Lily: Ontology Alignment Results for OAEI 2009

Peng Wang[1], Baowen Xu[2,3]

[1]College of Software Engineering, Southeast University, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, China
[3]Department of Computer Science and Technology, Nanjing University, China
pwang@seu.edu.cn, bwxu@nju.edu.cn

Abstract. This paper presents the alignment results of Lily for the ontology alignment contest OAEI 2009. Lily is an ontology mapping system, and it has four functions: generic ontology matching, large scale ontology matching, semantic ontology matching and mapping debugging. In OAEI 2009, Lily submited the results for four alignment tasks: benchmark, anatomy, directory and conference.

1 Presentation of the system

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. In the past year, we did not improve Lily significantly but revised some bugs according to the reports from some users.

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 matching process mainly contains three steps: (1) In preprocess, Lily parses ontologies and prepares the necessary data for the subsequent steps. (2) In computing step, Lily uses suitable methods to calculate the similarity between elements from different ontologies. (3)In post-process, the alignments are extracted and then refined by mapping debugging. The architecture of Lily is shown in Fig. 1.

The lasted version of Lily is V2.0. Lily V2.0 provides a friendly graphical user interface. Fig.2 shows a snapshot when Lily is running.

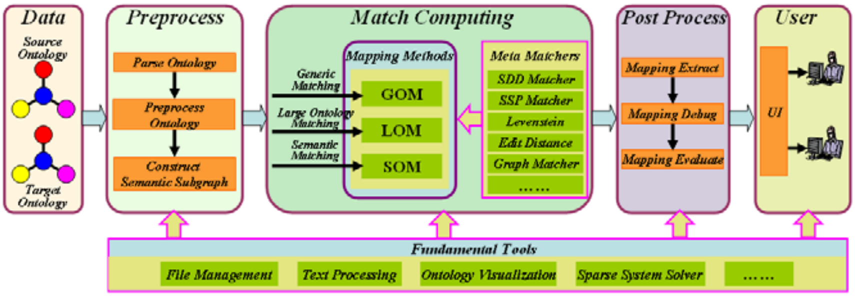
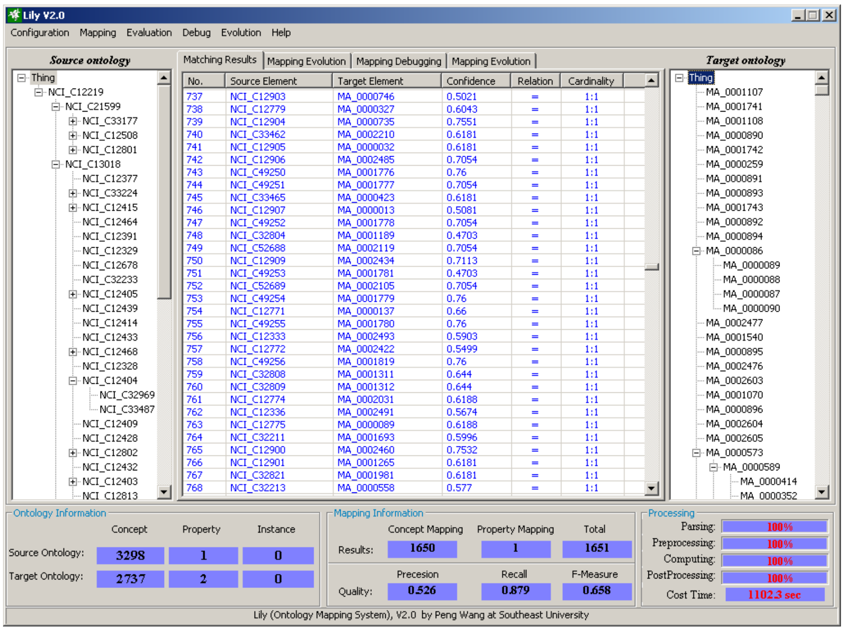


Fig. 1. The Architecture of Lily



No.	Source Element	Target Element	Confidence	Relation	Cardinality
737	NCI_C12903	MA_0000746	0.5021	=	1:1
738	NCI_C12779	MA_0000327	0.6043	=	1:1
739	NCI_C12904	MA_0000735	0.7551	=	1:1
740	NCI_C33462	MA_0002210	0.6181	=	1:1
741	NCI_C12905	MA_0000032	0.6181	=	1:1
742	NCI_C12906	MA_0002485	0.7054	=	1:1
743	NCI_C49250	MA_0001776	0.76	=	1:1
744	NCI_C49251	MA_0001777	0.7054	=	1:1
745	NCI_C33465	MA_0000423	0.6181	=	1:1
746	NCI_C12907	MA_0000013	0.5081	=	1:1
747	NCI_C49252	MA_0001778	0.7054	=	1:1
748	NCI_C32804	MA_0001189	0.4703	=	1:1
749	NCI_C52688	MA_0002119	0.7054	=	1:1
750	NCI_C12909	MA_0002434	0.7113	=	1:1
751	NCI_C49253	MA_0001781	0.4703	=	1:1
752	NCI_C52689	MA_0002105	0.7054	=	1:1
753	NCI_C49254	MA_0001779	0.76	=	1:1
754	NCI_C12771	MA_0000137	0.66	=	1:1
755	NCI_C49255	MA_0001780	0.76	=	1:1
756	NCI_C12333	MA_0002493	0.5903	=	1:1
757	NCI_C12772	MA_0002422	0.5499	=	1:1
758	NCI_C49256	MA_0001819	0.76	=	1:1
759	NCI_C32808	MA_0001311	0.644	=	1:1
760	NCI_C32809	MA_0001312	0.644	=	1:1
761	NCI_C12774	MA_0002031	0.6188	=	1:1
762	NCI_C12336	MA_0002491	0.5674	=	1:1
763	NCI_C12775	MA_0000089	0.6188	=	1:1
764	NCI_C32211	MA_0001693	0.5996	=	1:1
765	NCI_C12900	MA_0002460	0.7532	=	1:1
766	NCI_C12901	MA_0001265	0.6181	=	1:1
767	NCI_C32821	MA_0001981	0.6181	=	1:1
768	NCI_C32213	MA_0000558	0.577	=	1:1

Ontology Information

	Concept	Property	Instance
Source Ontology:	3298	1	0
Target Ontology:	2737	2	0

Mapping Information

	Concept Mapping	Property Mapping	Total
Results:	1650	1	1651
	Precesion	**Recall**	**F-Measure**
Quality:	0.526	0.879	0.658

Processing

Parsing:	100%
Preprocessing:	100%
Computing:	100%
PostProcessing:	100%
Cost Time:	1102.3 sec

Lily (Ontology Mapping System), V2.0 by Peng Wang at Southeast University

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 [3], 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 [6] is base 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 [7]. 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 2009, 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), Lily used LOM matcher to discover the alignments. Lily can determine which matcher should be chose 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 2009 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.

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	221-247	248-266	301-304	Average	H-mean
Precision	1.00	0.99	0.99	0.94	0.83	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.

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.

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.

3 General comments

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.: Research on the Key Issues in Ontology Mapping (In Chinese). PhD Thesis, Southeast University, Nanjing, 2009.
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. Li, K., Xu, B., and Wang, P. An Ontology Mapping Approach Using Web Search Engine. Journal of Southeast University, 2007, 23(3):352-356.
7. 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		0.96	0.84
201-2		1.00	1.00	251-8		0.99	0.83
201-4		1.00	1.00	252		0.95	0.77
201-6		1.00	1.00	252-2		0.98	0.94
201-8		1.00	1.00	252-4		0.98	0.94
202	No names, no comment	1.00	0.84	252-6		0.98	0.94
202-2		1.00	0.95	252-8		0.97	0.93
202-4		1.00	0.92	253		0.85	0.62
202-6		0.98	0.88	253-2		1.00	0.93
202-8		0.98	0.84	253-4		1.00	0.91
203	Misspelling	1.00	0.98	253-6		0.94	0.82
204	Naming conventions	1.00	1.00	253-8		0.98	0.82
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	0.98	254-6		1.00	0.61
209		0.97	0.87	254-8		1.00	0.42
210		1.00	0.88	257		1.00	0.12
221	No hierarchy	1.00	1.00	257-2		1.00	0.97
222	Flattened hierarchy	1.00	1.00	257-4		1.00	0.94
223	Expanded hierarchy	0.98	0.97	257-6		0.87	0.82
224	No instances	1.00	1.00	257-8		0.85	0.67
225	No restrictions	1.00	1.00	258		0.76	0.56
228	No properties	1.00	1.00	258-2		0.99	0.96
230	Flattening entities	0.94	1.00	258-4		0.96	0.88
231	Multiplying entities	1.00	1.00	258-6		0.95	0.83

232	No hierarchy no instance	1.00	1.00	258-8		0.94	0.80
233	No hierarchy no property	1.00	1.00	259		0.91	0.73
236	No instance no property	1.00	1.00	259-2		0.97	0.94
237		1.00	1.00	259-4		0.97	0.94
238		0.98	0.98	259-6		0.96	0.93
239		0.97	1.00	259-8		0.97	0.94
240		0.97	1.00	260		0.94	0.55
241		1.00	1.00	260-2		0.93	0.93
246		0.97	1.00	260-4		0.90	0.93
247		0.94	0.97	260-6		0.93	0.86
248		1.00	0.81	260-8		0.95	0.69
248-2		1.00	0.95	261		0.61	0.33
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		1.00	0.87	261-6		0.88	0.91
249		0.76	0.73	261-8		0.88	0.91
249-2		1.00	0.97	262		NaN	0.00
249-4		0.98	0.91	262-2		1.00	0.76
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		1.00	0.55	262-8		1.00	0.21
250-2		1.00	1.00	265		0.80	0.14
250-4		1.00	1.00	266		0.50	0.09
250-6		1.00	1.00	301	BibTeX/MIT	0.87	0.81
250-8		0.90	0.79	302	BibTeX/UMBC	0.84	0.65
				303	Karlsruhe	0.63	0.75
				304	INRIA	0.96	0.96