The Results of Falcon-AO in the OAEI 2006 Campaign

Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, and Yuzhong Qu

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

Abstract. In this paper, we briefly introduce the architecture of Falcon-AO (version 0.6) and highlight two major improvements in the current version. Falcon-AO successfully completes all the five alignment tasks in the OAEI 2006 campaign: `benchmark`, `anatomy`, `directory`, `food`, and `conference`, and some preliminary results are also reported in this paper. In the end, we present some comments about our results and lessons learnt from the campaign towards building a comprehensive ontology alignment system.

1 Presentation of the System

As an infrastructure for the 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. As a prominent component of Falcon, Falcon-AO is an automatic tool for aligning ontologies, which is dedicated to aligning the Web ontologies expressed in OWL Lite/DL. To date, Falcon-AO is continually being improved and elaborated, and currently the latest version is 0.6.

1.1 State, Purpose, General Statement

Falcon-AO is an automatic ontology alignment tool. There are three elementary matchers implemented in the current version: V-Doc [4], I-Sub [5], and GMO [1]. In addition, an ontology partitioner, PBM [2], is integrated into Falcon-AO to cope with large-scale ontologies. In order to coordinate all the elementary matchers with high quality, we devise a novel central controller, which is based on the observation of the linguistic comparability as well as the structural comparability. The architecture of Falcon-AO (version 0.6) is illustrated in Fig. 1.

Compared with our previous prototype (version 0.3) [3], Falcon-AO (version 0.6) is extended mainly in two aspects. One is the integration of PBM. The other is the design of the central controller. The details about the two improvements are presented in the next subsection. Besides, it is worthy of noting that we also refine the implementation of the elementary matchers to save the runtime of matching process.

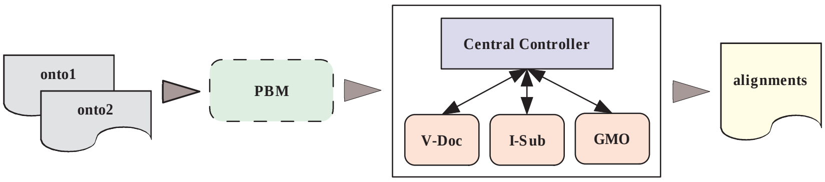

Fig. 1. The architecture of Falcon-AO (version 0.6)

1.2 Specific Techniques Used

To fit the requirements of different application scenarios, we have integrated three distinguishing elementary matchers, V-Doc, I-Sub, and GMO, which are regarded as independent components that make up of the core matcher library of Falcon-AO. Due to the space limitation, we only describe the key features of them. The technical details can be found in the related papers.

- V-Doc [4] discovers alignments by revealing the usage (context) of the domain entities in the ontologies to exploit their intended meanings. More precisely, words from the descriptions of domain entities as well as their neighboring information are simultaneously extracted to form the vectors in the word space, and the similarities between domain entities can be calculated in the Vector Space Model.

- I-Sub [5] is a light-weighted matcher simply based on the string comparison techniques. Its novelty is not only the commonalities between the descriptions of domain entities are calculated but also their differences are examined. Furthermore, it is stable to small diverges from the optimal threshold taking place.

- GMO [1] uses RDF bipartite graphs to represent ontologies, and measures the structural similarities between the graphs by the similarity propagation between domain entities and statements. An interesting characteristic is that GMO can still performs well even without any predefined alignment as input.

More importantly, two major improvements are taken in Falcon-AO (version 0.6). One is the integration of PBM for large-scale ontologies, while the other is the design of central controller.

PBM Due to the size and the monolithic nature of large-scale ontologies, exploiting alignments directly on the whole of them is quite difficult, inefficient, and also unnecessary. We develop an efficient ontology partitioner, PBM [2], to block matching of large-scale ontologies. In PBM, large-scale ontologies are hierarchically partitioned into blocks based on both the structural affinities and linguistic similarities, and then blocks from different ontologies are matched via predefined anchors. The overview of PBM is exhibited in Fig. 2. By applying V-Doc, I-Sub and GMO to the block mappings, we are finally able to generate alignments for large-scale ontologies more quickly while without loss of much accuracy.

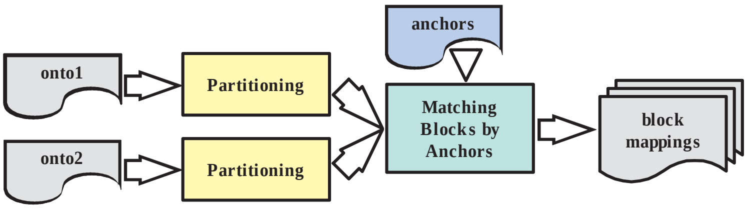

Fig. 2. The overview of PBM

Central Controller As presented above, we have introduced the features of the three elementary matchers, V-Doc, I-Sub and GMO. The question raised naturally here is how to integrate these matchers with ideal performance?

We propose a flexible integration strategy, which depends on the observation of the linguistic comparability as well as the structural comparability. Here, the linguistic comparability is computed by examining the proportion of the candidate alignments against the minimum number of domain entities in the two ontologies.

The calculation of the structural comparability is more complex. It firstly compares the built-in vocabularies used in the two ontologies. The basic assumption is the more built-in vocabularies are mutually included in the two ontologies, the more similar they might be in structure. But only measuring this is inadequate, we also compare the alignments found by V-Doc or I-Sub with high similarities to the alignments discovered by GMO, thus the reliability of the results of GMO can be estimated roughly.

The linguistic and structural comparability can be divided into three categories respectively: low, medium and high. If the comparability is low, it means that the alignments are probably unreliable. If the comparability is medium, the alignments with high similarities would be accepted by Falcon-AO. Otherwise, most of the alignments should be involved into the final output.

When the alignments generated by V-Doc, I-Sub and GMO are obtained, Falcon-AO integrates these alignments by considering the categories of the linguistic and structural comparability, following the rules below:

1. If the linguistic comparability is higher than the structural comparability, the outputted alignments mainly come from V-Doc and I-Sub.
2. If the linguistic comparability is lower than the structural comparability, the outputted alignments largely derived from GMO.
3. Otherwise, the outputted alignments are generated by making a combination among V-Doc, I-Sub and GMO with a weighting scheme.

1.3 Adaptations Made for the Evaluation

We don't make any specific adaptation for the tests in the OAEI 2006 campaign. All the alignments outputted by Falcon-AO are based on the same set of parameters.

1.4 Link to Falcon-AO

The latest version of Falcon-AO (version 0.6) is available at
http://xobjects.seu.edu.cn/project/falcon/matching/resources/falcon.zip,
or http://www.falcons.com.cn/falcon/falcon.zip.

1.5 Link to the Set of Provided Alignments

Full experimental results for all the tests in the OAEI 2006 campaign can be downloaded from http://xobjects.seu.edu.cn/project/falcon/matching/experiments/2006.zip,
or http://www.falcons.com.cn/falcon/2006.zip.

2 Results

The tests provided by the Ontology Alignment Evaluation Initiative (OAEI) 2006 campaign are composed of six categories, including: (a) `benchmark`; (b) `anatomy`; (c) `jobs`; (d) `directory`; (e) `food`; and (f) `conference`. Due to the `jobs` test needs to be further evaluated and discussed, in this section we only present the results of Falcon-AO (version 0.6) in the other five tests, i.e., `benchmark`, `anatomy`, `directory`, `food`, and `conference`.

2.1 `Benchmark`

The `benchmark` test might 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. Some more detailed results are listed in Appendix.

#101–104 Falcon-AO performs perfectly on the tests of this group. Please pay attention to #102, Falcon-AO could automatically detect the two candidate ontologies are totally different since 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 are altered in these tests. 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 heavily, so the tests in this group might be the most difficult ones in all the benchmark tests. In some tests, Falcon-AO doesn't perform well, but indeed, in these cases, it is really hard to recognize the correct alignments.

#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 but the structural comparability is moderate. It indicates that the outputs of Falcon-AO mainly come from V-Doc or I-Sub. Alignments from GMO with high similarities are also reliable to be integrated.

The summary of the average performance of Falcon-AO (version 0.6) on the benchmark test is depicted in Table 1.

Table 1. The average performance of Falcon-AO on the benchmark test

	1xx	2xx	3xx	H-mean	Time
Precision	1.00	0.97	0.98	0.92	472s
Recall	1.00	0.97	0.78	0.86	

2.2 Anatomy

The anatomy real world test bed covers the domain of body anatomy and consists of two ontologies, OpenGALEN and FMA, with approximate sizes of several 10,000 classes and several dozens of relations, respectively. By using PBM, Falcon-AO partitions OpenGALEN and FMA into 39 and 407 blocks, separately. Primarily 2,512 alignments are spotted as anchors, and then 42 block mappings are generated. After running further elementary matchers on these block mappings, totally 2,518 alignments are outputted in the end. The complete process takes over 5.5 hours. The experimental results of Falcon-AO (version 0.6) are exhibited in Table 2.

Table 2. The performance of Falcon-AO on the anatomy test

	Blocks	Anchors	Pairs	Alignments	Time
OpenGALEN	39	2512	42	2518	5.5h
FMA	407				

Most of these alignments seem credible since the labels of the two entities are the same when they are put into lowercase letters and the punctuation characters are taken out. But due to lack of domain knowledge about the field of anatomy, we couldn't make any further investigation.

