Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1651–1658
Expert Systems with Applications
www.elsevier.com/locate/eswa

A novel method for measuring semantic similarity for XML schema matching

Buhwan Jeong, Daewon Lee, Hyunbo Cho, Jaewook Lee *

Department of Industrial and Management Engineering, Pohang University of Science and Technology (POSTECH), San 31, Hyoja-dong, Pohang 790-784, South Korea

Abstract

Enterprises integration has recently gained great attentions, as never before. The paper deals with an essential activity enabling seamless enterprises integration, that is, a similarity-based schema matching. To this end, we present a supervised approach to measure semantic similarity between XML schema documents, and, more importantly, address a novel approach to augment reliably labeled training data from a given few labeled samples in a semi-supervised manner. Experimental results reveal the proposed method is very cost-efficient and reliably predicts semantic similarity.

Keywords: Integrated similarity; NNPLS; Schema matching; Semantic similarity; Semi-supervised learning; XML

1. Introduction

Along with advances in the internet technology, we have been daily facing countless information over the world, without opportune evaluation of its relevance. Especially, in the context of business-to-business (B2B) applications integration, XML schema has been used as the standard means to express and exchange information among enterprise applications. The profusion of XML schemas, however, hinders enterprises from seamless and interoperable integration. For this reason, it is very important to identify proper XML schema(s) for a particular integration need. The identification process, so-called semantic matchmaking or schema matching in short, is a reasoning process to produce a set of semantic mappings among input schemas with support of semantic similarity measures. The applications of schema matching include, but not limited to, schema/ontology integration, XML message mapping, e-catalog mapping, web service discovery and composition, agent communication, and further enterprises integration, wherever heterogeneity in syntax, system and/or semantics exists (Sheth, 1999; Shvaiko & Euzenat, 2005). The key prerequisite to the success of schema matching is a reliable semantic similarity measure among XML schemas. To this end, we envision a supervised measuring tool (i.e., a classifier) used to predict semantic similarity that incorporates and synthesizes various pieces of information attributing XML schemas.

The paper envisages a schema matching framework based on semantic similarity between XML schemas. Specifically, it first presents an integrated similarity measure between structured documents (i.e., XML schema) by incorporating prominent supervised learning (i.e., neural network-based partial least squares, NNPLS), which judges the latent similarity from various similarity measures. And, more importantly, the paper addresses a novel approach, in a semi-supervised manner, to augment a reliable dataset necessary for training the NNPLS classifier. To encourage potential researchers to use the semi-supervised approaches, we reviewed a number of

* Corresponding author.
E-mail addresses: bjeong@postech.ac.kr (B. Jeong), woosuhan@postech.ac.kr (D. Lee), hcho@postech.ac.kr (H. Cho), jaewookl@postech.ac.kr (J. Lee).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.025

semi-supervised algorithms and conducted exhaustive comparative experiments with both artificial data and industrial data.

The rest of the paper is configured as follows: Section 2 provides an overview of similarity measures for XML schemas. A conceptual schema matching framework is presented in Section 3, and followed by reviews of the NNPLS classifier and various semi-supervised techniques, used for the proposed schema matching, in Section 4. Section 5 addresses the empirical studies and discussions. Finally, the paper summary with future works is given in Section 6.

2. Similarity for XML documents

2.1. Motivating example

Take a simple example of schema discovery problem (Lu & Jung, 2003) that matches a schema – Schema (b) or (c) – for a query schema a user wants to retrieve (i.e., schema (a)), as shown in Fig. 1. A näive matcher may return Schema (b) as the answer because it has the same root element name (i.e., Paper). However, this reasoning is misled because the query represents a schema for a conference paper, while Schema (b) for a newspaper article. A more intelligent matcher absolutely answers that Schema (c) is equivalent to the query (a) because Schema (c) has a similar structure (including same child elements), even though different, but semantically same, root element labels (i.e., Paper vs. Article). This example illustrates that a schema matcher must be capable of interpreting various information, i.e., contextual relationship and semantics on terms (e.g., Paper and Article), as well as schema labels.

