IJCSC International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp. 61-66

ALIGNING ONTOLOGIES INTELLIGENTLY

Kalpana Nigam[1] and Monica Mehrotra[2]

[1]Maharaja Agrasen College, University of Delhi, India, *E-mail: klpn_shankar@yahoo.com*

[2]Department of Computer Science, Jamia Millia Islamia, Delhi, India, *E-mail: drmehrotra2000@yahoo.com*

ABSTRACT

In the semantic web, ontology plays an important role to provide formal definitions of concepts and relationships. Due to the presence of several similar ontologies in the same domain, there might be several definitions for a given concept. Ontology alignment overcomes these difficulties by exploring a map between similar entities that refer to the same concept in two different ontologies. This paper proposes a method to combine similarity measures of different categories such as string, linguistic, structural and instance based similarity measures. To align different ontologies efficiently, K Nearest Neighbor (KNN), Support Vector Machine (SVM) and Decision Tree (DT) classifiers and Baysian network are investigated. Each classifier is optimized based on the lower cost and better classification rate. Experimental results demonstrate that the F-measure criterion improves up to 98% using feature selection and combination of classifiers, which is highly comparable, and outperforms the previous reported F-measures.

Keywords: Ontology alignment, support vector machine, decision tree, K-nearest neighbor.

1. INTRODUCTION

Data on the semantic web is represented by ontologies, which typically consist of a number of classes, relations, instances and axioms. Ontologies have been a solution to the problem of managing the distributed information across the web. However, reuse of the existing ontologies has been addressed recently. Different attitudes of ontology designers bring about several similar ontologies in every particular domain [1, 2]. It is unlikely to find two ontologies describing one thing (concept) with a perfect overlap. This makes communication and interoperability either difficult or impossible [3]. Ontology alignment overcomes these difficulties through exploring a map between similar entities that refer to the same concept in two different ontologies [4, 5]. Therefore the importance of ontology alignment methods becomes more non-trivial, considering the fact that communication and inter-operability are necessary for a wide variety of areas. These areas include web service integration, agent communication, information retrieval from heterogeneous multimedia databases [6], learning resource management systems [1, 2], improving web-based search [7], business processes management systems [8] and so on.

An ontology alignment process usually comprises six steps: (1) feature engineering, (2) search step selection, (3) similarity computation, (4) similarity aggregation, (5) interpretation and (6) iteration [9]. Manual solution of this process is usually time-consuming and expensive. Therefore, having an automated solution becomes necessary. The current ontology alignment has applied automatic techniques in two parts: (1) training and generating the model; and (2) classification process [8]. ML techniques help to perform the last three steps of the above more efficiently. Different well-known categories of similarity methods used to measure the similarity of two ontologies include: string-based, linguistic, structural and instance-based methods. Each similarity measure is considered as a feature of the input sample, thus it is important to select effective similarity measures (features) from different categories (steps (1) and (2)). There are several works which have already exploited ML techniques towards ontology alignment. In [9] a multi-strategy learning was used to obtain similar instances of hierarchies to extract similar concepts using Naive Bayes (NB) technique. In [10], following a parameter optimization process on SVM, DT and neural networks (NN) classifiers, an initial alignment was carried out. Then the user's feedback was noticed to improve the overall performance. In [11], some string-based and linguistic (using WordNet) measures were utilized as input features. It then used CART, NN and DT based classifiers to align ontologies. In [12], string based, linguistic and structural measures (total 15 features) were used to obtain the data set of pair entities, and then the SVM, KNN and ADABoost algorithms were applied to classify the data set samples However, paper [13] presented a method for improving alignment results via not choosing a specific alignment method but applying ML techniques to an ensemble of alignment methods. Some research works [7, 8, 15, 16] have applied ontology instances in conjunction with the instance-based methods of similarity. Other studies use rule sets, RDF graph analysis, data mining and ML techniques to aggregate similarity measures of each individual category [17]. In this paper, we have combined different individual similarity metrics (features) of string-based, linguistic,

structural categories and instance based into one input sample. As each individual similarity measure is able to determine partial similarity of the whole feature space, considering all the measures simultaneously will probably achieve higher classification accuracy. The ensemble method is an active research area which gives better performance than a single classifier [18]. Some research works have shown that using a single classifier performing well may not be the optimal choice [19]. It may lose potentially valuable information contained in the other less accurate classifiers. Thus ensemble approach is proposed as a solution to combine several less accurate classifiers in this research.

