

Lily: Ontology Alignment Results for OAEI 2008

Peng Wang, Baowen Xu

School of Computer Science and Engineering, Southeast University, China
{pwangseu@gmail.com}

Abstract. This paper presents the alignment results of Lily for the ontology alignment contest OAEI 2008. Lily is an ontology mapping system, and it has four main features: generic ontology matching, large scale ontology matching, semantic ontology matching and mapping debugging. In the past year, Lily has been improved greatly for both function and performance. In OAEI 2008, Lily submitted the results for seven alignment tasks: benchmark, anatomy, fao, directory, mldirectory, library and conference. The specific techniques used by Lily are introduced briefly. The strengths and weaknesses of Lily are also discussed.

1 Presentation of the system

Currently more and more ontologies are distributedly used and built by different communities. Many of these ontologies would describe similar domains, but using different terminologies, and others will have overlapping domains. Such ontologies are referred to as heterogeneous ontologies, which is a major obstacle to realize semantic interoperation. Ontology mapping, which captures relations between ontologies, aims to provide a common layer from which heterogeneous ontologies could exchange information in semantically sound manners.

Lily is an ontology mapping system for solving the key issues related to heterogeneous ontologies, and it uses hybrid matching strategies to execute the ontology matching task. Lily can be used to discovery the mapping for both normal ontologies and large scale ontologies.

1.1 State, purpose, general statement

In order to obtain good alignments, the core principle of the matching strategy in Lily is utilizing the useful information effectively and rightly. Lily combines several novel and efficient matching techniques to find alignments. Currently, Lily realized four main functions: (1) Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies. (2) Large scale Ontology Matching method (LOM) is used for the matching tasks with large size ontologies. (3) Semantic Ontology Matching method (SOM) is used for discovering the semantic relations between ontologies. Lily uses the web knowledge to recognize the semantic relations

through the search engine. (4) Ontology mapping debugging is used to improve the alignment results.

The alignment process mainly contains three steps: (1) Preprocessing step parses the ontologies, and prepares the necessary data for the subsequent steps. (2) Match computing step uses suitable methods to compute the similarity between elements from different ontologies. (3) Post processing step is responsible for extracting, debugging and evaluating mappings. The architecture of Lily is shown in Fig. 1.

The latest version of Lily is V2.0. Comparing with the last version V1.2, Lily has been enhanced greatly at both function and performance. Lily V2.0 provides a friendly graphical user interface. Fig.2 shows a snapshot when Lily is running.

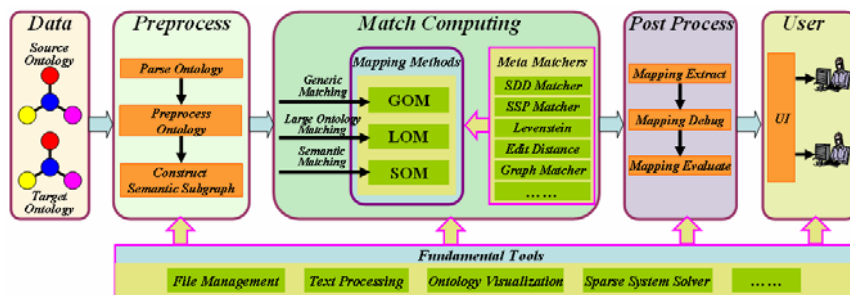


Fig. 1. The Architecture of Lily

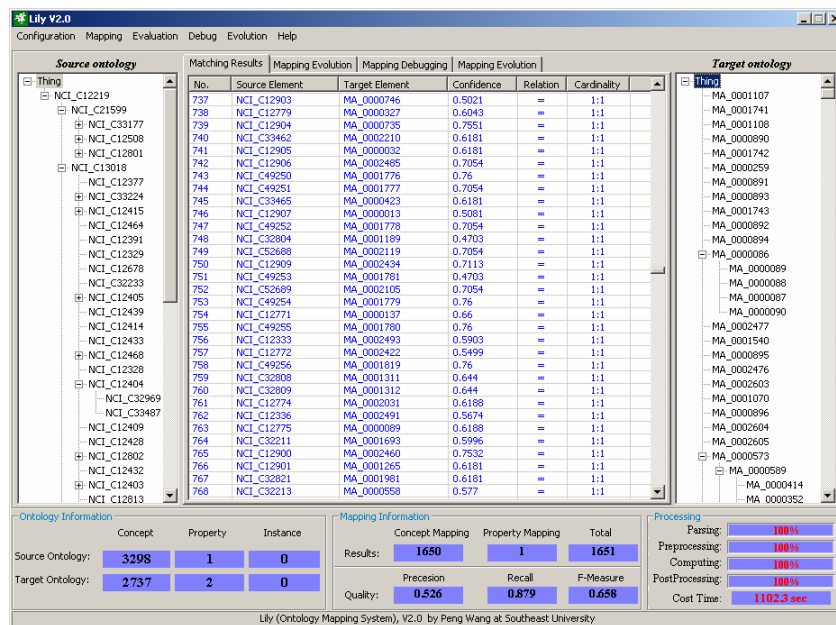


Fig. 2. The user interface of Lily

1.2 Specific techniques used

Lily aims to provide high quality 1:1 alignments between concept/property pairs. The main specific techniques used by Lily are as follows.