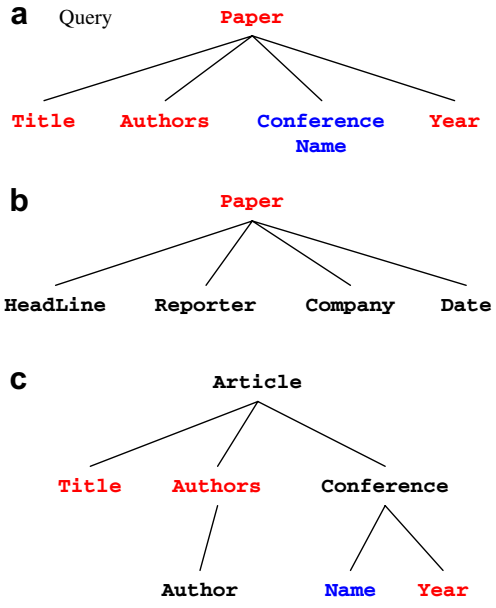

Fig. 1. A simple example to match a query Schema (a) to target Schemas (b) and (c).

2.2. Similarity measures

Similarity is formally defined as an increasing function of commonality and decreasing function of differences among objects to be compared. Similarity for structured documents, i.e., XML schema, is very complicated to define. Traditional similarity measures for XML schemas fall into one of lexical, structural, or logical category (Jeong, Kulvatunyou, Ivezic, Cho, & Jones, 2005).

A lexical similarity measure quantifies the commonality between individual XML schema labels using purely lexical information (Jeong et al., 2005). Commonly used lexical similarity measures are also divided into lexical form-based measures (e.g., affix, *n*-gram, edit distance) (Shvaiko & Euzenat, 2005; Do & Rahm, 2003) and semantic information-based measures (e.g., word sense and synonym, (weighted) edge counting, and information content-based one) (Castano, Antonellis, & Capitani, 2001; Jarmasz & Szpakowicz, 2003; Pedersen, Patwardhan, & Michelizzi, 2004; Resnik, 1995). It is noted that for the second group a lexical knowledge resource (e.g., thesaurus) is absolutely required.

A structural similarity measure quantifies the commonality between XML schemas by taking into account the lexical similarities of multiple, structurally related sub-elements/attributes of these terms (Jeong et al., 2005). A structural similarity metric typically provides a more conservative measure than a lexical similarity, because it looks beyond the individual labels. The tree structure is a native structure for XML documents; hence, it is most related to our problem context. Commonly used structural similarity measures include node, edge and/or path matching, inclusive path matching, tree edit distance (TED), (weighted-) tag similarity, weighted tree similarity, and Fourier transformation-based approach (Buttler, 2004; Zhang, Li, Cao, & Zhu, 2003; Bhavsar, Boley, & Yang, 2003). It is noted that a tree should be uniquely labeled and ordered for some measures (e.g., TED, weighted-tree similarity). Otherwise, a matching tool must be able to re-order the tree or relax the structure in local areas, which is an NP-complete problem.

It is a natural and safe way to aggregate several measures when no one knows which measure is the best to quantify the true similarity, possibly in a domain context-contingent case. In addition, using a single measure (or measures in a category) may fail to obtain optimal results. For example, two documents are not matched even though their labels are same (a high lexical similarity), when their structures are totally different (very low structural similarity) for different meanings. Another case is that two documents are highly related when their structures are very similar, in spite of their lexical similarity. Note that here we omit the description about the logical similarity category for few studies on this category have been done.

3. The proposed schema matching framework

The conceptual framework of schema matching is depicted in Fig. 2. Schema matching looks for a repository