2. FEATURE SELECTION

String-based, linguistic, structural and instance-based methods are four different categories of measuring similarities (features) in ontology alignment. Here, we have used 18 effective similarity methods in total from all four categories. Each method returns a similarity value in the range of [0, 1] for a given entity pair from two ontologies. These methods are briefly introduced in the following subsections.

2.1. String-Based Methods

There are several string-based methods in ontology alignment field. These techniques focus on entity's name (string) and find similar string entities. Here, the most popular methods which are already implemented in Alignment API and SecondString API have been selected [9, 20]. Because of low accuracy of each string based method, more methods from this category are used compared to the others so that each method calculates a different view of similarity (distinct feature). The overall performance can be increased through having a diversity of distinct features. The study's experimental results indicate that the following methods provide the more accurate outcomes. These methods are performed on two entity names (two strings).

2.2. Language-Based Methods

Apart from similar appearance of entity names which has been measured through the string based methods, there are some semantic similarities between which reflect the applied language in ontologies. For example, although "car" and "automobile" have almost no string-based similarity but they refer to the same concept from a linguistic point of view. WordNet is the most popular lexicon in English [5]. It arranges the word semantically rather than morphologically. WordNet is a network which has several synset. Every synset includes words with the same sense. Here, WordNet's package of Alignment API tool has been used to measure possible linguistic similarities of corresponding entity names.

2.3. Structural Methods

Ontology alignment solely based on string and linguistic similarities may fail because these similarities only investigate the entity names without considering the entity's relation to the other entities in its ontology. For instance, the result of applying the string and linguistic methods on two entities named "jackpot" from two given ontologies shows they are equal entities, while investigation of each entity in its own ontology, supposedly from two different ontologies like kitchenware ontology and game ontology, may result the opposite. Thus, structural methods are defined to evaluate the similarity of entities and relations in two ontologies. The current research has investigated two structural methods from the OLA's tool [4]. These methods compute the similarity measure of class names and their property locally, which are then aggregated into one particular measure.

2.4. Instance-Based Method

Instance based ontology matching is a new approach which uses the extensions of concepts to solve matching problems [14]. Instance matching is the task of recognizing similarity between instances with different ontology descriptions for the same real world objects. It is crucial in this ontology matching area like other data integration issues, and the quality of its algorithm can extremely affect the performance of this way of matching because it can change the overlap of extensional information; that is, common instances of concepts among different ontologies. In this paper, we investigated a distributional instance similarity measure which improves the quality of instance mapping in ontology matching scenario.

3. MACHINE LEARNING TECHNIQUES

Once the similarity features of two given entities from two ontologies are selected and measured, they will be aggregated. There are several techniques to compute the optimal aggregation for different types of similarity measures such as fuzzy, weighted product, weighted sum, Minkowski, etc. [7]. However choosing the optimum parameters of these techniques such as thresholds and other constraints is difficult. ML provides another possibility to combine different similarity measures. Here, supervised ML methods are utilized to extract the optimal model of compound metrics. Thus, the alignment problem is transformed into a supervised ML task. The basis of any ML-based ontology alignment system is a classifier. So far, numerous classifiers have been developed and applied to ML-based decision making problems. Here, the ontology alignment (classification) is regarded as a probability density function modeling. In this way, a parametric approach is used, in which explicit assumptions are made about underlying model characteristic [23]. This includes some parameters that

need to be optimized by fitting the model to the data set. In this paper, the performance of several classifiers such as SVM, KNN, DT and Naïve Bayesian techniques are analyzed to select the one with the most accurate results. These techniques are briefly introduced in the following sub-sections.

3.1. Support Vector Machine (SVM)

Given a set of training instances, which are marked as two categories of alignment and non-alignment, an SVM training algorithm builds a model that predicts the category into which a new instance falls. Intuitively, an SVM model is a representation of the instances as points in space so that the instances of separate categories are divided by a clear gap that is as wide as possible. A new instance is then mapped into that same space, and its category is predicted [24]. In other words, an SVM constructs a hyperplane or a set of hyperplanes in a high or infinite dimensional space which can be used for classification, regression or other tasks. A good separation is achieved by a hyperplane that has the largest distance to the nearest training data sets of any class. For a separable classification task, the idea is to map the training data into a higher-dimensional feature space using a kernel function where a separating hyperplane (w, b) with w standing for weight vector and b standing for bias, can be found which maximizes the margin or distance from the closest data points.

