KOSIMap: Ontology alignments results for OAEI 2009

Quentin Reul[1] and Jeff Z. Pan[2]

[1] VUB STARLab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2] University of Aberdeen, Aberdeen AB24 3FX, UK
Quentin.Reul@vub.ac.be; jeff.z.pan@abdn.ac.uk

Abstract. Ontology mapping has been recognised as an important approach to identifying similar information in heterogeneous ontologies. The Knowledge Organisation System Implicit Mapping (KOSIMap) approach relies on DL reasoning (i) to extract background knowledge about every entity, and (ii) to remove inappropriate correspondences from an alignment. The main assumption is that the use of this background knowledge reduces erroneous mappings, thus increasing coverage. In this paper, we provide an overview of KOSIMap, and present the result of our system for its first participation to the Ontology Alignment Evaluation Initiative (OAEI).

1 Presentation of KOSIMap

Ontology mapping has been recognised as an important means to identify similar information in different ontologies, thus achieving semantic interoperability on the Web. Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, the task of mapping one ontology to another is that of finding an entity in $\mathcal{O}_1$ that matches an entity in $\mathcal{O}_2$ based on their intended meaning. Many approaches to schema/ontology matching have been proposed over the years [5, 7, 10]. Furthermore, surveys reviewing these approaches, techniques and tools have been provided [4, 1]. The Knowledge Organisation System Implicit Mapping (KOSIMap) approach differs from existing approaches by relying on DL reasoning (i) to extract background knowledge about every entity, and (ii) to remove inappropriate correspondences from an alignment.

1.1 State, Purpose, General Statement

KOSIMap is an extensional and asymmetric matching approach implemented in Java. Given two consistent ontologies, KOSIMap aligns entities in the source ontology to entities in the target ontology by extracting background knowledge about entities based on DL reasoning. More specifically, a DL reasoner (e.g. FaCT++ [12], Pellet [9]) deduces logical consequences about an entity based on the asserted axioms defined in an ontology. Moreover, we investigate the use of DL reasoning to remove inappropriate correspondences from an alignment. The main assumption is that the use of these logical consequences reduces erroneous mappings, thus increasing coverage.

The current KOSIMap implementation of produces a set of *homogeneous* correspondences, where classes are mapped to classes. object properties to object properties, and datatype properties to datatype properties. More specifically, the approach computes the similarity between two entities based on their respective sets of features (e.g. subsumption). Note that KOSIMap only considers the equivalence mapping relation between two entities.

1.2 Specific Techniques Used

The KOSIMap system calculates the similarity between entities for a pair of ontologies by analysing three features; namely lexical description (i.e. label), hierarchical structure (subsumers for concepts, and super-properties), and internal structure (inherited properties for classes, domains and ranges for object properties, and domains for datatype properties). The measures obtained by comparing these three features are then combined into a single value using a weighted sum in a similar manner to [2]. These weights are set by a user depending on the input ontologies, and requirements for the output.

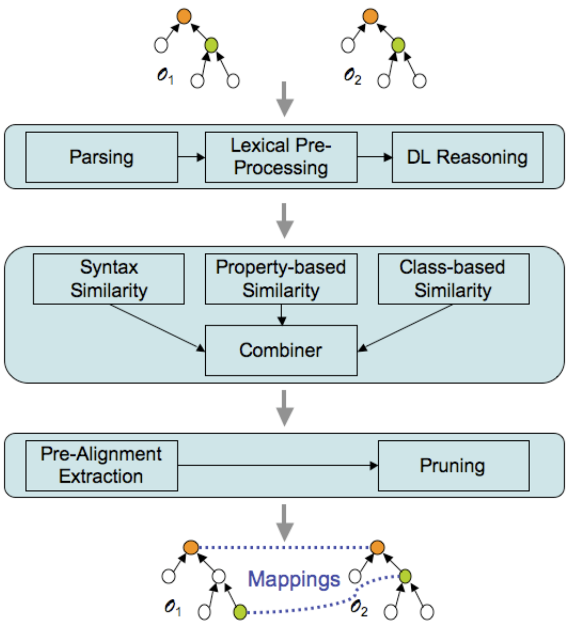

Fig. 1. The architecture of KOSIMap.

