YAM++ : (not) Yet Another Matcher for Ontology Matching Task

DuyHoa Ngo, Zohra Bellahsene
Université Montpellier 2, INRIA, LIRMM
Montpellier - France
{firstname.name@lirmm.fr}

July 19, 2012

Abstract

In this paper, we present the capability of our ontology matching tool YAM++. We show that YAM++ is able to discover mappings between entities of given two ontologies by using machine learning approach. Besides, we also demonstrate that if the training data are not available, YAM++ can discover mappings by using information retrieval techniques. Finally, we show that YAM++ is able to deal with multi-lingual ontologies matching problem.

1 Introduction

Ontology matching is a key solution to deal with the semantic heterogeneity problem. It discovers the mappings between semantically related entities of ontologies. By agreeing with these mappings, a common vocabulary and a unified understanding can be used to describe and ultimately analyze data [2]. Many diverse solutions of matching have been proposed so far; however, there is no integrated solution that is a clear success [8]. Therefore, ontology matching still attracts a lot of interest and attention of researchers.

There are many challenges in ontology matching task. A matcher tool can be seen as a combination of three sub-matchers such as: Element level matcher, Structural matcher and Semantical matcher. Generally, element level matcher discovers mappings by comparing annotation (i.e., labels, comments) of entities. It may use many different similarity metrics to handle the high terminological heterogeneity of ontologies. A difficult challenge here is how to combine different metrics effectively. Additionally, if labels of entities in ontologies are represented by different languages, the matching process is even more difficult. Structural matcher discovers mappings of entities based on analyzing their structural information. However, according to [2], most of them don't work well when the structures of ontologies are different. Moreover, structural matcher is error-prone, since it strongly depends on initial mappings provided by element level matcher. Semantical matcher is mainly used to refine the candidate mappings. It exploits the semantic constraints between entities in ontologies in order to remove inconsistent mappings.

To handle these challenges, we propose our solution as follows:

- If labels of entities are written by different languages, we use a multi lingual translator to translate labels from other languages into English.
- We use machine learning based approach to combine different similarity metrics at element level matcher. In case we don't have training data, we propose a similarity metrics based on information retrieval techniques.
- We use a graph matching, in particular similarity propagation method, which is known as the most stable method dealing with structural information to discover additional mappings.
- In terms of semantic refinement, we use the Global Optimal Diagnosis method [3].

The rest of the paper is organized as follows. In Section 2, we present an overview of YAM++ system. Section 3 contains the demonstration scenarios. In section 4, we summarize our contributions.

2 YAM++ Overview

The YAM++ system is an extension of our previous system YAM - not Yet Another Matcher for schema matching [1] to deal with ontology matching task. In the new system, we maintain the basis idea of using machine learning technique to combine different similarity metrics. However, YAM++ was also extended to work without using machine learning (i.e., when learning data are not available). In this extension, YAM++ exploits the intrinsic textual features of ontologies in order to provide a similarity metrics based on information retrieval techniques. Moreover, we add similarity propagation method and semantic verification module in order to discover and refine mappings between entities based on their structural and semantic information.

