Alignment Results of SOBOM for OAEI 2010

Peigang Xu, Yadong Wang, Liang Cheng, Tianyi Zang

School of Computer Science and Technology

Harbin Institute of Technology, Harbin, China

peigang.xu@gmail.com, ydwang@hit.edu.cn, chl198478@126.com,tianyi.zang@gmail.com

Abstract. In this paper we give a brief explanation of how Sub-Ontology based Ontology Matching (SOBOM) method gets the alignment results at OAEI2010. SOBOM deal with an ontology from two different views: an ontology with is-a hierarchical structure *O'* and an ontology with other relationships *O''*. Firstly, from the *O'* view, SOBOM starts with a set of anchors provided by a linguistic matcher. And then it extracts sub-ontologies based on the anchors and ranks these sub-ontologies according to their depths. Secondly, SOBOM utilizes Semantic Inductive Similarity Flooding algorithm to compute the similarity of concepts between different sub-ontologies derived from the two ontologies according the depth of sub-ontologies to get concept alignments. Finally, from the *O''* view, SOBOM gets relationship alignments by using the concept alignment results in *O''*. The experiment results show SOBOM can find more alignment results than other compared relevant methods.

1 System presentation

Currently more and more ontologies are distributedly built and used by different organizations. And these ontologies are usually light-weighted [1] containing lots of concepts especially in biomedicine, such as anatomy taxonomy NCI Thesaurus. The Sub-ontology based Ontology Matching (SOBOM) is designed for matching light-weight ontologies that has is-a hierarchy as their backbones. It matches an ontology from two views: *O'* and *O''* that are depicted in Fig. 1. The unique feature of our method is combining sub-ontology extraction with ontology matching.

1.1 State, purpose, general statement

SOBOM is developed to match ontology automatically for general purpose. Based on two different views, we design three elementary matchers in current version. The first one is a anchor generator which is used to find anchors; the second one is a structure matcher SISF (Semantic Inductive Similarity Flooding) which is inspired by Anchor-Prompt [3] and SF [4] algorithms and is exploited to flood similarity among concepts. The last one is a relationship matcher which utilizes the results of SISF to get relationship alignments. In addition, a Sub-ontology Extractor (SoE) is integrated into SOBOM to extract sub-ontologies according to the anchors got by linguistic matcher and rank them by their depths descendingly. Overall SOBOM is a sequential method, so it does not care how to combine the results of different matchers. The overview of the method is illustrated in Fig. 2.

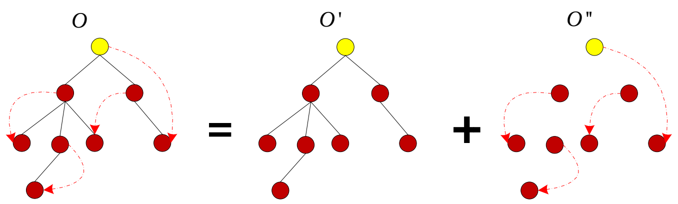

Fig. 1. Two views of an ontology in SOBOM

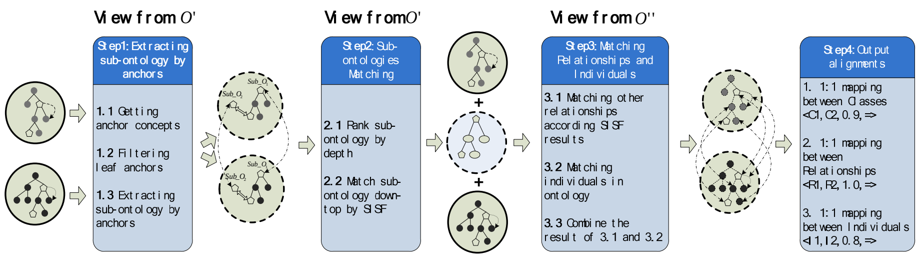

Fig. 2. The processing overview of SOBOM

For simplicity, we define some notations used in the report.

Ontology: An ontology *O* consists of a set of concepts *C*, properties (relations) *R*, instances *I*, and axioms *A*. We use entity *e* to denote either $c \in C$ or $r \in R$. Each relation *r* has a domain and range defined as following:

$$Domain\ (r) = \{c_i \mid c_i \in C\ \ and\ \ having\ \ the\ \ relationsh\ \ ip\ \ r\}$$

$$Range\ (r) = \{c_i \mid c_i \in C\ \ and\ \ can\ \ be\ \ value\ \ of\ \ r\}$$

Anchor: An anchor is defined as a pair of assumed equivalent non-leaf concepts across ontologies. Given two ontologies O_1, O_2, $c_1 \in O_1$, $c_2 \in O_2$, if $c_1 \equiv c_2$ (means that c1 is identical with c2),and c1, c2 are both not leaf nodes in the hierarchies of O1 and O2 respectively, then an anchor, X is defined as a pair of concepts $< c_1, c_2 >$.

Sub-Ontology: Let ontology $O = (C,R, H^C,H^R,)$. A sub-ontology O^x is a subset of O whose elements all come from *O*, called $O^x = (C_1,H_1^C)$, where $C_1 \subseteq C, H_1^C \subseteq H^C$, x is the root of H^C. Indeed, a sub-ontology in our method is a hierarchical taxonomy, and its root is an anchor concept.

Sub-ontology Depth. *The depth of sub-ontology* O^x *is the maximal length of path from the anchor x to the root* r_i *of the taxonomy* H_i^C *which contains it in the original ontology O.*

$$Depth(O^x) = Max(|x \rightarrow ..., \rightarrow r_i|),\ r_i \in H_i^C,\ H_i^C \in O$$

1.2 Specific techniques used

SOBOM aims to provide high quality of 1:1 alignments between concept and property pairs. We have implemented SOBOM algorithm in java and integrated three distinguishing constitutional matchers. They are independent components in core matcher library of SOBOM. Due to the space limitation, we only describe the key features of them. The details can be found in the related paper [8].

- Our anchor generator is based on the local context of an entity in ontology. In details, the local context of an entity including the following aspects: the textual information (label, id, comments and so on), the structure information (the number of super or sub concepts, the number of constraints) and the individual information (the number of individual if existing). We consider that the local context of an entity can express the meaning of it. Consequently, we get three similarity matrixes respectively, and we choose the maximal of them as the final results.

- SISF uses the RDF statement to represent the ontology and utilizes the anchors to inducting the construction of similarity propagation graph for the sub-ontologies. SISF handles the ontology from the view *O'* and only generate concept-concept alignment.
- R-matcher is a relationship matcher base on the definition of the ontology. It combines the linguistic and semantic information of a relation. From the *O''* view, it utilizes the is-a hierarchy to extend the domain and range of a relationship and uses the result of SISF to generate the alignment between relationships.

More importantly, SoE is integrated into SOBOM and extracts sub-ontologies according to the anchors [5,6]. SoE ranks extracted sub-ontologies according to their depths. As we extract sub-ontologies for ontology matching, the rules of extracting sub-ontology in SoE are as following: only sub-concepts of anchor are included in the sub-ontology. In other words, a sub-ontology is a taxonomy which has anchor as root.

If one of the two concepts in an anchor is a leaf node in the original ontology, we do not use SISF to deal with it actually. Because this phenomenon just represents a one-to-many mapping. After extracting sub-ontologies, SOBOM will match these sub-ontologies according to their depth in original ontology. We first match the sub-ontologies with larger depth value. By using SoE, SOBOM can reduce the scale of ontology and make it easy to operate sub-ontologies in SISF.

1.3 Adaptations made for the evaluation

We don't make any specific adaptation for the tests in the OAEI 2010 campaign. All the alignments outputted by SOBOM are based on the same set of parameters.

1.4 Link to the system and parameters file

The current version of SOBOM is available at: http://mlg.hit.edu.cn:8080/Ontology/Download.jsp, and the parameters setting is illustrated in the reading me file.

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

We deploy our matcher as a web service, our web service name is: eu.sealsproject.omt.ws.matcher.AlignmentWSImpl. The endpoint of our web service can be found at: http://mlg.hit.edu.cn:8080/SOBOMService/SOBOMMatcher?wsdl.

2 Results

In this section, we describe the results of SOBOM algorithm against the benchmark, directory and anatomy ontologies provided by the OAEI 2010 campaign. We use Jena-API to parse the RDF and OWL files. The experiments were carried out on a PC running Windows vista ultimate with Core 2 Duo processors and 4-gigabyte memory.

2.1 Benchmark

On the basis of the nature, we can divide the benchmark dataset into five groups: #101-104, #201-210, #221-247, #248-266 and #301-304. SOBOM is a sequential matcher. If the linguistic matcher gets no results, SOBOM will produce no result. We described the performance of our SOBOM algorithm over each group and overall performance on the benchmark test set in Table 1.

#101-104 SOBOM plays well for these test cases.

#201-210 In this group, some linguistic features of candidate ontologies are discarded or modified, their structures are quite similar. SOBOM is a sequential matcher, our anchor generator matches concepts based on their local context not only the linguistic information. So, although without linguistic information SOBOM also gets relatively high precision and recall.

#221-247 The structures of the candidate ontologies are altered in these tests. However, SOBOM discovers most of the alignments from the linguistic perspective via our anchor generator, 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. So, SOBOM does not very well.

#301-304 This test group are four real-life ontologies of bibliographic references. SOBOM can only find equivalence alignment relations.

Table 1. The performance on the benchmark

	101-104	201-210	221-247	248-266	301-304
Precision	1.0	0.99	0.99	0.87	0.77
Recall	1.0	0.85	0.99	0.57	0.70

Compared to our previous results (OAEI2009), the recall of every group is highly improved. This is enhanced by our redesigned anchor generator.

2.2 Anatomy

The anatomy real world test bed covers the domain of body anatomy and consist of two ontologies, Adult Mouse Anatomy (2247 classes) and NCI Thesaurus (3304 classes). These are relatively large compared to benchmark ontologies. This type ontologies is what SOBOM suitable for handling, it generated 268 sub-ontologies and 1249 alignments between MA and NCI, consumed 19min3s to complete the matching task. It is obvious that most of the alignment appears in the leaf nodes in ontologies (834 leaf node alignments). The experiment result shows in Table 2.

Table 2. The performance of SOBOM on the anatomy test

	Leaf node alignments	Sub-ontologies	Total Alignments	Time consuming

NCI MA	834	268	1249	19min3s

2.3 Conference

There are 120 pairs of ontologies in this track. Most of them are blind tests (i.e. there no reference alignment available). The whole results are available at: http://seals.inrialpes.fr/platform/;jsessionid=FD1E3A5CE8DA43C1D52DB21079EAECF3?wicket:bookmarkablePage=:eu.sealsproject.omt.ui.Results&endpoint=http://219.217.238.162:8080/SOBOMService/SOBOMMatcher?wsdl&evaluationID=http://219.217.238.162:8080/SOBOMService/SOBOMMatcher?wsdl2010/10/03+02:09:03&track=Conference+Testsuite.

3 General comments

In this section, we want to introduce comments on the results of SOBOM algorithm and the way to improve it.

3.1 Comments on the results

Strengths SOBOM deals with ontology from two different views and combines results of every step in a sequential way. If the ontologies have regular literals and hierarchical structures, SOBOM can achieve satisfactory alignments. And it can avoid missing alignment in many partitioning matching methods as illustrated in [7].

Weaknesses SOBOM needs anchors to extract sub-ontologies. So it depends on the precision of anchors. In current version, we use a linguistic matcher to get anchor concept, if the literals of concept missed, SOBOM will get bad results.

3.2 Discussions on the way to improve the proposed system

SOBOM can be viewed as a frame of ontology matching. So many independent matchers can be integrated into it. Now, we have enhanced the anchor generator by not considering the textual information but also the structure information. Our next plan is to integrate a more powerful matcher to produce anchor concepts or develop a new method to get anchor concepts. Meanwhile, we plan to develop a mapping debugging method to refine the results of SOBOM.

4 Conclusion

Ontology matching is very important part of establishing interoperability among semantic applications. This paper reports our participation in OAEI2010 campaign. We present the alignment process of SOBOM and describe the specific techniques for ontology matching. We also show the performance in different alignment tasks. The strengths and the weaknesses of our proposed approach are summarized and the possible improvement will be made for the system in the future. We propose a brand new algorithm to match ontologies. The unique feature of our method is combining sub-ontology extraction with ontology matching based on two different views of an ontology.

References

1. Fausto Giunchiglia and Ilya Aihrayeu : Lightweight Ontologies. Technical Report. (2007) DIT-07-071.

2. G.Stoilos, G. Stamou, S. Kollias: A string metric for ontology alignment, In Proc. Of the 4th International Semantic Web Conference(ISWC'05). (2005) 623-637.

3. N.F. Noy, M.A. Musen: Anchor-PROMPT: using non-local context for semantic matching, In Proc. Of IJCAI2001 Workshop on Ontology and Information Sharing, (2001) 63-70.

4. S. Melnik, H.G. Molina and E. Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm, In Proc 18th Int'l Conf. Data Eng. (ECDE'02) (2002) 117-128.

5. Julian Seidenberg and Alan Rector: Web Ontology Segmentation: Analysis, Classification and Use, WWW2006, (2006).

6. H. Stuckenschmidt and M. Klein: Structure-Based Partitioning of Large Class Hierarchies. In Proc of the 3rd International Semantic Web Conference (2004).

7. W. Hu, Y. Qu: Block matching for ontologies, In Proc of the 5th International Semantic Web Conference, LNCS, vol. 4273, Springer (2006) 300-313.

8. P.G. Xu, H.J. Tao: SOBOM: Sub-ontology based Ontology Matching. To be appear.