3.2. K-Nearest Neighbours (KNN)

The KNN classifier has been broadly used in ML applications due to its conceptual simplicity, and general applicability [23]. A KNN classifier is trained by storing all training patterns presented to it. During the test stage, the K stored entity pairs closest to the test entity pair are found using the Euclidian distance measure. A vote is then taken amongst those K neighbours, and the most frequent class is assigned to that test entity pair. This assignment minimizes the probability of the test entity pair in question being wrongly classified. The reader is referred to [24] for the details of this algorithm. In KNN classification, the number of neighbours i.e. K needs to be pre-defined. A single nearest neighbour technique (K = 1) is primarily suited to classifications where there is enough confidence in the fact that class distributions are non overlapping and the features used are discriminatory. But in most practical applications, such as ours, more than one nearest neighbour is necessary for majority vote. A reasonable and practical approach would be to use trial and error to identify K in such a way that it gives the lowest misclassification error rate. This is performed with different K values ranging from 1 to 9 to find the optimum value (section 5).

3.3. Decision Tree (DT)

Different methods exist to build DTs, which summarize given training data in a tree structure with each branch representing an association between feature values and a class label. The most famous and representative one is perhaps the C4.5 algorithm [23]. It works by recursively partitioning the training data set according to tests on the potential of feature values in separating the classes. The core of this algorithm is based on its original version, named the ID3. So, to have a basic understanding of how this algorithm works, ID3 method is outlined below. DT is learned from a set of training instances through an iterative process, of choosing a similarity measure (i.e. feature) and splitting the given data set according to the values of that feature. The key question here is which feature is the most influential in determining the classification to be chosen first. Entropy measures or information gain is used to select the most influential feature which is intuitively deemed to be feature of the lowest entropy (or of the highest information gain). In more detail, the learning algorithm works by: (a) computing the entropy measure for each feature, (b) partitioning the set of examples according to the possible values of the feature that has the lowest entropy, and (c) for each subset of instances repeating these steps until all features are partitioned or the other given termination conditions are met. In order to compute the entropy measures, frequencies are used to estimate probabilities. Note that although feature tests are chosen one at a time in a greedy manner, they are dependent on the results of previous tests explaining the results is one of the most popularity reasons of DT classifier in ontology alignment domain. It can be easily converted to set of rules or expression logic.

3.4. Bayesian Network

Bayesian networks (BNs), also known as belief networks (or Bayes nets for short), belong to the family of probabilistic graphical model's (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, BNs combine principles from graph theory, probability theory, computer science, and statistics. GMs with undirected edges are generally called Markov random fields or Markov networks. These networks provide a simple definition of independence between any two distinct nodes based on the concept of a Markov blanket. Markov networks are popular in fields such as statistical physics and computer vision [14]. BNs correspond to another GM structure known as a directed acyclic graph (DAG) that is popular in the statistics, the machine

learning, and the artificial intelligence societies. BNs are both mathematically rigorous and intuitively understandable. They enable an effective representation and computation of the joint probability distribution (JPD) over a set of random variables.

4. PROPOSED ALIGNMENT METHOD

The proposed system is being experimented in *SPINMap, Alignment API framework and Matlab. SPINMap* is a SPARQL-based language to represent mappings between RDF/OWL ontologies. These mappings can be used to transform instances of source classes into instances of target classes. The data sets are taken from Ontology Alignment Evaluation Initiative (OAEI) which provides framework in ontology alignment. These data sets are produced for alignment contest and provide several formats [25].

Series #301-304 represent real-life ontologies for bibliographic references found on the web. Out of these #301 is selected as training dataset, while #302-304 series are considered as test datasets. To construct the similarity matrix, similarity measures (section 2) are applied to a pair of ontologies selected from the data sets. Similarity matrix is a table with m rows and n columns, where m is the number of given entity pairs and n is the number of applied features (similarity measures). The truth alignment of each entity pair corresponding to each row of similarity matrix is called actual value. This value is defined by the expert and takes a value of 1 (i.e. aligned) or 0 (i.e. not aligned).

Figure 1: Snap Shots from SPINMap.

Having provided the similarity matrix and target values, the problem would be reduced to a supervised learning task comprised of training and testing phases. In this research, a binary classification with the objective of achieving the best possible alignments in an automatic and efficient way is introduced. Within the test stage, the trained optimum model is used to classify the new unseen similarity matrixes (test data) into two classes i.e. aligned or not aligned. This type of alignment is named System Alignment. Each classifier is quantitatively evaluated by independent test data; otherwise the evaluation would become biased and would not provide a fair assessment of the classifier performance. To assess the classifier generalization ability and consequently measure the classification accuracy, system alignment and actual value of each entity pair are compared.