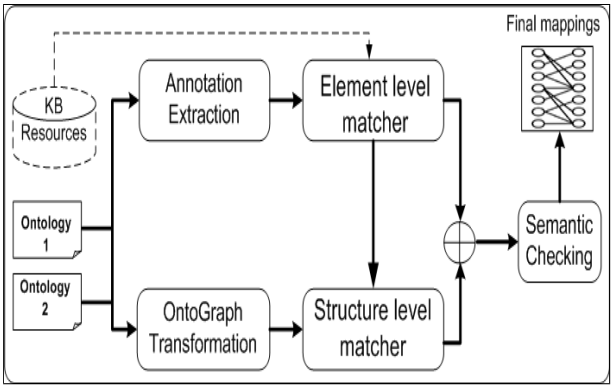

Figure 1: YAM++ architecture

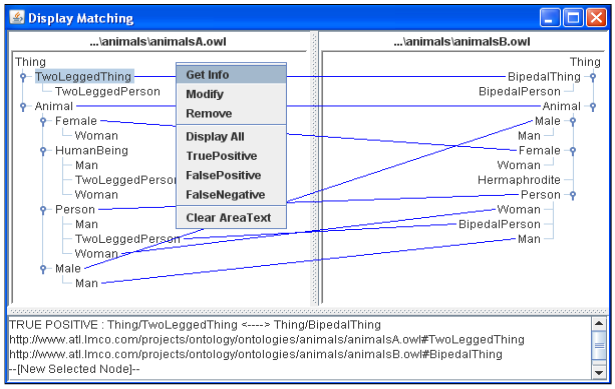

Figure 2: Graphical User Interface

Fig. 1 depicts the main components of YAM++ system. YAM++ discovers mappings between two input ontologies by two matchers: element level matcher and structural level matcher. The combination results of element level and structural level are then revised by the semantic checking in order to remove inconsistent mappings.

- At element level, input ontologies are processed in order to extract annotation information for every entity. Based on these information, similarity score between entities are computed by different terminological metrics. Here, similarity metrics can work independently or can be combined by combination methods in order to produce mappings at element level. Currently, YAM++ support machine learning based combination methods such as Decision Tree, SVM, NaiveBayes[1], etc. The training data are provided by user or are taken from knowledge base (KB) resources.
- At structural level, input ontologies are parsed and transformed into graph data structure. Then, YAM++ takes result obtained from element level as initial mappings to run a similarity propagation process. The propagation algorithm here is inspired from the well known Similarity Flooding algorithm [4]. See [6] for more detail about our similarity propagation method.
- In semantical checking module, we make use of global constraint optimization method proposed in Alcomox tool[2]

The resulting mappings of two ontologies alignment are displayed in a graphical interface. The user can judge a mapping as correct or not by their knowledge of ontologies'domain. Then, he/she can also modify, remove incorrect mappings or add new mappings with the help of command operations shown in system's menu (Fig. 2)

3 Demonstration Scenarios

YAM++ has been implemented in Java, offering a GUI to select different configuration options and display matching results. In this demo, we show the capabilities of YAM++: (i) matching with

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://web.informatik.uni-mannheim.de/alcomo/

machine learning method, (ii) matching with information retrieval method, and (iii) matching with multi-lingual ontologies.

3.1 Experiment Settings

Evaluation Metrics

Test Cases In demonstration, YAM++ is able to discover mappings between any two input ontologies provided by the user. In order to evaluate the matching quality of YAM++, we use two data sets widely used in the ontology matching field to compare the matching quality of other participants of OAEI campaign.

- The **Conference**[3] track contains 16 ontologies from the same domain (conference organization) and each ontology can be matched against with other ontology. Due to the high heterogeneity of these ontologies, finding mappings between them is very difficult.
- The **Multifarm**[4] data sets, which has been designed as a comprehensive benchmark for multilingual ontology matching. This dataset is composed of a subset of the Conference dataset, translated in eight different languages (Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish).

3.2 Matching with machine learning method

In the first scenario, we assume that the user has several gold standard data sets, which consist of two ontologies and a corresponding alignment provided by an expert of domain. The user may think that he/she can study some matching patterns from the existing data sets to discover new mappings from new matching scenario with the to-be-matched ontologies. Obviously, manually finding mappings is not applicable given the big size of ontologies. Therefore, the user would like to use the existing data as training data to train a machine learning model. Then, the learning model will automatically examine every pair of entities from to-be-matched ontologies and classify them into match or not.

