Optimization and comparison of schema matching solutions

PETER MARTINEK, BALAZS VILLANYI, BELA SZIKORA
Department of Electronics Technology
Budapest University of Technology and Economics
Goldman Gy. tér 3., 1111 Budapest
HUNGARY
martinek@ett.bme.hu

Abstract: - As the number of data schemas grows, the need to find correct mapping from one to another is also becoming more stressing. Due to the representational heterogeneity of schemas, the solution is far from trivial. The ultimate solution is yet to come, but several promising algorithms have already been published. We have studied and implemented numerous schema matching approaches. It has turned out that their accuracy is strongly dependent on their configuration parameter settings and the specific input schemas. However in the current literature these factors are not taken into account by the performance analysis. Hence our goal is to set up a universal and correct benchmarking method. We have also developed methods that enable their correct parameter adjustment and permit the unbiased comparison of their performance. These techniques incorporate sophisticated mathematical formulas. Furthermore an indirect approach is also offered, which should also ease the correct parameter adjustment. Eventually we have developed some computational approaches that have been implemented and tested. We have conducted several experiments, which were performed on different kinds of test schemas and validated our algorithms.

Key-Words: - Optimization of schema matching algorithms, Determine possible accuracy maxima, Similarity measure methods

1. Introduction

Due to the lack of comprehensive overall standardization between data schemas, solutions dealing with the integration of diverse data structure have become an important issue in business informatics. For this task, the identification of semantically related concepts in schemas is fundamental. This process however encompasses a lot of consideration, e.g. assessing schema similarity through diverse schema properties. Furthermore, current schema matching approaches simply do not offer a universal solution. The best possible result could only be acquired through the proper understanding of the scenario, the schema, the context, the task and maybe several miscellaneous user needs. The problem of schema mapping is referred to as schema matching, and several solution candidates have been aired, enumerating some cardinal features.

However most of the presented solutions are not reliable in the meaning that their output should be supervised before they could be deployed, so the human intervention cannot be set aside. This is not only inconvenient but manifests in a substantial overhead. They evaluation process should be improved, so as to crack down the superfluous time expense. The available algorithms incorporate a lot of parameters and occasionally fine interpretative distinctions. We were aimed at the identification of these and harnessing them so that they produce better results without the prior human schema analysis and solver fine-tuning.

Every known experiment that should evaluate the performance of the schema matcher seems to be conducted under artificial conditions, distorting the conclusion. We can only draw a righteous and evenhanded conclusions, if we warrant that every candidate face the very same test conditions. That is to say they are given the same input schema, and they are all optimized for the specific scenario, so that we acquire their best possible result. This probe defines the correct order of algorithm accuracy performance, enabling us ranking the existing solutions.

The paper presents the computational methods that enable the scenario based optimization of schema matching algorithms. The paper is structured as follows: the second chapter lists some related works. The detailed description of the accuracy evaluation of schema matcher methods is presented in chapter three, while chapter four encompasses the accuracy enhancing methods. Our conclusion is detailed in chapter five.

2. Related Work

Numerous researches concerning the schema matching are available [1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]. During our experiment we analyzed these approaches and implemented the approaches presented in [1, 7, 8]. These algorithms are detailed in chapter four.

The solution in [4] is a generic schema matching tool, called COMA+. The peculiarity of this proposal is
that it is not a schema matcher by itself, but constitutes a sophisticated platform in which several others can be integrated. Authors believe that with this platform the combined advantage of schema matcher can be exploited. The individual matchers are arranged in a library. The solution provides scalability by fragment schemas into subsets, trying to capitalize on the divide and conquer principle. This approach also contributes to the flexibility of the platform. On the other hand, it is not fair when comparing algorithms, because it does not include parameter optimization.

A promising approach to the schema matching problem is presented in [1]. In this paper the algorithm called Cupid is described. It has a complex evaluator incorporating a composite structural matching and a linguistic matcher. The latter one provides initial value based on string-based node comparison. According to the comparative study presented in the paper, the Cupid outperforms the DIKE[13] and the MOMIS-ARTEMIS[12]. This comparison however lists only capabilities and does not tell us about their accuracy performance. An actual result of a test measuring accuracy on test schemas is obviously missing. The capability comparison does not provide a clear view of the goodness of the individual solutions.

