Falcon-AO: Results for OAEI 2007

Wei Hu, Yuanyuan Zhao, Dan Li, Gong Cheng, Honghan Wu, and Yuzhong Qu

School of Computer Science and Engineering, Southeast University,
Nanjing 210096, P.R. China
{whu,yyzhao,dli,gcheng,hhwu,yzqu}@seu.edu.cn

Abstract. In this paper, we present a brief overview of Falcon-AO (version 0.7): a practical ontology matching system with acceptable to very good performance, a flexible architecture, and a number of unique features. We also show some preliminary results of Falcon-AO for this year's OAEI campaign: evaluation on seven different matching tasks.

1 Presentation of the system

As an infrastructure for Semantic Web applications, Falcon is a vision of our research group. It desires for providing fantastic technologies for finding, aligning and learning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is still under development in our group.

1.1 State, purpose, general statement

As a prominent component of Falcon, Falcon-AO starts as an automatic ontology matching system to help enable interoperability between (Semantic) Web applications using different but related ontologies. It has since become a very practical and popular tool for matching Web ontologies expressed in RDFS or OWL. To date, Falcon-AO is continually being improved and elaborated, and the latest version is 0.7.

1.2 Specific techniques used

Falcon-AO is implemented in Java, and it is an open source project under the Apache license. Fig. 1 exhibits the architecture of Falcon-AO (version 0.7). It consists of five components: the *Repository* to temporarily store the data during the matching process; the *Model Pool* to manipulate ontologies and to construct different models for different matchers; the *Alignment Set* to generate and to evaluate exported alignments; the *Matcher Library* to manage a set of elementary matchers; the *Central Controller* to configure matching strategies and to execute matching operations. Furthermore, Falcon-AO provides a graphical user interface (GUI) to make it easily accessible to users.

Due to space limitation, we only provide a brief overview of Falcon-AO's features in this paper. For more details, we refer the reader to the technical papers [1–6], and our website: http://iws.seu.edu.cn/projects/matching/.

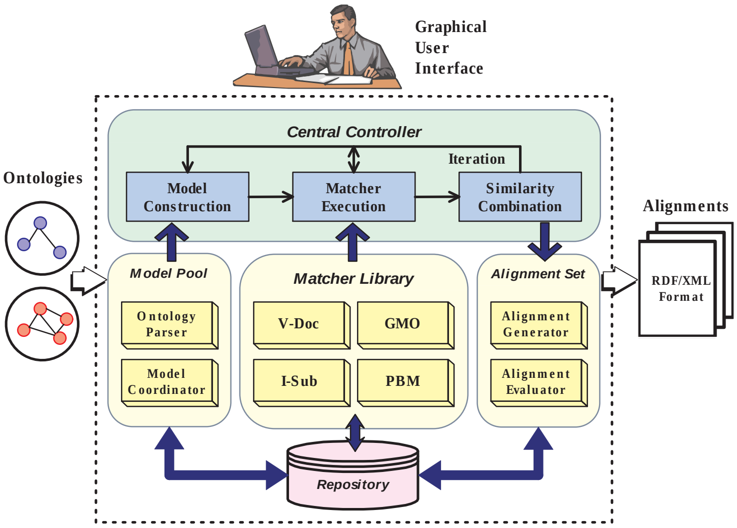

Fig. 1. System Architecture

– V-Doc, I-Sub, GMO, PBM: The four distinguishing elementary matchers make up the core matcher library of Falcon-AO. V-Doc [5] discovers alignments via revealing the usage (context) of domain entities in ontologies (i.e. the neighboring information). I-Sub [6] is a light-weighted matcher based on a refined string comparison technique (i.e. considering both commonality and difference). GMO [2] measures the structural similarity between RDF bipartite graphs based on propagating similarities between domain entities and statements. PBM [3] follows the divide-and-conquer idea to partition large ontologies into small blocks and construct mappings between the blocks for further matching with V-Doc, I-Sub, and GMO.

– Model Coordinator [4]: 21 coordination rules are used to eliminate useless axioms and reduce structural heterogeneity between the ontologies to be matched. Specifically, Falcon-AO can apply different coordination rules to different matchers. As an example, for I-Sub, Falcon-AO only removes ontology headers. As another example, for GMO, Falcon-AO uses the *rdfs:member* property to express the relationship between a list and each of its members, instead of using RDF collection vocabularies (*rdf:first*, *rdf:rest* and *rdf:nil*).

