## FORPM: Boosting Users' Effect on Ontology Matching

Dunwei Wen<sup>1,2</sup>, Xiaohu Fan<sup>1</sup>, Fuhua Lin<sup>2</sup>

<sup>1</sup> School of Information Science and Engineering, Central South University, Changsha, Hunan 410083, China dwwen@mail.csu.edu.cn, fanxiaohu17@gmail.com <sup>2</sup> School of Computing and Information Systems, Athabasca University, Athabasca, Alberta T9S 3A3, Canada {dunweiw, oscarl}@athabascau.ca

**Abstract.** In this paper, we attempt to view the ontology matching task from an information gaining angle. In our opinions, the information used for matching mainly comes from the matching tools as well as the human experts. With this understanding, we believe that by making good use of user efforts, we can also accelerate the matching process. Hence we present a prototype system named FORPM. First, it ranks the entities of the ontology. Important entities are chosen as centroids to form fragments. Then, users can use those centroids' information to estimate the content of the fragments and initially match them. Finally, automatic matching is carried out among those matched fragments. Experiment results obtained so far show that with a few user efforts, our approach significantly improves the matching efficiency while the loss of accuracy is acceptable.

## 1 Motivation

Ontology matching aims at finding semantic relationships between entities of different ontologies for solving the interoperation problem. From the viewpoint of information theory, we view a matching problem  $\Delta$  as a process of information gaining with uncertainty  $\Omega(\Delta)$ . Let  $\varphi$  denotes the information obtained by the matching tool (from ontology itself as well as external source like WordNet),  $\omega$  denotes the information provided by users in the validation step (we hold the same kind of opinions with [1] that fully automatic ontology matching is still impossible). To obtain matching results of high quality, we believe that the following equation has to be satisfied:

$$\omega + \varphi \ge \Omega(\varDelta). \tag{1}$$

On one hand, to our best knowledge, recently numerous researches have been focused on how to maximize  $\varphi$  and made great progresses. On the other hand, we believe that the human users, especially the domain experts are capable of discovering complex relationships, such as more general ( $\supseteq$ ), less general ( $\subseteq$ ) etc., between candidate pairs. This extra information, however, is always ignored and human effort is simply used in the validation step to judge simple relations such as matching or not matching. With these understandings, our work is aiming at making good use of the information provided by users, thus accelerating the matching process.

In this paper we propose: 1) an information theory based model for concept ranking and centroid extraction; 2) a clustering algorithm for ontology partitioning. To test our approch, we also introduce a prototype system called FORPM.

# 2 FORPM (Framework for Ontology Ranking, Partitioning and Matching)

FORPM is implemented with Java under JDK 5.0 and Eclipse 3.1.2. The system architecture is shown below in Fig 1. First, two ontologies are input and then transformed into DAG (Directed Acyclic Graph), where the "is-a" relations are transformed into arcs and concepts are transformed into nodes. After the four main process steps in the dash line, the result and a reference-mapping file are sent to the evaluation module, in which the evaluation results are generated automatically and presented to the user.

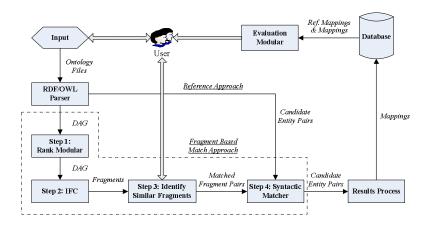


Fig. 1. System Overview

**Step 1: Entity Ranking** Based on our observation, the amount of information provided by a user equals the sum of the amount of information  $I_i$  provided in T times of the user's validation.

$$\omega = \sum_{\{i|i\in[1,T],i\in\mathbb{N}\}} I_i.$$
 (2)

If we assume that the cost of the user's every validation be the same, then one intuitive way to improve the matching efficiency is to maximize  $I_i$  wisely. Hence fundamental to our ranking approach is the ability to measure how much information is conveyed in a node thereby giving a sense of how much information the computer would "gain" by being informed about a discovered matching pair.