Matcher	Prec.	$F_{0.5}$Meas.	Rec.	Prec.	F_1Meas.	Rec.	Prec.	F_2Meas.	Rec.
YAM++	.8	.73	.53	.78	**.65**	.56	.78	.59	.56
CODI	.74	.7	.57	.74	.64	.57	.74	.6	.57
LogMap	.85	**.75**	.5	.84	.63	.5	.84	.54	.5
AgrMaker	.8	.69	.44	.65	.62	.59	.58	**.61**	.62
$BaseLine_2$	**.79**	**.7**	**.47**	**.79**	**.59**	**.47**	**.79**	**.51**	**.47**
MaasMtch	.83	.69	.42	.83	.56	.42	.83	.47	.42
$BaseLine_1$	**.8**	**.68**	**.43**	**.8**	**.56**	**.43**	**.8**	**.47**	**.43**
CSA	.61	.58	.47	.5	.55	.6	.5	.58	.6
CIDER	.67	.61	.44	.64	.53	.45	.38	.48	.51
MapSSS	.55	.53	.47	.55	.51	.47	.55	.48	.47
Lily	.48	.42	.27	.36	.41	.47	.37	.45	.47
AROMA	.35	.37	.46	.35	.4	.46	.35	.43	.46
Optima	.25	.28	.57	.25	.35	.57	.25	.45	.57
MapPSO	.28	.25	.17	.21	.23	.25	.12	.26	.36
LDOA	.1	.12	.56	.1	.17	.56	.1	.29	.56
MapEVO	.27	.08	.02	.15	.04	.02	.02	.02	.02

Figure 3: Comparison result on Conference 2011 track

Based on this idea, YAM++ provides different kinds of similarity metrics, which can be used to represent different features of each pair of entities from two to-be-matched ontologies. The existing data selected by the user are then transformed into training data. Then, YAM++ performs a training process on the user's selected machine learning model to produce a trained classifier. The classifier produces a set of mappings. If the user selects a structural method in the next step, these mappings will be passed as input to the similarity propagation process. Finally, the mapping results will be shown in the display for the user's judgment. For more detail about similarity metrics and machine learning based combination, we refer reader to our paper [5, 7].

[3] http://oaei.ontologymatching.org/2011/conference/index.html
[4] http://web.informatik.uni-mannheim.de/multifarm/

Fig. 3 shows the comparison result of YAM++ with other participants on the Conference track in OAEI 2011 campaign[5]. This was the first time we participate in the OAEI competition. At this time, YAM++ stayed in Top 2 among all participants. Especially, in terms of F_1measure, it achieved the best matching tool title.

3.3 Matching with information retrieval method

In this second scenario, we assume that related gold standard data sets are not available. In that case the method of using machine learning model is not applicable. In order to overcome this weakness, YAM++ supports a discovering method based on information retrieval techniques. In particular, YAM++ studies the annotation information of entities in both to-be-matched ontologies. Based on the frequency appearance of a word, YAM++ determines amount of informativeness of that word within the ontology. Then, YAM++ provides a metric to compare similarity between entities' labels. Moreover, YAM++ studies the context to which entities belong. The context information is also used to compute similarity between entities. Both labels method and context method are inspired from information retrieval techniques. Similar to the first scenario, the user can select similarity propagation for next step to discover more mappings by exploiting structural information of entities.

Fig. 4 shows the comparison result of YAM++ with the other participants on the Conference track in OAEI 2011.5 campaign[6]. This was the second time we participate to the OAEI competition with non learning YAM++ system. At this time, YAM++ obtained the best matching result and dominated all other participants.

Matcher	Threshold	Precision	F0.5-measure	F1-measure	F2-measure	Recall
YAM++	0	0.8	0.78	0.74	0.71	0.69
LogMap	0	0.82	0.75	0.66	0.59	0.55
CODI	0	0.74	0.7	0.64	0.6	0.57
Hertuda	0	0.79	0.7	0.6	0.53	0.49
WeSeE	0.33	0.71	0.65	0.59	0.53	0.5
Baseline2	0	0.79	0.7	0.59	0.51	0.47
LogMapLt	0	0.73	0.67	0.59	0.53	0.5
GOMMA	0	0.84	0.71	0.58	0.49	0.44
AUTOMSv2	0	0.79	0.68	0.56	0.47	0.43
Baseline1	0	0.8	0.68	0.56	0.47	0.43
MaasMtch	0.89	0.74	0.64	0.54	0.46	0.42
MapSSS	0	0.5	0.5	0.5	0.51	0.51
MapPSO	0.67	0.1	0.08	0.07	0.06	0.05
MapEvo	0.82	0.04	0.03	0.03	0.02	0.02