We have observed flaws by the comparison method presented in [7]. Although the evaluation in the paper introduces the Precision, Recall and F-measure, the accuracy result is given based on a single test schema. The question whether other candidate solutions were optimized for test scenario remains open. They also fail to mention the particular output of the matcher may not be compatible with that of other approaches. They distinguish two type of nodes based on their path, but some other approaches only distinguish types. How this controversy should be resolved remains the subject of personal judgment. The problem how linguistic similarity values are elicited from the WordNet [3] is not detailed enough, implementations may provide various similarity values which have serious impact on the result.

3. Evaluating Accuracy

Having results at disposal is in itself not expressive when goodness and quality comes in to question. In order to decide on their accuracy, the results have to be in a compatible, comparable form, what will be covered in this section.

The semantic distances – which are similarity characteristic values ranging from 0 to 1 – does not tell us whether they are meant to be matches or not. So the problem expands, encompassing the need of determining a limit called threshold that cuts the result set into two halves. Obvious that the adequate calculation of this value is at least as important as the proper parameter set of the solutions.

The results are stored in matrices, so that the values are more expressive to the human supervisor. Only the compatible format should be warranted, which entails some consideration, among others the format difference [1] and [7] must be untangled. This format enables to discuss the result accuracy relatively to each other effectively. After injecting the threshold into the similarity matrix, it includes only 0 and 1 which makes it easy to compare with the reference solution.

Other important requirement we should fulfill is the definition of the reference solution. It is not so straightforward because of the aforementioned reasons. To be unbiased we decided to make a survey involving some twenty evaluators, whose task was to solve problem under fairly similar conditions as it would be in the real life. The willing volunteers submitted their solutions which then were summarized after the necessary filtering. As the evaluators varied on a large scale, it has turned out that some of them clearly lacked the professional skills to give a perfect match. This is the reason why the necessity of the filtering ensued. This inaccuracy is not uncommon among human users, the survey clearly shores up our assumption on the complexity of the task.

Having the algorithm and reference results available, every condition needed to evaluate the efficiency is met. To calculate performance we need accuracy measures extracted from the result matrices. We used the most prevailing measures: the Precision, the Recall and the F-measure. They are best known for their usage in information retrieval, but they are not unheard of in other scopes of computational accuracy measurement. After making a brief survey, we found that these measures are utilized to describe the goodness of the individual solutions by the overwhelming majority of schema matching performance analysis. The measures are calculated with the formulas as follows: (1) Precision Formula, (2) Recall Formula and (3) F-measure formula.

\[
\text{Precision} = \frac{|\text{proposed matches} \cap |\text{relevant matches}|}{|\text{proposed matches}|} \quad (1)
\]
\[
\text{Recall} = \frac{|\text{proposed matches} \cap |\text{relevant matches}|}{|\text{relevant matches}|} \quad (2)
\]
\[
F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)
\]

Accuracy is then defined as the value of either the Precision or the Recall or the F-measure. Our main objective was to define methods which maximize these measures for a given scenario and algorithm. Note that this is not always possible as [8] does not define intermediate similarities which then could be
summarized by a weighted sum. In this case other methods are necessary which include the possible partial revision of the algorithm in question. In this article, we focus on the weight-based calibration; on the other hand we should note that we came up with amending suggestions to [8].

The main goal is to maximize a measure for a given algorithm and scenario. To achieve this, we should seek the optimal weights and threshold. We call this task calibration or parametrization, referring to possible analogy to other scopes of science. For the problem to be manageable we should find correspondence among the algorithm properties. The most important of them is given by the definition of the measures. We are able to calculate them for every parametrization, thus when observing the behavior of these measures during the fine-tuning of the factors we witness the correspondence between them. More specific is the characteristic that weights complement each other to one. It implies that we should care about one less weight, as the last one can be expressed as the function of the others. In our case it means that we should tackle the problem for two instead of three weights. Other relieving fact is that the threshold can be expressed as the lowest matching or the highest non-matching value (more accurately a little higher than the exact value in the latter case). This proves to be useful when the involvement of the threshold calculation substantially complicates the problem.