– Alignment Generator [1, 4]: Alignments are generated in terms of the observation of the linguistic comparability (LC) and the structural comparability (SC). LC is calculated by examining the proportion of the candidate alignments against the minimum number of the domain entities in the two ontologies; while SC is computed by comparing the built-in vocabularies in the two ontologies as well as estimating the correct alignments found by GMO in the portion of the reliable ones from V-

Doc and I-Sub. Falcon-AO uses these two kinds of comparability to automatically adjust the thresholds for selecting and combining alignments.

1.3 Adaptations made for the evaluation

We do not make any specific adaptation in the OAEI 2007 campaign. All the alignments outputted by Falcon-AO are uniformly based on the same parameters.

1.4 Link to the system and parameters file

The latest version of Falcon-AO can be downloaded from our website:
http://iws.seu.edu.cn/projects/matching/res/falcon.zip.

1.5 Link to the set of provided alignments (in align format)

The alignments for all the tasks in this year's OAEI campaign are available at:
http://iws.seu.edu.cn/projects/matching/res/falcon_2007.zip.

2 Results

In this section, we present the results of Falcon-AO (version 0.7) for the tasks provided by the OAEI 2007 campaign.

2.1 *Benchmark*

The *benchmark* task can be divided into five groups: #101–104, #201–210, #221–247, #248–266 and #301–304. The results of Falcon-AO are reported on each group in correspondence. The summary of the average performance of Falcon-AO (version 0.7) on the *benchmark* task is depicted in Table 1. For more details, please link to Appendix.

Table 1. Summary of the average performance on the *benchmark* task

	1xx	2xx	3xx	Average	H-mean	Time
Precision	1.00	0.93	0.89	0.93	0.92	300s
Recall	1.00	0.80	0.77	0.81	0.86	

#101–104 Falcon-AO performs perfectly on this group. Please pay attention to #102, Falcon-AO can automatically detect the two candidate ontologies are completely different, because both the linguistic comparability and the structural comparability between them are extremely low.

#201–210 Although in this group, some linguistic features of the candidate ontologies are discarded or modified, their structures are quite similar. So GMO takes much effect on this group. For example, in #202, 209, and 210, only a small portion of alignments are found by V-Doc or I-Sub, the rest are all generated by GMO. Since GMO runs much slower, it takes Falcon-AO more time to exploit all the alignments.

#221–247 The structures of the candidate ontologies in this group are altered. However, Falcon-AO discovers most of the alignments from the linguistic perspective via V-Doc and I-Sub, and both the precision and recall are pretty good.

#248–266 Both the linguistic and structural characteristics of the candidate ontologies are changed significantly, thus the tests in this group are the most difficult ones in the *benchmark* task. In some cases, Falcon-AO rarely finds any alignments. But indeed, in these cases, it is really hard to recognize the correct alignments due to lack of clues.

#301–304 Four real-life ontologies of bibliographic references are taken in this group. The linguistic comparability between the two candidate ontologies in each test is high while the structural comparability is medium. It indicates that the outputs of Falcon-AO are mainly from V-Doc or I-Sub. Alignments from GMO with high similarities are also reliable to be integrated.

2.2 *Anatomy*

The *anatomy* real world case is to match the Adult Mouse Anatomy (denoted by *mouse*) and the NCI Thesaurus describing the human anatomy (tagged as *human*). *mouse* has 2,744 classes, while *human* has 3,044 classes. Falcon-AO firstly partitions *mouse* and *human* into 122 and 14 blocks respectively, and then finds 16 block mappings based on 1,139 anchors. After further running elementary matchers on such block mappings, 715 alignments are finally generated as output. The whole process spends about 12 minutes. The summary of the performance is exhibited in Table 2.

Table 2. Summary of the performance on the *anatomy* task

	Precision	Recall	Time
Mouse vs. Human	0.96	0.59	12m

2.3 *Directory*

The *directory* task consists of Web sites directories like Google, Yahoo! or Looksmart. To date, it includes 4,639 tests represented by pairs of OWL ontologies, where classification relations are modeled as *rdfs:subClassOf* relations. Falcon-AO is quite efficient in this task, and it only takes less than 2 minutes (110 seconds) to complete all the tests. The average performance is summarized in Table 3.

Table 3. Summary of the average performance on the *directory* task

	Precision	Recall	Time
Directory	0.55	0.61	110s

2.4 *Food*

The *food* task includes two SKOS thesauri – *AGROVOC* and *NALT*. Since Falcon-AO focuses on Web ontologies expressed in RDFS and OWL, we have to adopt two OWL version ontologies transformed by campaign organizers in this task. *AGROVOC* owns 28,439 classes, while *NALT* owns 42,326 classes. Falcon-AO partitions them into 442 and 235 blocks, respectively. Supported by 15,353 anchors, Falcon-AO discovers 154 block mappings and conducts elementary matchers on them. Finally, 15,300 alignments are generated, where 14,615 alignments hold equivalence relationships, 685 ones hold subsumption relationships (558 broad relationships and 127 narrow relationships). The whole process costs nearly 6 hours. The performance is shown in Table 4.

Table 4. Summary of the performance on the *food* task

	Precision	Recall	Time
AGROVOC vs. NALT	0.84	0.45	5.75h

2.5 *Environment*

Three SKOS thesauri are collected in this task – a new thesaurus named *GEMET* (5,284 classes, is partitioned into 17 blocks), and the two ones in the *food* task. When matching *GEMET* and *AGROVOC*, Falcon-AO discovers 18 block mappings via 1,352 anchors, and generates 1,384 alignments, including 1,360 alignments hold equivalence relationships and 24 ones hold narrow relationships. The whole process spends nearly 33 minutes. When matching *GEMET* and *NALT*, Falcon-AO discovers 22 block mappings via 1,230 anchors, and generates 1,374 alignments, including 1,353 alignments hold equivalence relationships and 21 ones hold narrow relationships. The whole process spends about 1.2 hours. The performance is shown in Table 5.

Table 5. Summary of the performance on the *environment* task

	Precision	Recall	Time
GEMET vs. AGROVOC	0.88	0.39	33m
GEMET vs. NALT	0.86	0.30	1.2h

2.6 *Library*

Participants of this task are expected to match two Dutch thesauri used to index books from two collections held by the National Library of the Netherlands (KB). *Brinkman* owns 5,221 classes, and *GTT* owns 35,194 classes. Falcon-AO firstly partitions them into 232 and 2,211 blocks respectively, and then exploits 223 block mappings based on 3,641 anchors. After further running elementary matchers on such block mappings, 3,697 alignments are finally generated as outputs, where 3,661 alignments hold equivalence relationships, and 36 ones hold subsumption relationships (including 23 broad relationships and 13 narrow ones). The whole process spends about 40 minutes. The summary of the performance is exhibited in Table 6.

Table 6. Summary of the performance on the *library* task

	Precision	Recall	Time
Brinkman vs. GTT	0.97	0.87	40m

2.7 *Conference*

91 matching tasks are generated from 14 different ontologies with respect to conference organization. Falcon-AO takes 160 seconds to complete all the tests. Some statistics of the average performance are summarized in Table 7.

Table 7. Summary of the performance on the *conference* task

	Precision	Recall	Time
Conference	0.73	0.57	160s

3 General comments

In this section, we present some comments on Falcon-AO's results as well as the OAEI 2007 test cases.

3.1 Comments on the results

Here, we would like to make a rough comparison between Falcon-AO's results in this year and the results in the OAEI 2006 campaign (see Fig. 2). It can be seen that on these four tasks, Falcon-AO has more or less improvement.

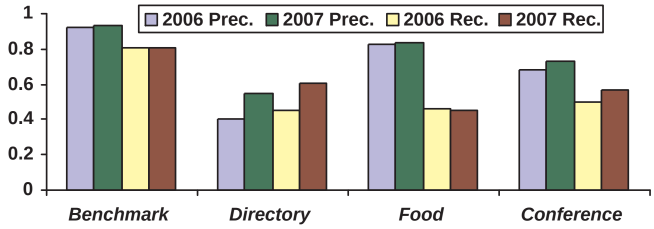

Fig. 2. Results in 2007 vs. Results in 2006

3.2 Comments on the OAEI 2007 test cases

The proposed matching tasks cover a large portion of real world domains, and the discrepancies between them are significant. Doing experiments on these tasks are helpful to improve algorithms and systems. In order to enhance applicability, we list some warnings as well as our modifications occurring in our experiment procedure, which might aid organizers to correct the problems in the future: (i) the prefix “rdfs” is not bound in “gemet_oaei2007.owl” in the *environment* task; and (ii) the encoding is inappropriate in the *library* task, and our modification is replacing “utf-8” by “iso-8859-1”.

4 Conclusion

Ontology matching is an important way to establish interoperability among (Semantic) Web applications using different but related ontologies. We implement a practical system for ontology matching called Falcon-AO. From the experimental experience in the OAEI 2007 campaign, we conclude that Falcon-AO (version 0.7) performs quite well and balancing on most of tasks. In the future, we look forward to making a stable progress towards building a comprehensive ontology matching system.

Acknowledgements

The work is funded by the NSFC under Grant 60573083. We would like to thank the following people for their valuable comments to Falcon-AO: Hai Leong Chieu, Antoine Isaac, Michel Klein, Adam Saltiel, Rubén Tous, Shenghui Wang, Songmao Zhang.

References

1. Hu, W., Cheng, G., Zheng, D., Zhong, X., and Qu, Y.: The results of Falcon-AO in the OAEI 2006 campaign. In Proc. of the ISWC 2006 workshop on Ontology Matching. (2006) 124–133
2. Hu, W., Jian, N., Qu, Y., and Wang, Y.: GMO: A graph matching for ontologies. In Proc. of the K-CAP 2005 workshop on Integrating Ontologies. (2005) 41–48
3. Hu, W., Zhao, Y., and Qu, Y.: Partition-based block matching of large class hierarchies. In Proc. of the 1st Asian Semantic Web Conference (ASWC'06). (2006) 72–83

4. Jian, N., Hu, W., Cheng, G., and Qu, Y.: Falcon-AO: Aligning ontologies with Falcon. In Proc. of the K-CAP 2005 workshop on Integrating Ontologies. (2005) 85–91
5. Qu, Y., Hu, W., and Cheng, G.: Constructing virtual documents for ontology matching. In Proc. of the 15th International World Wide Web Conference (WWW'06). (2006) 23–31
6. Stoilos, G., Stamou, G., and Kollias, S.: A string metric for ontology alignment. In Proc. of the 4th International Semantic Web Conference (ISWC'05). (2005) 623–637

Appendix: Raw Results

Tests are carried out on an Intel Core 2 Duo 2.13GHz desktop machine with 2GB DDR2 memory under Windows XP Professional operating system and Java 1.6 compiler.

Matrix of Results

In the following table, the results of Falcon-AO (v0.7) in the *benchmark* test are shown by precision (Prec.), recall (Rec.) and machine processing time. Here, the machine processing time is the sum of the time in model construction, matcher execution, similarity combination and results evaluation.

#	Name	Prec.	Rec.	Time
101	Reference alignment	1.00	1.00	2.7s
102	Irrelevant ontology	NaN	NaN	1.9s
103	Language generalization	1.00	1.00	1.2s
104	Language restriction	1.00	1.00	1.2s
201	No names	1.00	0.95	1.2s
202	No names, no comments	0.87	0.87	24.9s
203	No comments	1.00	1.00	0.7s
204	Naming conventions	0.98	0.98	1.2s
205	Synonyms	1.00	0.98	1.2s
206	Translation	1.00	0.93	1.1s
207		0.98	0.91	1.1s
208		1.00	1.00	0.7s
209		0.79	0.78	24.2s
210		0.81	0.80	24.3s
221	No specialization	1.00	1.00	1.1s
222	Flattened hierarchy	1.00	1.00	1.1s
223	Expanded hierarchy	1.00	1.00	1.1s
224	No instance	1.00	0.99	0.9s
225	No restrictions	1.00	1.00	1.1s
228	No properties	1.00	1.00	0.5s
230	Flattened classes	0.94	1.00	1.1s
231	Expanded classes	1.00	1.00	1.2s
232		1.00	0.99	1.0s
233		1.00	1.00	0.5s
236		1.00	1.00	0.4s
237		1.00	0.99	0.9s
238		1.00	0.99	1.1s
239		1.00	1.00	0.5s
240		1.00	1.00	0.6s
241		1.00	1.00	0.4s
246		1.00	1.00	0.4s
247		1.00	1.00	0.5s
248		0.85	0.84	23.5s
249		0.87	0.87	23.6s
250		1.00	0.27	0.4s
251		0.56	0.56	27.2s
252		0.71	0.71	26.3s
253		0.85	0.84	23.0s
254		1.00	0.27	0.5s
257		1.00	0.27	0.4s
258		0.54	0.54	26.4s
259		0.70	0.70	25.6s
260		1.00	0.31	0.4s
261		0.89	0.24	0.5s
262		1.00	0.27	0.4s
265		1.00	0.31	0.4s
266		0.89	0.24	0.5s
301	Real: BibTeX/MIT	0.91	0.82	0.9s
302	Real: BibTeX/UMBC	0.90	0.58	0.4s
303	Real: Karlsruhe	0.77	0.76	0.7s
304	Real: INRIA	0.96	0.93	15.9s