Cluster-based Similarity Aggregation for Ontology Matching

Quang-Vinh Tran[1], Ryutaro Ichise[2], and Bao-Quoc Ho[1]

[1] Faculty of Information Technology, Ho Chi Minh University of Science, Vietnam
{tqvinh,hbquoc}@fit.hcmus.edu.vn
[2] Principles of Informatics Research Division, National Institute of Informatics, Tokyo, Japan
ichise@nii.ac.jp

Abstract. Cluster-based similarity aggregation (CSA) is an automatic similarity aggregating system for ontology matching. The system have two main part. The first is calculation and combination of different similarity measures. The second is extracting alignment. The system first calculates five different basic measures to create five similarity matrixes, i.e, string-based similarity measure, WordNet-based similarity measure... Furthermore, it exploits the advantage of each measure through a weight estimation process. These similarity matrixes are combined into a final similarity matrix. After that, the pre-alignment is extracted from this matrix. Finally, to increase the accuracy of the system, the pruning process is applied.

1 Presentation of the system

In the Internet, ontologies are widely used to provide semantic to data. Since they are created by different users for different purposes, we need to develop a method to match multiple ontologies for integrating data from different resources [2].

1.1 State, purpose, general statement

CSA (**C**luster-based **S**imilarity **A**ggregation) is the automatic weight aggregating system for ontology alignment. The system is designed to search for semantic correspondence between heterogeneous data sources from different ontologies. The current implementation only support one-to-one alignment between concepts and properties (including object properties and data properties). The core of CSA is utilizing the advantage of each basic strategy for the alignment process. For example, the string-based similarity measure works well when the two entities are similar linguistically while the structure-based similarity measure is effective when the two entities are similar in their local structure. The system automatically combines many similarity measurements based on the analysis of their similarity matrix. Details of the system are described in the following parts.

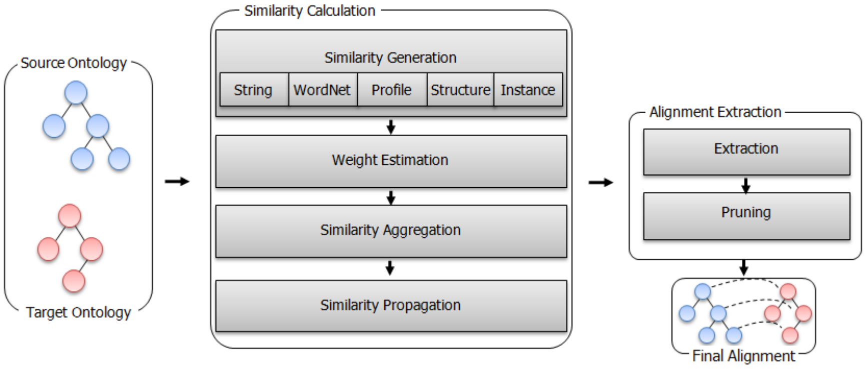

Fig. 1. The main process of CSA

1.2 Specific techniques used

The process of the system is illustrated in Figure 1. First, we calculate five basic similarity measures. These similarities are String edit distance, WordNet based, Profile, Structure, and Instance based. Second, the weight for each similarity is estimated through a weight estimation process. We then aggregate these similarities based on their weights. After that, we propagate the similarity to get the final similarity matrix. The pre-alignment is then extracted from this matrix. Finally, we apply the pruning process to get the final alignment.

Similarity Generation The similarity between entities in the two ontologies is computed by five basic measures. The String edit distance measures the lexical feature of the entity's name. The WordNet [3] exploits the similarity between words occur in the entity's name. We use the method of Wu and Palmer for calculating the WordNet similarity [8]. The profile similarity makes use of the id, label, and comments information contained in an entity. The profile for a class takes their properties, instances into account. The profile for a property includes their domains and their ranges. We then construct the weight feature vector using tf-idf. The similarity is then calculated by the cosine similarity of the two vectors. The structure similarity is calculated for class only. This similarity measures the difference in the local structure of an entity. We implement the method introduced in [7] for the structure measure. This calculation is based on the difference of number of class's children, number of a class's siblings, the normalized depth from the root and the number of properties restricted to this class. The instance-based measure is similar to the profile except that we only utilize the content of instances that belong to classes and the properties appear in these instances.

Weight estimation Weight estimation is the core of CSA. In this step, we analyze each similarity matrix of each basic measure to find which one is actually effective for the alignment. This process is based on two sources of information. First, for each single method, the process of finding a threshold for distinguishing matching pairs from non matching pairs can be viewed as a binary classification problem [5]. The positive class contains matching pairs and the negative class contains non matching ones. Second, in one-to-one ontology alignment, the maximum number of matching pairs is equal to the minimum number of entities in the two ontologies. If a single method is effective, its correspondent similarity matrix must have the two criteria: The matrix that can distinguish matching from non matching pairs and the number of matching pairs must approximate the minimum number of entities in the two ontologies.

On the basis of these criteria, we model the weight estimation process for concept as follows: First, for each similarity matrix we use the K-means algorithm to cluster the similarity values into two different classes (k = 2). The feature is the similarity value of each pair of classes. The cluster with higher mean represents the matching set, and the lower one represents the non matching set. We filter out all the values that belong to the non matching set. What remains is the similarity matrix with the higher values. We then we calculate the number of row that has value in the matrix. These row represent the possible matching pairs. Because in our case we consider the one-to-one matching, one concept from source ontology is only matched up to one concept from target ontology. Finally, the weight is estimated by the ratio of the number of rows over the number of matched values in the filtered matrix.

$$weight = \frac{|\textit{number of row that has value}|}{|\textit{number of value in matching set}|} \tag{1}$$

The weight estimation for property similarity matrix is calculated in the same manner.