Figure 1 shows the architecture of our approach. KOSIMap consists of three main steps; namely *Pre-Processing*, *Similarity Generation*, and *Alignment Ex-*

traction. The pre-processing step includes three sub-tasks. It first parses the two ontologies with the OWL API [6]. The OWL API provides an interface to access the explicit information for each entity defined in an ontology. The API supports several representations including XML/RDF, KRSS and OBO flat files. Secondly, natural language techniques (i.e. *elimination*, *lemmatization*, and *transformation*) are applied to each entity to obtain their most basic form. Entities are not only defined by annotation properties, but also by the semantics provided by the axioms in the ontology. Thus, the final pre-processing sub-task extracts logical consequences (i.e. background information) resulting from asserted axioms. The current implementation uses the FaCT++ API[3] to classify the different ontologies.

Definition 1 (Degree of Commonality Coefficient). *Given two sets* $\mathtt{S}_s$ *and* $\mathtt{S}_t$*, the degree of commonality coefficient between them, denoted DoCCoeff(*$\mathtt{S}_s$*,* $\mathtt{S}_t$*) is defined as:*

$$DoCCoeff(S_s, S_t) = \frac{1}{\max(|S_s|, |S_t|)} \sum_{e_i \in S_s} \max_{e_j \in S_t} sim(e_i, e_j) \tag{1}$$

where S_s *is the source set,* S_t *is the target set, and* $sim(e_i, e_j)$ *computes the similarity between pair of elements in the two sets.*

Secondly, the similarity generator computes three kinds of similarities; namely *syntax similarity*, *property-based similarity*, and *class-based similarity*. The most basic feature of entities is their labels. Labels are human identifiers (i.e. words) expressed in a vocabulary shared by experts in the same domain. Therefore, we assume that equivalent classes are likely to be modelled using similar labels (or names). KOSIMap relies on string similarity (e.g. Jaro-Winkler [14], Q-Gram [11], Monge-Elkan [8], and SMOA [10]) to calculate the label similarity for each pair of entities. The SimMetrics API[4] provides a library of normalised and optimised similarity (or distance) metrics. The property-based similarity and the class-based similarity both rely on the *degree of commonality coefficient* (Definition 1) to provide an similarity value between two sets of complex objects. The property-based similarity focuses on features containing properties (i.e. set of super-properties for `OWLObjectProperty` and `OWLDataProperty` and the set of inherited properties for `OWLClass`), while the class-based similarity focuses on features containing classes (i.e. set of subsumers for `OWLClass` and the set of binary relation containing their domain and range for `OWLObjectProperty`). The results of the different similarity approaches are then aggregated for each pair of entities and stored into a n*m matrix, where is the number of element in the source ontology and m is the number of elements in the target ontology. The aggregated similarity score for a pair of entities is obtained by applying a weighted function (see Equation 2), where the weights (i.e. w_k) for each measure

[3] `http://code.google.com/p/factplusplus/`

[4] `http://sourceforge.net/projects/simmetrics/`

is in the range [0,1] and their total is 1.

$$sim(e_1, e_2) = \sum_{k=0}^{n} w_k sim_k(e_1, e_2) \quad (2)$$

After the similarity aggregation, we have a n*m matrix containing pairs of entities with a similarity value. The problem is to extract a set of relevant mappings from the matrix. This is normally achieved by discarding all candidate mappings below a threshold ζ. However, this method may return multiple mappings for each entity in the source ontology. In KOSIMap, we follow a two-step approach to extract mappings. First, the approach extracts a pre-alignment from the matrix, by selecting the maximum similarity score for each row in the matrix (i.e. for each n). This pre-alignment is then passed through a refinement process, which eliminates inappropriate mappings. In KOSIMap, we use DL reasoning to extract the local implication as part of the mapping extraction process. This approach extends the work by Wang and Xu [13], which only checked whether local implications were asserted in an ontology. As our approach only supports equivalent mapping relations, we focus on removing *inconsistent* mappings from the pre-alignment. Inconsistent mappings occur when the local consistency of an ontology is violated by the introduction of a correspondence between two ontologies. For example, an local inconsistency would occur if several entities in the source ontology are mapped to the same entity in the target ontology, and that the two classes are not recognised as equivalent by a DL reasoner.

