Web Semantics: Science, Services and Agents on the World Wide Web 6 (2008) 237–239

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Falcon-AO: A practical ontology matching system

Wei Hu*, Yuzhong Qu

School of Computer Science and Engineering, Southeast University, Nanjing 210096, PR China

ARTICLE INFO

Article history:
Received 18 October 2007
Received in revised form 21 January 2008
Accepted 29 February 2008
Available online 11 April 2008

Keywords:
Ontology matching
Data integration
Semantic interoperability

ABSTRACT

In this paper, we introduce a general overview of Falcon-AO: a practical ontology matching system with acceptable to good performance and a number of remarkable features. Furthermore, Falcon-AO is one of the best systems in all kinds of tests in the latest three years' OAEI campaigns. Falcon-AO is written in Java, and is open source.

1. Introduction

Falcon is an infrastructure for Semantic Web applications, which aims at providing fundamental technologies for finding, aligning and learning ontologies, and, ultimately, for capturing knowledge from the Web via an ontology-driven approach.

Falcon-AO, a prominent component of Falcon, is an automatic ontology matching system that helps actualize interoperability between (Semantic) Web applications that use different but related ontologies. Recently, it has become a very practical and popular choice for matching Web ontologies expressed by RDF(S) and OWL. Falcon-AO is implemented in Java, and, presently, it is an open source project under the Apache 2.0 license.

In this paper, we introduce the system architecture of Falcon-AO (Section 2), present its unique features and capabilities (Section 3), point out its strengths in performance (Section 4) and, finally, conclude with some future work (Section 5). For technical details, we refer the reader to the papers [4–8] and the website: http://iws.seu.edu.cn/projects/matching/.

2. System architecture

Fig. 1 illustrates the system architecture of Falcon-AO, which consists of five components:

- *Model Pool* parses input ontologies into models (in memory) by Jena,[1] and adjusts models by using a set of coordination rules [6].
- *Matcher Library* manages a collection of elementary matchers. As of today, four representative matchers are integrated to make up the matcher library: V-Doc [7] and I-Sub [8] are two light-weight linguistic matchers; GMO [4] is an iterative structural matcher; PBM [5] adopts the divide-and-conquer strategy to find block mappings between large-scale ontologies.
- *Alignment Set* generates alignments by using a widely-accepted RDF/XML format [2] and evaluates generated alignments against reference alignments based on the conventional precision/recall metrics.
- *Central Controller* allows manual configuration of matching strategies. Also, it executes matchers and combines similarities based on measures of linguistic comparability and structural comparability.
- *Repository* stores reusable data during the matching process.

Furthermore, Falcon-AO implements a *graphical user interface* (GUI) to make the provided functionality, such as setting matching parameters, viewing and manipulating exported alignments, easily accessible to users.

3. Features and capabilities

In this section, we introduce some distinguishing features and capabilities of Falcon-AO, including multiple elementary matchers (V-Doc, GMO and PBM), the coordination rules, and the similarity combination strategy.

3.1. Linguistic matching

V-Doc [7] takes a linguistic approach to ontology matching. Its novelty is the idea of constructing virtual documents. Basically, as

* Corresponding author. Tel: +86 25 5209 0908; fax: +86 25 5209 0880.
E-mail addresses: whu@seu.edu.cn, whu1982@gmail.com (W. Hu), yzqu@seu.edu.cn (Y. Qu).

[1] http://jena.sourceforge.net/.