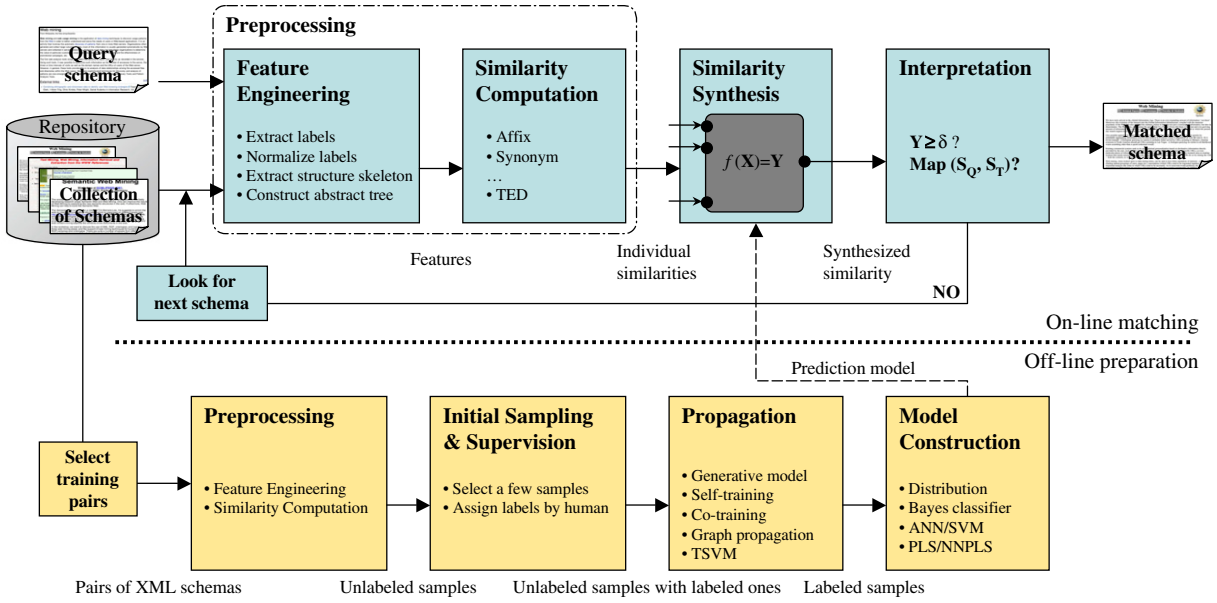

Fig. 2. Conceptual framework of real-time schema matching.

(or over the Web) and gets the most similar one to the query. To enable real-time schema matching, we present a two-phase design, i.e., online matching and offline preparation, because to obtain reliable training data and to construct a prediction model (i.e., a supervised classifier) is time consuming. The online phase is, in a narrow sense, schema matching to find the most plausible schema(s) based on the semantic similarity. The matching procedure will be detailed below. On the other hand, the offline phase supports the online matching by providing a robust supervised classifier that predicts semantic similarity from various measures. In this paper, a more important point is that this preparation phase generates reliably labeled data to train the classifier. Since labeled data require enormous expenses to collect, we adopt a semi-supervised strategy that augments unlabeled training data from a few labeled samples.

3.1. Online matching procedure

Schema matching is an engineering process of measuring similarity among schemas and then selecting the most plausible schema(s) based on the similarity (Shvaiko & Euzenat, 2005; Jeong et al., 2005; Do & Rahm, 2003; Castano et al., 2001; Jeong, 2006). In this perspective, the matching procedure is as follows:

1. *Feature engineering*: Read two XML schemas, extract features such as labels (e.g., schema or root element name) and structural data, and represent them in an internal format digestible by similarity computation. In this step, since labels are often recommended to be concatenated with several words, so-called a compound word (e.g., PurchaseOrder), such words must be normalized through a process of 'tokenization', 'lemmatization' and 'elimination' to compute lexical similarity. Moreover, the structure data used in this study is not the native XML tree structure, i.e., DOM (document object model) tree, of a schema document. Rather, we use a structure skeleton, namely an abstract tree representation, that actually captures an intrinsic structure to instance documents derived from the schema (Jeong, 2006).
2. *Similarity computation*: Select a subset of individual similarity measures and compute similarities. The selection of a subset is to reduce the computation time due to a large number of measures available. We assume that an operator has rough knowledge about which measures are more prominent to his/her own problem.
3. *Similarity synthesis*: Synthesize the individual similarity measures using a supervised classifier (i.e., NNPLS in the paper). This synthesis can be viewed as classifier construction and prediction. To construct a robust and accurate classifier, training data need to be augmented via a semi-supervised method.
4. *Interpretation and iteration*: Using a threshold, determine whether or not the schemas are matched to the query. If not, go back to step 1 with different candidate schemas from the repository or to step 2 with a different combination of similarity measures.

3.2. Offline preparation procedure

The offline preparation mainly consists in training data augmentation and classifier construction. Of them, obtaining a sufficient number of reliably labeled samples necessary for training a classifier is the main burden – requires huge efforts of human experts to determine whether two XML schemas are similar or not. Therefore, we adopt a

semi-supervised strategy to augment a number of labeled data. The propagation procedure is as follows:

