Combining Ontology Alignment Metrics Using the Data Mining Techniques

Babak Bagheri Hariri and **Hassan Sayyadi** and **Hassan Abolhassani** and **Kyumars Sheykh Esmaili**[1]

Abstract. Several metrics have been proposed for recognition of relationships between elements of two Ontologies. Many of these methods select a number of such metrics and combine them to extract existing mappings. In this article, we present a method for selection of more effective metrics – based on data mining techniques. Furthermore, by having a set of metrics, we suggest a data-mining-like means for combining them into a better ontology alignment.

1 Introduction

Ontology Alignment is an essential tool in semantic web to overcome heterogeneity of data, which is an integral attribute of web. In [2], Ontology Alignment is defined as a set of correspondences between two or more ontologies. These correspondences are expressed as mappings, in which *Mapping* is a formal expression, that states the semantic relation between two entities belonging to different ontologies. There have been several proposals for drawing mappings in Ontology Alignment. Many of them define some metrics to measure *Similarity* or *Distance* of entities and find existing mappings using them [4]. To extract mappings, in most of these methods, couples having Compound Similarity higher than a predefined threshold – after applying a number of constraints – are selected as output. [4] contains a number of such methods.

In this paper, given several similarity metrics we are trying to determine which of them is best for a particular data set, using data mining techniques. In order to do that, we train our techniques on some mappings for which we have a *gold standard* alignment, determining which metric is the best predictor of the correct alignment. We consider such metrics to be the best, and calculate *Compound Similarity* using them.

The rest of this article is organized as follows. In section 2, a review of related works in evaluation of existing methods and calculation of compound similarity are given. Section 3 reports our proposed method. In section 4 an example of applying this method is shown. Finally in section 5, discusses on its advantages and disadvantages are explained.

2 Existing Works

Works on metric evaluation as well as a method for aggregating results of different metrics is introduced in this section.

2.1 Alignment Evaluation Techniques

Many of the algorithms and articles in Ontology Alignment context uses *Precision* and *Recall* or their harmonic mean, referred to as *F-Measure*, to evaluate the performance of a method [4]. Also in some articles, they are used to evaluate alignment metrics[12]. In such methods after aggregation of results attained from different metrics, and extraction of mappings – based on one of the methods mentioned in [4] – the resulting mappings are compared with actual results.

In [8] a method for evaluation of Ontology Alignment methods – *Accuracy* – is proposed. This quality metric is based upon user effort needed to transform a match result obtained automatically into the intended result.

$$Accuracy = Recall \times (2 - \frac{1}{Precision}) \tag{1}$$

2.2 Calculation of Compound Similarity

The work closest to ours is probably that of Marc Ehrig et al. [3]. In *APFEL* weights for each feature is calculated using *Decision Trees*. The user only has to provide some ontologies with known correct alignments. The learned decision tree is used for aggregation and interpretation of the similarities.

3 Proposed Method

We first proposed a method to select appropriate metrics among existing set, and then introduce a method to combine them as a compound similarity. To use Precision, Recall, F-measure and Accuracy for metrics evaluation, it is needed to do mapping extraction. It depends on the definition of a *Threshold* value and the approach for extracting as well as on some defined constraints. Such dependencies results in in-appropriateness of current evaluation methods, although methods like what defines in [12] used to compare quality of metrics. We propose a new method for evaluation of metrics and creating a compound metric from some of them, featuring independent of mapping extraction phase, using learning.

[1] Semantic Web Laboratory, Computer Engineering Department, Sharif University of Technology, Tehran, Iran, email: {hariri,sayyadi}@ce.sharif.edu, abolhassani@sharif.edu, shesmail@ce.sharif.edu

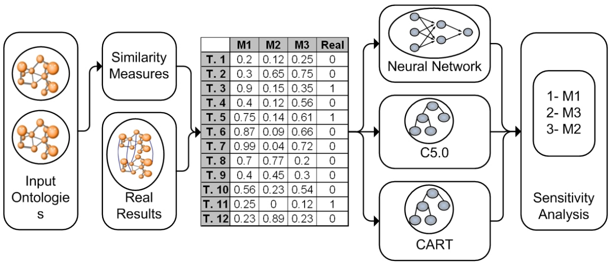
Usually String and Linguistic based metrics are more influential than others and therefore if we want to select some metrics among existing metrics, most of the selected ones are linguistic which results in lower performance and flexibility of algorithm on different inputs. Therefor as a input for the training set, a number of metrics with their associated category is considered. Categories are for example *String Metric*, *Linguistic Metric*, *Structural Metric* and so on. Proposed algorithm selects one metric from each category. Furthermore, to enforce the algorithm to use a specific metric we can define a new category and introduce the metric as the only member of it. Like other learning based methods, it needs an initial training phase. In this phase a train set - an ontology pair with actual mappings in them - is given to the algorithm.

3.1 Learning Phase

In our algorithm, selection of appropriate metrics and aggregation of them are done based on *Data Mining* Techniques.

3.1.1 Reduction to a Data Mining Problem

For a pair of Ontologies a table is created with rows showing comparison of an entity from first ontology to an entity from the second one. For each metric under consideration a column is created in such a table with values showing normalized metric value for the pair of entities. An additional column with true or false values shows the existence of actual mapping between the two entities is also considered.



	M1	M2	M3	Real
T. 1	0.2	0.12	0.25	0
T. 2	0.3	0.65	0.75	0
T. 3	0.9	0.15	0.35	1
T. 4	0.4	0.12	0.56	0
T. 5	0.75	0.14	0.61	1
T. 6	0.87	0.09	0.66	0
T. 7	0.99	0.04	0.72	0
T. 8	0.7	0.77	0.2	0
T. 9	0.4	0.45	0.3	0
T. 10	0.56	0.23	0.54	0
T. 11	0.25	0	0.12	1
T. 12	0.23	0.89	0.23	0

Figure 1. Proposed evaluation technique in detail

One table is created for each pair of Ontologies in the training set. Then all of such tables are aggregated in a single table. In this table, the column representing actual mapping value between a pair of entities is considered as target variable and the rest of columns are predictors. The problem now is a typical data mining problem and so we can apply classic data mining techniques to solve it. Fig. 1 shows the process. In this figure *Real Results* part shows the real mappings among entities of ontologies which are required during learning phase, and the *Sensitivity Analysis Rectangle* shows the results which are gain after sensitivity analysis, showing the appropriateness of metrics on the given train set.

3.1.2 Selection of Appropriate Metrics

In what following, we analysis the problem using Neural Networks as well as $CART^2$ and $C_{5.0}$ decision tress[6]. As mentioned before, columns of the table corresponding to values of metrics are considered as Predictors and the actual mapping value is the target variable. Fig. 1 shows the process. The aim is to find metrics having most influence in prediction of the target variable using Data Mining Models:

Neural Networks: *Sensitivity Analysis* for any problem is applied after a model has been constructed. With varying the values of input variables in the acceptable interval, the output variation is measured. With the interpretation of the output variation it is possible to recognize most influential input variable. After giving average value for each input variable to the model and measuring the output of the model, Sensitivity Analysis for each variable is done separately. To do this, the values of all variables except one in consideration are kept constant (their average value) and the model's response for minimum and maximum values of the variable in consideration are calculated. This process is repeated for all variables and then the variables with higher influence on variance of output are selected as most influential variables. For our problem it means that the metric having most variation on output during analysis is the most important metric.

Decision Trees: After creating the root node, in each iteration of the algorithm, a node is added to the decision tree. This process is repeated until the expansion of the tree is not possible anymore considering some predefined constraints. Selection of a variable as next node in the tree is done based on information theory concepts – in each repetition a variable with higher influence is selected among candidates. Therefore as a node is more near to the root, its corresponding variable has higher influence on the value of target variable. Hence from the constructed decision tree, it is possible to say that the metric in the root node has the highest influence.

3.1.3 Calculation of the Compound Metric

According to the results, and considering step 3-1, the problem is reduced to a Data Mining problem with the goal of finding an algorithm to compute target variable based on the predictor variables. In the Data Mining area several solutions have been proposed for these kind of problems. Among existing Data Mining solutions, we can refer to *CART* and $C_{5.0}$ [6] decision trees, A Priori for Association Rules generation [1] and Neural Networks [6]. Based on initial results among these methods, only *Neural Networks* has showed acceptable results. Neural Networks, have similar behavior with popular Alignment methods and they calculate Compound Similarity in the form of Weighted Sum with the weights is adjusted during learning.

Similar to the evaluation method, a table is constructed. As before, columns are the values selected metrics and an additional column records the target variable (0 or 1) showing the existence of a mapping between two entities. Now having such training samples a *Neural Network Model* is built. It is like a combined metric from the selected metrics which can be used as a new metric for the extraction phase.

[2] Classification And Regression Trees

4 Using the Proposed Method

To simplify the problem only String Based similarity metrics are considered. For the initial set of metrics we consider following metrics: the *Levenshtein* distance [7] which used the *Edit Distance* to match two strings, the *Needleman-Wunsch* distance[10], which assigns a different cost on the edit operations, the *Smith-Waterman* [11], which additionally uses an alphabet mapping to costs, the *Monge-Elkan* [9], which uses variable costs depending on the substring gaps between the words , the *Stoilos* similarity [12] which try to modify existing approaches for entities of an ontology, *Jaro-Winkler* similarity [5, 14], which counts the common characters between two strings even if they are misplaced by a "short" distance, and the *Sub-string* distance [4] which searches for the largest common substring. EON$_{2004}$ [13] data set is used as the training set which is explained below: *Group 1_{xx}:* We only use test 103 from this group. Names of entities in this group is remaining without any changing and cause this group not to be a suitable data set for evaluation of string metrics. *Group 2_{xx}:* The reference ontology is compared with a modified one. Tests 204, 205, 221, 223 and 224 are used from this group. *Group 3_{xx}:* We use tests 302, 303 and 304 from this group. The reference ontology is compared with real-life ontologies. *All:* We merged all the data from described sets.

