

# IAMA Results for OAEI 2013

Yuanzhe Zhang<sup>1</sup>, Xuepeng Wang<sup>1</sup>, Shizhu He<sup>1</sup>,  
Kang Liu<sup>1</sup>, Jun Zhao<sup>1</sup>, and Xueqiang Lv<sup>2</sup>

<sup>1</sup> Institute of Automation, Chinese Academy of Sciences, China

{yzzhang, xpwang, shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn

<sup>2</sup> Beijing Key Laboratory of Internet Culture and Digital Dissemination Research

lxq@bistu.edu.cn

**Abstract.** This paper presents the results of IAMA on OAEI 2013. IAMA (Institute of Automation's Matcher) is an ontology matching system with the capability to deal with large scale ontologies. IAMA is designed to find out the correspondences between two ontologies by using multiple similarity measures. Candidate filtering technique is adopted when processing ontologies at large scale.

## 1 Presentation of the system

### 1.1 State, purpose, general statement

Large amount of ontologies has been published since the semantic web emerged. However, managing the heterogeneity among various ontologies is still a problem [1]. For example, many ontologies describe the same entity (i.e., class or property) using different terminologies, while the entities having the same name belonging to different ontologies may refer to disparate objects. Finding the matching between different ontologies is still challenging. Ontology matching, as a solution to the aforementioned problem, has received great interests in these years.

The principal goal of IAMA is to discover equivalent entities rapidly between different ontologies. We use efficient terminology matching techniques and do not turn to any external resource at this stage. IAMA is able to match classes and properties of two input ontologies. The system could achieve qualified results, though neglecting the structural information. The Matching process takes little time to cope with small ontologies. When processing large scale ontologies, IAMA could still, with the help of candidate filtering, yield the alignment in reasonable time. We tend to make an universal and extensible system, so more matching methods could be conveniently incorporated in the future.

### 1.2 Specific techniques used

IAMA employs various similarity measures to take advantage of the available information in the ontologies. The entities in two ontologies are pairwise compared, and lexical similarities and structural similarities are calculated respectively. In the current version, only 1:1 alignment is considered.

Let  $O_1$  and  $O_2$  denote the two input ontologies, and  $e_1$  is an entity in  $O_1$ . Each entity  $e_2$  in  $O_2$  has a similarity with  $e_1$  indicated as  $\text{sim}(e_1, e_2)$ . We are able to find out the maximum value as  $\text{sim}(e_1, \hat{e})$ . If  $\text{sim}(e_1, \hat{e})$  is greater than a predetermined threshold  $t_1$ , entity pair  $(e_1, \hat{e})$  will be added to the alignment. In the following paragraphs, we will present the used similarity measures in our system.

### Lexical Similarity

The system extracts local names, labels, and comments of the entities in the two input ontologies as lexical features. For most situations, the lexical information is effective.

*Local Name similarity* measures the similarity between the names of two entities. We get rid of the spaces and other punctuations because the entity name is comprised of multiple words or contains hyphens at times. All the letters are turned to lower case simultaneously. *Label Similarity* measures the similarity between the labels. Not all the entities have labels, and many entities have a label exactly the same as its local name. *Comment Similarity* measures the similarity between the comments. A comment of an entity is usually a brief descriptive sentence, which is helpful when the two ontologies name their entities with quite different style. Both labels and comments are processed as local names, thereby treated as a single word.

IAMA uses Levenshtein [2] distance, which is proved competent in [3], to calculate lexical similarities. For the three lexical similarities mentioned above, we do not take them equally. Each similarity is assigned a weight intuitively. Local name similarity has a greater weight than label similarity, while comments similarity has the lowest weight.

### Individual Similarity

Between the classes that have individuals, *Individual Similarity* is additionally calculated. The names of individuals that belong to a class are extracted to a set of string. Assume  $S_1$  and  $S_2$  are two sets, then the similarity between them is computed as follows:

$$\text{sim}(S_1, S_2) = 2 \times \frac{\#(S_1 \cap S_2)}{\#S_1 + \#S_2} \quad (1)$$

For example, if  $c_1$  is a class in ontology  $O_1$ , and  $c_2$  is a class in ontology  $O_2$ . The names of the individuals belonging to  $c_1$  is a set of string  $i_1 = \{s_1, s_2, s_3\}$ , and similarly we get  $i_2 = \{s_2, s_3, s_4, s_5\}$ . The individual similarity  $\text{sim}_i(c_1, c_2)$  is:

$$\text{sim}_i(c_1, c_2) = 2 \times \frac{\#(i_1 \cap i_2)}{\#i_1 + \#i_2} = 2 \times \frac{2}{3 + 4} = 0.571$$

IAMA adopts the maximum value of all the similarities as the final similarity of the entity pair. It is worth noting that other similarities such as superclass similarity, subclass similarity, domain similarity and range similarity are also tested in our earlier attempts. But they contributed little considering the time increased. They could be added easily if needed, which makes IAMA extensible.

