet for semantic similarity. Hungarian or Kuhn Mukres algorithm is applied in the bipartite graph score matrix computation (Fig. 3). The result has shown a slight better result in term of precision, recall and F-measure compared with its predecessor.

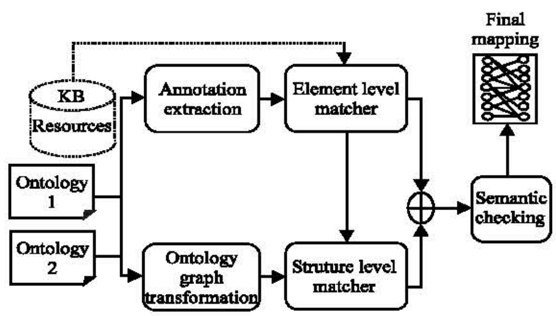

Fig. 2: YAM++ system architecture (Ngo *et al*., 2012)

iMatch: Another graph-based ontology matching model developed is called iMatch. iMatch applies Markov networks in which suitable in catering complex computation by using approximate reasoning. In addition, the alignment process is on data basis rather than involving domain expert.

Compared with extended Falcon-AO, YAM++ and Shiva++, iMatch offers user interaction in which user is asked to accept or reject processed matching candidates. Based on the selection the result will be updated accordingly.

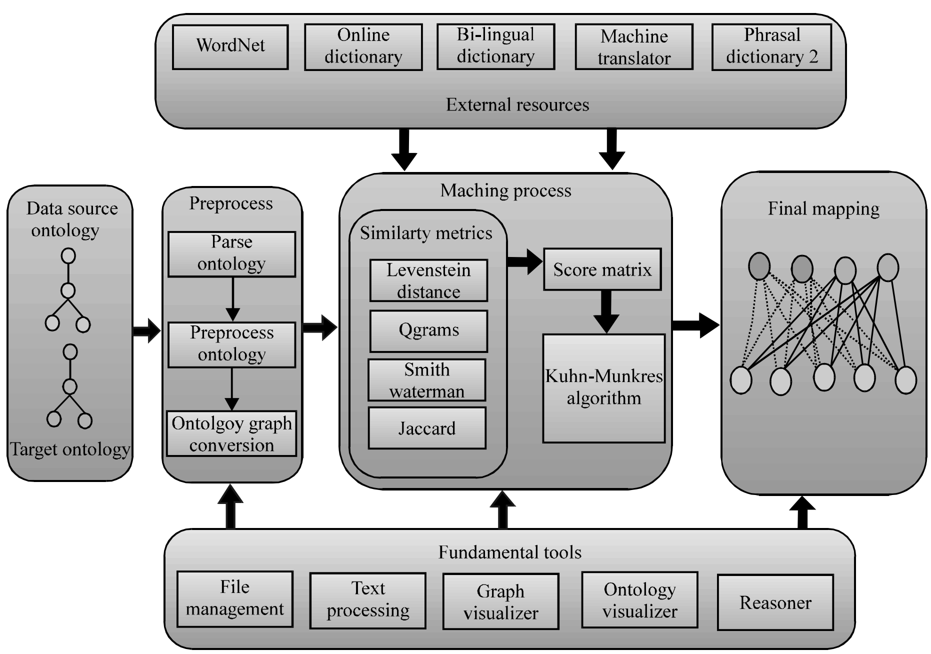

Fig. 3: Shiva++ system architecture (Mathur *et al*., 2014)

Based on the experiment result performed against other matching models, basically, iMatch perform average in term of precision and recall. However, iMatch manage to outperform other matchers for F-measure. iMatch can be further improved by including linguistic matching process.

Affinity-Preserving Random Walks Model (APRWM): This model developed its own matching process based on Markov random walks algorithm. This algorithm originally used to compute web pages ranking. Comparing with other models described before, based on review APRWM solely focusing on structural-level matching.

Based on the experiment result, APRWM outperforms most of the other models in term of F-measure. However, one of other model that outperforms APRWM, MapSSS is utilizing external resources as an example Google queries hence yielding better result.

Referring to the result, APRWM has shown a significant performance. However, this model could be improved to process alignment on the element and semantic-level.

RESULTS AND DISCUSSION

Table 1 shows the comparison between the selected models to be reviewed which are extended Falcon-AO, YAM++, Shiva++, iMatch and APRWM Model. The maturity column indicates in continuation the evolution of the model enhanced over the years. Most of the system selected here has gone through several improvements from the initial version. The ontology matching here can consist of three types of matching technique which are element, structure and semantic level. Different system will have various techniques to perform the matching process.

The experiment setup between all systems is different because of different data used. However most of them are using Ontology Alignment Evaluation Initiative (OAEI) data set such as Benchmark and Conference track. Due to the number of ontology matching increasing over the years, a standard data set is essential to be used for evaluation. OAEI is providing a standard data set as a benchmark for testing. It consists of data sets designed from reference ontologies of different sizes and domains. The conference track consists of 16 ontologies based on conference organization. It is commonly used for ontology matching evaluation due to their heterogeneous character of origin.

In this review even though the data set are similar however the criteria for each experiment are dissimilar such as Shiva++ in which tested lightweight and heavyweight ontologies. Lightweight ontologies consist of small numbers of entities involves while heavyweight ontologies contain a complex number of ontologies. Other than that iMatch evaluate its model by having the

Table 1: Comparison table between selected models

Model names and researchers	Maturity	Element- level matching technique	Structural-level matching technique	Semantic matching technique	Result: Experiment setup	Precision	Recall	F-measure
Extended Falcon-AO Alhassan *et al.* (2015)	Jian *et al.* Jauro *et al.* Alhassan *et al.* (2015)	V-DOC I-sub	GMO PBM	-	6 sets of OAEI conference track ontologies	0.56845960	-	-
YAM++ Ngo *et al.* (2012)	Duchateau *et al.* Ngo *et al.* (2012)	Multiple machine learning based combination methods Decision tree Support vector Naive Bayes	Similarity propagation method	Global constraint optimization	OAEI 2011 Benchmark and conference tracks data set	Benchmark Conference: e: 0.79	Benchmark Conference: e: 0.55	Benchmark: 0.83 Conference: 0.66
Shiva++ Mathur *et al.* (2014)	Mathur *et al.* (2014)	Multiple string matching algorithm Levensthein Edit distance Qgrams Smith waterman Jaccard's coefficient	Hungarian method	WordNet	OAEI 2013 evaluation task including anatomy track data set (lightweight and heavyweight ontologies)	Lightweight: 0.92021975 Heavyweight: 1	Lightweight: 0.49182442 Heavyweight: 0.86604625	Lightweight: 0.86228542 Heavyweight: 0.927884
iMatch Albagli *et al.*	-	-	Markov network algorithm	-	OAEI conference track data set	Threshold 0.5: low precision Threshold 0.7: low precision	Threshold 0.5: Outperform other systems Threshold 0.7: low threshold	Threshold 0.5: Outperform other systems Threshold 0.7: Outperform other systems

Table 1: Continue

Model names and researchers	Maturity	Element- level matching technique	Structural-level matching technique	Semantic matching technique	Result Experiment setup	Precision	Recall	F-measure
APRWM Sui *et al*.	-	-	Markov random walk algorithm	-	OAEI 2012 and 2013 benchmark biblio data set	-		OAEI 2012: Outperform other systems except for MapSSS OAEI 2013: Outperform other systems except for YAM++ and CroMatcher

threshold input of 0.5 and 0.7. These thresholds are user defined in order to match thresholds reported by another existing model being tested. In addition, some of the experiments done did not completely evaluate the precision, recall or F-measure rate. As an example for, extended Falcon-AO only precision rate are being evaluated while APRWM only considers F-measure as the performance metrics to be compared with other existing models.

In Table 1 are the summary of the experiment results conducted for each model. However, the results here are computed by average for easiness and not depicted in detail. In term of precision, Shiva++ displays the best result for lightweight and heavyweight ontologies which are 92.03 and 100%, respectively. However, for recall Shiva++ still has a low result for lightweight but higher for heavy weight ontologies. iMatch displays high recall for a 0.5 threshold. Most selected model for review here has a high rate of F-measure. Based on the review and analysis done Shiva++ displays the best result in term of precision and F-measure. The recall rate for this model can be further enhanced in order to achieve a balance result with the precision and F-measure rates.

CONCLUSION

This study is conducted to with the purpose to critically review on available graph based ontology matching models. In order to achieve this objective, several ontology matching models has been selected to be analysed which are extended Falcon-AO, YAM++, Shiva++, iMatch and affinity-preserving random walks model. These models have been selected due to the nature of the matching process which is using structured-based matching technique. All these models are compared with each other in term of precision, recall and F-measure. At the end Shiva++ has been considered the best graph-based ontology matching model but can be further improved to achieve higher F-measure rate.

RECOMMENDATIONS

Based on this study done, future research that can be conducted for the ontology matching model here is to include semantic matching technique. Most of the models here only process ontologies on the element and structural level. In this way, the entities and relationships in the ontology can be analysed on the semantic level hence producing higher accuracy rate and better result.

ACKNOWLEDGEMENTS

This study is funded by “A New Holistic Service Oriented Model and Sustainable Flood Management: From Monitoring to Emergency Response” under Long Term Research Grant Scheme (Grant ISO No. 12491) from Ministry of Education Malaysia.

REFERENCES

Alhassan, B.B., S.B. Junaidu and A. Obiniyi, 2015. Extending an ontology alignment system with a lexical database. Sci. Res. J., 3: 12-17.

Gracia, J. and E. Mena, 2012. Semantic heterogeneity issues on the web. IEEE. Internet Comput., 16: 60-67.

Guizzardi, G., G. Wagner and H. Herre, 2004. On the foundations of uml as an ontology representation language. Proceedings of the 14th International Workshop on European Knowledge Acquisition Vol. 3257, October 5-8, 2004, Springer, Whittlebury Hall, UK, pp: 47-62.

Hayes, J. and C. Gutierrez, 2004. Bipartite graphs as intermediate model for RDF. Proceedings of the 3rd International Conference on Semantic Web Vol. 3298, November 7-11, 2004, Springer, Hiroshima, Japan, pp: 47-61.

Jauro, F., 2014. Design and implementation of Falcon-AO++: An improved ontology alignment system. Ph.D Thesis, Ahmadu Bello University, Zaria, Nigeria.

Mathur, I., N. Joshi, H. Darbari and A. Kumar, 2014. Shiva++: An enhanced graph based ontology matcher. Intl. J. Comput. Appl., 92: 30-34.

Ngo, D., Z. Bellahsene and R. Coletta, 2012. Yam++-a combination of graph matching and machine learning approach to ontology alignment task. J. Web Semant., 2012: 1-16.