4. Algorithms

We selected the most promising solutions available. This entailed a comprehensive analysis of published schema matchers. The main issue was to embrace a large scale of matching aspect. On the other hand, too much matcher on the benchmark is not advantageous, that is why we did not want too much solutions involved. One of our main contributions is the detailed examination and ranking of these schema matchers.

Our choice fell on three algorithms. The first one is a complex matcher [7] encompassing the Name, related Terms and Attributes (called NTA in the rest of the paper) similarity assessment, eventually the semantic distance delivered by the weighted sum. Peculiarity of this solution is that it is relatively simple, though effective and also involves recursive elements. The evaluation is based on scores given to string comparison based similarity (0, 0.5, 1 respectively), which is then combined in to set similarity values with methods described by equations. The second algorithm is the Similarity Flooding (called SF in the rest of the paper) [8]. This solution presumes that the more similar neighbor nodes of two concepts are the more similar the concepts themselves are. This presumption is implemented by defining an extended similarity propagation net, in which iterative flooding of similarity values along weighted edges is executed. The iterative flooding is delimited by a stop condition. Another candidate proposing a trustworthy solution is the WordNet-based Matcher (called WN in the rest of the paper)[1]. The hallmark of this solution is the capitalization of the complex conceptual synonym vocabulary called WordNet [3]. The usage of which forebodes the best possible linguistic similarity, for it not only exploits the string-based similarity but also the meaning based one. It also defines a complex structural matching, trying to take into account the relative and absolute position of the concept in the structure. It designates three contexts in which then mingled linguistic and complex path similarity is performed. Where applicable also set similarity values are expressed, counting the best forthcoming values. The three context similarities is then combined in to the semantic distance by building the weighted some of them.

5. Accuracy measurement methods

In this section we will describe methods, which we propose to the problem. In the followings we assume that all necessary prerequisites are met (result matrices elicited from the semantic distances and a reference table). When describing the solutions we will do with notations used for the NTA. This does not make a difference as the solutions are applicable unaltered to the WN. The N, T, A matrices contain the similarity values of the name, related term and attribute similarities. The R, P, I matrices contain the reference values, the matches retrieved by the algorithm and the retrieved relevant matches respectively. The w1, w2, w3 and τ values denote the weights and the threshold. The S matrix, s(m) and s(n)T vectors denote the matrix and the vectors consisting of value one. The sum of the elements of matrix X is marked with ||X||.

5.1 Accuracy maximization with reference-approximation

This first method is mention-worth for its simplicity. The result is given by formulas, easily calculable even with simple computational tools, e.g. calculator. Consequently, it is effective and can be determined instantly even for larger schemas. It returns a single value, other possible results are not rendered. This however does not diminish the solution, as in most cases a single good result is satisfactory.

It uses an indirect approach. The key idea is that the F-measure value can be maximized by seeking the weight distribution where the ensuing result matrix nearest approximates the reference table. This
approximation is understood as the aggregation of the element differences between the two matrices. In other words, we should build the average of the quadratic deviation of every element, and minimize by means of mathematical analysis. This approach has the benefit of resulting in exact formulas, which have coefficients ready to be substituted for values. It is not hard to see, that the method guarantees the extrema is not achieved through a few low deviation values, but all involved. The threshold is not included in the calculation, it can be obtained as the minimum of the matching values.

The base task can be formulated as follows. Expression (4) shows the base task of the reference-approximation and expressions (5-6) show the final formulas of the reference-approximation and accuracy maximization with measure maximization respectively.

$$\min (w_1N + w_2T + w_3A - R)$$

(4)

$$w_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{t_{ij} - a_{ij}}{2a_{ij} + n_{ij}}$$

(5)

$$w_2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \left( \frac{t_{ij} - a_{ij}}{1 - \frac{1}{t_{ij} - 2a_{ij} + n_{ij}}} \right)$$

(6)

### 5.2 Accuracy maximization with F-measure maximization

This method is an approach to manipulate and eventually maximize the measure directly with the adjustment of the weights. For the present let us discuss the F-measure maximization. This task however is not so straightforward as the former one. The task entangles because of the conversion of semantic distances to result values. Clear that this task involves a not continuous function as decision about a value being match or not is involved. This prevents the usage of the means used in the case of the reference-approximation, yet with computational means approximation of the output list possible. This results in a much more complex approach than the reference-approximation, though it returns all possible solutions. The goal is then to derive the formula of the F-measure so that it has only the weights and threshold as unknowns. The final formula is then generated with the help of the following auxiliary formulas (7-9).

$$F = \frac{I}{\|I\|}$$

(7)

$$R = \frac{\|I\|}{\|P\|}$$

(8)

$$P = \text{sgn}(\text{sgn}(w_1N + w_2T + w_3A - \tau S) + S)$$

$$I = \text{sgn}(\text{sgn}(P + R - 2S) + S)$$

(9)

The base task and the ultimate formula are given below in formula 10 and 11. Formula 10 shows the base task of the F-measure maximization and formula 11 shows the base task of the precision and recall maximization.

$$\max(F) = \max \left( \frac{2PR}{P + R} \right)$$

(10)

$$\max(P) \max(R)$$

(11)

With the above listed formulas the accuracy maximization problem unfolds in to a far less complex problem. Each maximization target measure may deliver different solution for the same scenario. For every single calibration task, the measure on which the optimization is performed should be decided based on the schema matching objective.

### 6. Experimental Results

We have conducted several calibration experiments on test schema. Hereby we would like focus on the results returned for the NTA algorithm on three test schemas. In table below the reference-approximation weights and thresholds are presented. The runtime costs of computation of these values were negligible. We can observe how schema dependent the ideal parameters are.

| Table 1 Optimal parameters given by our approximation method |
|-------------|-------------|-------------|-------------|-------------|
|             | w1         | w2         | w3         | Threshold   |
| Company     | 0,149      | 0,225      | 0,625      | 0,291       |
| University  | 0,837      | 0,109      | 0,054      | 0,975       |
| Trader      | 0,043      | 0,372      | 0,584      | 0,511       |

We have analyzed the impact of weight adjustment on the deviation from the reference table. Figure 1 and 2 shows the phenomenon for the Company and the University test schemas. The graphs show how seriously the actual choice of the parameters influences the deviation on a schema. Based on what the table and graph suggest, it is obvious how unfair a not optimized schema matcher comparison can be. Thus all scenarios need different parameter set in order to achieve their best possible result. If it is not taken into account it may a serious aftermath. See the deviations from the reference in figure 1 and the changing of F-measure values for the university test scenario.
The actual weight setting not only influences the deviation from the reference table but also the measures. Given above is graph of f-measure values in the function of weights. The required threshold value is defined as the one returned by the reference approximation approach.

7. Conclusion

We scrutinized the tested methods for three test scenarios. Such schemas were selected that as a whole they represent a wide range of real life schemas. These schemas were examined in pairs, each pair constituting the input for a given scenario. The investigated scenarios are the company, the university and auto trader schemas.

We have calibrated the algorithms for the given scenario (if possible), and evaluate their performance under these fair conditions. Other factors, such as the
effect of the weight adjustment to the diversion from the reference table and also to the measure values were examined and expressed. This showed us how the scenario influences the accuracy measurement outcome. We found that the factors may vary on a large of scale or just stay seemingly untouched by the modification of the weights based on the choice of the test schemas.

Finished with all tests we have summarized the accuracy values gained in order to assess their accuracy. The performed experiments showed the applicability of our method and formulas. Using our approach it is now already possible to evaluate and compare the accuracy of different schema matching algorithms in a correct way.

References: