CMC: Combining Multiple Schema-Matching Strategies based on Credibility Prediction

KeWei Tu and Yong Yu

Department of Computer Science and Engineering,
Shanghai JiaoTong University, Shanghai, 200030, P.R.China
{tkw,yyu}@apex.sjtu.edu.cn

Abstract. Schema matching is a key operation in data engineering. Combining multiple matching strategies is a very promising technique for schema matching. To overcome the limitations of existing combination systems and to achieve better performances, in this paper the CMC system is proposed, which combines multiple matchers based on credibility prediction. We first predict the accuracy of each matcher on the current matching task, and accordingly calculate each matcher's credibility. These credibilities are then used as weights in aggregating the matching results of different matchers into a combined one. Our experiments on real world schemas validate the merits of our system.

1 Introduction

Given two schemas, schema matching finds semantic correspondences between their elements. With the increasing request of knowledge sharing, numerous schema matching techniques have been developed [1–4]. Combining multiple matching strategies in a single system could achieve better performance, because in this way every possible kind of information about the schemas to be matched can be utilized. Two representative such systems are LSD [2] and COMA [3].

LSD is used in schema integration, i.e. finding mappings between various local schemas to the same mediated schema. To combine individual matchers it performs meta-learning with the *stacking* technique. COMA employs quite straightforward methods such as average and maximization for combination, so that it could avoid the burden of learning. Apart from their merits, LSD and COMA still suffer from some limitations.

For the LSD system, (1) There is one meta-learner for each element of the target schema, thus each element must have a training set. To include enough positive training samples, LSD collects equivalent elements from other schemas which have already been mapped to the target schema. However, in applications other than schema integration, such existing mappings may be scarce. (2) Meta-learners are associated with particular schemas. For applications other than schema integration, matching is performed on arbitrary two schemas, so if the schemas are new to the system, then a set of meta-learners must be trained from scratch, which is very time-consuming. (However, it is possible to release

such association at the cost of performance, i.e. to use one meta-learner for any schemas, as implemented for comparison purpose in Sect.3.) (3) Meta-learners must be re-trained if adding/removing base matcher(s). Unfortunately, adding or removing base matchers may be necessary in many scenarios. For example, if the matching task has a time limit, then matchers with iteration or training phase are undesirable and should be removed. (4) Meta-learners in LSD perform weighted sum of the results from base matchers. The weights are obtained by training, and then kept unchanged regardless of what source element is being matched. This, however, is improper because a matcher's effectiveness is always determined by both the target element and the source element.

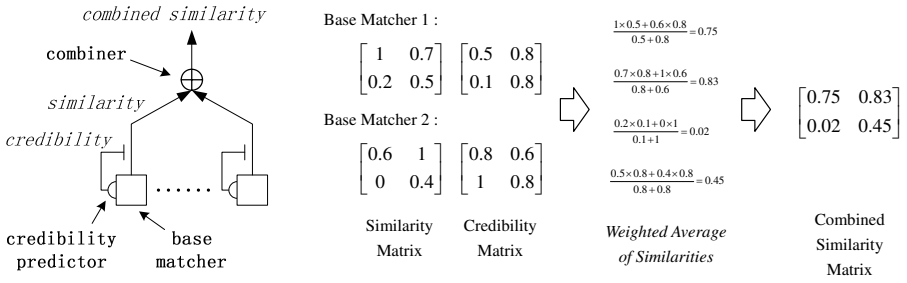
For the COMA system, (1) The combination methods in COMA, e.g. average and maximization, may be inadequate for complex situations, because in fact each base matcher has very different performance in different conditions, and these simple methods couldn't capture such performance variation. (2) If better performance is needed, users of COMA have to manually choose and configure the combination methods, such as specifying weights for matchers.

In order to overcome these limitations as well as to further improve the performance of matcher combining, in this paper we propose the CMC (Credibility-based Matcher Combiner) system. The matcher combination procedure of CMC is based on the observation that every base matcher has quite different performance in different matching tasks. For example, a matcher exploiting schema structure information would perform well for elements from XML schemas with rich structures, but the same matcher would become unreliable when it comes to "flat" schemas. Therefore, CMC dynamically predicts the accuracy of each matcher based on the characteristics of the current matching task, and accordingly calculates the matcher's credibility. Then, the results from various base matchers are aggregated based on their credibilities.

2 The CMC System

As a matcher combining system, CMC contains a set of base matchers. Most schema-matching techniques developed so far can serve as base matchers. As in LSD and COMA, a base matcher takes two target schemas S_1 and S_2 as input, and outputs a similarity between 0 and 1 for each pairwise combination of S_1 elements and S_2 elements, constituting a $m \times n$ similarity matrix, where m and n are element numbers of S_1 and S_2 respectively. Based on whether an initial similarity matrix is needed or not, base matchers can be divided into two classes.

