On Applying Matching Tools to Large-Scale Ontologies

Heiko Paulheim

SAP Research
heiko.paulheim@sap.com

Abstract. Many existing ontology matching tools are not well scalable. In this paper, we present the *Malasco* system, which serves as a framework for reusing existing, non-scalable matching systems on large-scale ontologies. The results achieved with different combinations of partitioning and matching tools are discussed, and optimization techniques are examined. It is shown that the loss of result quality when matching with partitioned data can be reduced to less than 5% compared to matching with unpartitioned data.

1 Introduction

The need for matching large-scale ontologies arises in different fields. In electronic business, several large ontologies representing business standards are in use [1]. Another example is the field of medical research where large databases and ontologies exist in which taxonomies, definitions, and experimental results are stored.

To the best of our knowledge, there are only three works which explicitly address the scalability issue: The schema matching system COMA++ [2], the ontology matching tool Falcon-AO [3], and the MOM approach [4]; however, there seems to be no publicly available implementation of the latter. All of them address the scalability problem by first partitioning the input ontologies into smaller sub-ontologies and then performing the actual matching task on the partitions. This approach seems promising, although one must take care to implement the partitioning step in a way that large ontologies can be processed, in order not to replace one bottleneck with another.

2 Scalability of Existing Matching Tools

To examine the scalability of existing ontology matching tools, we used two pairs of large ontologies: the e-business standards *eClass* (375K triples) and *UNSPSC* (83K triples), and the medical ontologies *GO* (465K triples) and *NCI thesaurus* (543K triples). From the large variety of matching tools, we chose tools that are publicly available and widely known, two of which focus explicitly on the matching of large-scale ontologies. We conducted tests with the above mentioned

COMA++ and Falcon-AO, as well as FOAM, INRIA, PROMPT, and CROSI (using a simple string comparator), on a standard desktop PC.

The business pair of only be processed by COMA++ and Falcon-AO. The larger medical pair could not be processed by any of the tools examined. Most of the tools suffered from a lack of memory. These experiments show that matching large ontologies is a severe problem with many of the tools that are currently available.

3 The Malasco System

The system introduced in this paper is called *Malasco* (**Ma**tching **la**rge **sc**ale **o**ntologies). It allows matching large-scale ontologies by first partitioning the input ontologies. The actual matching is then carried out on the smaller partitions.

3.1 Design

This approach has also been implemented in COMA++ and Falcon-AO. Unlike those systems, our implementation follows a more modular design, which allows the use of different existing systems both for partitioning and for matching the partitions. This approach has several advantages:

- Existing matching and partioning tools can easily be reused. This lowers the effort of setting up a matching solution and offers the possibility to benefit from future developments without having to modify the system.
- Different matching tools provide results of various quality, depending on the nature of the input ontologies. Therefore, building a system that can work with different matching tools is a promising approach for creating a versatile tool.
- From an academical point of view, the approach allows experiments on different combinations of partitioning and matching tools.

3.2 Partitioning approaches

As a simple partitioning approach, we implemented a naive baseline algorithm which iterates over the RDF sentences [5] and creates chunks of N triples. While that approach is rather naive (as it does not create clusters of concepts that are semantically related), two more sophisticated algorithms are used in the prototype: the islands algorithm developed by Stuckenschmidt and Klein [6], implemented in the tool *PATO* and the ε-connections algorithm proposed by Grau et al. [7], implemented in the tool *SWOOP*.

4 Evaluation

The Malasco system has been evaluated in two respects: the ability to process large-scale ontologies, and the quality of the matching results achieved.

4.1 Scalability

To demonstrate that our system is capable of matching large-scale ontologies, we used the test ontologies and test environment described in section 2. As an example, we used the baseline algorithm with a maximum partition size of 5,000 statements and the INRIA matching system. Our system could process both pairs; the largest amount of time – more than 100 times longer than the rest of the process – was consumed by the pairwise matching of partitions.

4.2 Result Quality

While it is obvious that element-based matching algorithms can be run on partitions of the input ontologies with unchanging results (given a covering partitioning), most matching systems are structure-based and will thus produce different (and probably worse) results on partitioned data. To evaluate how big the loss of quality is when working on partitioned data, we ran the example matching tools both on unpartitioned and on partitioned ontologies and compared the results. Since the matching tools examined can only work on smaller-scale ontologies, such a comparison is only possible on smaller-scale data sets.

Six pairs of ontologies of a size between 600 and 2,000 statements were used for evaluation. For partitioning, we used two variants of the baseline algorithm (with 250 and 500 statements as a maximum), two variants of the islands algorithm (with 50 and 100 classes per island as a maximum), the ε-connections algorithm[1], and the unpartitioned ontologies for comparison. For pairwise matching of the partitions, we used INRIA [8] and FOAM [9] in their respective standard configurations (both of which partly rely on structure-based algorithms).

To evaluate the results, we calculated recall, precision, and F-measure. While the recall value achieved on partitioned data is as high as (and in some cases even slightly higher than) the result on unpartitioned data, the precision value is less than 50% than that achieved on unpartitioned data, caused by a very high number of false positives.

4.3 Optimization I: Using overlapping partitions

To achieve better results, in particular better precision values, we tested two optimization approaches. The first one is motivated by the insight that structure-based matching approaches use information on neighboring elements. For partioned ontologies, those are missing for elements on the border of a partition. Hence, for the first optimization approach, we added the direct neighbors for each concept contained in a partition, thus creating overlapping partitions. The matching is then performed on the overlapping partitions. Mapping elements found between the neighboring elements are discarded, because the matching algorithm only has partial information about those elements.

When using overlapping partitions, it can be observed that using overlapping partitions causes a significant improvement of the precision value (the loss

[1] For the ε-connections algorithm, various problems can be observed [7]; two ontologies could not be partitioned at all. Therefore, that algorithm is considered not suitable and not regarded any further in the following results.

of precision can be limited to less than 20%), almost without any negative affection of the recall value. On the other hand, since the overlapping partitions are larger, the matching phase runs up to four times as long as for non-overlapping partitions.

4.4 Optimization II: Thresholding

The second approach to improve our system's results' quality is the use of a lower threshold. As most matching system provide a confidence parameter with each mapping element, a lower threshold can be employed to discard all elements with a confidence value below that threshold in order to improve the results [10]. This approach has been motivated by the observation that the average confidence value is significantly lower for false positives than for true positives.

To determine an optimal lower threshold τ, we calculated precision, recall, and F-measure for threshold values between 0 and 1 and determined the average optimal (w.r.t. F-measure) threshold values for each partitioning algorithm, including the unpartitioned case for comparison.

Thresholding the results leads to a significant improvements in precision and F-measure. Fig. 1 shows the results using the matching system FOAM[2]. The improvement is stronger than using overlapping partitions, more than 95% of the F-measure achieved on unpartitioned data can be reached (even up to 99% for INRIA). Applying a filter which is optimal for a given partitioning technique leads to almost the same results, thus, the choice for an actual partitioning algorithm is of marginal effect.

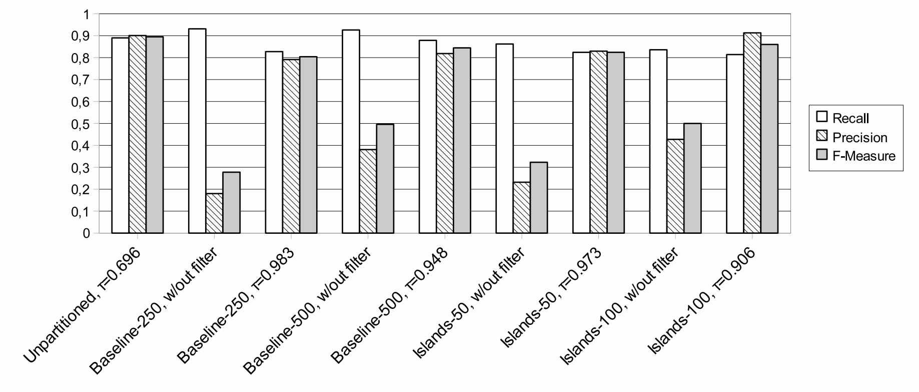

Fig. 1. Results with FOAM and thresholding

Using the thresholding optimization is less costly than using overlapping partitions: the matching system does not have to work on larger partitions, and the runtime complexity of applying the threshold is only linear in the number of results. Combining both overlapping partitions and thresholding leads only to

[2] The results with INRIA were in most of the cases comparable to those achieved with FOAM and are therefore not shown separately.

minimal improvements (less than 5%) compared to thresholding alone. Since using overlapping partitions is rather costly, thresholding alone is the more approriate approach in most usage scenarios.

5 Conclusion

In this paper, we have presented the Malasco framework which allows using existing matching tools for matching large-scale ontologies. Its modular architecture allows for using arbitrary partitioning and matching tools, including domain-specific tools for particular matching tasks.

In our evaluation, we have shown that our system is actually capable of matching large ontologies, that the choice of a particular partitioning algorithm is only of minor importance, and that the quality deviation compared to the results which would be achieved on the unpartitioned ontologies (given a matcher that could process them) can be reduced to less than 5%.

Acknowledgements

The author would like to thank the members of the e-business integration group at Hochschule Darmstadt and the members of the knowledge engineering group at TU Darmstadt for their valuable and helpful remarks.

References

1. Rebstock, M., Fengel, J., Paulheim, H.: Ontologies-based Business Integration. Springer (2008)
2. Do, H.H., Rahm, E.: Matching large schemas: Approaches and evaluation. Information Systems **32**(6) (2007) 857–885
3. Hu, W., Zhao, Y., Qu, Y.: Partition-based block matching of large class hierarchies. [11]
4. Wang, Z., Wang, Y., Zhang, S., Shen, G., Du, T.: Matching large scale ontology effectively. [11]
5. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on rdf sentence graph. In Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J., eds.: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, ACM (2007) 707–716
6. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. [12] 289–303
7. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing OWL ontologies. In Sleeman, D., Alani, H., Brewster, C., Noy, N., eds.: Proceedings of the KCAP-2005 Workshop on Ontology Management. (2005)
8. Euzenat, J.: An API for ontology alignment. [12] 698–712
9. Ehrig, M.: Ontology Alignment - Bridging the Semantic Gap. Semantic Web and Beyond. Computing for Human Experience. Springer, New York (2007)
10. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Berlin, Heidelberg, New York (2007)
11. Mizoguchi, R., Shi, Z., Giunchiglia, F., eds.: The Semantic Web - ASWC 2006, First Asian Semantic Web Conference. Number 4183 in LNCS, Springer (2006)
12. McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: The Semantic Web - ISWC 2004: Proceedings of the Third International Semantic Web Conference. Number 3298 in LNCS, Springer (2004)