In information theory [2], the amount of information contained in an event is measured by the negative logarithm of the probability of occurrence of the event. Thus if  $\chi$  is an event that has possible outcome values  $x_1, x_2, ..., x_n$  occurring with probabilities  $pr_1, pr_2, ..., pr_n$ , the amount of information gained or uncertainty removed by knowing that has the outcome  $x_i$  is given by:

$$I(\chi = x_i) = -\log(pr_i).$$
(3)

Based on this we can build a model to measure the amount of information of a node in an ontology graph by considering the concept as an event and "is-a" relations as its outcomes. Assume that in the ontology graph G(Arcs, Nodes), where Arcs is the set of all "is-a" relations and *Nodes* is the set of all concepts in ontology *O*. Then for any arc  $a_i \in Arcs$ , its probability is given by

$$\Pr\left(\chi = a_i\right) = \left(\frac{1}{|arcs_i|}\right).\tag{4}$$

Here  $|arcs_i|$  is the number of arcs connecting with Node  $n_i$ . Thus the amount of information contained in arc  $a_i$  is:

$$I(\chi = a_i) = -\log(\chi = a_i).$$
 (5)

If we have a node  $n_i \in Nodes$ , then the amount of information contained in  $n_i$  is:

$$I(n_i) = \sum_{a_i \in arcs_i} I(a_i).$$
(6)

We use the amount of information to rank nodes. The node contains the most amount of information is defined as an information center.

**Definition 1** (Information Center/Centroid Node). Let G(N, A) be an ontology graph, N be the node set, A be the arcs set, then node  $I_c \in N$  is an information center if for any node  $n_i \in N$ :

$$I(I_c) \ge I(n_i). \tag{7}$$

#### Step 2: IFC (Information Flooding theory based Clustering)

The goal of this step is to form fragments from centroids. We noticed that in the "isa" hierarchy tree, semantic similarity between two concepts often decays as the distance between them increases. In our work, we define an information flooding function to measure how strong a source node could affect a target node.

**Definition 2** (Information Flood). Let G(N, A) be an ontology graph,  $n_i, n_j \in N$ , we define the information flood from  $n_i$  to  $n_j$  as:

$$InfoFlood(n_i, n_j) = F(Dist_{ij})I(n_i).$$
(8)

Where  $I(n_i)$  is the information contained in  $n_i$ ,  $F(Dist_{ij})$  is a quadratic experiential decay function to simulate the attenuation of similarity defined as

$$F(Dist_{ij}) = \frac{1}{a \times Dist_{ij}^2 + b \times Dist_{ij} + c}.$$
(9)

and  $Dist_{ij}$  is the number of arcs between node  $n_j$  and node  $n_j$  in the "is-a" relation hierarchy tree. In our experiment, we have a = 0.25; b = 0.5; c = 0.

**Definition 3** (**Fragment**). Let *O* be an Ontology,  $G_i(N, A)$  be a graph representing part of *O*, where *N* is the node set, *A* is the arc set. Let dList be the set of all centroid nodes in *O*. If for any  $d_i \in dL$  and all  $n_k \in N$ , we have  $d_i \in dL$  ist which satisfies

$$InfoFlood(d_i, n_k) \ge Max(InfoFlood(d_j, n_k)).$$
(10)

Then we say  $G_i$  is a fragment  $f(d_i, m_i)$  with  $m_i$  as its size and  $d_i$  as its centroid node.

Procedure (G, Max, Min) for each node in Graph G Ranking nodes, the first Max nodes with highest rank are centroids; end while the centroid list does not change or iteration times < SetValue Compute the InfoFlood $(n_i,n_j)$  from centroids to other normal nodes; Assign normal nodes to centroids which maximize the InfoFlood $(n_i,n_j)$ ; Form the fragments set F; Recalculate the centroids list; end end

We briefly describe the partitioning algorithm in Table 1. The algorithm receives two parameters, Max, the upper bound of the number of the fragment, and Min, the lower bound of the size of the fragment. We set a max iteration number to ensure the stop of the algorithm. Also a merge algorithm is implemented to deal with the fragments whose size is below the lower bound.

#### Step 3: Manual Matching.

In this step, users use those centroid nodes to estimate the content of the fragments and then match them manually. A centroid node may have more than one counterparts with relations such as equivalence (=), more general( $\supseteq$ ), less general( $\subseteq$ ), mismatch ( $\perp$ ) and overlapping ( $\cap$ ). Two centroids are considered semantically matched [3] if the relation between them is not mismatch.

#### Step 4: Automatic Matching.

Two fragments are viewed as matched if their centroids are semantically matched, the remaining matching work between two matched fragments is the same as a normal task but of smaller sizes. Various approaches could be adopted here to finish the task. In FORPM, we employ the same string-based tech in [4] for demonstration.

### **3** Evaluation and Discussion

In our experiment, we adopted a dataset from [4]. The Russia1a contains 151 concepts while Russia1b contains 162 concepts, with 64 human confirmed mappings (concepts only). We used F-Measure [cf.4] and Cost (see blow) as quility metrics.

$$Cost = \frac{\#Compare\_Times}{\#Found\_Marched\_Pairs}.$$
(11)

In FORPM, users can tune the system by changing the value of Max and Min in step2. We can see in Fig. 2 that the more fragments there are, the more likely users are to make right judgements which lower the cost. Meanwhile more human work is required (The user has to do Max\*Max times validations at most). According to our experience, it seems that the program performs best when the parameter Max is set between  $5\% \sim 10\%$  of the total number of the concepts.

 Table 1. IFC (Information Flooding theory based Clustering)

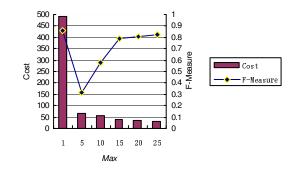


Fig. 2. FORPM with different Max (Min = 4)

## 4 Concluding Remarks

In this paper, we have proposed an information gaining theory based framework for ontology matching. We have shown that with a few user efforts, our approach is effective in reducing the matching complexity.

Our work is inspired by data mining technology. We gain our idea of information model from [5]. Our tool refers to [6]'s work in implementation, while [7] propose an automatic block based matching approach. Both [7] and our tool employ a ranking step to label the blocks. However, [7]'s rank step is after the partitioning step, while in FORPM, ranking step is firstly carried out since we employ an extraction-like clustering algorithm.

## 5 Acknowledgements

We thank National Science and Engineering Research Council of Canada, Academic Research Council of Athabasca University for their financial support for the research, and thank anonymous reviewers for their detailed and constructive comments.

## References

- 1. N.F.Noy and M.A.Musen: The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*.2003,59(6):983-1024.
- 2. C.E.SHANNON: A Mathematical Theory of Communication. *The Bell System Technical Journal*.Vol. 27, pp. 379-423,623-656,July, October,1948.
- Paolo Avesani, Fausto Giunchiglia, and Mikalai Yatskevich: A Large Scale Taxonomy Mapping Evaluation. *ISWC 2005*.LNCS 3729, pp. 67-81, 2005.
- Marc Ehrig and Steffen Staab: QOM- Quick Ontology Mapping. The Semantic Web Proceedings ISWC'04.pp. 289-303, 2004. Springer-Verlag LNCS 3298.
- Kemafor Anyanwu Angela Maduko and Amit Sheth : SemRank: Ranking Complex Relationship Search Results on the Semantic Web In: WWW 2005 May 10-14, 2005, Chiba, Japan.
- Heiner Stuckenschmidt and Michel Klein: Structure-Based Partitioning of Large Concept Hierarchies. In: *The SemanticWeb Proceedings ISWC'04* pp. 289-303, 2004.
- W. Hu, Y. Zhao, Y. Qu: Partition-Based Block Matching of Large Class Hierarchies In Proceedings of ASWC 2006, Beijing, China, 2006.