1. Randomly select a number of pairs of XML schemas from a repository to be eventually used as a training data set for the (online) schema matching tool.
2. *Preprocessing*: Represent the pairs in a numeric matrix, each row of which consists of a number of individual similarity measures (assuming that a row vector represents a sample).
3. *Sampling and supervision*: Select a few samples and assign their true similarity by human experts. Since it is desirable that training data sufficiently span the solution space, for initial sampling we first cluster the whole samples and then pick out a few samples from each cluster (Fung & Mangasarian, 1999).
4. *Propagation*: Apply semi-supervised techniques, that is, train a classifier (e.g., ANN, clustering) or estimate distribution with the unlabeled samples as well as the labeled samples, and propagate to assign labels to the rest of unlabeled samples. Since all the samples labeled in this way will be used for future training for which, once again, reliability is the key, they are required to go through verification. If the classifier with the resulting samples shows poor performance, the following strategies (among which 4.2 and 4.3 require more human intervention) may be invoked.
   - 4.1. In this own step, use a different classifier, a näive Bayes classifier or other distributions, for example, for labeling.
   - 4.2. Go back step 3, and select and assign labels to a different (or additional) set of unlabeled samples.
   - 4.3. Alternatively, consider an iterative strategy used in active learning from the beginning (Brinker, 2004).
5. *Model construction*: Finally, with the training data, which are labeled by the propagation step as well as human experts, construct a supervised classifier (e.g., Bayes classifier, multilayer perceptron, NNPLS, etc.) that will be used in online similarity synthesis.

4. Methods for schema matching

This section details the proposed method that constructs a schema matching model and enables data augmentation for the propagation step.

4.1. NNPLS-based similarity synthesis

To synthesize various similarity measures, we propose to use a supervised classifier that predicts the semantic similarity between XML schemas. In particular, we advise to use NNPLS (neural network-based partial least squares) because multiple input variables (i.e., similarity measures) are usually mutually collinear and a high dimension of input variables requires much computation time. However, as described below, NNPLS overcomes such problems through the process of PCA-like dimension reduction (Bennett & Embrechts, 2003).

NNPLS is a supervised and efficient learning algorithm that extracts latent variables from observed input variables. Its basic principle is (1) to reduce the original input variable space ($\mathbf{X}$ and $\mathbf{Y}$) into a smaller PC-like latent variable space ($\mathbf{V}$ and $\mathbf{U}$), and then (2) to relate the correlated latent variables using SISO (single-input–single-output) neural networks. The dimension reduction makes it possible to build a robust prediction model from collinear and 'fat and short' data, while the use of neural networks enables to build the nonlinear relationship between the input variables. The original variables are decomposed as follows:

$$\mathbf{X} = \mathbf{V}\mathbf{P}^{\mathrm{T}} + \mathbf{E}$$
$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{F}$$
$$\mathbf{V} = \mathbf{X}\mathbf{W}$$
$$\mathbf{U} = N(\mathbf{V})$$

where $\mathbf{V}(n \times a)$ and $\mathbf{U}(n \times a)$ are the score matrices, $\mathbf{P}(m \times a)$ and $\mathbf{Q}(r \times a)$ are the loading matrices, and $\mathbf{E}(n \times m)$ and $\mathbf{F}(n \times r)$ are the residual matrices of $\mathbf{X}$ and $\mathbf{Y}$, respectively. And, $\mathbf{W}(m \times a)$ is used for constructing orthogonal score vectors of $\mathbf{X}$. A nonlinear mapping (i.e., $N(\cdot)$, a SISO network) is established between input and output score vectors whenever a pair of latent variables is extracted. This mapping procedure is repeated until the desired number of latent variables reaches. This approach circumvents the problems of over-parameterization and convergence to local optima. In addition, this NNPLS model can be collapsed into an equivalent feedforward artificial neural network with one hidden layer. More detailed descriptions about NNPLS and its collapse procedure are referred to Jeong, Lee, and Cho (2005) and Jeong and Cho (2006).

Fig. 3 depicts an NNPLS model to synthesize a number of similarity measures. The input vector consists of m individual similarity measures used (i.e., $\mathbf{X} \in \mathbf{R}^m$) and the output vector consists of only a single value (i.e., semantic similarity, $\mathbf{Y} \in \mathbf{R}$) over $[0, 1]$, where 1(0) means two XML documents are equal (totally different). Since it is hard to interpret the meaning of a value near 0.5, for example, the output vector can alternatively be defined in $\mathbf{R}^3$, for example. In such a case, three ideal output vectors, i.e., $[1\ 0\ 0]^{\mathrm{T}}$, $[0\ 1\ 0]^{\mathrm{T}}$, and $[0\ 0\ 1]^{\mathrm{T}}$, are possible to specify two schemas to be compared are matched, somehow related, or not related at all, respectively.

