Fourth International Conference on Semantics, Knowledge and Grid

Ontology Mapping: As a Binary Classification Problem

Ming Mao
SAP Research
ming.mao@sap.com

Yefei Peng
Yahoo!
ypeng@yahoo-inc.com

Michael Spring
U. of Pittsburgh
spring@pitt.edu

Abstract

Ontology mapping seeks to find semantic correspondences between similar elements of different ontologies. Ontology mapping is critical to achieve semantic interoperability in the WWW. To solve the ontology mapping problem, this paper proposes a non-instance learning-based approach that transforms the ontology mapping problem to a binary classification problem and utilizes machine learning techniques as a solution. Same as other machine learning based approaches, a number of features (i.e., linguistic, structural and web features) are generated for each mapping candidate. However, in contrast to other learning-based mapping approaches, the features proposed in our approach are generic and do not rely on the existence and sufficiency of instances. Therefore our approach can be generalized to different domains without extra training efforts. To evaluate our approach, two experiments (i.e., within-task vs. cross-task) are implemented and the SVM algorithm is applied. Experimental results show that our non-instance learning-based ontology mapping approach performs well on most of OAEI benchmark tests when training and testing on the same mapping task; and the results of approach vary according to the likelihood of training data and testing data when training and testing on different mapping tasks.

1. Introduction

The World Wide Web (WWW) is widely used as a universal medium for information exchange. However, semantic interoperability in the WWW is still limited due to the heterogeneity of information. Ontology, a formal, explicit specification of a shared conceptualization [5], has been suggested as a way to solve the problem. With the popularity of ontologies, *ontology mapping* that aims to find semantic correspondences between similar elements of different ontologies has attracted many research attentions from various domains. Different techniques have been examined in ontology mapping, e.g., using linguistic techniques to measure the lexical similarity of concepts in ontologies [13], treating ontologies as structural graphs [10], taking the advantage of information retrieval techniques [9], applying heuristic rules to look for specific mapping patterns [6], and learning to map ontologies through machine learning techniques [2][3]. Comprehensive surveys of ontology mapping approaches can be found in [4][8].

Previous learning-based approaches have achieved high accuracy in prediction of correct mappings in the cases reported in [2][3]. However the approaches either have a limitation that it heavily relies on the availability of instance data when measuring the similarity of classes/attributes, or require new training data to rebuild their model when domain changes and thus restrict the universality of the model.

To overcome the limitations, we treat the ontology mapping problem as a binary classification problem. We learn a generic mapping model, which does not require the existence of instances and domain constraints. To learn a model, a variety of features that can reflect the characteristics of mapping pairs are generated, and then the SVM algorithm is applied. Experimental results show that our non-instance learning-based ontology mapping approach performs well in most of OAEI benchmark tests when training and testing on the same mapping task; and the results of approach vary according to the likelihood of training data and testing data when training and testing on different mapping tasks.

2. Problem Statement

Ontology is a formal, explicit specification of a shared conceptualization in terms of classes, attributes and relations [5]. Ontologies are typically represented as taxonomic trees that include *classes*, *properties*, and *relations*, and associated with *instances*. Two sample bibliographic ontologies are

978-0-7695-3401-5/08 $25.00 © 2008 IEEE
DOI 10.1109/SKG.2008.101

shown in Fig. 1, in which the ellipses indicate classes (e.g., "Reference"), the dashed rectangles indicate properties (e.g., "publisher"), the lines with arrowhead indicate "subClassof" relation between two classes, and the solid rectangles indicate instances of class (e.g., "Object-oriented data modeling"). Each class and property can also have descriptive information (e.g., ID, label, comment) and restrictions (e.g., title, publisher) as indicated in the brace next to "Book".

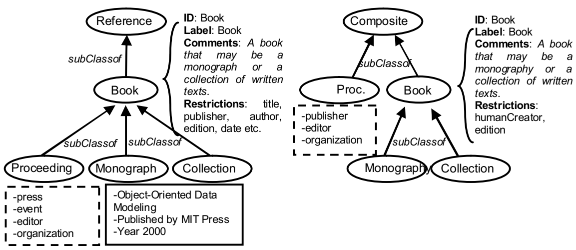

Fig. 1. ***Two sample bibliographic ontologies***

The process of ontology mapping is to find semantic correspondences between similar elements in two homogeneous ontologies, and many ways can be used to judge the quality of a mapping result. In this paper, we refer to the "*correspondence*" as an "=" relationship, the "*elements*" as "*classes*" and "*properties*" of an ontology, and we judge the mapping result by its correctness, i.e., either *correct* or *incorrect*, which can be depicted as a binary set {+1,-1}. Therefore the ontology mapping problem can be easily transformed as a binary classification problem represented as following statement:

$$m(e_{1i}, e_{2j}, r) \rightarrow \{+1,-1\}$$

where e_{1i} is element e_i from ontology O_1, e_{2j} is element e_j from ontology O_2, and r is the mapping relation (i.e. correspondence) between e_{1i} and e_{2j}. According to the statement, candidate mappings in Fig. 1 can be evaluated as followings: $m(Book_{right}, Book_{left}, =) \rightarrow \{+1\}$, $m(Proceedings, Proc, =) \rightarrow \{+1\}$, $m(Monograph, Monography, =) \rightarrow \{+1\}$, $m(Proceedings, Talks, =) \rightarrow \{-1\}$, $m(Proceedings, Monography, =) \rightarrow \{-1\}$, etc.

3. Our Approach

3.1 Overview

The insight of our approach is to treat ontology mapping problem as a binary classification problem, and thus we can take advantage of machine learning techniques. Generally speaking, our approach has 5 steps, see detailed description in 4.4.1 and 4.4.2.

1. Generate various domain independent features (i.e., linguistic, structural and web features) to describe the characteristics of ontologies.
2. Randomly generate training and testing set for OAEI benchmark tests.
3. Train a SVM model on training set.
4. Classify testing data on the trained SVM model.
5. Extract mapping results of testing data using naïve descendant extraction algorithm [11].
6. Evaluate testing data against ground truth.
7. Finally, repeat step 2-6 10 times and get the average evaluation result to eliminate bias.

3.2 Feature Generation

Applying machine learning techniques to ontology mapping context raises the question of what types of information should be used in the learning process. Many different types of information can contribute toward deciding the correspondence of a mapping pair. Two principles are followed to select features:

- The feature should not be limited to instances. It could be generated from classes, properties and/or instances in ontologies.
- The feature should be general enough and domain independent so that the model could be generalized to other applications regardless of the variety of domain.

In the approach, 3 categories, i.e., linguistic features, structural features and web features, and total 26 features are generated for each mapping pair.

3.2.1 Linguistic Features

Linguistic features are selected according to the principle described in [7]. Totally 16 linguistic features are generated, which can be divided into two types (We do not list all linguistic features due to the space limit):

1 Isolated characteristics of elements in mapping pair, e.g. length of elements, number of tokens, etc.

2 Syntactic characteristics of mapping pair, e.g. (normalized) length difference between elements, Levensthtein edit distance between two elements, the proportion of word change between elements, number of common tokens in the pair, the cosine similarity of the profile [9][10] of elements, etc.

3.2.2 Web Features

Bollegala, Matsuo *et al*. [1] proposed a page count based co-occurrence measure i.e., *WebDice*, to compute semantic similarity, which is defined as following, where the notation *H(X)* and *H(Y)* denote the page counts for query *X* and *Y* respectively in a search engine, $H(X \cap Y)$ denotes the page counts for the conjunction query *X AND Y*, *c* is a predefined threshold (e.g. *c=5*) to reduce the adverse effects caused by random co-occurrences.

$$WebDice(X,Y) = \begin{cases} 0 & \text{if } H(X \cap Y) \le c \\ \dfrac{2H(X \cap Y)}{H(X) + H(Y)} & otherwise \end{cases}$$

3.2.3 Structural Features

Structural information is important in estimating the similarity of ontologies. Table 1 lists the structural features of a mapping candidate.

