Schema Matchers' Performance Improvement

Balázs Villányi, Péter Martinek

Budapest University of Technology and Economics, Hungary
baloowee@gmail.com, martinek@ett.bme.hu

Abstract: The growing number of business solutions has lead to heterogeneity. This heterogeneity has several aspects one of which is the database nonuniformity. Several applications, such as the B2B e-commerce, require the efficient communication between these systems. The problem roots at the database representational ambiguities and can be resolved by the matching of database schemas. This task referred as schema matching, has become one of the main the focus points of business interest. Several possible solutions have been aired lately, though the scope includes some not yet resolved problems. This then have several detrimental effects on the performance. The solutions so far introduced need the human supervision as their result set is not always reliable. We have recognized that their performance is strongly dependent on their parameters and other factors. That is why we endeavored to find means of pre-run scenario optimization and propose methods which better harness the current solutions. According to our recommendation the schema matchers should be calibrated with a proper parameter set before their execution. The optimal choice of threshold is crucial, we have proposed a method which should handle the problem in a sophisticated way.

Keywords: schema matching, algorithm optimization and calibration, machine learning

1 Introduction

As the heterogeneity of the enterprise data schemas has become a more and more stressing issue, the need to invent and implement such methods and algorithms which are able to cope with the problem has become inevitable. The remedy to this problem compromises such tools which should leverage the schema matching by means of identification of entities and matching of them one-by-one. This then should result a global communication between schemas.

The communication between schemas is highly sought after in case of web shops, for example. Let us look at the scenario where a portal should render the products of many different vendors. The exact palette of business partners is not available at design-time and we should assume that it varies from time to time. These conditions then require the dynamic recognition and integration of products on which no information is available at the moment. The communication is only possible after a proper schema analysis and matching. The solutions should handle many different schemas as the vendors may use highly different schemas to describe their products. They may use different structures and elements to describe the very same real world concept and they may use highly resembling concepts to describe varying entities. The cost of the after-run correction is a very important factor as it appears every time a new supplier is introduced to the system. There are also those special product searching engines which look for the very same product in several shops in order that the user gain a uniform picture of available stores. The listing then is used to seek the lowest price, the nearest store etc. This application is simply not feasible if there is no efficient implementation of schema matchers. Should the engine fail, the product is listed with illegal attributes and may mislead the user which is a violation of business rules. The scenario could be analyzed more in depths though the necessity of the area is well depicted by this scenario and a brief insight may also be gained.

The integration by means of schema matching require the matching among all the schemas involved, thus a global communication between schemas [7] is implemented. This matching or pairing is performed by identifying related entities and the transformation of them into each other under constraints given by the structural features. These entities define their interfaces, so the task also involves the wiring of them. Only with the proper wiring is the seamless communication secured. The entities can be divided into two categories based on whether they refer to other entities when defining their interface. In accordance with the de facto standard, the XSD (XML Schema Description), the simple types only have base types – such integer, string, decimal etc – in their argument, whereas the complex types have more detailed arguments. This referencing implies a tree structure of nodes which are the complex types and simple types are located in the leaves. Only after the traverse of the sub-tree can be defined the exact interface required by the complex type. That is the reason why schema matching require the efficient graph algorithms and some of them also enumerate recursive elements or visits (neighbor) nodes.

Several algorithms have been published which solve the task more or less completely. Some of them show pretty convincing accuracy values tested on artificial schemas. However, these schemas do require the human supervision and the proper check of the result set. As a direct consequence these methods are best described as semi-automatic solutions as the control crew's intervention cannot be set aside. Their exclusion from the process is highly desirable due to several reasons. On the drawback side the first item is the expense factor of the human

work, which is considerably larger than that of the machine. Regarding the run-time needed, the advantage of the machine-based evaluation is even more significant and the difference between the two evaluation methods is drastic. Considering the practical aspect of the phenomenon, the human process can consume so much time that it exceeds the limit defined by the business conditions.

On the other hand, we should not forget about the feasibility. After a certain schema extent, the human assessment is not possible, as the evaluators are not able to comprehend the schema dependencies and inner associations which consists the fundamental part of the task. As a direct result, these procedures are not applicable by certain scenarios. This then redirects the attention to those procedures where this after-run data process falls of. Secondly, the real time usage is also required in given areas, particularly in those, where the set of schemas involved varies dynamically, as exemplified above by the web shop scenario.