4.2. Semi-supervised learning techniques

From experiences, we know that supervised learning, using only labeled data to train a classifier, is outperforming in a variety of classification applications under a critical precondition that reliably labeled training samples are sufficiently ready. However, satisfying this precondition is

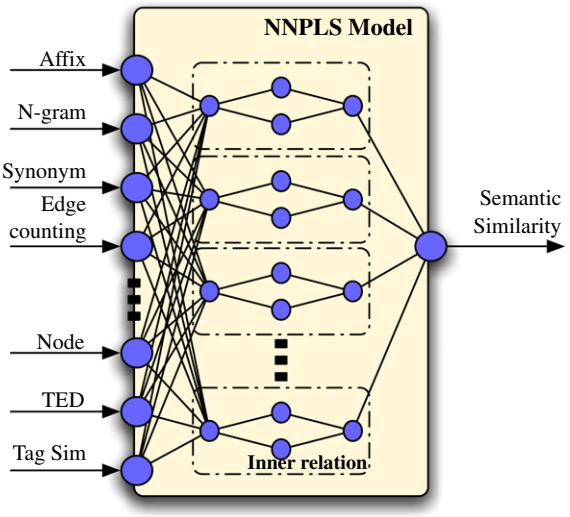

Fig. 3. NNPLS-bases similarity synthesis.

nearly impossible in most of real applications because it is very expensive and time consuming to assign desirable labels to each sample – requiring significant involvement of experienced human experts (more is better for reliability), and/or the labeling procedure may be error-prone even for experts. Meanwhile unlabeled data may be relatively easy to gather, but less useful nor practical for classification applications of interest. For these reasons, we often have to envision a novel compromised approach, namely a semi-supervised approach, that is, to use a large number of unlabeled samples (u) with a few number of labeled samples ($l \ll u$). The semi-supervised approach incrementally assigns a label to each unlabeled sample by estimating from an initial set of labeled samples. They are known to require less human involvement while to be able to achieve higher accuracy (Zhu, 2006). Here, we thoroughly reviewed various semi-supervised approaches including a generative model (e.g., expectation maximization (EM) algorithm, cluster-and-label), self-training, co-training, graph-based propagation, transductive support vector machine (TSVM), and active learning.

The generative model, the oldest semi-supervised approach, assumes a model $p(x,y) = p(y)p(x|y)$, where $p(x|y)$ is an identifiable mixture distribution. Ideally speaking, if a mixture distribution is identified, then an unlabeled sample is said to be in the same distribution. The EM algorithm is a typical one using a mixture of distributions (Zhu, 2006). Another generative approach is Cluster-and-Label, which first clusters the whole dataset, followed by label assignment to each cluster with labeled data. It is worthy noting that although they can perform well if the particular clustering algorithms (Guldemir & Sengur, 2006) match the true data distribution, these approaches are hard to analyze due to their algorithm nature.

Self-training (or self-teaching) is a common technique, in which a classifier is first trained with a small amount of labeled data available, and then guesses labels of the unclassified data gradually. After an iteration, self-training selects the most confident previously unlabeled samples, together with their predicted labels, and adds them to the training set. The classifier is re-trained and the procedure is repeated. However, it is noted that this approach is very sensitive to initial labeled samples as well as it may reinforce itself with inaccurate data (Zhu, 2006).

Co-training has two (or more) complementary classifiers trained with conditionally independent training datasets, respectively. The basic procedure is (1) to split the labeled samples into two separate datasets by features (not by samples), (2) for each dataset to train a classifier (one may train different type classifiers while other may train the same type classifier), and consequently (3) for each classifier to teach the other classifier with a few most confident unlabeled samples with corresponding predicted labels. A critical assumption is that each labeled set (or sub-features) is good enough to trust the labels by each classifier on unlabeled samples. To hold this assumption, each set must be conditionally independent so that one classifier's high confident data points are *iid* samples for the other classifier (Seeger, 2002).

Graph propagation has recently gained great attention. It constructs a graph where the nodes are the labeled and unlabeled data points, and edges reflect their proximity of corresponding nodes. It has a basic assumption that two nodes probably have a similar label if they are similar (high proximity on their connecting edge). With this assumption, labels propagate though adjacent nodes. However, a whole graph must be re-constructed to inductively assign labels to unseen data (Zhu, 2006).

TSVM copes with the weakness of discriminative methods, i.e., a semi-supervised learning cannot work well when $p(x)$ and $p(y|x)$ do not share parameters, by means of the connection between $p(x)$ and the discriminative decision boundary by not putting the boundary in high density regions. As an extension of the standard SVM to unlabeled samples, TSVM assigns labels to unlabeled samples in order for the hyperplane to maximize margin on the whole (both originally labeled and newly labeled) samples. In other words, TSVM also minimizes the generalization error bound on unlabeled samples. It is noted that the exact TSVM (i.e., integer programming or non-convex optimization problem) is NP-complete (i.e., $\mathcal{O}(2^n)$) (Chapelle & Zien, 2005).

Active learning is a special type of semi-supervised learning, and resembles to co-training having different types of classifiers without feature separation. Unlike to general semi-supervised algorithms, active learning assumes labeled samples are not initially given. Rather, the algorithm selects a few number of unlabeled samples and requests an oracle (e.g., human expert) to assign their labels. Hence, the initial sampling is critical to the performance (see Zhu, 2006; Lee & Lee, 2007a; Lee & Lee, 2007b; Lee & Lee, 2006; Lee & Lee, 2005; Seeger, 2002 for the detailed and updated algorithms of other learning techniques including semi-supervised learning).

5. Experimental analysis and discussion

For demonstration purposes, we conduct experiments with two datasets – one is a word similarity dataset[1] to prove the feasibility of using semi-supervised approaches in synthesizing several measures and the other is a real industrial dataset[2] to show the performance of the proposed schema matching in real applications. The first dataset, denoted as D_1, consists of 327 samples and ten similarity measures to compute the similarities for each sample (i.e., $D_1 \in \mathbf{R}^{327\times10}$). From the 327 samples, we use 60 samples as a labeled training set, 190 samples as an unlabeled set for assigning labels by using semi-supervised learning, and the rest 77 samples as a test set for evaluating the semantic similarity predicted by the proposed method. It is noted since the supervised labels are continuous in [0, 1], we stratify them with three discrete classes (to be a classification problem). For the second dataset, denoted as D_2, 200 pairs of XML schemas are randomly selected to score their relation (i.e., 0 for not related, 1 for somehow related, and 2 for strongly related) by four human experts, and 18 similarity measures, both in lexical and structural categories, are used (i.e., $D_2 \in \mathbf{R}^{200\times18}$). Among them, we use 60 samples as a labeled set, 90 samples as an unlabeled set, and the rest 50 samples as a test set. We use five particular semi-supervised algorithms (typical or predominant ones from each approach) – generative mixture model using EM (GMMEM), self-training using NNPLS (Self), low density separation (LDS)-based graph (Chapelle & Zien, 2005) (Graph), co-training using NNPLS (Co), and gradient-based TSVM (Chapelle & Zien, 2005) (TSVM). Each experiment is replicated 20 times.

Fig. 4 depicts the results on the first dataset (in box-plot), which evaluate how correctly a semi-supervised algorithm can assign labels to unlabeled samples (i.e., unlabeled set) as well as how precisely the supervised NNPLS classifier can predict unknown test samples (i.e., test set), in terms of correlation coefficients between supervised labels and predicted labels. The first two boxplots, 'Labeled' and 'Supervised' are obtained from NNPLS classifiers trained with 60 labeled samples and 250 labeled samples by human experts, respectively. They may be considered as the baseline and benchmarking performance, respectively. While others are obtained from NNPLS classifiers trained with 250 samples augmented from 60 labeled samples and 190 unlabeled samples using each of semi-supervised algorithms. The figure shows that the training data augmented by semi-supervised algorithms are very reliable enough to construct a supervised classifier. Clearly the five 'semi-supervised' plots are, on the median, larger than the first 'labeled' plot. This

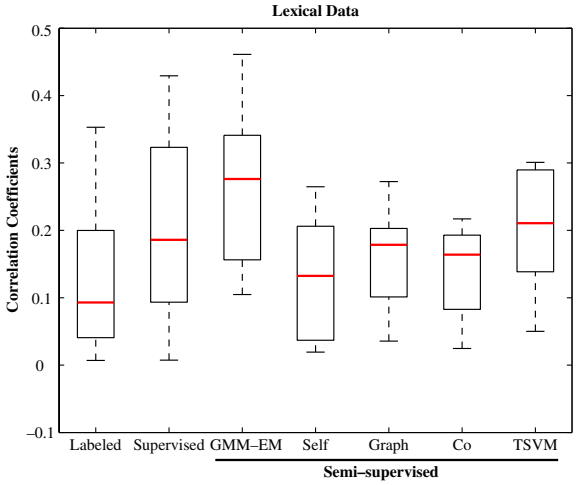

Fig. 4. Lexical similarity synthesis case: correlation coefficients between supervised labels and NNPLS-predicted labels.