1.3 Adaptations Made for the Evaluation

As stated in Section 1.1, KOSIMap is an asymmetric matching approach. The asymmetricity results from Equation 1, which consider the maximum value for each element in the source set. However, the organisers of OAEI campaign requested that we delivered a symmetric set of alignments. As a result, we modify the similarity generation for the property-based and class-based similarity to consider the biggest set as the source set. Moreover, we implemented a Java class to run the different tracks in batch mode. Moreover, the parameters taken by the approach (i.e. weights and thresholds) were tuned and set depending on the type of information contained in the ontologies to be mapped. For example, the property-based similarity was not calculated for the directory track as no properties were defined.

1.4 Link to the Set of Provided Alignments (in align format)

The results of the 2009 OAEI campaign for the KOSIMap system can be found at `http://www.csd.abdn.ac.uk/~qreul/research/OAEI2009.zip`.

2 Results

In this section, we present the results of the 2009 OAEI campaign obtained by the KOSIMap system. KOSIMap was used to generate alignments for four tracks, namely benchmark, anatomy, conference and directory. Note that the full results of the Alignment Evaluation Initiative (OAEI) 2009 Campaign can be found in [3]. The experiments were carried on a Mac Book with an Intel Core 2 Duo processor (2.13GHz) and 4GB RAM running Mac OSX. The minimum memory for the Java Virtual Machine was set to 512MB, while its maximum was set to 1GB. In this experiment, we used FaCT++ as the default DL reasoner unless stated otherwise.

2.1 Benchmark

The benchmark data set contains 111 alignment tasks. KOSIMap follows the approach defined in Section 1.2. In this experiment, we used the Q-Gram similarity measure to compute the syntax similarity and as the similarity function for the degree of commonality coefficient. The FaCT++ reasoner returned an exception (`NonSimpleRoleInNumberRestrictionException`) for some tests (i.e. 222, 230, 237, 251, 258, and 304), so we used the Pellet reasoner for this test. The threshold was set to 0.2, while the weights were set as follows:

- Weight for syntax similarity: 0.3
- Weight for property-based similarity: 0.2
- Weight for class-based similarity: 0.5

KOSIMap gets near perfect alignment (Precision and Recall is 0.99) for tests 101, 103 and 104 (Table 1). Although KOSIMap performs quite well in the 2xx tests, it yields very low recall ($\leq$ 0.1) when labels in the target ontology have been scrambled (i.e. tests #202 #248, #249, #25x, and #26x). Note that KOSIMap yields high recall (i.e. $\geq$ 0.9) for tests #221 to #247. For the real ontology data set (i.e. 3xx), KOSIMap yields 0.815 for Precision and .425 for Recall. Finally, KOSIMap achieves a much better harmonic mean precision than edna even though our system yields the same recall.

Table 1. Results for KOSIMap at the OAEI 2009 campaign for the benchmark test case.

Tool	KOSIMap		edna	
Test	Prec.	Rec.	Prec.	Rec.
#1xx	0.99	0.99	0.96	1.0
#2xx	0.94	0.57	0.41	0.56
#3xx	0.72	0.50	0.47	0.82
H-Mean	0.91	0.59	0.43	0.59

2.2 Anatomy

The anatomy data set consists of two large scale anatomy ontologies. On the one hand, the Adult Mouse Anatomical Dictionary[5] represents the anatomical structure of the postnatal mouse and contains 2744 classes organised hierarchically

[5] `http://www.informatics.jax.org/searches/AMA_form.shtml`

by “is-a” and “part-of” relationships. On the other hand, the NCI Thesaurus[6] is a reference terminology and biomedical ontology covering clinical care, translational and basic research, public information, and administrative activities. This ontology contains a subset of the classes defined in the thesaurus (i.e. 3304 classes). Note that the property-based similarity was discarded for this track as these ontologies only contain a very small number of properties.