We have focused on the performance analysis of three algorithms [16]. They were selected carefully, so they together muster the state-of-the-art approach of schema rule-based schema matching. After the evaluation of their performance, we have realized how it varies depending on the test schemas. It has also turned out that their original performance estimation was somewhere far too optimistic. Their fair comparison can only be performed under the same test conditions. That means that they are tested on the same scenario and their parameter set is optimized to the schema in question. This procedure was later understood as a key point towards an accurate schema matching, so we defined this pre-run optimization phase as calibration.

The calibration task is necessary as it should not constitute a considerable run-time factor. Otherwise what is gained on the one side is severely punished on the other. After emphasizing the importance of runtime, we should argue with the accuracy when this former aspect is not fulfilled. Nevertheless, the calibration needs a lot of intuition, so the human execution is once again the reasonable solution regarding the accuracy, but not the runtime. Because of this latter aspect we should rule it out and define automatic solutions which can effectively parameterize the schema matchers before the run. This methods are learner-based solutions, so their decision is done by previously familiarize them with good matching.

Nonetheless we should remain fair and unbiased when assessing the matchers accuracy. It is only clear that measuring how many result values are correct is not enough. Fortunately there are several accuracy measures, some of which are widely used. Hence our decision to evaluate matchers' accuracy with the most prevailing measures, that is precision, recall and f-measure. As performed the result set quality evaluation, we realized that sometimes the correctly matched pairs and from the result set excluded pairs tend to appear in the closest vicinity of threshold. This phenomenon has two consequences. Firstly they are regarded and marked as all the other matching pairs. This is not fair, because the matcher could be unsure about a definite matching pair, which is apparently not the same case as

marking them rightly strong related. Secondly, the choice of the threshold value becomes a crucial point. It is indeed so vital that a slight deviation from the optimal value may result in a complete disaster, leading to an utterly false conclusion about the accuracy. While the choice of the threshold value still remains an important factor it should not influence the accuracy so drastically.

2 Algorithms Used

The philosophy behind the similarity flooding method [8] is that two concepts are related if their adjacent nodes are similar. To define the adequate neighbors, it constructs a graph representation of the schemas. In the resulting DAG the linguistic matching is estimated through the common prefixes and suffixes. This produces the initial values of the flooding. In order to execute iterative value exchange among the nodes, it constructs a similarity propagation graph. After setting a stop condition (either iteration number or difference vector variation threshold) the iterative flooding of similarity starts. Although the concept behind this approach is interesting in its simplicity, it cannot overcome the cardinality related problems of the direct ancestors.

The NTA (name, related terms, attributes) algorithm [7] provides a more elaborated approach. It defines three comparison aspects. The name comparison is a simple string matching. For every concept it defines a related term set which should encompass expression associated with the concept. This should further accentuate the (un)relatedness of two schema entities. The definition of related terms has a great impact if the schema granularity is low, and decision cannot be made based solely on name and structure comparison. The related term method tries to find pairs – that is to say (partially) identical expressions – and the quantity is then normalized. The complex type attributes contains its children. The evaluation differs for diverse types involved (simple or complex); as a consequence correlation assessment of two attribute sets is divided into four cases. It also uses recursive methods, for it capitalizes the NTA value of complex children if it is possible. By optimal choosing the weights for the NTA, the algorithm is hard to compete, but a good performance is often preceded by the refinement of parameters. However we are facing serious problems if we cannot prepare the method properly, and then its performance falls back to good-average.

The WordNet-based complex matching [2] is yet another candidate trying to come up with the ultimate solution. Its advanced methodology consists of linguistic, structural and optionally constraint-based matching. For the linguistic matching, it exploits the benefits of a well-known English semantic vocabulary, called WordNet. In the chain of synonyms, it searches for the shortest path and based on path length it evaluates the relatedness. The merit is obvious: no string matching technique can approximate the precision of a synonym based one. The structural

matching part of this method is evaluated through three contexts: the ancestor, the children and the leaves. These three contexts together represent the whole structural environment of the concept, and define its absolute and relative position in the schema. The similarity end result is the linear combination of the three context relatedness, where weights can be optimized for the deployment scenario. Though this method is comprehensive and meticulous, its possibly high performance does not compensate for its enormous runtime costs. While the other algorithms fulfill their tasks so fast on smaller schemas that real-time application is also feasible, this method is very time consuming even by half a dozen of entities to compare. This phenomenon roots at the myriad comparison necessary to obtain a single concept similarity. The result set quality depends on the choice of the context weights and threshold used to filter the matrix.

3 Accuracy Measurement

In order to express quality and goodness we should thoroughly investigate the result and define means of assessing accuracy. This task requires that the results are in a compatible form, which is a vital step towards the comparison of result values.

In most cases the result set is stored in matrix. Only the compatible format should be warranted, which entails some consideration as in some cases there are different pragmatics to describe entities. For example one should decide whether entity instances are distinguished or only their type.

Another issue is that the algorithms return so called semantic distances. That is they decide to which extent two entities are related and not whether they are related. The semantic distances are similarity characteristic values ranging from 0 to 1 and should be converted the matching and non-matching pairs. So the problem intrigues as it encompasses the need to determine a limit called threshold that cuts the result set into two halves. It is now obvious that the adequate calculation of this value plays a key role. After injecting the threshold into the similarity matrix, the result matrix consists of only 0 and 1. This fact makes the result set easily comparable with reference solution and is pretty descriptive for human inspectors.

It is all right that the result set is easily comparable with the reference solution, but what is exactly the reference. It turns that there is no reference solution available, or at least it is not obvious what it should be. To remain unbiased we decided to make a survey involving some twenty human 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.

With the method described above with acquired reference values with which the result set were compared. In order to assess accuracy we needed accuracy measures. We used the most prevailing measures: the Precision, the Recall and the F-measure. They are best known for their usage in information retrieval, and they are used widely. By making a brief search we found that these measures are utilized to describe the goodness in the majority of cases. The measures are calculated with the formulas as follows: (1) Precision Formula, (2) Recall Formula and (3) F-measure formula.

$$\mathrm{Precision} = \frac{|\{\mathrm{proposed\_matches}\} \cap \{\mathrm{relevant\_matches}\}|}{|\{\mathrm{proposed\_matches}\}|}$$

Formula 1
The formula of the precison

$$\mathrm{Recall} = \frac{|\{\mathrm{proposed\_matches}\} \cap \{\mathrm{relevant\_matches}\}|}{|\{\mathrm{relevant\_matches}\}|}$$

Formula 2
The formula of the recall

$$\mathrm{F\_measure} = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Formula 3
The formula of the f-measure

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.

4 Calibration with Reference Approximation

We denote the process of optimizing the algorithm for a given scenario as calibration. As earlier mentioned this has a beneficial impact on the performance. The goal is to find automated solutions which carry out this pre-run task run-time efficiently.

The to be set up parameters encompass the weights of the partial similarity matrices and the threshold as default. As turned out, several possible value

distributions are possible, which are equally good. Nonetheless, one correct value set satisfies our needs.

In order that the problem to be manageable we should find correspondence among the algorithm attributes. One of the most important and prevailing characteristic is that the weights included 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). It implies that we should not care about the threshold when it is not involved in the calculation process as a variable. This is the case by the reference maximization.

This method in question is an indirect approach. The behind lying 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.

This process, however, may need a pre-run filtering. The behind lying reason is that we use the same parameters for each pairs, obviously. This fact may impede to reach beyond an upper accuracy limit. For example when every linear combination has a positive tendency, that is they converge to the corresponding value of the reference matrix, the task can be easily solved. Should emerge one or two “reluctant” value which have an opposite tendency, we should relinquish the possibility of full match. No matter what kind of parameters we use, these opposite tendencies foil the accuracy. By filtering out these elements we should achieve our original goal with the addition that the accuracy value 1 is no more possible. We can easily filter out these opposite tendencies with the proximity measure, introduced below. We should seek the negative values, which set then defines exactly those pairs which are wrongly assessed by the algorithm.

In the rest of the paper we use the following notations. Let N denote the number of weights and M the number of entity comparisons. We define w as vector containing the weights. Our objective is to define the elements of this vector. The R vector contains the values relative to which the accuracy is measured. In accordance with the earlier it only contains 0 and 1. The matrix is NxM matrix which contains the partial similarity values returned by the algorithm. In other

words we were aimed at finding the weigh distribution by which the aggregation of the rows of the matrix best approximates the R vector.