strongly indicates that using unlabeled samples can considerably improve the performance of a classifier if only the unlabeled samples are properly augmented. In other words, the use of semi-supervised algorithms is a feasible approach when only few labeled samples are available. Even though some algorithms, i.e., Self-training and Co-training, are not outperforming than the second 'supervised' one, we can conclude that the semi-supervised algorithms in general are a good method to augment reliable training data. It is hard to expect for the semi-supervision to outperform the supervision, but for all that we can certainly obtain a reliably labeled dataset with lower costs. In a particular case, we dramatically reduced the costs necessary to assign labels 76% (i.e., 60 labeled samples only), compared with the costs when human experts assign labels to 250 samples, while preserving the reliability of the training data.

Fig. 5 summarizes the results on the second experiment. Similar to the first experiment, the results, except Co-training, designate that the use of semi-supervised learning when a few labeled data exist is a feasible, and the best, way in real applications. The Co-training shows a poor performance due to poor feature separation. TSVM often provides as a good performance as the supervised one, but the tendency to converge to a local minimum keeps it from an excellent performance. The other methods show stable and fair results, even though not so good as the supervised one. However, by using such semi-supervised methods, we expect a significant cost reduction to obtain such fairly reliable training data.

Some comments can be made on the characteristics of various algorithms for a potential use of semi-supervised learning in schema matching and its labeled training data augmentation as follows. The selection of a classifier for self-training and co-training is critical to determine their performance in terms of both accuracy and efficiency. With the same classifier, co-training takes much longer computation time than self-training. In the first particular case above, co-training needs 212 sec., on the average, to assign

[1] A list of word pairs widely used in evaluating the accuracy of lexical similarity measures (available online via http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353). Note that we remove some meaningless samples.

[2] A list of XML schema pairs from OAGIS 9.0 BOD specifications (http://www.openapplications.org).

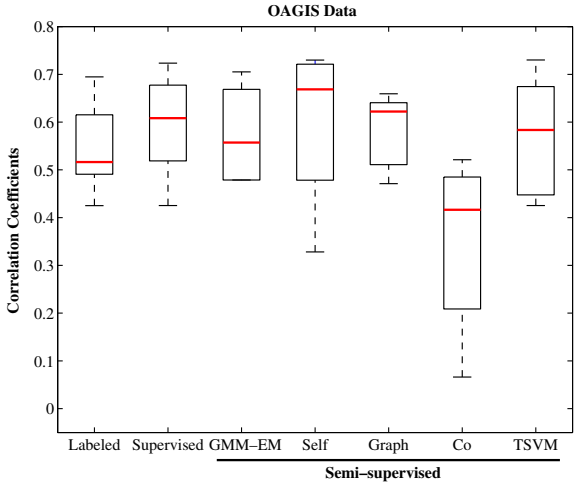

Fig. 5. OAGIS case: correlation coefficients between supervised labels and NNPLS-predicted labels.

labels, approximately eight times larger than self-training of 25 s. In the point of computation time, the others are very efficient algorithms (i.e., 0.017 s, 0.706 s and 0.572 s with respect to GMMEM, Graph-based, and TSVM, in the same case). The separation of features in co-training makes it less stable than self-training. A generative model is very efficient but relies heavily on the initial data distribution. The graph-based algorithm works well for data lying on a low dimensional manifold, but graph construction is the initial annoyance and inability to inductive propagation for unseen samples is the main difficulty to use. As shown in Fig. 5, a relaxed implementation of the original TSVM (i.e., $\mathcal{O}(2^n)$) often converges into local minima, hence its performance varies from iterations.

6. Conclusion

Schema matching is a crucial process in many applications from XML message mapping to enterprises integration and the core task of successful schema matching is a correct measurement of similarity among XML schemas. This paper proposed a similarity-based approximate schema matching, used a supervised classifier (NNPLS in the paper) to get semantic similarity by synthesizing various similarity measures, and more importantly, presented a robust approach to obtain reliable training data when only a few supervised labels are available. In particular, we utilized unlabeled data for this data preparation in a semi-supervised manner due to the limited number of labeled training samples. The results on experiments show that the proposed method is reliable and cost-efficient. More investigations on large scale real XML schema matching cases are still present.

References

Bennett, K., & Embrechts, M. (2003). An optimization perspective on kernel partial least squares regression. In J. Suykens, G. Horvath, J. Basu, C. Micchelli, & J. Vandewalle (Eds.), *Advances in Learning Theory: Methods, Models and Applications. NATO Science Series III: Computer & Systems Sciences* (vol. 190, pp. 227–250). Amsterdam: IOS Press.

Bhavsar, V., Boley, H., & Yang, L. (2003). A weighted-tree similarity algorithm for multi-agent systems in e-business environments. In *Proceedings of the business agents and the semantic web (BASeWEB) workshop*.

Brinker, K. (2004). Active learning with kernel machines. Ph.D. thesis, Computer Science and Mathematics, University of Paderborn, November 2004.

Buttler, D. (2004). A short survey of document structure similarity algorithms. In *Proceedings of the 5th international conference on internet computing (IC2004)*.

Castano, S., Antonellis, V., & Capitani, S. (2001). Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering, 13*(2), 277–297.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. In *Proceedings of the 10th international workshop on artificial intelligence and statistics* (pp. 57–64).

Do, H., & Rahm, E. (2003). COMA – a system for flexible combination of schema matching approach. In *Proceedings of the 29th international conference on very large data base (VLDB)* (pp. 610–621).

Fung, G., & Mangasarian, O. (1999). Semi-supervised support vector machine for unlabeled data classification. Tech. Rep. 99-05, Data Mining Institute, October 1999.

Guldemir, H., & Sengur, A. (2006). Comparison of clustering algorithms for analog modulation classification. *Expert Systems with Applications, 30*, 642–649.

Jarmasz, M., & Szpakowicz, S. (2003). Roget's thesaurus and semantic similarity. In *Proceedings of conference on recent advances in natural language processing (RANLP)* (pp. 212–219).

Jeong, B. (2006). Machine learning-based semantic similarity measures to assist discovery and reuse of data exchange XML schemas. Ph.D. thesis, Department of Industrial and Management Engineering, Pohang University of Science and Technology, June 2006.

Jeong, B., & Cho, H. (2006). Feature selection techniques and comparative studies for large-scale manufacturing processes. *International Journal of Advanced Manufacturing Technology, 28*(9), 1006–1011.

Jeong, B., Kulvatunyou, B., Ivezic, N., Cho, H., & Jones, A. (2005). Enhance reuse of standard e-business XML schema documents. In *Proceedings of international workshop on contexts and ontology: theory, practice and application (C& O'05) in the 20th national conference on artificial intelligence (AAAI'05)*.

Jeong, B., Lee, J., & Cho, H. (2005). Efficient optimization of process parameters in shadow mask manufacturing using NNPLS and genetic algorithm. *International Journal of Production Research, 43*(15), 3209–3230.

Lee, D., & Lee, J. (2007a). Domain described support vector classifier for multi-classification problems. *Pattern Recognition, 40*, 41–51.

Lee, D., & Lee, J. (2007b). Equilibrium-based support vector machine for semi-supervised classification. *IEEE Transactions on Neural Networks, 18(2)*, in press.

Lee, J., & Lee, D. (2005). An improved cluster labeling method for support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(3), 461–464.

Lee, J., & Lee, D. (2006). Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(11), 1869–1874.

Lu, E., & Jung, Y. (2003). XDSearch: an efficient search engine for XML document schemata. *Expert Systems with Applications, 24*, 213–224.

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet: Similarity – measuring the relatedness of concepts. In *Proceedings of the 19th national conference on artificial intelligence (AAAI'04)*.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on artificial intelligence (IJCAI-95)* (pp. 448–453).

Seeger, M. (2002). Learning with labeled and unlabeled data. Tech. rep., University of Edingburgh, December 2002.

Sheth, A. (1999). Changing focus on interoperability in information systems: From system, syntax, structure to semantics. In M. Goodchild, M. Egenhofer, R. Fegeas, & C. Kottman (Eds.), *Interoperating Geographic Information Systems* (pp. 5–30). Kluwer, Academic Publishes.

Shvaiko, P., & Euzenat, J. (2005). A survey of scham-based matching. *Journal of Data Semantics IV, 3730*, 14–171.

Zhang, Z., Li, R., Cao, S., & Zhu, Y. (2003). Similarity metric for XML documents. In *Proceedings of workshop on knowledge and experience management (FGWM2003)*.

Zhu, X. (2006). Semi-supervised learning literature survey. Tech. rep. Computer Sciences TR 1530, University of Wisconsin, April 2006.