1570-8268/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2008.02.006

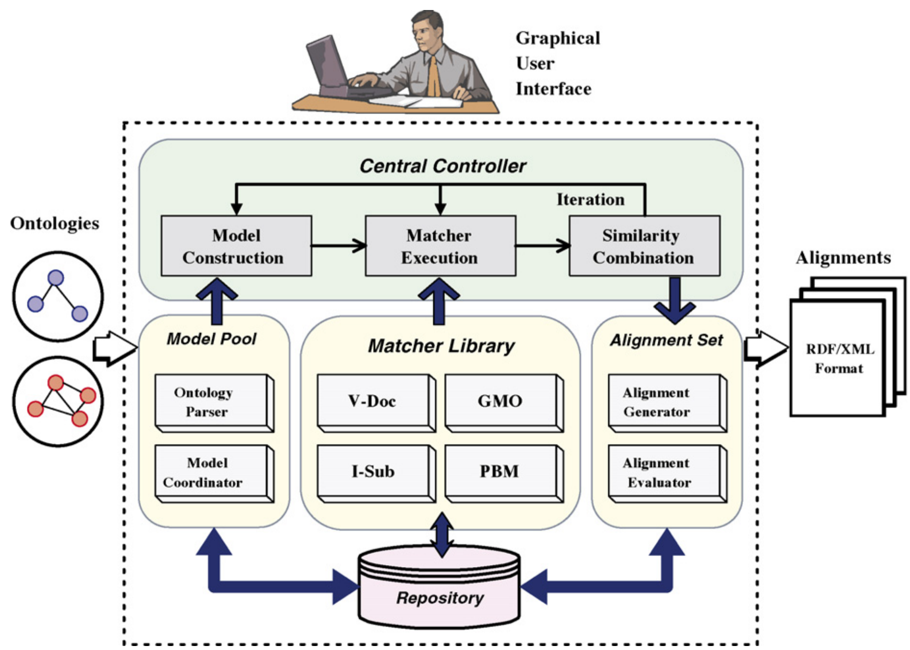

Fig. 1. System architecture.

a collection of weighted words, the virtual document of a domain entity (e.g., a class or a property) in an ontology contains not only the local descriptions, but also the neighboring information to reflect the intended meaning of the entity. Document similarity can be calculated via traditional vector space techniques, and further be used in certain similarity-based approaches to ontology matching. Specifically, the RDF graph structure is exploited to obtain the description information from neighboring domain entities.

3.2. Structural matching

GMO [4] is an iterative structural matcher. It uses RDF bipartite graphs to represent ontologies and computes structural similarities between domain entities and between statements (triples) in ontologies by recursively propagating similarities in the bipartite graphs. GMO takes a set of external alignments as input, which are typically found previously by other matchers (in current implementation, the external alignments are the ones with high similarities that are from V-Doc and I-Sub), and incrementally generates extra alignments as output. The performance of GMO improves as the precision of external alignments increases.

3.3. Partition-based block matching of large-scale ontologies

Large-scale ontologies raise a big challenge to existing ontology matching systems because of their size and their monolithic nature. PBM uses a divide-and-conquer approach to finding block mappings between large-scale ontologies [5], which has two major advantages: (1) it avoids our matching system suffering from lack of memory; and (2) it decreases the execution time without loss of quality, because it is likely that large portions of one or both input ontologies have no matching counterparts.

In particular, PBM firstly partitions domain entities of each ontology into a number of small clusters based on their structural proximity (e.g., the distance between classes in the class hierarchy, and the overlapping between the domains of properties), and then builds blocks by assigning RDF sentences to the clusters. RDF sentences can provide more integrated syntactic and semantic structures than RDF statements, because they can encapsulate blank nodes into them. Finally, blocks are matched via anchors (i.e., pre-found alignments by I-Sub) and only block pairs with high similarities are further matched by V-Doc and GMO.

3.4. Coordination rules

Due to the heterogeneous ways in expressing semantics and the various inferencing capabilities of ontology languages, ontologies are often represented differently. So, it is necessary to adjust ontologies before executing elementary matchers. Falcon-AO implements 21 coordination rules to eliminate superfluous axioms and reduce structural heterogeneity between the ontologies to be matched.

Specifically, three categories of coordination rules can be assigned to elementary matchers [4]: (1) removing redundant statements; (2) inferring omitted statements, e.g., the ones involving *owl:inverseOf*; and (3) reconstructing List structures, e.g., using the *rdfs:member* property to describe the relationship between a list and each of its members, instead of RDF collection vocabularies (*rdf:first*, *rdf:rest* and *rdf:nil*).

3.5. Similarity combination strategy

Similarity combination is an important and difficult issue in building ontology matching systems. Falcon-AO develops an approach to gradually tune up the thresholds (cutoffs) based on the measures of both the linguistic comparability and the structural comparability, which makes Falcon-AO robust in a variety of matching scenarios.

The linguistic comparability is calculated by examining the proportion of the candidate alignments against the minimum number of domain entities in the ontologies. The intuition is that if the number of alignments is close to the number of domain entities in the smallest ontology, then we are almost done with matching, and it is not necessary to run GMO any more. The structural comparability is calculated by comparing which built-in properties are used in the ontologies, and how often. Furthermore, it estimates the number of correct alignments from GMO in proportion to the ones from V-Doc and I-Sub.

Table 1
Performance of Falcon-AO in OAEI 2007

	Benchmark	Anatomy	Directory	Food	Environment	Library	Conference
Precision	0.92	0.96	0.55	0.84	0.87	0.97	0.73
Recall	0.86	0.59	0.61	0.45	0.35	0.87	0.57
Total time	300 s	12 min	110 s	5.75 h	1.75 h	40 min	160 s

Falcon-AO considers these two kinds of comparability to automatically determine the similarity combination strategy. For example, if the linguistic comparability is high, Falcon-AO would lower the thresholds of V-Doc and I-Sub, so that more alignments from V-Doc and I-Sub can be combined to the final alignments.