Each comparison of two strings is assigned a similarity degree. After collecting output for each metric, we evaluate them for each data set as it is described in Sect. 2. Fig 2 shows the results of applying *Sensitivity Analysis* on each test set after normalization. Levenshtein similarity is the most important one. Besides Sensitivity Analysis, Decision Tree models are also used to confirm the results. In Table 1 we compare results of these techniques. All of three tests agree about importance of Levenshtein similarity on the test set. Neural Network chooses *Levenshtein* while $C_{5.0}$ and *CART* select it as second suitable metric. According to the presented algorithm and considering the fact that only one category is introduced as input, only Levenshtein is selected. In a more real situation the above steps are done for each category and one metric from each category is selected. *Levenshtein* and *Jaro-Winkler* are selected (from two imaginary categories). After creating a neural network with 4 layers and evaluation of the model on 3_{xx} test set, we got the convincing results.

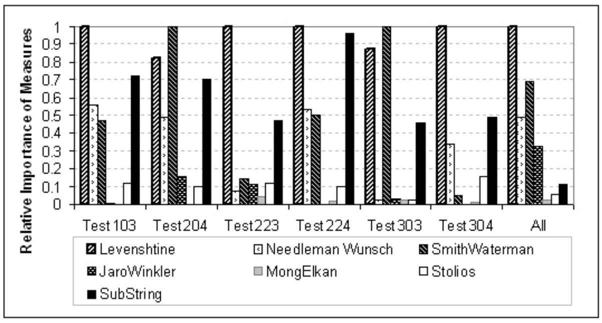


Figure 2. Evaluation of string metrics using Neural Networks

Neural Network	**CART**	**C5.0**
Levenshtein	Jaro-Winkler	Needleman-Wunsch
SubString	Levenshtein	Levenshtein

Table 1. Most 2 important metrics

5 Conclusions

One advantage of the evaluation method is the uniform treatment of Similarity and Distance metrics so that we don't need to differentiate and process them separately. This is because in Data Mining evaluation, methods, there is no difference between a variable and a linear form of it. The alignment method can be improved when new metrics are introduced. In such cases it is only needed to add some new columns and do learning to adjust weights. Some of the researchers have emphasized on clustering and application of metrics for clusters as their future works. Another advantage of this method is that we can add cluster value as a new column to influence its importance for combination of metrics.

REFERENCES

[1] R. Agrawal, T. Imielinski, and et al, 'Mining association rules between sets of items in large databases', in *ACM SIGMOD Intl. Conf. Management of Data*, (May 1993).

[2] Paolo Bouquet, Marc Ehrig, and et al, 'Specification of a common framework for characterizing alignment', deliverable 2.2.1, Knowledge web NoE, (2004).

[3] Marc Ehrig, Staab Staab, and York Sure, 'Bootstrapping ontology alignment methods with APFEL', in *Proceedings of the 4th International Semantic Web Conference (ISWC-2005)*, eds., Yolanda Gil, Enrico Motta, and Richard Benjamins, Lecture Notes in Computer Science, pp. 186–200, (2005).

[4] Jérôme Euzenat, Thanh Le Bach, and et al, 'State of the art on ontology alignment', deliverable D2.2.3, Knowledge web NoE, (2004).

[5] M. Jaro, 'Probabilistic Linkage of Large Public Health Data Files', *Molecular Biology*, **14**, 491–498, (1995).

[6] Daniel T. Larose, *Discovering Knowledge In Data*, John Wiley and Sons, New Jersey, USA, 2005.

[7] Vladimir Levenshtein, 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals', *Soviet Physics-Doklady*, **10**, 707–710, (August 1966).

[8] S. Melnik, H. Garcia-Molina, and et al, 'A versatile graph matching algorithm', in *Proceedings of ICDE*, (2002).

[9] Alvaro E. Monge and Charles P. Elkan, 'The Field-Matching Problem: Algorithm and Applications', in *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, (1996).

[10] S.B. Needleman and C.D. Wunsch, 'A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins', *Molecular Biology*, **48**, (1970).

[11] T.F. Smith and M.S. Waterman, 'Identification of Common Molecular Subsequences', *Molecular Biology*, **147**, (1981).

[12] Giorgos Stoilos, Giorgos Stamou, and et al, 'A String Metric for Ontology Alignment', in *Proceedings of the ninth IEEE International Symposium on Wearable Computers*, pp. 624–237, (October 2005).

[13] Y. Sure, O. Corcho, and et al, eds. *Proceedings of the 3rd Evaluation of Ontology-based tools*, 2004.

[14] William E. Winkler, 'The State Record Linkage and Current Research Problems', Technical report, U. S. Bureau of the Census, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, (1999).