4. Evaluations

4.1 Test Ontologies

Our test ontologies are OAEI benchmark tests ontologies [1], originating from the bibliography domain. The OAEI benchmark tests include one reference ontology O_R dedicated to the very narrow domain of bibliography, multiple test ontologies O_T manually discarding various information from the reference ontology in order to evaluate how algorithms behave when information is lacking, and 4 real world bibliographic ontologies that are generated by MIT [2], UMBC [3], University of Karlsruhe [4] and INRIA [5] respectively. The OAEI benchmark tests are *open* tests, which mean the expected results are provided for all participants.

4.2 Evaluation Criteria

We follow the evaluation criteria used by the OAEI ontology matching campaign 2007. That is, standard information retrieval evaluation measures, i.e., *precision*, *recall* and *f-measure*, are computed against the reference alignment. The *precision*, *recall* and *f-measure* are defined as follows.

Precision $p = \dfrac{\#correct\_found\_mappings}{\#all\_found\_mappings}$

Recall $r = \dfrac{\#correct\_found\_mappings}{\#all\_possible\_mappings}$

F-measure $f = \dfrac{2 \times p \times r}{p + r}$

4.3 Experimental Design Motivation

Two experiments were designed. The motivation of them is:

- The 1st experiment investigates how the approach performs in the situation where people have manually marked some mapping results for a specific mapping task, but they need help from automatic mapping tools to find the rest of mappings.
- The 2nd experiment investigates whether a model trained on one mapping task can work on another mapping task(s). Moreover, we are interested in which benchmark test(s) is more suitable as a training model. The motivation for the 2nd experiment is: in most ontology mapping cases, no ground truth is available for a specific mapping task, but a general model has been learned that can be used to find mappings. Thus, to save users' time and effort, we want to find out mapping results using the existing model.

4.4 Experimental Methodology and Results

4.4.1 1st Experiment – Within-task

The methodology of the 1st experiment is:

1. For each OAEI benchmark test, we generate candidate mapping pairs by simply combine all elements from two ontologies.
2. For each mapping candidate, we mark down their correctness according to the reference alignment (i.e. the ground truth). Simultaneously we generate various features (i.e., linguistic, structural and web features) to describe the characteristics of the mapping pair.
3. We split all mapping pairs into two groups (i.e., one is for training purpose and the other is used as testing set) by randomly choosing (e.g. 50% vs. 50%). We train two SVM models (i.e., *SVM-Class* and *SVM-Property*) on training set using *SVM-Light* package [6].

[1] http://oaei.ontologymatching.org/2007/benchmarks/
[2] http://visus.mit.edu/bibtex/0.1/
[3] http://ebiquity.umbc.edu/
[4] http://www.aifb.uni-karlsruhe.de/ontology
[5] http://oaei.ontologymatching.org/2007/benchmarks/fr.inrialpes.exmo.rdf.bib.owl
[6] http://svmlight.joachims.org/

Table 1. Structural features

Elements	Features	Description
Classes	DirPropNumDiff	The normalized difference between the numbers of the classes' direct properties
	DirPropSim	The edit distance based similarity between the classes' direct properties, i.e., $\text{DirPropSim} = \text{Avg}_i(\max_j(EditDistSim(p_{1i}, p_{2j})))$, where p_{1i} and p_{2j} are direct properties of class C_1 and C_2.
	chNumDiff	The normalized difference between the numbers of the classes' subclasses.
	chSim	The edit distance based similarity between the classes' subclasses, i.e., $\text{chSim} = \text{Avg}_i(\max_j(EditDistSim(subC_{1i}, subC_{2j})))$, where $subC_{1i}$ and $subC_{2j}$ are subclasses of class C_1 and C_2.
	paSim	The edit distance based similarity between the classes' super classes, i.e., $\text{paSim} = \text{Avg}_i(\max_j(EditDistSim(paC_{1i}, paC_{2j})))$, where paC_{1i} and paC_{2j} are super classes of class C_1 and C_2.
	depDiff	The normalized difference between the depth to root of the classes
Properties	domainSim	The edit distance based similarity between the properties' domain
	rangeSim	The edit distance based similarity between the properties' range
	motherSim	The edit distance based similarity between the properties' mother class

4. We classify testing data on two models.
5. We extract mapping results of testing data using naïve descendant extraction algorithm [11] and evaluate the results against reference alignment.
6. Finally to eliminate the bias caused by randomly choosing mapping pairs to generate training and testing data in step 3, we repeat step 3-5 10 times and report the average result as our final result.

In the experiment, two SVM models (i.e., *SVM-Class* model for classes and *SVM-Property* model for properties) are trained separately due to the difference between the structure of classes and properties. As a result, the mapping pairs of *classes* are tested on *SVM-Class* model and the mapping pairs of *properties* are tested on *SVM-Property* model. Moreover, since the number of negative examples is much larger than the number of positive examples in training data, we use a fixed cost factor (i.e. 10) in *SVM-Light* to equalize the distribution and ensure training errors on positive examples outweigh those on negative examples.

Fig. 2 shows the average f-measure of *classes* of each OAEI benchmark task tested on *SVM-Class* model. Fig. 3 shows the f-measure of *properties* of each OAEI benchmark task tested on *SVM-Property* model, in which the f-measures of benchmark tests #226, #233-#237, #240-#247, #250, #254-#257, #260-#266 are 0 is because there is no property existing for those tests. For comparison purpose both Fig 2 and 3 include the f-measure of *classes/properties* running by PRIOR+ approach [10], a non learning based ontology mapping approach.

The observations from Fig. 2 and 3 are:

1. On Test #101-#104 and #221-#247, both *SVM-Class* model and *SVM-Property* model perform as well as PRIOR+. This is because the linguistic information of these test ontologies is highly similar with that of the reference ontology and there is much less interference such as randomly generated name of classes/properties. Thus it is easy for both *SVM-Class* and *SVM-Property* model to catch useful features like edit distance that can contribute to learning models.
2. On Test #201-#210, both *SVM-Class* and *SVM-Property* model perform relatively worse than the PRIOR+ (especially on #201, #202, #208, #209). This is because the linguistic information changes too much on these tests so that it is hard to catch its linguistic and web characteristics in the training model. Meanwhile the structural feature is relatively weak.
3. On Test #248-#266, both *SVM-Class* and *SVM-Property* model perform much worse than the PRIOR+. This is because there is no name and no comments in the test ontologies at all, i.e., both linguistic features and web features are totally unavailable. The only feature available for SVM models is structural, which is relatively weak. Meanwhile, the PRIOR+ benefits from the profile enrichment process that integrates instance information, which keeps all descriptive information, to both classes and properties.
4. On real world cases #301-304, the SVM-Class model performs much better than the PRIOR+ and the SVM-Property model performs similarly as the PRIOR+ (i.e., slightly better on #301 and #302 but slightly worse on #303 and #304). The reason is our learning based approach utilizes

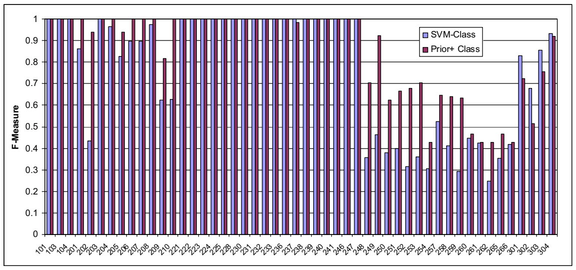

Fig. 2. Results of *classes* on *SVM-Class* model on all benchmark tests (*Within-task*)

