Calibration alternatives in schema matching

BALAZS VILLANYI, PETER MARTINEK, BELA SZIKORA
Department of Electronics Technology
Budapest University of Technology and Economics
H-1111 Budapest, Goldmann György tér 1-3
HUNGARY
villanyi@ett.bme.hu http://www.ett.bme.hu/

Abstract: - Schema matching is the task of finding related entities in various input schemas. This task is crucial in enterprise integration scenarios, as companies have different systems and the schemas of these systems are heterogeneous. In order to be able to initiate and maintain communication among enterprise systems the resolution of the heterogeneity is inevitable. We have analyzed several schema matching solutions and concluded that they performance may seriously vary in different integration scenarios. Our main focus fell on the reduction of this variation. Consequently, we have developed several techniques to able to cope with the task. In this paper we present our solution alternatives in the subfield of schema matcher calibration.

Key-Words: - schema matching, algorithm optimization, reference approximation, database integration

1 Introduction

After the computers became available on the mass market for reasonable price, companies have strived to automate their business process using various information systems. However, this endeavor resulted in non-compatible systems as the communication of systems was not issue at the beginnings. For a few decades, these systems operated on their own, without any interfaces to other systems. If communication was indeed needed, the operator had to take care of the data transfer between systems, usually with manual, laborious effort.

As information technology developed, the communication between systems has become inevitable. With the appearance of networks, further stress was imposed on the architects to design systems which are capable of communication. However, there was the problem of system heterogeneity. The lack of compliance between systems and the lack of industry standards has lead to the evolution of incompatible systems. In order to carry out communication, the heterogeneity was to be resolved. Even the introduction of industry standards would not help, as companies tend to preserve their legacy systems because of various reasons.

The heterogeneity can be resolved by means of integration techniques. Integration can be carried out at several system tiers. One of these system tiers is the database. Databases can be described with schema definitions, which stores information about the database entities, database structure and constraints, etc. Database level integration involves the formulation of the integrated database schema, which consists of the union of the input schemas.

Schema matching helps the implementation of this process by identifying semantically related entities. It is a pre-integration task, whereby the potential semantical overlapping among schemas is inspected. This an essential step towards integration as the consistent communication among systems does require a mapping among schemas, otherwise information may not be transferred correctly from one system to the other.

Schema matchers are classified into three major categories. Linguistic matchers make decision on the relatedness of entities based on syntactic and semantic evaluators applied to strings, like the name and type of the entity. Structural matchers, on the other hand, perform the same job by comparing the structure of schema, the sequence of entities and their order of descendance. And lastly, the constraint-based matcher support the decision of the latter two evaluators by evaluating the constraints involved in the input schemas.

Several schema matchers have been published in the last years. Some of them are really powerful and are capable of outstanding performance. Their performance is measured both by their required runtime and their accuracy. Out of the two, accuracy seems to be the decisive factors in papers like [2, 4, 7, 9, 10]. The phenomenon can be explained with the following reasoning. The accuracy of schema matchers is crucial in enterprise integration scenarios, where even smaller errors are not tolerated when working on confidential data – as it
is the case most of the time. Consider for example a business-to-business integration scenario where a bank is involved. Consequently, we can conclude that so long that perfectly accurate schema matching cannot be provided, runtime remains a background factor.

The accuracy is measured through accuracy measures. The three most wide-spread measures are the precision, recall and the f-measure [1]. This latter incorporates the former two as the f-measure is formulated as the harmonic mean of precision and recall, which are calculated based on the confusion matrix.

There is another factor, which is strongly related to the inaccuracy and to the need for impeccable mapping. Namely, a human evaluator is required for a posteriori correction of the matching. This is not only expensive, but also time consuming. There is also a natural limit to the size of the schema matching problem, beyond which the human evaluator might fail completely. As a direct consequence, the task of the a posteriori human supervisor shall be radically reduced and eventually completely eliminated.

Our goal is then to provide techniques which would enable the performance optimization of existing solutions. We have already published several techniques in the subject. In this paper we present our new technique and compare them with each other.

The paper is divided into section as follows. This first section serves as a prelude to the problem formulation describing schema matching and our goal statement. The second section is dedicated to the related works. The third chapter explains the calibration and why it is needed. The forth chapter proposes reference approximation techniques, while the fifth introduces the membership function based threshold. Our experimental results are detailed in chapter six. We present our conclusion in section seven.

2 Related Works
Several papers have published in the subject of schema matching, like [2, 4, 7, 9, 10]. In this chapter, we aim at the presentation of those works, which we either tested or it encompass techniques similar to ours.

The similarity flooding [11] is a schema matcher distinguished by the iterative flooding in a graph called extended similarity propagation net. The similarity propagation net is constructed from the input schemas by a complex conversion process. The iterative similarity flooding is delimited by either a predefined number of steps or a by a minimal difference between steps. Although the algorithm performs well in most of the cases, its lack of parameters is a huge drawback if optimization comes into question.

The NTA [10] allows more optimization. The main components it encompasses are the name similarity, the related term similarity and the attribute similarity. The structural evaluation is performed in the attribute similarity as it also includes a recursive approach and uses similarity values of lower schema level entities. Thus the similarities of the attributes influence the similarity of the entities themselves and this is true for every schema level. We soon became convinced that the NTA is capable of outstanding performance, but only if their parameters are chosen optimally. Even by close to optimal parameters, the performance may seriously lag back.

Also the WordNet based evaluator [4] can be appreciated for its outstanding performance characteristics. On one hand, one might prefer it because of the complex linguistic evaluation which uses an external dictionary, and also the context based structural matching appears to be hard to compete with. On the other hand, this complexity results in huge runtime requirements, which might hinder the practical applicability of this approach. Based on what we learned in [13], we substituted the dictionary based linguistic matcher with a syntactic based one and managed to retain the exceptional performance characteristics. Nevertheless, the accuracy cannot be expected to be too high without the optimal parameters.

Authors of [8] also came up with optimization solutions. They are also seeking the optimization of existing solutions. In their proposal they present a methodology, which allows us to find optimal linear combination of candidate techniques or – as they are referred to in the article – knobs. Although the linear combination also involves the analysis of optimal parameters, they do not explicitly address the optimal parameter set definition of existing schema matchers, only through the combination of them.

In [13] we have presented the basics of our schema matching optimization framework and also our main optimization approaches were presented. In our terminology, calibration is the methodology of finding optimal parameter sets, whereas comparative component analysis encompasses techniques which can be useful for optimal recombination of existing solutions. These two steps constitute two essential steps of our framework. We have further developed our framework in order to be even more potent in the optimization of schema
matchers. In this paper we present our new, extended set of techniques.

3 Problem Formulation
As presented in the introduction, the accuracy of schema matchers is crucial, because of the tremendous expense of correction. We have analyzed the techniques [4, 10, 11] among others. Our analysis showed that the performance of algorithms is strongly dependent on the actual scenario. Even by fine-tuning the parameters of a given solution, its accuracy was observed to be substantially higher in one scenario than the other.

We concluded that the nature of this performance variation lies in the sub-optimality of the schema matchers in given scenarios. Fortunately schema matchers do have various parameters. To these parameters belong the weights of the linear combination, which enable the components of schema matchers to have more influence on the end result if desirable. In papers like [2, 4, 7, 9, 10], authors provide general guidelines for the choice of these weights. Earlier [9] we showed that these general guidelines are not sufficient in certain scenarios, because the guidelines cannot be applied to every scenario. That is why we propose the methodology called schema matcher calibration, which are targeted at finding optimal parameter sets.

So far we have identified two subcategories in the calibration. The first category is called reference approximation, whereby the task is to find parameters with which the result returned by the algorithm best approximates the predefined reference. Another category in the schema matching calibration is identified as accuracy measure maximization, where the formulas used to evaluate the accuracy of schema matchers are derived so that they should include the parameters of the schema matching algorithm and exclusively them. Then the formula can be maximized and a subsequent maximization problem is gained.

Unfortunately, this maximization problem always involves a non-continuous function as a binary decision on the returned semantic value being match or non-match is part of accuracy measures. Thus the runtime efficient calibration alternative is the reference approximation.

4 Reference approximation
As already mentioned, the reference approximation encompasses techniques which enhance the accuracy through providing means by which the values returned by the schema matcher will be closest possible the reference values.

These methods follow an indirect approach, as the accuracy measures used to evaluate the accuracy are not involved in the optimization process. The behind lying assumption which enables the omission of measures is the following: the better the returned values approximate the reference, the higher the accuracy measures shall be. It is indeed always fulfilled, otherwise the accuracy measures do not qualify.

However, there can be more than one technique which can fulfill the afore-mentioned criteria as you will see in the subsections. The following enumeration is not supposed to be exhaustive, the purpose is to show some methods which we used during our experimentations.

4.1 Global approximation
This technique was the first one in our reference approximation repertoire, thus it may be the least complex approach among the others. The problem is formulated as the minimization of the aggregated difference of among reference and returned values. Having summed the difference as MSE (Mean Squared Error), we can minimize the problem. The problem is further explained through the equation below:

$$D = \min \left( \frac{\sum_{i=1}^{M} (\sum_{j=1}^{N} (r_i - w_j a_{ji}))^2}{M} \right)$$ (1)

where $r$ is the vector containing the reference values, $w$ is the vector of weights and matrix $A$ holds values returned by the components. The weights can be expressed and by means of mathematical analysis, we can calculate end formulas. Fortunately the gained end formulas for the most typical scenario with three weights, can be easily calculated even manually. The formula below shows how one weight is expressed as a function of the other parameters.

$$w_x = \sum_{i=1}^{M} \left( \sum_{k=1 \neq x}^{N} w_k a_{ki} \right)$$ (2)
4.2 Approximation with a two step linear program

Nevertheless, the general formula should be solved for each and every schema matching algorithm. However, the production of end-formulas can be tiresome, thus we inspected a more comfortable alternative to the formula based parameter computing. Our effort resulted in a two step linear program, where in the first part only the maximal approximation value is sought after, which is followed by the computing of weights which produce the best approximation and all fall below the maximal approximation value computed in the first part. This is two step solution is necessary, otherwise maximal approximation might be reached by local optimums, whereby extreme distance from the approximated value might occur, which is highly undesirable.

Consider for example a scenario, where we seek the optimal parameter set and the schema matching algorithm has three weights. In this particular example, the linear program for the maximal approximation would look like this:

\[
\begin{align*}
\text{min: } & x; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 + z_0 = 1; \\
& 1 \times w_0 + 0 \times w_1 + 1 \times w_2 - z_1 = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_2 = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_3 = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 + z_4 = 1; \\
& 0 \times w_0 + 0 \times w_1 + 0.125 \times w_2 - z_5 = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_6 = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_7 = 0; \\
& 0 \times w_0 + 0.0714285714285714 \times w_2 - z_8 = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_9 = 0; \\
& 0.5 \times w_0 + 0.5 \times w_1 + 0.5 \times w_2 + z_{10} = 1; \\
& 0 \times w_0 + 0.0625 \times w_1 + 0.0625 \times w_2 - z_{11} = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_{12} = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 - z_{13} - z_{14} = 0; \\
& 0 \times w_0 + 0 \times w_1 + 0 \times w_2 + z_{15} = 1; \\
& z_0 < x; \text{//for every z this condition must be met} \\
& 0 <= z_0 <= 1; \text{//for every z this condition must be met} \\
& \ldots \\
& \ldots \\
& \ldots \\
& \ldots \\
& \ldots
\end{align*}
\]

In this particular example the x (as the maximal approximation) is output as 0.5. This should be substituted into the task as an additional condition.

4.3 Weighted local approximation

Another alternative to the global approximation might be a local approximation. However there is no warranty that we can achieve the desired maximal approximation through local approximations. That is why we combined our local approximation approach with the employment of weights.

Weights add more than just the right balance among local maximal approximations. It gives one the chance the stress a few values more than the other. Consider for example that in an enterprise integration scenario, the best approximation of a few business schema objects is crucial. In this particular scenario for example, the application of weights is especially useful, provided the chosen schema matching does not allow good approximation because of its mediocre performance. The formula below presents the base task of weighted local approximation:

\[
D = \min \left( \frac{\sum_{i=1}^{M} w'_i (\Sigma_{j=1}^{N} (r_i - w_j a_{ij}))}{M} \right)
\]

where \(w'\) denotes the weights we have given to the value approximations. The higher the \(w'\), the more precisely the approximation will be. However, if every approximation is weighted with nearly the same values, we will end up having same approximations, which can be imprecise, even if we used high \(w'\) values.