The proximity measure and base task is defined below:

$$P = r - wA$$

Formula 1
The proximity measure

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

Formula 2
The base task of the reference approximation

With expression below we can optimize for a given weight. If taking into account all of them, that is optimizing for all of them, we gain a global optimal solution.

$$2\sum_{i=1}^{M}\left(w_x a_{xi} - r_i a_{xi} + \sum_{k=1}^{N} a_{xi} w_k a_{ki}\right) + \sum_{i=1}^{M}\sum_{j=1, j\neq x}^{N}\left(w_j^2 a_{ji}^2 - 2 r_i w_j a_{ji}\right) + 2\sum_{i=1}^{M}\sum_{k,j=1}^{N}\sum_{k\neq j, j\neq x} w_j a_{ji} w_k a_{ki} + \sum_{i=1, j\neq x}^{M} r_i^2 = 0$$

Formula 3
Optimization task to a given w_x weight

5 Multiple Thresholding, Threshold Blurring

According to our experiments the choice threshold value by dense similarity has such an impact on the outcome the choice of the weights added together. We have also observed that by the majority of the result sets this is indeed the case – the algorithms do return values which are very near to each other. That is the reason why our attention turned to a possible procedure which could blunt the weight of this decision.

The main issue is that the result does not divide into two clear subsections, such as the matching and non-matching set. A method which were more "lenient" with the values locate near to the threshold would be highly desirable.

The first step toward this goal is to define multiple thresholds, as it is not obvious for an accuracy assessor what is near to the threshold. As a set-off we defined two thresholds. The first is the absolute match line, which should denote the threshold for those pairs which are pairs beyond any possible doubt, that is to say they have considerably larger similarity values than the threshold. The second threshold called absolute non-match line works the same way, only the "absolutely non-matching" values are denoted. Between the values are those matches which are the source of trouble.

In order to decide on their classification we define the matching measure. This value is meant to express to which extent the matching is relevant. Under assessing conditions we proceed the following way. We inspect matched pairs from result set and decide whether they are also in the reference set. If so then we increment the number of correct matching. The first modification is the amount by which the correct matching value is incremented. When the pair is in the direct vicinity of the threshold matching or not found value is incremented, but by varying amounts. The following indicator function gives further insight:

$$I(v_i) = \begin{cases} 1, & if\, v_i > l_1 \\ 0, & if\, v_i < l_2 \\ otherwise, g(v_i) \end{cases}$$

Formula 4
The threshold blurring function

,where l_1 is the absolute match line, l_2 is the absolute non-match line and $g(v_i)$ is a function. We recommend that the $g(v_i)$ should be a linear function or a logarithmic function. These functions share the characteristic that by approaching the argument to the similarity value that is considered to be that of a matching pair the function returns higher and higher values. In this way the formula fulfills the blurring effect on the threshold which was originally set as a primary goal.

Our altered quality assessment functions are the followings:

$$Pr_{alt} = \frac{\sum_i I(v_i)}{|\{proposed\_matches\}|}$$

$$Recall_{alt} = \frac{\sum_i I(v_i)}{|\{relevant\_matches\}|}$$

Formula 5
The modified precision and recall. Used with the threshold blurring together

These altered accuracy measures can provide a better feedback on the actual accuracy, which not so dependent from the actual choice of the threshold, however it still remains a vital factor.

6 Experiment Results

We implemented the analyzed algorithms and have executed them on several test schemas in order to assess their performance. Performance is understood as run-time needed and accuracy achieved.

After executing initial performance measures the calibration units were linked in. We have evaluated the runtime overhead and the gain realized by the calibration. Regarding the time factor the result is quite convincing, as it was not increased substantially. It is crucial question, especially be runtime efficient solutions such as the NTA. On one hand it was expected by the simplicity of the formula which can be computed very easily by smaller calibration tasks such as we faced, on other hand the recommended parameter set adjusted the original set to a very surprising extent.

Table 1
Optimal weigh distribution defined by the reference approximation

	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

These values were attained after performing the required filtering, thus achieved very high precision values with this set, but in some cases lower recall values as some of the pairs – which show opposite tendencies – had to be ruled out. According to our observation the f-measure values – which we defined in our case as the ultimate accuracy measure – still remain high, so the filtering process does not have considerably adverse impact on the outcome. Other conclusion is that the filtering phase is by most of the time unnecessary as the entity pairs show the right tendency, only their weight contribution should defined by a simple formula calculation.

Concerning the multiple thresholding we have recalculated our accuracy values. It has result a slight change of the measures in case of the WordNet-based matcher and the similarity flooding. Interestingly the values remained nearly untouched in case of the NTA. When trying to identify reasons, it has turned out the NTA best divides the matching set from the non-matching set, hence the phenomenon. In case of the other two benefits is clear.

Table 2
F-measure values with single and multiple thresholding

	Single threshold				Multiple thresholds		
	NTA	SF	WN		NTA	SF	WN
Company	1	0,57	0,8		0,96	0,71	0,85
University	0,66	0,66	0,76		0,64	0,79	0,84
Trader	0,6	0,22	0,66		0,66	0,52	0,78

Their accuracy value is improved. We found this improvement justified as we also faced the problem that lead the introduction of multiple thresholding, namely we tried our best at defining the threshold, but the some values fell slightly under or over the threshold. This then resulted in a value distortion.

Conclusions

The reference approximation and the multiple treshold techniques are ment to improve the accuracy of schema matchers and let the performance evaluator gain a more realistic picture of the accuracy.

Our experiment shows that the realization of this goal is indeed possible. We evaluated our approches in test schemas. In the future we will test these approches on larger, industrial sized schemas to attain accuracy improvement also in real life sceanrios. Also the question of runtime overhead by larger schemas or more complex schema macthers which might have as 5 or 6 weights should be analized. We are convinced that these methods are easily scalable as they are very simple and offer straightforward solutions which are applicable also by larger schemas.

References

[1] Aida Boukottaya, Christine Vanoirbeek, Schema Matching for Transforming Structured Documents, Proceedings of the 2005 ACM symposium on Document engineering, 2005, pp. 101-110

[2] D. Buttler, A Short Survey of Document Structure Similarity Algorithms, Proceedings of the 5th international conference on internet computing, 2004

[3] Cognitive Science Laboratory, WordNet - a lexical database for the English language, at http://wordnet.princeton.edu/

[4] Hong-Hai Do, Erhard Rahm, Matching large schemas: Approaches and evaluation, Information Systems, Vol. 32, Issue 6, 2007, pp. 857-885

[5] Buhwan Jeong, Daewon Lee, Hyunbo Cho, Jaewook Lee, A novel method for measuring semantic similarity for XML schema matching, Expert Systems with Applications, Vol. 34, Issue 3, 2008, pp. 1651-1658

[6] J. Madhavan, P. A. Bernstein, E. Rahm, Generic Schema Matching with Cupid, Proceedings of the 27th International Conference on Very Large Data Bases, 2001, pp. 49-58

[7] Martinek P., Szikora B, Detecting semantically related concepts in a SOA integration scenario, Periodica Polytechnica, 2008 – in press

[8] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching, Proceedings of the 18th International Conference on Data Engineering, 2002, pp. 117-128

[9] Richi Nayak, Wina Iryadi, XML schema clustering with semantic and hierarchical similarity measures, Knowledge-Based Systems, Vol. 20, Issue 4, 2007, pp. 336-349

[10] Khalid Saleem, Zohra Bellahsene, Ela Hunt, Performance Oriented Schema Matching, Lecture Notes in Computer Science, Vol. 4653, 2007, pp.844-853

[11] H. Quyet Thang, V. Sy Nam, XML Schema Automatic Matching Solution, International Journal of Computer Systems Science and Engineering, Vol 4., Num. 1, pp. 68-74

[12] Bergamaschi, S., S. Castano, and M. Vincini: Semantic Integration of Semistructured and Structured Data Sources, SIGMOD Record, Vol. 28, Issue 1, 1999, Pp. 54-59

[13] Palopoli, L. G. Terracina, and D. Ursino: The System DIKE: Towards the Semi-Automatic Synthesis of Cooperative Information Systems and Data Warehouses, Lecture Notes in Computer Science, Vol. 2282, 2002, Pp. 228-276

[14] A. Salguero, et. al, Ontology based framework for data integration, WSEAS Transactions on Information Science and Applications, Volume 5, Issue 6, 2008, Pp. 953-962

[15] Peter Martinek, Balazs Villanyi, Bela Szikora, Calibration and Comparison of Schema Matchers, WSEAS Transactions on Mathematics, Vol. 8, Issue 9, 2009, pp.489-499.

[16] B. Villányi, P. Martinek - Analysing Schema Matching Solutions, microCAD Conference, 2009, pp. 127-132

[17] P.Martinek, B. Villanyi, B. Szikora - Optimization and Comparison of Schema Matching Solutions, 11th WSEAS International Conference on Mathemathical Methods, Computational Techniques and Intelligent Systems, 2009, pp. 258-263