### Candidate Filtering

Pairwise compare is time consuming. In most cases, calculating similarities for every entity pair is unnecessary. *Candidate Filtering* helps to find out a few promising entity pairs in advance, thus saving running time dramatically.

Assume the two input ontologies are  $O_1$  and  $O_2$ , and  $O_2$  has more entities than  $O_1$ . For each entity in  $O_1$ , we attempt to find out potential entities in  $O_2$  to construct a candidate set. The idea is implemented as follow. First, the lexical information in the bigger ontology  $O_2$ , namely name, label and comment is tokenized and indexed by Lucene<sup>3</sup>. Second, we construct search query for each entity in  $O_1$ . For instance, the lexical information of an entity in  $O_1$  is "Reference", "Reference", "Base class for all entries". We split it into index tokens, and every single token is searched in the constructed index, yielding top-k entities as a candidate set. Last, our system calculates the final similarity values pairwise.

The time used for indexing and searching is acceptable. For large input ontologies, candidate filtering improves the matching speed substantially. Take anatomy track for example, the difference can be seen in Table 1. The experiment is conducted on a computer with 4.7GHz Intel i5 CPU (4 core) and 8GB RAM.

	Precision	F-Measure	Recall	Runtime (ms)
IAMA without candidate filtering	0.994	0.719	0.563	117,503
IAMA with candidate filtering	0.995	0.713	0.555	5,376

**Table 1.** The impact of candidate filtering in Anatomy track

Candidate filtering could still miss some potential entity pairs though negligible. IAMA defined an alterable trigger threshold  $t_2$ , which is set to 500 empirically. Only both the two ontologies have more than 500 entities, candidate filtering is employed.

### 1.3 Adaptations made for the evaluation

There are two key parameters in IAMA (i.e.,  $t_1$  and  $t_2$ ). Specifically, if the final similarity of an entity pair is greater than  $t_1$ , the pair will be added to the alignment.  $t_2$  is the trigger threshold of candidate filtering component as mentioned before. In the version to participate in OAEI 2013,  $t_1$  is set to 0.9 and  $t_2$  is set to 500.

### 1.4 Link to the system and parameters file

The latest version of IAMA can be seen on <https://github.com/YuanzheZhang/IAMA>.

## 2 Results

This section presents the results of IAMA achieved in OAEI 2013. Our system mainly focuses on benchmark, anatomy, conference, and large biomedical ontologies. We do not provide multilingual support for the moment.

---

<sup>3</sup> <http://lucene.apache.org>

## 2.1 benchmark

The goal of the benchmark data set is to provide a stable and detailed picture of each algorithm[4]. The benchmark test library consists of several test suits. The test suites are generated from the usual bibliography ontology this year, and they are blind to participants. Table 2 shows the results of benchmark track. Pt F-m./s means the average F-measure point provided per second.

Precision	F-Measure	Recall	Time (s)	pt F-m./s
0.99	0.73	0.57	102	0.72

**Table 2.** Results for Benchmark track

Our system acquired its best results in this track. Concerning F-measure, IAMA ranked fourth in the 21 systems. The comparison with other top systems is shown in Table 3

System	Precision	F-Measure	Recall	Time (s)	pt F-m./s
YAM++	0.97	0.89	0.82	702	0.13
CroMatcher	0.95	0.88	0.82	1,114	0.08
CIDER-CL	0.85	0.75	0.67	844	0.09
IAMA	0.99	0.73	0.57	102	0.72
ODGOMS	0.99	0.71	0.55	100	0.71

**Table 3.** Comparison with other top systems in Benchmark track

## 2.2 anatomy

The task of anatomy track is to find the alignment between the Adult Mouse Anatomy and a part of the NCI Thesaurus. These two ontologies describe the mouse anatomy and the human anatomy respectively. The results of our system on anatomy are shown in Table 4.

Runtime	Size	Precision	F-Measure	Recall	Recall	Coherent
10	845	0.996	0.713	0.555	0.014	-

**Table 4.** Results for Anatomy track

Since both the two ontologies have the scale larger than 500 entities, candidate filtering is employed. As a result, IAMA finishes this track in 10 seconds. Only two systems are faster than IAMA. The simple use of lexical similarity generates mostly trivial correspondences, leading the low recall+ measure.

### 2.3 conference

Conference track contains sixteen ontologies from the conference organization domain. There are two versions of reference alignment. The original reference alignment is labeled as *ra1*, and the new reference alignment, generated as a transitive closure computed on the original reference alignment, is labeled as *ra2*. Table 5 shows the results of our system in this track.

	Precision	F-Measure	Recall
ra1	0.78	0.59	0.48
ra2	0.74	0.55	0.44

**Table 5.** Results for Conference track

IAMA finishes the conference track in 53 seconds. Candidate filtering has not been activated.

### 2.4 multifarm

The MultiFarm data set contains ontologies in eight different languages. These ontologies are translated from conference track. IAMA does not design a multilingual method specifically, thus obtained relatively poor results. We managed to utilize language detection and translation API. Unfortunately, it increased the processing time of our system and led to other problems. In the next version, IAMA will adopt specialized method to deal with multilingual ontologies. The results are presented in Table 6.

Average Precision	Average F-Measure	Average Recall
0.30	0.05	0.03

**Table 6.** Results for MultiFarm track

### 2.5 library

The task of library track is to match two real-world thesaurus, namely STW and TheSoz. IAMA does not provide particular method aiming at this track. The results can be seen in Table 7. IAMA does not apply particular method for this track.

Precision	F-Measure	Recall
0.78	0.04	0.08

**Table 7.** Results for Library track

## 2.6 large biomedical ontologies

Large Biomedical track challenges matching tools by offering large scale ontologies. The task of this track is to find alignments between Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). IAMA finishes the task in reasonable time owe to the use of candidate filtering. Table 8 shows the results.

<b>Task 1: Small FMA and NCI fragments</b>				
P	F	R	#Mappings	Runtime (s)
0.979	0.733	0.585	1,751	14
<b>Task 2: Whole FMA and NCI ontologies</b>				
P	F	R	#Mappings	Runtime (s)
0.901	0.708	0.582	1,894	139
<b>Task 3: Small FMA and SNOMED fragments</b>				
P	F	R	#Mappings	Runtime (s)
0.962	0.236	0.134	1,250	27
<b>Task 4: Whole FMA and SNOMED ontologies</b>				
P	F	R	#Mappings	Runtime (s)
0.749	0.227	0.134	1,600	218
<b>Task 5: Small SNOMED and NCI fragments</b>				
P	F	R	#Mappings	Runtime (s)
0.965	0.604	0.439	8,406	99
<b>Task 6: Whole SNOMED and NCI ontologies</b>				
P	F	R	#Mappings	Runtime (s)
0.917	0.593	0.439	8,843	207

**Table 8.** Results for Large Biomedical track

IAMA is one of the fifteen systems that are able to complete all six tasks, and provides the best results in terms of precision in task 1 and task 2. Furthermore, our system finishes all the tasks in 704 seconds, only slower than LogMapLt (371 seconds). The average results are shown in Table 9.

Precision	F-Measure	Recall	Incoherence	Total time (s)
0.912	0.517	0.386	46.4%	704

**Table 9.** The average results for Large Biomedical track

### 3 General comments

#### 3.1 Comments on the results

IAMA achieved qualified results in its first participation in OAEI. The results for benchmark, conference, and large biomedical track is better. Since the system does not design specific method to handle MultiFarm and library track, the results are relatively poor. It is evident that IAMA got relatively high precision but low recall. The reason is that the threshold  $t_1$  is fixed to a high value of 0.9. Candidate filtering, as already mentioned, cuts down the recall as well.

#### 3.2 Discussions on the way to improve the proposed system

IAMA remains much to be improved. First, the system does not take advantage of structural information, which is beneficial when lack of lexical information. We tried to calculate structural similarity like subclass similarity and superclass similarity, but did not receive expected results. The hierarchy information is also remained to be exploited. Second, predetermining all the parameters loses the flexibility. The influence of parameter  $t_1$  can be seen in Table 10. The experiment is conducted on a computer with 4.7GHz Intel i5 CPU (4 core) and 8GB RAM. A self-adjust mechanism is to be employed in the future. Third, the system lacks the ability to match ontologies in different languages. The next version will support multi-language inputs. We expect the optimized system would become an eligible universal ontology matching system.

	Precision	F-Measure	Recall
$t_1=0.7$	0.800	0.752	0.710
$t_1=0.8$	0.945	0.783	0.668
$t_1=0.9$	0.995	0.719	0.563

**Table 10.** Impact of parameter  $t_1$  in anatomy track (without candidate filtering)

### 4 Conclusion

This paper has reported the results of IAMA in OAEI 2013. The results reflect that IAMA has the ability to deal with a majority of ontologies, including large ones. On the other hand, for those disadvantages exposed, we discuss the possible solutions. By and large, IAMA achieved reasonable results for its first participation in OAEI, and it is promising to be much improved in the future.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61070106,61272332,61202329) and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research(ICDD201201).

## References

1. P Shvaiko and Jérôme Euzenat. Ontology matching: State of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, (99), 2012.
2. Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.
3. Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A string metric for ontology alignment. In *The Semantic Web-ISWC 2005*, pages 624–637. Springer, 2005.
4. José Luis Aguirre, Bernardo Cuenca Grau, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Robert Willem van Hague, Laura Hollink, Ernesto Jimenez-Ruiz, Christian Meilicke, Andriy Nikolov, et al. Results of the ontology alignment evaluation initiative 2012. In *Proc. 7th ISWC workshop on ontology matching (OM)*, pages 73–115, 2012.