Lily-LOM: An Efficient System for Matching Large Ontologies with Non-Partitioned Method

Peng Wang

School of Computer Science and Engineering, Southeast University, China
pwang@seu.edu.cn

Abstract. Since the high time and space complexity, most existing ontology matching systems are not well scalable to solve the large ontology matching problem. Moreover, the popular divide-and-conquer matching solution faces two disadvantages: First, partitioning ontology is a complicate process; Second, it will lead to loss of semantic information during matching. To avoid these drawbacks, this paper presents an efficient large ontology matching system Lily-LOM, which uses a non-partitioned method. Lily-LOM is based on two kinds of reduction anchors, i.e. positive and negative reduction anchors, to reduce the time complexity problem. Some empirical strategies for reducing the space complexity are also discussed. The experiments show that Lily-LOM is effective.

1 Introduction

Since high time and space complexity, most ontology matching systems cannot deal with large ontology matching (LOM) problem. First, matching process requires a large amount of memory space, which would cause the system to crash due to the out of memory error. The space complexity of a matching system usually is $O(n^2)$. Second, most ontology matching algorithms are $O(n^2)$ time complexity, i.e. it needs n^2 times similarity calculations.

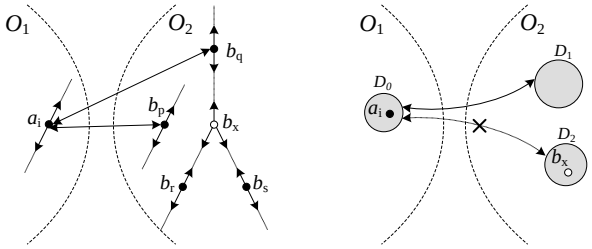
Divide-and-conquer strategy is a feasible solution for LOM problem. However, it also has two main issues to be resolved. First, we notice that some ontology partitioning approach cannot control the size of blocks, which may be too small or too large for matching. Second, the ontology partitioning idea also would cause another considerable issue, namely, the partitioning would make the elements on the boundaries of blocks lose some semantic information, that would in turn affect the quality of final matching results.

This paper presents Lily-LOM, a system for matching large ontologies, which is based on a non-partitioned method. Compared with the existing work, Lily-LOM has two distinct advantages: First, it needs not to partition large ontologies but it also has the high performance. Second, it is a general solution for LOM problem, namely, it can adopt most existing matching techniques.

2 Matching Large Ontologies Based on Reduction Anchors

During matching large ontologies, we notice two interesting facts: (1) a large ontology is often composed of the hierarchies organized by *is-a* or *part-of* properties, and a correct alignment should not be inconsistent with such hierarchies; (2) an alignment between two large ontologies has locality, i.e., most elements of region D_i in ontology O_1 will match to the elements of region D_j in ontology O_2. The two facts provide new ways for finding efficient solution about LOM.

In Fig. 1. (a), if high similarity values exist between a_i and b_p or b_q, we can decide that a_i matches b_p or b_q. This decision will bring a direct benefit: the subsequent similarity calculations between sub-concepts(/super-concepts) of a_i and super-concepts(/sub-concepts) of b_p or b_q can be skipped. This paper calls such concept pairs like (a_i, b_p) the positive reduction anchors(P-Anchors), which employ ontology hierarchy feature to reduce the time complexity in LOM.



(a) Positive Reduction Anchor (b) Negative Reduction Anchor

Fig. 1. Reduction anchors in large ontology matching

Fig. 1. (b) shows the locality phenomenon in LOM, where D_i refers to a region in the ontology. Suppose a_i in D_0 does not match b_x in D_2, then we can infer that the neighbors of a_i do not match b_x too, i.e., the similarity values between them are very low. As a result, we can skip the subsequent similarity calculations between the neighbors of a_i and b_x, which can also reduce the times of similarity calculations. This paper calls such concept pairs like (a_i, b_x) the negative reduction anchors(N-Anchors).

P-Anchors and N-Anchors provide two ways to design new efficient solutions for LOM. Based on the two kinds of anchors, matching process can skip many times of similarity calculations to reduce the time complexity significantly. Obviously, P-Anchors and N-Anchors cannot be identified in advance, so it needs to discover them dynamically in matching, then uses the anchors to predict the ignorable similarity calculations.

Let the P-Anchors of a_i is $PA(a_i) = \{b_1, b_2, ..., b_k\}$. We call all the ignorable similarity calculations predicted by $PA(a_i)$ the positive reduction set of a_i. The corresponding reduction set can be calculated by following formula, in which lub denotes least upper bound and glb is greatest lower bound.

$$PS(a_i) = [sub(a_i) \otimes sup(lub(b_1, ..., b_k))] \cup [sup(a_i) \otimes sub(glb(b_1, ..., b_k))]$$

We can prove that when the order of similarity calculations can divide the hierarchy path L into equal parts continually, the P-Anchors can generate the

maximum valid positive reduction set with $|L| * (|L| - 2)$ size [1]. It means the algorithm has the best time complexity $O(2n)$. Generally, the algorithm has $O((1 - \frac{\bar{d}}{n})n^2)$ time complexity, where $\bar{d}$ is the average depth of the ontology.

N-Anchors can also predict the ignorable similarity calculations, which are called the positive reduction set. If (a_i, b_j) is a N-Anchor, we can predict that neighbors of a_i are also irrelevant to b_j. The set of all ignorable similarity calculations predicted by this way are called the negative reduction set.

Let $NA(a_i)$ refer to the N-Anchors about a_i, the neighbors with $nScale$ distance to a_i constitute a set $Nb(a_i) = \{a_x | d(a_x, a_i) <= nScale\}$, the negative reduction set generated by a_i is:$NS(a_i) = NA(a_i) \otimes Nb(a_i)$. The time complexity of the algorithm is $O(\alpha n^2)$, where α is in $[0, 1]$ and is determined by size of negative reduction set.

3 Empirical Space Complexity Processing

Besides the time complexity, the space complexity is another challenge in LOM. We present some empirical methods for handling the space complexity problem, and it may be useful for other matching systems. The number of elements in large ontology is large, so we should avoid allocating a $n \times n$ similarity matrix. Considering the similarity matrix is a typical sparse matrix, it can adopt the compression techniques to replace it. It usually compresses a similarity matrix into several MBs. In our LOM algorithms, the size of reduction set will become bigger and bigger, which takes a large amount of space. We first replace the two dimension reduction set with one dimension style, then merge the continuous number of elements as a link. Memory space resource is valuable in LOM, so if a variable or a data structure is unused, we should free its space immediately. This principle will reduce the possibility of out of memory error.

4 Experimental Evaluations

All algorithms proposed in this paper are implemented in ontology matching system Lily-LOM. More information about Lily can be found at http://cse.seu.edu .cn/people/pwang/lily.htm.

We get some matching results on several real large ontologies by participating in OAEI[1]. Here we present the results of our LOM algorithms on three LOM tasks (*Anatomy*, *Fao*, and *Library*) in OAEI2008.

From 2007 to 2008 years, there are 13 systems participated in the anatomy task, but only three systems: Lily, Falcon-AO, and TaxoMap, used the special large ontology matching method. Falcon-AO proposed a divide-and-conquer method called PBM algorithm. TaxoMap uses the PBM algorithm, so it is a re-implement of PBM. We measure quality of the results with the classic F1-measure, and use *Recall+* [2] to measure how many non trivial correct alignments can be found.

Table 1 shows the results of three LOM systems. According to the results, we have four conclusions: (1) Lily is one of the LOM system can perform well in *Anatomy* task. (2) For the three LOM systems, Lily and Falcon-AO have similar quality, which are better than TaxoMap. (3) The running time of Lily has two parts: the special matcher used in Lily takes 3.1 hours for the preprocessing, but the matching computing and postprocessing only spend 13 minutes. It indicates that if we use other literal-based matchers, we would have close running time

[1] Ontology Alignment Evaluation Initiative http://oaei.ontologymatching.org/

Table 1. Matching results of systems on Anatomy

System	Runtime	Precision	Recall	F-Measure	Recall+
Label Eq.	–	0.981	0.613	0.755	0.000
Lily	3.1h+13min	0.796	0.693	0.741	0.470
Falcon-AO	12min	0.963	0.599	0.738	0.127
TaxoMap	25min	0.460	0.764	0.574	0.470

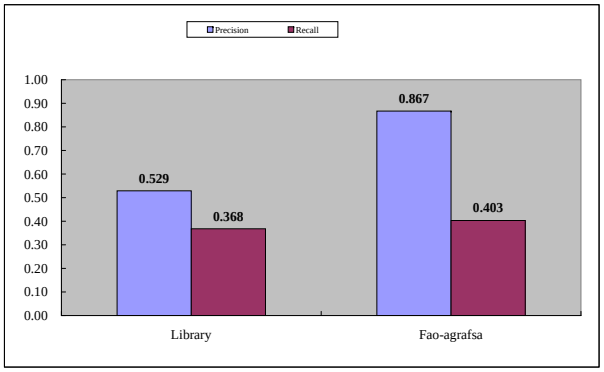


Fig. 2. Matching results of Lily on the real large ontologies

to other systems. (4) Lily and Taxomap have high Recall+ value, it means that they have the ability to discover the difficult alignments, but Lily has better F-measure.

The results of Lily on the Library and Fao tasks are showed as Fig. 2, which also demonstrates that it can discover some alignments in the two tasks.

5 Conclusion

This paper present a system Lily-LOM, which proposes a new large ontology matching method based on reduction anchors. The reduction anchors are useful to predict the ignorable similarity calculations during matching, that can reduce the high time complexity problem.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61003156).

References

1. Peng Wang and Baowen Xu. Matching large ontologies. Technical Report WIP-TR-2009-02, Southeast university, http://cse.seu.edu.cn/people/pwang/publication/WIP-TR-2009-02.pdf, 2009.
2. Caterina Caracciolo, Jrme Euzenat, Laura Hollink, Ryutaro Ichise, and et al. Results of the ontology alignment evaluation initiative 2008. In *The Third International Workshop on Ontology Matching (OM2008), Karlsruhe, Germany.*, 2008.