2.3 Directory

The directory case consists of Web sites directories (like Google, Yahoo! or Looksmart). To date, it includes 4,639 matching tasks represented by pairs of OWL ontologies, where classification relations are modeled as *rdfs:subClassOf* relations.

Table 3. The performance of Falcon-AO on the directory test

Tasks	Precision	Recall	F-Measure	Time
4369	40.50%	45.47%	42.85%	280s

Falcon-AO is quite efficient in this test, and it only takes less than 5 minutes to complete all the matching tasks. Based on the manual observation, a large portion of generated alignments come from the linguistic perspective, i.e., V-Doc or I-Sub. The precision of Falcon-AO is 40.50%, the recall is 45.47%, and the F-Measure is 42.85%. We also experiment on the previous test set provided by the OAEI 2005 campaign, and the mapping quality seems moderate. The performance of Falcon-AO (version 0.6) on the directory test is summarized in Table 3.

2.4 Food

The food test case includes two SKOS thesauri, AGROVOC and NALT. Since Falcon-AO aims at the Web ontologies expressed in OWL Lite/DL, we firstly transform them into OWL ontologies. The transformation rules are listed as follows. Each concept is transformed into an *owl:Class*. Each broad or narrow relation is transformed into an *rdfs:subClassOf* relation. Each label written in English is reserved. All the other SKOS relations are discarded. Please note that this transformation is incomplete and even sometimes inaccurate.

Table 4. The performance of Falcon-AO on the food test

	Blocks	Anchors	Pairs	Alignments	Precision	Time
AGROVOC	1141	11919	253	13009	0.83	5.5h
NALT	950					

Then, Falcon-AO partitions the two corresponding OWL ontologies into 1,141 and 950 blocks, respectively. Supported by 11,919 anchors, Falcon-AO discovers 253 block mappings and runs further elementary matchers on them. Finally, 13,009 alignments are outputted. However, we merely consider the exact matching (equivalence). Currently, the broad or narrow relationship is not addressed in Falcon-AO. The whole process costs nearly 5.5 hours. According to the evaluation by the organizers, the precision is 0.83. The performance of Falcon-AO (version 0.6) is shown in Table 4.

2.5 `Conference`

The collection of tests is dealing with `conference` organization. At present, it includes 45 matching tasks, which are all composed of small ontologies. By comparing to the reference alignments provisionally made by track organizers, the precision of the alignments generated by Falcon-AO is 0.68, while the relative recall is about 0.50. Here, the relative recall is computed as the ratio of the number of all unique correct alignments (sum of all unique correct alignments per one system) to the number of all unique correct alignments found by any of systems (per all systems). In addition, Falcon-AO spends 109 seconds to finish all the matching tasks. Some statistics of the performance of Falcon-AO (version 0.6) are presented in Table 5.

Table 5. The performance of Falcon-AO on the `conference` test

Tasks	Precision	Recall	Time
45	0.68	0.50	109s

3 General Comments

In this section, we summarize some features of Falcon-AO, and discuss the improvement directions towards building a comprehensive ontology alignment system.

3.1 Comments on the Results

Different integration strategies of V-Doc, I-Sub, GMO and PBM lead to significantly different performance of Falcon-AO. In Table 6, we list the most important components that take effect on each test.

Table 6. The most important components integrated in each test

Tests	Components
`Benchmark`	V-Doc, I-Sub, GMO
`Anatomy`	PBM, V-Doc, I-Sub
`Directory`	V-Doc, I-Sub, GMO
`Food`	PBM, V-Doc, I-Sub, GMO
`Conference`	V-Doc, I-Sub

According to the experimental results on these tests shown in the previous section and the integration strategy shown in Table 6, we can analyze some strengths and weaknesses of Falcon-AO (version 0.6) clearly.

Strengths

- Falcon-AO (version 0.6) is a quite flexible ontology alignment tool. It copes with not only ontologies with moderate sizes but also very large-scale ontologies. Moreover, Falcon-AO integrates three distinguishing elementary matchers to manage different alignment applications, and the integration strategy is totally automatic.

- It achieves a good performance in both effectiveness and efficiency. Based on the reference alignments provided by the organizers and the check of human observation, the precision and recall in most cases are sound. Besides, Falcon-AO runs so fast that it only takes a few seconds to complete for ontologies with moderate sizes. Even for large ontologies, it still finishes the alignment tasks in an acceptable time.