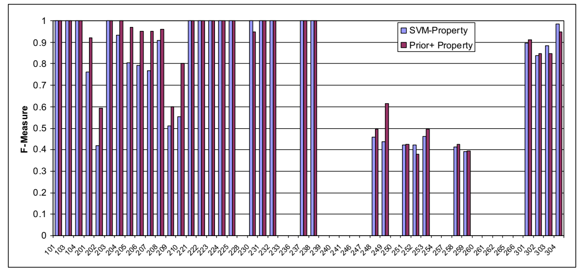

Fig. 3. Results of *properties* on *SVM-Property* model on all benchmark tests (*Within-task*)

Web feature to explore synonymous relations between concepts in ontologies. By contrast the PRIOR+ approach does not integrate any auxiliary thesaurus for such a purpose.

Our conclusion is: For learning-based approach (within-task), the performance is good when mapping task is relatively easy (i.e., #1xx and #221-247). When mapping task is more difficult, its performance is not as good as the PRIOR+ approach (i.e., #201-#210 and #248-#266). But the performance of this approach is better than the PRIOR+ on real world cases, which shows the features used in this approach make more sense on real world cases than on artificially constructed cases.

4.4.2 2nd Experiment – Cross-task

The methodology of the 2nd experiment is:

1. Same as step 1 in 1st experiment.
2. Same as step 2 in 1st experiment.
3. We train two SVM models (i.e. *SVM-Class* and *SVM-Property* for each benchmark mapping task, except #228, #233, #236, #239-#247, #250, #254, #257, and #260-#266, using *SVM-Light* package. This is because no properties exist in these test ontologies, and thus no *SVM-Property* model can be trained on them. And thus it does not make sense to test mapping tasks with both classes and properties on the model trained without property.
4. We classify testing data of all the other benchmark tests (excluding the one that has been used in training model) using the SVM models.
5. We extract mapping results of using naïve descendant extraction algorithm and evaluate the results against the reference alignment.
6. Finally we repeat step 3-5 10 times and report the average f-measure of a group of testing data (e.g., #1xx, #2xx, #3xx etc.) on each training model as our final result.

Fig. 4 shows the average f-measure tested on different data sets (i.e., all tests, #1xx, #2xx, #3xx, and more specific #201-#210, #221-#238, #248-#259). Our conclusion is: For learning-based cross-task approach, the performance is good when training data and testing data share similar characteristics. If the testing mapping task is very simple, it's easy to catch characteristics in the training model and thus get good performance with more difficult training task. Meanwhile if both training and testing tasks are difficult but with different characteristics, the performance is not as good as other approaches.

5. Related Work

Different approaches have been explored to solve ontology mapping problem, among which machine learning based method is efficient when the concepts in ontologies are associated with many instances, and it works better if many value of instances are text rather than references to other instances. In GLUE [2], a well-known machine learning based ontology mapping system, to measure the similarity of concepts the author needs to calculate the joint probability distribution of the concepts that heavily rely on the availability of instance. However, in most cases instances are just unavailable or insufficient, and it is more common to have references between instances than text description. Furthermore, the target of the GLUE is every element in the target ontology, which makes the model unable to be generalized to any application where domain has changed. Therefore they need new training data to re-build the model for each domain, which is usually unavailable.

Another approach using machine learning techniques for ontology mapping is QOM [3]. In QOM, the authors first calculate various similarities based on expert encoded rules, and then they use neural network to integrate all these similarity measures. In the contrast, the features we use are not limited to the variety of similarities.