Semantic subgraph An entity in a given ontology has its specific meaning. In our ontology mapping view, capturing such meaning is very important to obtain good alignment results. Therefore, before similarity computation, Lily first describes the meaning for each entity accurately. The solution is inspired by the method proposed by Faloutsos et al. for discovering connection subgraphs [1]. It is based on electricity analogues to extract a small subgraph that best captures the connections between two nodes of the graph. Ramakrishnan et al. also exploits such idea to find the informative connection subgraphs in RDF graph [2].

The problem of extracting semantic subgraphs has a few differences from Faloutsos's connection subgraphs. We modified and improved the methods provided by the above two work, and proposed a method for building an *n-size* semantic subgraph for a concept or a property in ontology. The subgraphs can give the precise descriptions of the meanings of the entities, and we call such subgraphs semantic subgraphs. The detail of the semantic subgraph extraction process is reported in our other work [3].

The significance of semantic subgraphs is that we can build more credible matching clues based on them. Therefore it can reduce the negative affection of the matching uncertain.

Generic ontology matching method The similarity computation is based on the semantic subgraphs, i.e. all the information used in the similarity computation is come from the semantic subgraphs. Lily combines the text matching and structure matching techniques [3].

Semantic Description Document (SDD) matcher measures the literal similarity between ontologies. A semantic description document of a concept contains the information about class hierarchies, related properties and instances. A semantic description document of a property contains the information about hierarchies, domains, ranges, restrictions and related instances. For the descriptions from different entities, we calculate the similarities of the corresponding parts. Finally, all separate similarities are combined with the experiential weights. For the regular ontologies, the SDD matcher can find satisfactory alignments in most cases.

To solve the matching problem without rich literal information, a similarity propagation matcher with strong propagation condition (SSP matcher) is presented, and the matching algorithm utilizes the results of literal matching to produce more alignments. Compared with other similarity propagation methods such as similarity flood [4] and SimRank [5], the advantages of our similarity propagation include defining stronger propagation condition, semantic subgraphs-based and with efficient and feasible propagation strategies. Using similarity propagation, Lily can find more alignments that cannot be found in the text matching process.

However, the similarity propagation is not always perfect. When more alignments are discovered, more incorrect alignments would also be introduced by the similarity propagation. So Lily also uses a strategy to determine when to use the similarity propagation.

Large scale ontology matching Large scale ontology matching tasks propose the rough time complexity and space complexity for ontology mapping systems. To solve this problem, we proposed a novel method [6], which uses the negative anchors and positive anchors to predict the pairs can be passed in the later matching computing. The method is different from other several large scale ontology matching methods, which are all based on ontology segment or modularization.

Semantic ontology matching Our semantic matching method [7] is based on the idea that Web is a large knowledge base, and from which we can gain the semantic relations between ontologies through Web search engine. Based on lexico-syntactic patterns, this method first obtains a candidate mapping set using search engine. Then the candidate set is refined and corrected with some rules. Finally, ontology mappings are chosen from the candidate mapping set automatically.

Ontology mapping debugging Lily uses a technique called ontology mapping debugging to improve the alignment results [8]. During debugging, some types of mapping errors, such as redundant and inconsistent mappings, can be detected. Some warnings, including imprecise mappings or abnormal mappings, are also locked by analyzing the features of mapping result. More importantly, some errors and warnings can be repaired automatically or can be presented to users with revising suggestions.

1.3 Adaptations made for the evaluation

In OAEI 2008, Lily used GOM matcher to compute the alignments for three tracks (benchmark, directory, conference). In order to assure the matching process is fully automated, all parameters are configured automatically with a strategy. For the large ontology alignment tracks (anatomy, fao, mldirectory, library), Lily used LOM matcher to discover the alignments. All parameters used by these tracks are same. Lily can determine which matcher should be chosen according to the size of ontology.

1.4 Link to the system and the set of provided alignments

Lily V2.0 and the alignment results for OAEI 2008 are available at <http://ontomappinglab.googlepages.com/lily.htm>.

2 Results

2.1 benchmark

The benchmark test set can be divided into five groups: 101-104, 201-210, 221-247, 248-266 and 301-304.

101-104 Lily plays well for these test cases. But for the irrelevant ontology 102, Lily returns several alignments because it cannot decide whether the two ontologies are irrelevant, so it tries to find any possible alignments.