A key operation of CMC is base-matcher combination. Unlike LSD and COMA, a credibility-based approach is employed in CMC. The underlying rationale is that every base matcher performs very differently in matching different kinds of schema element pair, so the matcher combiner should take into account the anticipated performance of each base matcher for the current matching task and accordingly assign different credit on them. For instance, structure matchers are more credible if the schema elements being matched are embedded in rich structures. To achieve this idea, in CMC each base matcher is attached



(a) The Architecture (b) Sample Procedure

Fig. 1. Matcher Combiner

with a credibility predictor, which dynamically predicts the matcher's credibility *for each pair of elements being matched*. In this way the combiner receives two matrices from each base matcher, i.e. the similarity matrix and the credibility matrix, then it aggregates all the similarity matrices into one matrix by weighted average, where the weights are determined by the credibility matrices. This procedure is illustrated in Fig.1. Notice that a matcher combiner itself could serve as a base matcher for another combiner.

With base matchers and combiners as modules, one could connect them freely. On the other hand, CMC also provides a default connection policy. There are two layers in this default structure. The bottom layer consists of base matchers that do not require initial similarity, and a combiner aggregates their outputs. Matchers requiring initial similarity constitute the upper layer, with the combined similarity of the first layer serving as their initial similarity. Finally a second combiner aggregates the results from the upper layer, as well as the result of the first combiner, and output the final similarity matrix.

To convert the final similarity matrix to the matching result, i.e. correspondences between schema elements, CMC adopts the method introduced in [3].

2.1 Credibility Prediction

In CMC, a matcher's credibility indicates how much the combiner should trust the matcher. There are two steps in predicting a base matcher's credibility: *accuracy predicting* and *converting accuracy to credibility*. An important feature of this mechanism is that the prediction procedure of one matcher is independent with the others, thus will not be affected by the addition, removal or relocation of the other matchers.

Accuracy Predicting. As the first step of credibility prediction, we predict a matcher's accuracy *for each inputted pair of schema elements*.

For a specific matcher, its matching accuracy in a matching task is correlated with several features of the task (here a *matching task* means the estimating of a pair of schema elements' similarity). For example, for some structure matchers, the number of edges connected to the element to be matched can serve as a

feature, because more edges usually indicate more structural information, leading to higher matching accuracy. With this knowledge, we predict a matcher's accuracy in the current matching task as *the mean accuracy of the set of tasks bearing the same features as the current task*. Given that a matcher's output is a numeric similarity, the mean accuracy is defined in terms of the *mean square error* (MSE) of that set of tasks: $MSE = E_{\mathcal{F}}[(sim - sim_{actual})^2]$. Here $\mathcal{F}$ is that set of matching tasks bearing the same features as the current task, and $E_{\mathcal{F}}$ represents the mathematical expectation on the set $\mathcal{F}$. sim_{actual} is 1 for matched element pairs and 0 otherwise. Obviously the less MSE is, the higher the accuracy is.

Two different strategies are presented here to estimate MSE.

Manual Rule. For some matchers, the MSE estimation is intuitive enough to be formulated manually.

Take for example the DataType matcher, which compares the data type of schema elements. Its outputted similarity is the only feature correlated to the matching accuracy, indicating the probability that the data types of the two elements are matched. If the data types are unmatched, then these two elements can't be matched at all. If the data types match, then the elements' being matched or not could be equally possible. Therefore, $MSE = (1-sim) \times (sim-0)^2 + sim \times \frac{(sim-1)^2+(sim-0)^2}{2} = \frac{1}{2}sim$.

Notice that the matcher combiner of CMC could also be regarded as a base matcher, so its accuracy must be calculated as well. With the formulas from [5], we could formulate the MSE of a matcher combiner as follows, under the assumption that base matchers are uncorrelated: $MSE = \sum_{i,j} w_i w_j C_{ij} = \sum_i w_i^2 C_{ii} = \sum_i w_i^2 MSE_i$. Here w_i is the weight of the i-th matcher, and C_{ij} is the correlation between the i-th and the j-th matcher, which is zero under our assumption if $i \neq j$, and equals MSE_i otherwise.

Learning to Predict. For most schema matchers, it is difficult, if not impossible, to manually formulate the MSE calculation. Therefore, we use machine-learning to predict MSE from the features of the current matching task.

It is important to select appropriate features of matching tasks for each matcher, and the feature set should include all the possible factors relevant to the matching accuracy. Take for example the Name matcher which compares the schema element labels: longer labels often convey more information, so the length of an element label is a feature; the outputted similarity is also a feature as matchers often have different reliability on different outputs.

With features selected, we train a learner which takes the values of the features as the input and outputs the estimated MSE. Any existing schema matches can be used to construct the training set: for each pair of elements $< e_1, e_2 >$ construct a training sample, where e_1 and e_2 come respectively from the two schemas of the existing match; let the input of the training sample be the feature values of $< e_1, e_2 >$, and the target output be the squared error of their estimated similarity by the base matcher. Since in actual schema matching the

matched element pairs are far less than the unmatched ones, we duplicate those samples constructed by matched element pairs, so that their number is equivalent to the number of samples constructed by unmatched element pairs. This last step is to avoid producing a predictor with an overwhelming bias.

Notice that although machine learning is used here, for a particular matcher once the predictor is trained, it could be applied for any matching task and no retraining is mandatory. Moreover, while all kinds of supervised learning methods can be used here, the online learning techniques [6] are preferred as the learner could improve itself in operation, thus further eliminating the worry of having insufficient existing schema matchings for training.

From Accuracy to Credibility. With accuracy (i.e. MSE) estimated, the credibility of each outputted similarity can be calculated as $e^{-C \times MSE}$. Here C is a non-negative constant, determining how fast the credibility falls with the increase of MSE. If C is positive, higher credibility is assigned to matchers with higher accuracy (i.e. lower MSE); if C is zero, the results from matchers are simply averaged, as in COMA. The empirical value of C is 1.0.

3 Evaluation

The CMC system used in the experiments consists of four base matchers, i.e. the Name, DataType, PathName and Leaves matchers, which are introduced in detail in [3]. The default structure of CMC is used, and the machine learning technique used in credibility prediction is the multilayer perceptron [6].

For comparison, another two combination methods are also tested. The *average-combination* method, which is the default combination method of COMA, simply averages the results from base matchers. The *meta-learning* method uses *stacking* for combination, and it trains only one meta-learner for all elements so as to avoid LSD's limitations discussed in Sect.1. When testing these two methods, only the combiners in the system are substituted while all the others remain unchanged. In addition, all the three methods employ the same converter, which is discussed in [3], to convert the similarity matrix to matches. For each method the converter's parameters are tuned to achieve the best performance, and it happens that all the three sets of parameters are the same as in [3].

The test data are five real-world XML schemas on purchase order, which were first used in [3]. In the experiment, we tested the three methods on all the ten matches between the five testing schemas. Considering that machine learning is used in CMC and the meta-learning method, we adopted a cross-validation strategy [6]: the ten matches were divided into five groups, and each time two successive groups were used for testing and the rest were used for training. Thus the testing was conducted for five times altogether. Four measures are adopted to evaluate the matching results, i.e. Precision, Recall, Overall [3], and F-measure [4]. Comparisons between CMC and the other two methods are respectively illustrated in Fig.2 and Fig.3, where the data is computed by subtracting the results of the contrast method from the results of CMC.

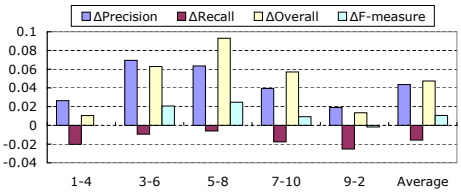


Fig. 2. Average-Combination vs. CMC

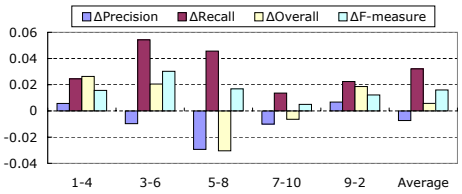


Fig. 3. Meta-Learning vs. CMC

On average the two integrated measures (Overall and F-measure) are increased significantly by CMC in both comparisons. For the two coupled measures (Precision and Recall), it is interesting to see that, CMC outperforms the average method by higher Precision while outperforms the meta-learning method mainly by higher Recall. We suppose this somewhat exposes the characteristic of the three methods, and CMC is more balanced between Precision and Recall.

4 Conclusion

CMC combines multiple schema-matchers based on their predicted credibility. It not only achieves better matching performance, but also overcomes the limitations of previous combination systems discussed in Sect.1: (1) When machine learning is used in accuracy prediction, arbitrary existing matches can be used for training, and online learning is also applicable, so collecting training set won't be a problem. Once trained, the predictor can work for arbitrary matching task and no retraining is obligatory, so the time for training is neglectable. Actually, as training can be accomplished by developers, the end users may even be unaware of it. (2) The credibility prediction for each base matcher is independent, so adding or removing matchers won't affect the combination. (3) Our combination method could take into account any available information of the current matching, which is specified as input features of credibility prediction. (4) The combination procedure is automatic, while permits users to do customization.

References

1. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal: Very Large Data Bases **10** (2001) 334–350
2. Doan, A., Domingos, P., Halevy, A.Y.: Reconciling schemas of disparate data sources: a machine-learning approach. SIGMOD Record **30** (2001) 509–520
3. Do, H.H., Rahm, E.: COMA — A system for flexible combination of schema matching approaches. In: VLDB 2002, Morgan Kaufmann Publishers (2002) 610–621
4. Berlin, J., Motro, A.: Database schema matching using machine learning with feature selection. In: Proc. of 14th Intl. Conf. on Advanced Information Systems Engineering (CAiSE). (2002)
5. Perrone, M.P., Cooper, L.N.: When networks disagree: Ensemble method for neural networks. In: Neural Networks for Speech and Image processing. (1993)
6. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)