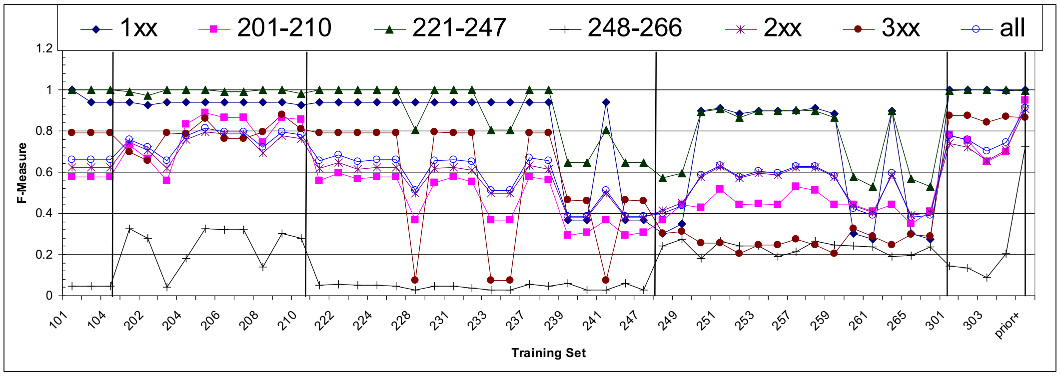

Fig. 4. Testing results of benchmark tests (Cross-task)

6. Conclusions and Future Work

In this paper, we examined a non-instance learning-based ontology mapping approach, which overcomes the limitations of previous learning-based ontology mapping approaches that either rely on the availability of sufficient instances or are domain-dependent. In the approach we treated the ontology mapping problem as a binary classification problem; generated a number of generic features; utilized these features to build training model; and conducted two experiments to investigate the performance of machine learning techniques in different situations. The experiment results show that our approach performs well on most of OAEI benchmark tests when training and testing on the same mapping task; and the results of approach vary according to the likelihood of training data and testing data when training and testing on different mapping tasks.

Future work may include: Leverage different features so as to achieve a robust semantic similarity measure, do feature selection procedure by maximizing the f-measure, and perform an active learning with Support Vector Machine algorithm.

7. References

1. Bollegala, D., Matsuo, Y. and Ishizuka, M. (2007) Measuring Semantic Similarity between Words Using Web Search Engines. Proceedings of the International World Wide Web Conference, Banff, Canada. 2007.
2. Doan, A., J. Madhaven, et al. (2003). "Learning to Match Ontologies on the Semantic Web." VLDB Journal 12(4): 303-319.
3. Ehrig, M. and S. Staab. QOM: Quick Ontology Mapping. In the Proceedings of the 3rd International Semantic Web Conference (ISWC). 2004.
4. Euzenat, J., Bach, T., et al. State of the art on ontology alignment, Knowledge web NoE. 2004
5. Gruber, T. "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition **5**(2): 199-220. 1993.
6. Hovy, E. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain. 1998.
7. R. Jones, B. Rey et. al. Generating Query Substitutions. In WWW '06: Proceedings of the 15th international conference on World Wide Web. New York, NY, USA, 387--396. 2006.
8. Kalfoglou, Y. and M. Schorlemmer "Ontology mapping: the state of the art." Knowledge Engineering Review **18**(1):1-31
9. Mao, M. Ontology Mapping: An Information Retrieval and Interactive Activation Network Based Approach. In Proceeding of ISWC 2007, LNCS 4825, pp. 931–935, 2007.
10. Mao, M. and Peng, Y. PRIOR+ System: Results for OAEI 2007. In Proceedings of ISWC 2007 Ontology Matching Workshop. 2007.
11. Meilicke, C. and Stuckenschmidt, H. Analyzing Mapping Extraction Approaches. In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.
12. Melnik, S., H. Garcia-Molina, et al. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. Proc. 18th International Conference on Data Engineering (ICDE). 2002.
13. Qu, Y., Hu, W., and Cheng, G. Constructing virtual documents for ontology matching. In Proceedings of the 15th International Conference on World Wide Web. 2006.