Figure 4: Comparison result on Conference 2011.5 track

3.4 Matching with multi-lingual ontologies

In the last scenario, we show the ability of YAM++ to work with multi-lingual ontologies matching. When the user provides two to-be-matched ontologies, YAM++ read annotations of entities in order to determine which language is used in each ontology. Once the languages are defined, YAM++ use Microsoft Bing Translator tool[7] to translate all labels from other languages to English. After that, YAM++ discovers mappings between entities based on their translated labels by our proposed information retrieval methods. The returned mappings are passed as input to similarity propagation process to discover more mappings.

Fig. 5 shows the comparison result between YAM++ and other participants of OAEI 2011.5 campaign on Multifarm data sets. There are two types of evaluation. In the first type, all matching tools deal with different ontologies with different languages. In this evaluation, YAM++ achieved the best matching quality (Fmeasure = 0.45). In the second type, all tools discover mappings of the same ontologies but translated in different languages. In this evaluation, YAM++ obtained the second position among all participants.

[5]http://oaei.ontologymatching.org/2011/
[6]http://oaei.ontologymatching.org/2011.5/
[7]http://www.microsofttranslator.com/

	Different ontologies (type i)				Same ontologies (type ii)			
Matching system	Size	Precision	Recall	F-measure	Size	Precision	Recall	F-measure
YAM++	1838	0.54	0.39	0.45	5838	0.93	0.48	0.63
AUTOMSv2	746	0.63	0.25	0.36	1379	0.92	0.16	0.27
WeSeE	4211	0.24	0.39	0.29	5407	0.76	0.36	0.49
CIDER	737	0.42	0.12	0.19	1090	0.66	0.06	0.12
MapSSS	1273	0.16	0.08	0.10	6008	0.97	0.51	0.67
LogMap	335	0.36	0.05	0.09	400	0.61	0.02	0.04
CODI	345	0.34	0.04	0.08	7041	0.83	0.51	0.63
MaasMtch	15939	0.04	0.28	0.08	11529	0.23	0.23	0.23
LogMapLt	417	0.26	0.04	0.07	387	0.56	0.02	0.04
MapPSO	7991	0.02	0.06	0.03	6325	0.07	0.04	0.05
CSA	8482	0.02	0.07	0.03	8348	0.49	0.36	0.42
MapEVO	4731	0.01	0.01	0.01	3560	0.05	0.01	0.02

Figure 5: Comparison on Multifarm 2011.5 track

4 Conclusion

In this paper, we present YAM++ - an ontology matching tool, which supports: (i) discovering alignment of ontologies by machine learning approaches; (ii) discovering alignment of ontologies by generic methods without using learning techniques; (iii) discovering alignment of ontologies represented in different languages. Moreover, our YAM++ tool produces the best matching quality on Benchmark, Conference and Multifarm tracks in comparison with other participants on OAEI 2011 and OAEI 2011.5 campaigns. The running tool can be found at http://www2.lirmm.fr/∼dngo/.

References

[1] Fabien Duchateau, Remi Coletta, Zohra Bellahsene, and Renée J. Miller. Yam: a schema matcher factory. In *CIKM Conference*, pages 2079–2080, 2009.

[2] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[3] Christian Meilicke. Alignment incoherence in ontology matching. In *Thesis*.

[4] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE*, pages 117–128, 2002.

[5] DuyHoa Ngo, Zohra Bellahsene, and Remi Coletta. A flexible system for ontology matching. In *Caise 2011 LBIP*, pages 79–94, 2011.

[6] DuyHoa Ngo, Zohra Bellahsene, and Remi Coletta. Yam++ results for oaei 2011. In *OM*, 2011.

[7] DuyHoa Ngo, Zohra Bellasene, and Remi Coletta. A generic approach for combining linguistic and context profile metrics in ontology matching. In *ODBASE Conference*, 2011.

[8] Pavel Shvaiko and Jérôme Euzenat. Ten challenges for ontology matching. In *OTM Conferences (2)*, pages 1164–1182, 2008.