201-210 Lily can produce good results for this test set. Even without right labels and comments information, Lily can find most correct alignments through making use of other information such as instances. Using few alignment results obtained by the

basic methods as inputs, the similarity propagation strategy will generate more alignments.

221-247 Lily can find most correct alignments using the labels and comments information.

248-266 This group is the most difficult test set. Lily first uses the SDD matcher to look for a few alignments. Then, using initial alignments as input, Lily exploits the SSP matcher to discover more alignments. In our experiments, too smaller and too bigger size semantic subgraph can not produce good alignments. *10-35* is a suitable size range in our experience. In 262, since almost all literal and structure information are suppressed, the similarity propagation can not find any results.

301-304 This test set are the real ontologies. Lily only finds the equivalent alignment relations.

The following table shows the average performance of each group and the overall performance on the benchmark test set.

Table 1. The performance on the benchmark

	101-104	201-210	222-247	248-266	301-304	Average	H-mean
Precision	1.00	1.00	0.99	0.93	0.86	0.95	0.97
Recall	1.00	0.95	1.00	0.76	0.79	0.84	0.88

2.2 anatomy

The anatomy track consists of two real large-scale biological ontologies. Lily can handle such ontologies smoothly with LOM method. Lily submitted the results for three sub-tasks in anatomy. Task#1 means that the matching system has to be applied with standard settings to obtain a result that is as good as possible. Task#2 means that the system generates the results with high precision. Task#3 means that the system generates the alignment with high recall.

Table 2 shows the performance of the task #1, #2 and #3 on anatomy test set, where Recall+ measures how many non trivial correct correspondences can be found in an alignment.

Table 2. The performance on the anatomy

	Runtime	Precision	Recall	Recall+	F-measure
Task#1	3h 20min	0.796	0.693	0.470	0.741
Task#2	3h 20min	0.863	0.640		0.664
Task#3	3h 20min	0.490	0.790	0.613	0.605

2.3 directory

The directory track requires matching two taxonomies describing the web directories. Except the class hierarchy, there is no other information in the ontologies. Therefore, besides the literal information, Lily also utilizes the hierarchy information to decide the alignments. Table 3 shows the performance on the directory test set.

Table 3. The performance on the directory

Precision	Recall	F-measure
0.59	0.37	0.46

2.4 conference

This task contains 15 real-case ontologies about conference. For a given ontology, we compute the alignments with itself, as well as with other ontologies. For we treat the equivalent alignment is symmetric, we get 105 alignment files totally. The heterogeneous character in this track is various. It is a challenge to generate good results for all ontology pairs in this test set.

The performance of Lily on this data set is shown as Table 4. The evaluation is based on two reference alignments.

Table 4. The performance on the conference based on reference mappings

	Precision	Recall	F-measure
Reference Alignment A	0.568	0.581	0.575
Reference Alignment B	0.432	0.500	0.463

2.5 fao

The task consists of several large scale ontologies about food and agricultural domain. The LOM method is used to find the alignments. Lily only provides the alignments between concepts or properties. Therefore, we did not submit the alignments for the subtask for finding the alignments between instances. Table 5 is the performance on fao data set.

Table 5. The performance on the fao

subtrack	Precision	Recall
agrafsa	0.867	0.403

2.6 library

This is a thesaurus mapping task. Lily only discovers the *extractMatch* alignments. Lily did not utilize the instance information provided in this year. Table 6 shows the evaluation results of Lily on this data set.

Table 6. The performance on the library

Evaluation Scenario	Precision	Coverage			
Thesaurus merging	0.529	0.368			
Annotation translation	Precision (book level)	Recall (book level)	Precision (annotation level)	Recall (annotation level)	Jaccard (annotation level)
	0.435	0.156	0.397	0.107	0.100

2.7 mldirectory

This task requires matching two web directories in different languages. For the reason that the ontologies provided by this task are hard to be parsed correctly, Lily only submits two alignment results for two subtasks (Auto and Movie) in English. Lily finds 377 alignments for Auto and 1864 alignments for Movie.

3 General comments

3.1 Comments on the results

Strengths For normal size ontologies, if they have regular literals or similar structures, Lily can achieve satisfactory alignments.

Weaknesses Lily needs to extract semantic subgraphs for all concepts and properties. It is a time-consuming process. Even though we have improved the efficiency of the extracting algorithm, it still is the bottleneck for the performance of the system.

4 Conclusion

We briefly introduce our ontology matching tool Lily. The matching process and the special techniques used by Lily are presented. The preliminary alignment results are carefully analyzed. Finally, we summarized the strengths and the weaknesses of Lily.