KOSIMap produces an alignment for three of the four sub-tasks of this track:

1. *Optimal solution*: The optimal solution is obtained with a threshold set to 0.6, the syntax similarity set to 0.6 and the class-based similarity set to 0.4. It took KOSIMap approximately 5 min to generate the alignment.
2. *Optimal precision*: The optimal solution is obtained with a threshold set to 0.7, the syntax similarity set to 0.6 and the class-based similarity set to 0.4. It took KOSIMap approximately 5 min to generate the alignment.
3. *Optimal recall*: The optimal solution is obtained with a threshold set to 0.6, the syntax similarity set to 0.6 and the class-based similarity set to 0.4. It took KOSIMap approximately 5 min to generate the alignment.

Table 2. Results for KOSIMap at the OAEI 2009 campaign for the anatomy test case.

Tool	Optimal solution				Optimal precision			Optimal recall		
	Runtime	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
KOSIMap	$\approx$ 5 min	0.87	0.62	0.72	0.91	0.45	0.60	0.87	0.62	0.72

The results of the anatomy track are shown in Table 2. KOSIMap takes around 5 minutes to extract mappings between the Adult Mouse Anatomical Dictionary and the NCI Thesaurus and the F-Measure is 0.72. We observer significant differences with regard to the trade-off between precision and recall. For instance, we observe that the recall obtained by KOSIMap falls from 0.62 to 0.45 when generating an alignment for optimal precision sub-task. As KOSIMap favours recall over precision, the score obtained for the optimal recall sub-task is the same as the optimal solution.

2.3 Conference

This track contains 15 ontologies covering the conference organization domain. These ontologies differ in terms of DL expressivity and size. For example, *ekaw.owl* is represented in $\mathcal{SHIN}$, while *paperdyne.owl* is expressed in $\mathcal{ALCHIN}$(D).

KOSIMap generated 105 non-empty alignments with parameters set as follows:

– Weight for syntax similarity: 0.3
– Weight for property-based similarity: 0.2
– Weight for class-based similarity: 0.5

[6] `http://www.cancer.gov/cancerinfo/terminologyresources/`

Table 3 shows the precision, recall, and F-measure computed for three different thresholds (0.2, 0.5, and 0.7). The results show that KOSIMap reaches an optimal solution with the threshold set to 0.5 before obtaining lower performances with higher thresholds. Moreover, the precision achieved by our system increases at the same time as the threshold.

Table 3. Results for KOSIMap at the OAEI 2009 campaign for the conference test case.

Tool	threshold=0.2			threshold=0.5			threshold=0.7		
	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
KOSIMap	0.18	0.56	0.27	0.41	0.43	0.41	0.70	0.23	0.33

2.4 Directory

The directory track consists of 4639 test cases. As no properties (object properties or datatype properties) are found in this track, the property-based similarity is discarded for this track. In this experiment, the threshold is set to .0, while the weights are set to 0.6 (for syntax similarity) and 0.4 (for the class-based similarity). Due of the low expressivity of the ontologies (i.e. $\mathcal{AL}$), we simplified the rules to retain the correspondence with the highest score when a class in the source ontology maps to several classes in the target ontology. KOSIMap takes just over 1 minute to generate the 4639 alignments. The preliminary results of this track yielded a score of 0.618 for Precision, 0.453 for Recall, and a F-Measure of 0.523.

3 General Comments

3.1 Comments on the Results

From the results we can see that KOSIMap can take advantage of all different features associated with entities. The lexical description is especially important to achieve high precision and recall, while the hierarchical and internal structure are used to refine the final alignment. For example, tests in the benchmark track with scrambled labels (i.e. tests 248 to 266) tend to yield very low recall.

Based on the anatomy track, we have demonstrated the scalability of our approach. Although the two ontologies are not very expressive (i.e. $\mathcal{ALE}$ for AMA and $\mathcal{ALE}+$ for the NCI thesaurus), we have shown that the use of a DL reasoner does not impact the scalability of our system. Thus, this result suggests that the use of a reasoner does not greatly increase the runtime of the mapping task. Note that testing on more expressive large-scale ontologies should be carried to further test this observation.

3.2 Discussions on the Way to Improve KOSIMap

KOSIMap uses different strategies to extract correspondences between two ontologies. Based on the test library, we have seen that some strategies (e.g. property-based similarity) were not always useful to extract alignments. One possible way to improve the current system would be to include a strategy selection module. With strategy selection, KOSIMap could avoid some some noise produced by some strategies when the information these strategies rely on is not adequate. For example, when no properties are defined in the ontology.

Another improvement to the system would be to include a module to fine-tune weights when combining the different similarity measure. The current approach relies on the user to analyse the information contained in the ontologies. It is important to note that this is process is both time-consuming and error prone. A solution to this problem would be to consider the DL expressivity of both ontologies to analyse the impact of each measure on the global similarity value.

3.3 Comments on the OAEI 2009 Test Cases

The advantage of the OAEI test library is that it provides a wide range of tests covering real word and modied ontologies. For example, the *benchmark* track allows anyone to clearly identify the strengths and weaknesses of their systems. The library also includes test cases for comparing large scale ontologies. However, the ontologies provided in the anatomy track are not very expressive. As a result, it is difficult to address the impact of using DL reasoners on large scale ontologies.

4 Conclusion

In this paper, we present the KOSIMap system, which aligns entities from two ontologies. This system relies on DL reasoning to (i) extract background knowledge about every entity, and (ii) to remove inappropriate correspondences from an alignment. KOSIMap consists of three main steps; namely *Pre-Processing*, *Similarity Generation*, and *Alignment Extraction*. It first parses the two ontologies, extracts the implicit structure of both ontologies using an OWL DL reasoner, and applies natural language techniques to lexical descriptions (i.e. labels). Next, it computes three different types of similarities for every pair of entities. These similarity values are then combined and stored in a n*m matrix from which a pre-alignment is extracted. This pre-alignment is then passed through a refinement process, which eliminates inconsistent mappings.

Secondly, we report the results obtained by KOSIMap for its first participation to the Ontology Alignment Evaluation Initiative. From the results of the benchmark test case, we can see that our system can take advantage of all different features associated with entities during the ontology mapping task. We have also shown that KOSIMap remains scalable despite using DL reasoning throughout the mapping process. However, testing on more expressive large-scale ontologies should be carried to further test this observation.

5 Acknowledgements

The IPAS project is co-funded by the Technology Strategy Board's Collaborative Research and Development programme (www.innovateuk.org) and Rolls-Royce (project No. TP/2/IC/6/I/10292).

References

1. S. Castano, A. Ferrara, S. Montanelli, G. N. Hess, and S. Bruno. State of the art on ontology coordination and matching. Report FP6-027538, BOEMIE, March 2007.
2. M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *Proceedings of the 1st European Semantic Web Symposium (ESWS 04)*, pages 76–91, 2004.
3. J. Euzenat, A. Ferrara, L. Hollink, V. Malaise, C. Meilicke, A. Nikolov, J. Pane, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Svab-Zamazal, V. Svatek, C. T. dos Santos, and G. Vouros. Preliminary results of the ontology alignment evaluation initiative 2009. In *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009)*, 2009.
4. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, Berlin, 2007.
5. J. Euzenat and P. Valtchev. An integrative proximity measure for ontology alignment. In *Proceedings of ISWC-2003 Workshop on Semantic Information Integration*, page 3338, 2003.
6. M. Horridge, S. Bechhofer, and O. Noppens. Igniting the OWL 1.1 touch paper: The OWL API. In *Proceedings of the 3rd OWL Experienced and Directions Workshop (OWLED 2007)*, Innsbruck, Austria, 2007.
7. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, pages 251–263, Siguenza, Spain, 2002.
8. A. E. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.
9. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.
10. G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *Proceeding of the 4th International Semantic Web Conference (ISWC 2005)*, 2005.
11. E. Sutinen and J. Tarhio. On using Q-Gram locations in approximate string matching. In *Proceedings of the 3rd Annual European Symposium on Algorithms (ESA 95)*, page 327340, 1995.
12. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, pages 292–297, Seattle, USA, August 2006.
13. P. Wang and B. Xu. Debugging ontology mappings: A static approach. *Computing and Informatics*, 22:1001–1015, 2003.
14. W. E. Winkler. String comparator metrics and enhanced decision rules in the Fellegi-Sunter Model of record linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Association, 1990.