Weaknesses

- The tuning of the algorithms within Falcon-AO is still a rigid process. For example, PBM performs well on the large ontologies with simple class hierarchy structures. But when the relations in ontologies are complicated (e.g., OpenGALEN), the partitioning quality of PBM is not sound.

- So far, we do not consider any domain knowledge in the current version of Falcon-AO. Hence, when Falcon-AO meets some applications from specific domains, it might fail to achieve a high quality result.

- Semantic relationship (e.g., equivalence, subsumption) offers general reasoning capability, which is the most prominent difference as compared to schema matching. But currently, Falcon-AO cannot provide alignments with semantic relationship.

3.2 Discussions on the Way to Improve the Proposed System

From the experiments we have learnt some lessons and plan to make improvements in the later versions. The following three improvements should be taken into account.

- While expressing the same thing, people may use synonyms and even different languages. So it is necessary to use lexicons or thesauri in the alignment process.

- The values of parameters used in Falcon-AO is mainly determined by manual setting. Some machine learning approaches can be involved to help automatic adjustment according to different application scenarios.

- The patching strategy for combining the alignments discovered by different matchers needs to be further discussed, e.g., adding some missing alignments, or deleting some wrong and redundant ones.

4 Conclusion

Ontology matching is a crucial task to enable interoperation between Web applications using different but related ontologies. We develop an automatic tool for ontology alignment, named Falcon-AO. From the experimental experience in the OAEI 2006 campaign, we can make a conclusion that Falcon-AO (version 0.6) performs well on most of tests. In our future work, we look forward to making a stable progress towards building a comprehensive ontology alignment system.

References

1. Hu, W., Jian, N., Qu, Y., and Wang, Y.: GMO: A graph matching for ontologies. In Proc. of the K-CAP workshop on Integrating Ontologies. (2005) 41–48
2. 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
3. Jian, N., Hu, W., Cheng, G., and Qu, Y.: Falcon-AO: Aligning ontologies with Falcon. In Proc. of the K-CAP workshop on Integrating Ontologies. (2005) 85–91
4. 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
5. 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 a PC running Windows XP with an Intel Pentium IV 3.0 GHz processor and 1GB memory.

Matrix of results

In the following table, the results of Falcon-AO in the `benchmark` test are provided with precision (Prec.), recall (Rec.) and machine processing time (Time). Here, the machine processing time is the sum of the time for ontology parsing, ontology matching, alignment generation and evaluation.

#	Name	Prec.	Rec.	Time
101	Reference alignment	1.00	1.00	5.6s
102	Irrelevant ontology	NaN	NaN	3.2s
103	Language generalization	1.00	1.00	2.2s
104	Language restriction	1.00	1.00	2.0s
201	No names	0.96	0.91	2.0s
202	No names, no comments	0.84	0.84	41.2s
203	No comments	1.00	1.00	1.2s
204	Naming conventions	0.96	0.96	1.9s
205	Synonyms	1.00	0.97	2.0s
206	Translation	0.98	0.93	2.0s
207		0.98	0.92	2.2s
208		1.00	1.00	1.1s
209		0.79	0.78	39.6s
210		0.81	0.80	39.2s
221	No specialization	1.00	1.00	1.9s
222	Flattened hierarchy	1.00	1.00	1.9s
223	Expanded hierarchy	1.00	1.00	2.1s
224	No instance	1.00	0.99	1.5s
225	No restrictions	1.00	1.00	1.8s
228	No properties	1.00	1.00	0.9s
230	Flattened classes	0.94	1.00	1.7s
231	Expanded classes	1.00	1.00	2.0s
232		1.00	0.99	1.5s
233		1.00	1.00	0.9s
236		1.00	1.00	0.7s
237		1.00	1.00	1.7s
238		1.00	1.00	2.0s
239		0.97	1.00	0.9s
240		0.97	1.00	1.1s
241		1.00	1.00	0.7s
246		0.97	1.00	0.8s
247		0.97	1.00	1.0s
248		0.86	0.85	38.2s
249		0.85	0.85	37.8s
250		1.00	0.27	0.8s
251		0.55	0.55	43.6s
252		0.71	0.71	42.3s
253		0.86	0.85	36.6s
254		1.00	0.27	0.8s
257		1.00	0.27	0.7s
258		0.56	0.56	43.6s
259		0.72	0.72	42.3s
260		0.90	0.31	0.8s
261		0.80	0.24	0.9s
262		1.00	0.27	0.7s
265		0.90	0.31	0.8s
266		0.80	0.24	0.9s
301	Real: BibTeX/MIT	0.89	0.80	1.5s
302	Real: BibTeX/UMBC	0.90	0.56	0.7s
303	Real: Karlsruhe	0.78	0.73	1.3s
304	Real: INRIA	0.95	0.92	25.4s