References

1. Faloutsos, C., McCurley, K. S., Tomkins, A.: Fast Discovery of Connection Subgraphs. In the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington (2004).
2. Ramakrishnan, C., Milnor, W. H., Perry, M., Sheth, A. P.: Discovering Informative Connection Subgraphs in Multirelational Graphs. ACM SIGKDD Explorations, Vol. 7(2), (2005)56-63.
3. Wang, P., Xu, B. A Generic Ontology Matching Method Based on Semantic Subgraph, to appear.
4. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In the 18th International Conference on Data Engineering (ICDE), San Jose CA (2002).
5. Jeh, G., Widom, J.: SimRank: A Measure of Structural-Context Similarity. In the 8th International Conference on Knowledge Discovery and Data Mining (SIGKDD), Edmonton, Canada, (2002).
6. Wang, P., Xu, B. A New Method for Matching Large Scale Ontologies, to appear.
7. Li, K., Xu, B., and Wang, P. An Ontology Mapping Approach Using Web Search Engine. Journal of Southeast University, 2007, 23(3):352-356.
8. Wang, P., Xu, B. Debugging Ontology Mapping: A Static Method. Computing and Informatics, 2008, 27(1): 21-36.

Appendix: Raw results

The final results of benchmark task are as follows.

Matrix of results

#	Comment	Prec.	Rec.	#	Comment	Prec.	Rec.
101	Reference alignment	1.00	1.00	251		0.96	0.76
103	Language generalization	1.00	1.00	251-2		0.99	0.96
104	Language restriction	1.00	1.00	251-4		0.99	0.90
201	No names	1.00	1.00	251-6		0.94	0.83
201-2		1.00	1.00	251-8		0.99	0.85
201-4		1.00	1.00	252		0.96	0.79
201-6		1.00	1.00	252-2		0.97	0.94
201-8		1.00	1.00	252-4		0.97	0.94
202	No names, no comment	1.00	0.84	252-6		0.97	0.94
202-2		1.00	0.97	252-8		0.97	0.94
202-4		1.00	0.92	253		0.81	0.59
202-6		0.98	0.87	253-2		0.98	0.93
202-8		0.98	0.85	253-4		1.00	0.92
203	Misspelling	1.00	1.00	253-6		0.95	0.81
204	Naming conventions	1.00	1.00	253-8		0.95	0.79
205	Synonyms	1.00	0.99	254		1.00	0.27
206	Translation	1.00	0.99	254-2		1.00	0.82
207		1.00	0.99	254-4		1.00	0.70
208		1.00	0.99	254-6		1.00	0.61
209		0.97	0.88	254-8		1.00	0.42
210		1.00	0.89	257		0.50	0.06
221	No hierarchy	1.00	1.00	257-2		1.00	0.97
222	Flattened hierarchy	1.00	1.00	257-4		0.94	0.88
223	Expanded hierarchy	0.98	0.98	257-6		0.84	0.79
224	No instances	1.00	1.00	257-8		0.89	0.76
225	No restrictions	1.00	1.00	258		0.80	0.60
228	No properties	1.00	1.00	258-2		0.97	0.94
230	Flattening entities	0.94	1.00	258-4		0.96	0.88
231	Multiplying entities	1.00	1.00	258-6		0.95	0.82
232	No hierarchy no instance	1.00	1.00	258-8		0.94	0.78
233	No hierarchy no property	1.00	1.00	259		0.89	0.70
236	No instance no property	1.00	1.00	259-2		0.98	0.95
237		1.00	1.00	259-4		0.98	0.95
238		0.99	0.99	259-6		0.98	0.95
239		0.97	1.00	259-8		0.98	0.95
240		0.97	1.00	260		0.94	0.55
241		1.00	1.00	260-2		0.96	0.93
246		0.97	1.00	260-4		0.93	0.93
247		0.94	0.97	260-6		0.96	0.79

248		1.00	0.81	260-8		0.88	0.72
248-2		1.00	0.95	261		0.67	0.48
248-4		1.00	0.92	261-2		0.88	0.91
248-6		1.00	0.88	261-4		0.88	0.91
248-8		0.98	0.85	261-6		0.88	0.91
249		0.83	0.66	261-8		0.88	0.91
249-2		0.98	0.95	262		NaN	0.00
249-4		0.98	0.91	262-2		1.00	0.79
249-6		0.98	0.87	262-4		1.00	0.61
249-8		0.95	0.82	262-6		1.00	0.42
250		0.90	0.58	262-8		1.00	0.21
250-2		1.00	1.00	265		0.80	0.14
250-4		1.00	1.00	266		0.30	0.09
250-6		1.00	1.00	301	BibTeX/MIT	0.94	0.82
250-8		1.00	0.88	302	BibTeX/UMBC	0.89	0.65
				303	Karlsruhe	0.65	0.71
				304	INRIA	0.95	0.97