4. Performance

The Ontology Alignment Evaluation Initiative (OAEI)[2] is an international campaign commencing in 2004, which aims at establishing a consensus on the evaluation and comparison of matching systems. The proposed matching tasks cover a large portion of real world domains, and the differences between them are significant (e.g., ontology scalability, evaluation modality). Therefore, it could be considered as a comprehensive test of existing matching systems.

Falcon-AO has continuously participated in the last three years' OAEI campaigns (i.e., OAEI 2005, 2006, and 2007). It has been recognized by the organizers [3] as one of the best ontology matching systems (the other remarkable systems include COMA++ [1] and RiMOM [9]). As compared to other systems, Falcon-AO has three strengths: (1) it can accomplish various matching tasks, especially matching large-scale ontologies; (2) it can stably achieve very good precision and recall on both systematic and blind tests. For instance, the average precision and recall of Falcon-AO in OAEI 2007 are 0.83 and 0.61, and the actual performance for all the tasks is not far from this average (see Table 1); and (3) it is efficient, all the tasks can be carried out in a reasonable time only on an ordinary personal computer. For small ontologies, Falcon-AO can complete the matching process within several seconds, even for large-scale ontologies, Falcon-AO can accomplish them within a few hours.

5. Conclusion

In this paper, we have introduced a new generic ontology matching system, Falcon-AO, which offers a comprehensive infrastructure to solve a large part of the ontology matching problem. Following a flexible architecture, Falcon-AO supplies a library of representative matchers, and provides a robust combination of their alignments. Through participating in the latest three years' OAEI campaigns, Falcon-AO has demonstrated its practicability for matching a variety of ontologies.

In the near future, we look forward to extending Falcon-AO in several different directions. Firstly, we are planning to integrate some new kinds of matchers, especially instance-based approaches. Secondly, we would like to go beyond ontology matching, and discover alignments between ontologies and relational database schemas. Finally, we hope to support data transformation by deriving executable mapping expressions from schema matching.

Acknowledgments

This work is funded in part by the NSFC under Grant 60573083, and in part by the 973 Program of China under Grant 2003CB317004. We thank the following people for their valuable contributions to the implementation: Ningsheng Jian, Gong Cheng, Xinyu Zhong, Dongdong Zheng, Yuanyuan Zhao, and Dan Li. We appreciate Willem van Hage and Spyros Kotoulas for their valuable comments.

References

[1] H.H. Do, E. Rahm, COMA: a system for flexible combination of schema matching approaches, in: Proceedings of the 28th International Conference on Very Large Data Bases, ACM Press, 2002, pp. 610–621.
[2] J. Euzenat, An API for ontology alignment, in: Proceedings of the 3rd International Semantic Web Conference, LNCS, vol. 3298, Springer, 2004, pp. 698–712.
[3] J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Šváb, V. Svátek, W.R. van Hage, M. Yatskevich, Results of the ontology alignment evaluation initiative 2007, in: Proceedings of ISWC+ASWC Workshop on Ontology Matching, 2007, pp. 96–132.
[4] W. Hu, N. Jian, Y. Qu, Y. Wang, GMO: a graph matching for ontologies, in: Proceedings of K-CAP Workshop on Integrating Ontologies, 2005, pp. 41–48.
[5] W. Hu, Y. Zhao, Y. Qu, Partition-based block matching of large class hierarchies, in: Proceedings of the 1st Asian Semantic Web Conference, LNCS, vol. 4185, Springer, 2006, pp. 72–83.
[6] N. Jian, W. Hu, G. Cheng, Y. Qu, Falcon-AO: aligning ontologies with Falcon, in: Proceedings of K-CAP Workshop on Integrating Ontologies, 2005, pp. 85–91.
[7] Y. Qu, W. Hu, G. Cheng, Constructing virtual documents for ontology matching, in: Proceedings of the 15th International World Wide Web Conference, ACM Press, 2006, pp. 23–31.
[8] G. Stoilos, G. Stamou, S. Kollias, A string metric for ontology alignment, in: Proceedings of the 4th International Semantic Web Conference, LNCS, vol. 3729, Springer, 2005, pp. 624–637.
[9] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, K. Wang, Using Bayesian decision for ontology mapping, Journal of Web Semantics 4 (4) (2006) 243–262.

[2] http://oaei.ontologymatching.org/.