4.1. Evaluation Criteria and Experiments

In ontology alignment task, precision and recall and F-measure criteria are generally used to evaluate the system's performance [28]. These measures are defined as followings:

$$\text{Preci sion} = \frac{|\text{alignment given} \cap \text{correct alignment}|}{|\text{alignment given}|}$$

$$\text{Recall} = \frac{|\text{alignment given} \cap \text{correct alignment}|}{|\text{correct given}|}$$

F-measure is basically the harmonic means of precision and recall, which defined as under

$$\text{F-measure} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}$$

F-measure is a common performance measure in information retrieval which balances precision and recall. Indeed, Alignment API provides a utility to evaluate the result of alignment [2, 5].

Here, the experiments address an aspect which has its impact on the training model and final results. Furthermore, each experiment is carried out using different classifiers (DT, SVM, KNN and Bayesian Network) and the results are compared against each other.

4.2. Experiments

Here, two experiments have been conducted. First experiment addresses an aspect which has its impact on the training model and is carried using different classifiers (DT, SVM, KNN and Bayesian Network) and the second focuses on testing the training dataset.

4.2.1. First Experiment

The first experiment has simply chosen the optimum model based on total 18 similarity measures based on string, linguistic, structural and instance based similarity measures, which are mentioned in section 2. The obtained similarity matrix then aggregated via classification. After adjusting classifiers' parameters, training model was obtained.

4.2.2. Second Experiment

Within the test stage, the trained optimum model is used to classify the new unseen similarity matrices into two classes i.e. aligned or non-aligned. This experiment explores the effect of training samples quantity on the quality of final trained model In this experiment, the number of entity pairs is increased by using other ontologies such as #102, #103, i.e. entity pairs extracted from (#101, #102) and (#101, #103). So the diversity of instances in training phase is widened. To avoid training the model with similar input samples, those samples from #102 and #103 ontologies which represent the highest variances are selected.

5. RESULTS AND EVALUATION

This study optimizes the classifiers. If every parameter of each classifier tunes well, the alignment results will be more accurate. The design of the SVM classifier architecture is simple and mainly requires choosing the kernel and its associated parameters. There are currently no technique available to learn the form of kernel, thus a Gaussian RBF kernel function has been employed. We construct a set of SVM classifiers with a range of values for the kernel parameter σ and with no restriction on the Lagrange multipliers αi. Having defined classification rate as the system alignment over the truth alignment, the most classification accuracy is achieved when $\sigma = 0.1$.

In KNN classification, the number of neighbours, i.e. K needs to be pre-defined. A reasonable and practical approach would be to use trial and error to identify K in such a way that it gives the lowest misclassification rate. We performed such an experiment with different K values ranging from 1 to 9 (K is chosen to be odd to avoid tie votes), and found $K = 3$ as the optimum K value for the application at hand.

In DT, having minimum tree without losing accuracy significantly decreases the costs. Therefore once the DT is constructed, it is configured to estimate the minimum tree with the lowest cost for every test set. Here, the minimum tree size is experimentally found to be 12.

Table 1 summarizes the best F-measure performances obtained against the test set #302,#303 and #304 for the used classifiers.

Table 1
F-measure Values

Test Set	*KNN*	*DT*	*SVM*	*BN*
#302	0.93	0.87	0.90	0.80
#303	0.87	0.85	0.76	0.86
#304	0.98	0.95	0.96	0.98

Table 2 shows the results taken from [15], that uses instance based similarity measure with DT and NB classifiers. It may be observed that the above results clearly prove our point of combining similarity measures from all the categories.

Table 2
F- Measures from Different Methods

Test Set	*DT[15]*	*NB[15]*
#302	0.759	0.753
#304	0.816	0.860
#304	0.960	0.960

In general, F-measure values which are obtained against test sets #304 and #303 are the best and worst results respectively. This is due to the fact that test # 304 has similar structure and vocabularies so to the reference ontology, i.e. #101, while test set #303 has the least vocabularies and linguistics information. This trend also validates the recent attention on reusing the existing ontologies.

6. CONCLUSION

This paper proposes an efficient method for ontology alignment based on the combination of different categories of similarity measures, such as string, linguistic, structural and instance based in one input sample. This, in turn, increases the discrimination ability of the model and enhances the system's overall accuracy. Through a comprehensive optimization process of operational parameters, our proposed model does not require any user intervention, and it has a consistent performance for both aligned and non-aligned entities. Although, this research uses very few similarity measures in its optimum model, but the possible impacts of feature reduction has been compensated by enlarging the diversity of training set samples, and choosing more effective similarity measures such as instance based similarity measures. This grants more accuracy and less computation cost which makes the proposed model appropriate even for an online ontology alignment task.

REFERENCES

[1] J. Euzenat, T. L. Bach, J. Barrasa, P. Bouquet, J. De Bo, R. Dieng, M. Ehrig, M. Hauswirth, M. Jarrar, R. Lara, D. Maynard, A. Napoli, G. Stamou, H. Stuckenschmidt, P. Shvaiko, S. Tessaris,S. Van Acker, I. Zaihrayeu, "D2.2.3: State of the Art on Ontology Alignment", *Knowledge Web*, pp. 5-12, 2004.

[2] J. Euzenat, P. Shvaiko, "Ontology Matching", Springer, 2007.

[3] L. Predoiu, C. Feier, F. Scharffe, J. de Bruijn, F. Recuerda, D. Manov, M. Ehrig, "D4.2.2 Stateof-the Art Survey on Ontology Merging and Aligning V2", *SEKT Consortium*, 2005.

[4] J. Euzenat, "Alignment API and Server", INRIA & LIG, pp. 32, 2008.

[5] R. Zhang, Y. Wang, J. Wang, "Research on Ontology Matching Approach in Semantic Web", *International Conference on Internet Computing in Science and Engineering*, pp. 1, 2008.

[6] X. Li, J. Zhang, T. Huang, "A Standardized Learning Resources Retrieval System Based on Ontology Matching", Springer-Verlag Berlin Heidelberg, 2007.

[7] A. Doan, J. Madhavan, P. Domingos, A. Halevy, "Learning to Map Ontologies on the Semantic Web", Proceeding of www, 2002.

[8] M. Ehrig, S. Staab, Y. Sure, "Bootstrapping Ontology Alignment Methods with APFEL", Springerlink, 2005.

[9] J. David, F. Guillet, H. Briand, "Association Rule Ontology Matching Approach", *International Journal on Semantic Web & Information Systems*, **3**, Issue 2, 2007.

[10] B. Bagheri Hariri, H. Sayyadi, H. Abolhassani, "A Neural-Networks-Based Approach for Ontology Alignment", *Proceedings of the Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems*, Japan, 2006.

[11] B. Bagheri Hariri, H. Sayyadi, H. Abolhassani, "Combining Ontology Alignment Metrics using the Data Mining Techniques", *Proceeding of the 17th European Conference on Artificial Intelligence, International Workshop on Context and Ontologies* (C&O' 2006), Trento, Italy, 2006.

[12] B.Shadgar, A. Haratian, A. Osareh, "Ontology Alignment using Machine Learning Techniques", *International Journal of Computer Science and Information Technology*, **3**, April, 2011.

[13] K. Eckert, C. Meilicke, H. Stuckenschmidt, "Improving Ontology Matching using Meta-Level Learning", *Proceedings of ESWC*, 2009.

[14] Stich, T. (2004)., "Bayesian Networks and Structure Learning", Diploma Thesis, *Computer Science and Engineering*, University of Mannheim, 2004.

[15] M. Mao, "Ontology Mapping: Towards Semantic Interoperability in Distributed and Heterogeneous Environment", Ph.D. Thesis, 2008.

[16] U. Straccia, R. Troncy, "oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies", Springer-Verlag Berlin Heidelberg, LNCS 3806, 2005, pp. 133–147.

[17] J. Euzenat, P. Guegan, P. Valtchev, "OLA in the OAEI 2005 Alignment Contest", 2005.

[18] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, "On Combining Classifiers", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, **20**, pp. 226-239, 1998.

[19] K. Tumer, J. Ghosh, "Classifier Combining: Analytical Results and Implications", *Working notes from the Workshop, Integrating Multiple Learned Models., 13th National Conference on Artificial Intelligence*, 1996, Protland, Oregon.

[20] SecondString Project Page, <http://secondstring. sourceforge-.net.

[21] G. Stoilos, G. Stamou, S. Kollias, "A String Metric for Ontology Alignment," Springer-Verlag Berlin Heidelberg ISWC 2005, LNCS 3729, pp. 624–637, 2005.

[22] W. W. Cohen, P. Ravikumar, S. E. Fienberg, "A Comparison of String Metrics for Matching Names and Records", *American Association for Artificial Intelligence*, 2003.

[23] R. O. Duda, P. E. Hart, D. J. Storke, "Pattern Classification", John Wiley & Sons, New York, 2001, ISBN:0471056693.

[24] E. Alpaydin, "Introduction to Machine Learning", MIT press, 2004.

[25] "Ontology Alignment Evaluation Initiative", (2009) <http://oaei.ontologymatching.org>.