Similarity Aggregation The similarity combination can be defined as the weight average of the five basic measures. The weight for each measure is estimated in the previous step.

$$Sim_{combine}\left(e_1, e_2\right) = \frac{\sum_{i=1}^{n} weight_i \times Sim_i\left(e_1, e_2\right)}{\sum_{i=1}^{n} weight_i} \tag{2}$$

Similarity Propagation This step considers the impact of structural information on the similarity between each entity pair in the aggregated matrix. The intuition is that the more similar in structure two entities are, the more similar they are. To exploit the structure information, we use Descendant Similarity Inheritance [1].

Extraction In our system, only one-to-one matching is allowed. The final similarity matrix can be viewed as a bipartite graph with the first set of vertices are entities from source ontology and the second set of vertices are entities from target ontology. Thus, the alignment extraction can be modeled as the process of finding the mapping from the bipartite graph. To solve this, we apply the stable marriage problem algorithm [4]. We model the two set of entities as sets of men and women. For each man and each woman, in the correspondence set, a list of priority of men and women is created based on their similarity value. The stable marriage algorithm is then applied to find the stable mapping between two sets. The result is the pre-alignment.

Table 1. Performance of CSA on benchmark track

Test	Prec.	Rec.
101	1.0	1.0
201-202	0.83	0.73
221-247	0.98	1.0
248-252	0.79	0.61
253-259	0.81	0.55
260-266	0.70	0.49
H-mean	0.82	0.65

Pruning This is the final step of our system. In this step we filter out a proportion of entities pair that have low confidence to increase the precision of our system. For the threshold, we set it manually. The result is the final alignment of the two ontologies.

1.3 Adaptations made for the evaluation

We do not make any specific adaptation for the OAEI 2011 campaign. The three track are run in the same set of parameter.

1.4 Link to the system and parameters file

The CSA system can be downloaded from seal-project at `http://www.seals-project.eu/`.

1.5 Link to the set of provided alignments (in align format)

The result of CSA system can be downloaded from seal-project at `http://www.seals-project.eu/`.

2 Results

In this section, we present the results of the CSA system. We participate in the three tracks of benchmarks, anatomy, and conference. The result is in the following part.

2.1 Benchmarks

On the benchmarks of 2011, the reference ontology are different that the previous year. Since the descriptions, restrictions and instances are limited, it affects our algorithm very much. The result is shown at Table 1. We group the test into six groups based on their difficulty. The result shows the harmonic means precision and recall for each group.

Table 2. Performance of CSA on anatomy track

Precision	Recall	F-measure
0.465	0.757	0.576

Table 3. Performance of CSA on conference track

Prec.	F_1Meas.	Rec.
0.5	0.55	0.6
Prec.	F_2Meas.	Rec.
0.5	0.58	0.6
Prec.	$F_{0.5}$Meas.	Rec.
0.61	0.58	0.47

2.2 Anatomy

The anatomy dataset consists of two large ontologies of adult mouse anatomy with 2744 classes and a part of NCI Thesaurus for describing human anatomy with 3304 classes. The CSA result is shown in Table 2. Because of the high cost of computation, the execution time is quite high (4685s). In this track our system is high in recall (0.76) but the precision is quite low (0.47).

2.3 Conference

The results of conference track are shown in Table 3. It is difficult to archive the good results since ontologies from this track are real and developed by different organizations for different purposes.

3 General comments

This is the first time CSA has participated in the OAEI tracks, and our systems new to the seals platform. Further, the same set of parameter for all test in all tracks are difficult, because for each track the ontologies have a different characteristics to be processed. Thus, for any given dataset we need a different method for defining the threshold to extract the final alignment.

3.1 Comments on the results

Strengths CSA can be used to automatically combine different similarity measures. Our system does not need any external resources or training data for estimating the weight in the aggregation step.

Weaknesses The structure based similarity included in CSA is not strong enough to distinguish the different between matching and non-matching pairs. There are no structure similarity for properties. Further, we have not yet integrated any semantic verification or constraints in our system.

3.2 Discussions on the way to improve the proposed system

Our system is new and there are many opportunities to improve our method. First, we can integrate more basic similarity measures for aggregating. Second, for the pruning step, we can find the way for automatic defining a threshold rather than manually tuning. Finally, we can use some semantic verification as in [6] to pruning the low confidence matching pairs.

4 Conclusion

This is the first time the CSA has participated in OAEI campaign. In this year, we have participated in three tracks of benchmarks, anatomy and conference. We have introduced a new method for aggregating different similarity measures. The results show that our method is promising.

References

1. Isabel F. Cruz and William Sunna. Structural alignment methods with applications to geospatial ontologies. *Transactions in GIS*, 12(6):683–711, 2008.
2. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
3. Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.
4. David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.
5. Ryutaro Ichise. Machine learning approach for ontology mapping using multiple concept similarity measures. In *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science*, pages 340–346, Washington, DC, USA, 2008. IEEE Computer Society.
6. Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7:235–251, September 2009.
7. Ming Mao, Yefei Peng, and Michael Spring. An adaptive ontology mapping approach with neural network based constraint satisfaction. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8:14–25, March 2010.
8. Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.