5 Accuracy measure maximization with the application of membership functions

The other calibration method is the accuracy measure optimization. As already briefly discussed, thereby the goal was to derive such formulas from the measure formulas which would enable the expression of accuracy with the algorithm parameters involved. The obtained formulas turned out to be complex, and – due to the compulsory threshold injection – and they encompass non-continuous functions. As a direct result, the maximization problem can only be solved with approximation techniques, which are very resource consuming. The decision which measure to maximize, depends on the actual scenario. This approach then explicitly contains the threshold of the algorithm as parameter. This gives rise to two consequences. Firstly, the maximization problem won’t be continuous, thus it can only be
resolved through approximation. Secondly, the semantic values near the threshold might be wrongly classified as matching or non-matching, while their actual presence near the threshold is completely neglected.

In order to reconcile this inaccurate cutting effect of the threshold, we propose the usage of membership functions. They are particularly appropriate for this task as strict threshold can be left out. We have also researched the most adequate membership functions. Our research embraced the potential usage of interpolation, spline, spline interpolation curves. However their usage did not prove to be beneficial for schema matching. We have also considered techniques like smoothing and curve fitting. The recurrent problem was that the result curve did not match the ideal ones.

Hence we tried to find a curve which best approximates the ideal curve and parameterize it in a way, so that it can applied in arbitrary chosen schema matching scenario. Our choice fell on the sigmoid function, the form of which especially fits that of the ideal curve. After parameterization, we gained the following formula for general scenarios. In the formula below, $x_1$ and $x_2$ denote two values identically positioned from the threshold:

$$f(x) = \frac{1}{1 + e^{\frac{12}{x_2 - x_1}(x - \frac{x_1 + x_2}{2})}} \quad (4)$$

6 Experimental results

The goal of our experiments was to find out which calibration method performs best. We have tested and analyzed our reference approximation solutions on six different test schemas with three different schema matcher solutions.

In our approach, there were two measures of goodness. We characterized accuracy with the value of approximation and the f-measure subsequently counted on the returned values. In order to be able to count the f-measure, a threshold was also needed. We strived to focus on the performance of the reference approximation values and that is why we did not want to distort the acquired approximations with too complex threshold injection. As threshold is not provided by reference approximation, we defined it as the semantic value of the lowest matching pair. We might as well have the semantic value of the highest non-matching pair or a value between the matching and non-matching pairs.

The approximation was calculated as follows. For every given entity we have calculated the distance of the returned value from the reference and subsequently their average was calculated. This average is regarded as the measure of approximation.

As you can see on the diagrams below, the approximation and the f-measure values convey similar, but not the same information. The behind lying reason is that the f-measure is calculated using a fix threshold, while the approximation does not include it. The distorting nature of the threshold was already detailed in the previous section. The diagram below shows the average approximation values produced with the NTA in six different scenarios:

![Fig. 1. The average approximation values obtained with the reference approximation techniques in six different scenarios](image)

The other diagram below shows the calculated f-measure values produced with the NTA in the same six scenarios:

![Fig. 2. The f-measure values obtained with the reference approximation techniques in six different scenarios](image)

Both diagrams show similar results, but in different ways. In case of the approximation the lower the value, the more accurate the calibrated algorithm. While in case of the f-measure the higher the obtained f-measure value, the more accurate the calibrated algorithm.
According to what can be observed on the diagrams, we concluded that the weighted local approximation is outperformed by the global approximation. The linear programming solution is an even better alternative to both of them. On the other hand the weighted local approximation is especially useful in scenarios, where the approximation of a few values is more crucial than the others.

These diagrams also convinced us that different approximation and thus accuracy can be obtained in different scenarios even with the same optimized algorithm. From the available calibration techniques always the most appropriate should be chosen as there were cases where for example the global method outperformed the linear programming based one.

7 Conclusion and future works
In this paper we have presented new schema matcher calibration techniques, which provide further alternatives for the parameter optimization of schema matchers.

We also compared their performance and concluded that most of the time the linear program based approximation conveys the best results, closely followed by the global approximation. However, the weighted local approximation is useful, when the approximation of a few important entities is essential.

We also described the distortion effect of the threshold used for the evaluation of the accuracy and provided a possible remedy for the phenomenon: we have investigated several functions and their applicability in the resolution of the problem. One especially appropriate candidate is the sigmoid function, which can resolve the beneficial distortion provided it is parameterized correctly.

In the future, we will strive to propose even more calibration techniques especially in the field of accuracy measure maximization, where the huge runtime requirement of the optimization algorithms might hinder the practical applicability of them.

References: