

Improved Accuracy Evaluation of Schema Matching Algorithms

Balázs Villányi, Péter Martinek

Department of Electronics Technology, Budapest University of Technology and Economics, Egrý József utca 18, H-1111 Budapest, Hungary
e-mail: villanyi@ett.bme.hu, martinek@ett.bme.hu

Abstract: Application integration is one of the most relevant topics in enterprise computing today. To support Enterprise Application Integration, various algorithms, methods and complex systems are applied. Automated schema matchers support application integration by identifying the semantically related entities of the input schemas. In this article, we present the notion of the cutting threshold problem in schema matching and propose a solution. Our approach incorporates the definition of a threshold function, which is conceptually similar to a fuzzy membership function. We have also redefined the accuracy measures most commonly used for accuracy evaluation of schema matchers. Employing the threshold function, we managed to obtain a 9.85% average accuracy improvement. The introduction of the threshold function also enables the redefinition of the schema matching accuracy maximization problem.

Keywords: Enterprise Application Integration; Schema Matching; Algorithm Optimization; Fuzzy Logic; Threshold Function; Accuracy Measure Improvement

1 Introduction

Currently, organizations have heterogeneous software components in their system landscapes covering functions like logistics, financial, production, etc. Enterprises aim at having an integrated system landscape, in order to reduce process times and improve information availability, while reducing redundancy. On the other hand, heterogeneous systems tend to publish different interfaces. These diverse interfaces have to be matched before the communication can be launched. The matching of interfaces involves the identification of semantically related schema entities of interfaces. This related entity identification in heterogeneous system schemas is referred to as schema matching [1] and is used to be carried out manually involving a handful of schema matching experts. Needless to say, that this type of schema matching execution is cumbersome, time-consuming and is error-prone. Therefore, efficient automated schema matching solutions are required.

There are many current, published automated schema matching approaches. These approaches can be categorized as follows. The linguistic matcher evaluates the semantic relatedness of textual elements found in schemas. We treat two types of linguistic matchers, based on whether they use external dictionaries, thesauri or ontologies: First, syntactic matchers do not use any kind of external taxonomy, but assess only string relatedness (e.g. common prefix/suffix-based, substring based etc.). Second, vocabulary approaches are linked with various kinds of external ontology, dictionary or thesaurus, like the WordNet [2]. Beyond linguistic matching, another main type of schema matchers is the structural matcher. Structural matchers detect the possible resemblance in the structure of the schema. Most typically, they analyze the ascendant-descendant relations of semantically related nodes in the schemas being compared.

These types of schema matchers are not used on their own and are combined using weights. Practically, every schema matcher provide a similarity value (also called semantic distance), which expresses the relatedness of the entities being compared and subsequently these values are summed using predefined weights. A schema matcher incorporating other matchers is called hybrid schema matcher and is the prevailing type among viable schema matcher proposals.

In order to match schemas, the semantic distance serves only as a basis for classification. Another predefined value, called threshold, is required to decide whether the entities being compared will constitute a match or no (i.e. whether they represent the same real world entity/business object or not): entity pairs obtaining semantic values higher than a specific threshold are treated as “*match*” while the rest is classified as “*non-match*”. Consequently, the optimal choice of the threshold is crucial for the schema matching, but it differs from the weights used to combine individual semantic distances.

The performance of schema matching solutions is expressed through accuracy. It is measured using accuracy measures such as precision, recall and F-measure [1]. Precision expresses the rate of real matches among all entity pairs classified as match, recall is the rate of retrieved true matches among all reference matches, while f-measure is the harmonic mean of the former two.

Our first problem (*Prob. 1*) formulation is as follows. The schema matcher weights and threshold should be defined so that all entity pairs be classified correctly. This task is particularly difficult if the two result set (labeled “*match*” and “*non-match*”) are near or overlapping. In case of the overlapping the threshold value application may cause accuracy value degradation as the classic accuracy measures (precision, recall and F-measure) assess accuracy based on crisp set membership. Entity pairs near to the threshold are excessively punished/rewarded for being on the improper/proper side of the threshold, however, their classification is not 100% certain according to the relatedness evaluation of the schema matcher. This uncertainty then may lead to a false accuracy feedback of the real schema matcher performance. See the problem in details in Section 3. We refer to this problem as the cutting threshold problem.

There is also another problem (*Prob. II*), which has encouraged us to conduct threshold related experiments: the non-continuous objective function of the accuracy measure maximization problem. It is treated in full length in [3], where we propose a methodology to optimize the parameters of schema matchers. A subgroup of our parameter optimization techniques comprises the accuracy measure maximization, whereby the objective function is a formula derived from a given accuracy measure so that it contains exclusively the schema matcher parameters (the weights and the threshold). The main drawback of the presence of a constant threshold value is that it makes the objective function non-continuous.

We provide a solution for both of these problems in Chapter 5, by proposing the introduction of schema matching threshold function. In order to be able to validate our idea of how the introduction of a threshold function resolves our accuracy measure maximization related problem (*Prob. II*), the accuracy measures are also described in Section 4.

This article is divided into six sections as follows. This first chapter provides introductory insight into the problem formulation. The second chapter presents some of the related works. The third chapter proposes a solution for the cutting threshold problem. The fourth chapter introduces the reader to our revised accuracy measure maximization technique. The fifth chapter is about the results obtained with the proposed accuracy measures. Lastly, we provide some closing thoughts on the subject.

2 Related Works

In this chapter, we present some of our related work which have inspired us. Many schema matching has been published in the last decades like [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. The matcher in [6] is a complex schema matcher consisting of three components which are the name similarity, the related term similarity and the attribute similarity. The name similarity is a rough, scoring based similarity assessor. The solution [4] provides context based matching, which is a very efficient technique backed by the WordNet lexical database [2]. Nevertheless, the processes of context based matching increase complexity and there are several parameters to be set before the matcher could perform optimally. The similarity flooding [10] is a key technique which involves the redistribution of concept similarities in the similarity propagation net.

Matching is also involved in [14] where authors provide a solution to reconcile the semantics of structured and semi-structured data. This solution is also geared towards flexibility and knowledge accumulation. Another solution for the matching of ontologies can be found in [15] which has linguistic and structural matching components like many other schema matchers. This solution also makes

use of the above listed lexical database [2]. Another ontology related application can be found in [16] where a special ontology system is proposed, which helps storing fuzzy information. The management of heterogeneous information is carried out using this ontology. A special ontology is introduced in [17] which is closely related to semantic networks and UML diagrams. Schema and ontology matching are closely related [18], consequently our observations and results of this current paper may apply to ontology mapping as well.

An excellent semantic integration technique can be found in [19]. Schema matching can be applied in the same context. In [19] authors propose a solution which is able to automatically translate XML schema representations of the business objects into OWL based ontology. Schema matching can be used to identify semantically related entities in the XML schemas, thus it can be applied even in those situations where highly diverse, heterogeneous schemas are to be integrated.

The solution in [20], is a generic schema matching tool, called COMA+. It is not a schema matcher by itself, but is a complex platform designed for integrating schema matching components. The individual matchers are arranged into a library. The solution provides scalability by fragmenting schemas into subsets, which is an application of the “divide and conquer” principle to the field of schema matching. This allows for the platform to be flexible. On the other hand, it does not include parameter optimization which would further improve its efficiency.

The above solutions bear in common that they have fixed threshold values. Based on our observation, this attribute may be limiting in several cases. It is also true, that the employed fixed threshold values do not take into account the cutting threshold problem introduced in the next section. Consequently, our method presented in this paper, can be used to enhance the performance evaluation of all of the above listed schema matching solution.

3 The Cutting Threshold Problem

As introduced in the first section (*Prob. 1*), the cutting threshold problem lies in the presence of a mixed labeled - i.e. “*match*” or “*non-match*” - entity pair cluster around the threshold value. The employment of the classic sets in the schema matcher accuracy evaluation may lead to the serious distortion of the schema matching accuracy because of the followings. In certain schema matching scenarios the result sets will be overlapping even when using optimal schema matcher parameters. In other words there will always be entity pairs on the wrong side of the threshold based on the semantic value given by the schema matcher — i.e. “*match*” labeled entity pairs having semantic values lower than the threshold or “*non-match*” labeled entity pairs having semantic values higher semantic values than the threshold. See Fig. 1.

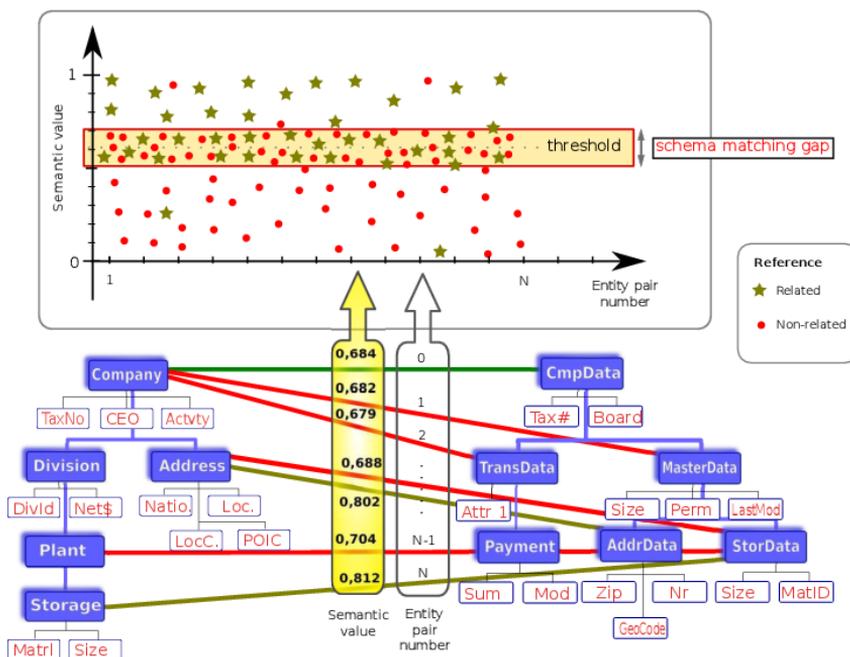


Figure 1
Depiction of the cutting threshold problem

Fig. 1 depicts the overlapping result sets by optimal schema matcher weights. On the graph, the x-axis represents the entity pairs being matched (by simply ordering them in a chosen sequence), while y-axis is the semantic value (given by the schema matcher for a given entity pair). There is also another information on this graph: the ground truth or reference match. The green star represents those entity pairs, which should be labeled “match” by reference, i.e. they represent the same real world entity and are thus considered to be related. Conversely, the red dot represent those entity pairs, which should be labeled “non-match” by reference, i.e. the red dots are those entity pairs which are considered non-related by reference.

The problem is also seen in Fig. 1. Even if the schema matcher are optimal, we cannot specify an optimal threshold because the (reference) result sets overlap. Part of this problem is the distortion effect of the threshold: should we specify a suboptimal threshold in the intersection of the result sets – we call it schema matching gap, see Fig. 1 –, the accuracy measure evaluation will not be precise since at least some of the entity pairs in the schema matching gap will be wrongly classified. This is especially true for the F-measure evaluation. As already stated, we refer to this problem as cutting threshold problem.

Stepping one step further, the classification of the entity pairs in the schema matching gap is uncertain, which is not reflected in the classic accuracy measures leading to a distortion in the accuracy assessment. Conversely, the concept of a threshold value is still useful and thus, applicable, without any modifications required, for the cases of entity pairs falling outside of the schema matching gap. This is because their schema matcher classification is firm, no matter how we specify the threshold in the schema matching gap.

There is agreement among the above mentioned observations and aspects: the concept of the threshold value can be retained if the classification is clear for semantic values outside the schema matching gap, while the cutting threshold problem can be resolved if we introduce a continuous result set transition for entity pairs in the schema matching gap.

These considerations have led us to the definition of threshold function. At the time of the threshold function formula the recurrent problem was that the result curve did not fit the needs stated above or the curve formula became so complex that it seriously hindered the practical applicability of the approach. In the end, we have found that the most appropriate threshold function is the sigmoid function as it is continuous, has a simple formula and there is a continuous transition between semantic values of matching {1} and non-matching {0} entity pairs in the schema matching gap. It is also important that this function can be easily parameterized so that it fit a large scale of schema matching scenarios. Its formula has been reformulated in order that it fit our needs, resulting in Eq. 1. We have also involved the concept of schema matching gap l_{sg} . For further references, see Fig. 2. ($\tau = 64$ and $l_{sg} = 0.2$). In the following Eq. 1, τ denotes the threshold and g denotes the granularity:

$$f(x) = \frac{1}{1 + e^{g(x-\tau)}} \quad (1)$$

where g can be defined as a function of the schema matching gap l_{sg} , see Eq. 2:

$$g = \frac{128}{l_{sg}} \quad (2)$$

For the practical application, the curve should be parameterized considering the followings. The transition interval from match certainty zero to one should include all semantic value of the overlapping result set intersection (by definition of the schema matching gap) and the transition interval should be centered on the original threshold value if any — see Fig. 1. Thus, entity pairs having semantic distance equal to the threshold will have uncertainty value 0.5, fairly reflecting the fact that they may as well be “*match*” or “*non-match*” according to the semantic value given by the schema matcher. Furthermore, entity pairs falling outside the schema matching gap will have semantic values {0;1} indicating that their classification is sure.

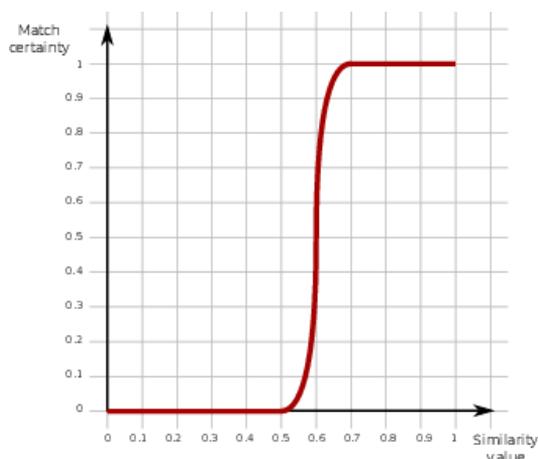


Figure 2

The sigmoid function substituting the threshold

Also the threshold function may serve for the accuracy evaluation for related entities (the “*match*” set), since on a $[0, 1]$ scale it gives null accuracy for those related entity pairs which have semantic values lower than the lower bound of the schema matching gap, and one accuracy for those related entity pairs which have semantic values higher than the higher bound of the schema matching gap. The rest is given accuracy in-between. By the same token, the $f(-x)$ can be used as the accuracy evaluation functions for the set “non-match”. These membership functions ($f(x)$ and $f(-x)$) have the common inflexion point located at the threshold value, resulting in a characteristic “X” shape.

Later we have also realized [21] that our concept of threshold function is very similar to that of the membership functions in fuzzy logic, thus we may as well use the aforementioned accuracy evaluation functions as membership functions for the result sets.

4 Redefinition of the Accuracy Measures

In this section, we will present our ideas on how to use the schema matching threshold function to adjust the formula of existing accuracy measures [1] to solve the accuracy measure maximization problem.

We will use the following denotations in Eq. 3 and Eq. 4. Let M denote the set of matches. M_p (proposed) will denote the match set returned by the schema matcher, while M_r (reference) is the reference match set (ground truth). In concordance with the earlier, g and τ denote the granularity and the threshold respectively. c_w denotes the number of weights used by the algorithm, which is the same number

as the number of schema matcher components [3]. w_{iu} shall denote u^{th} weight in case of the i^{th} matching pair, c_{iu} is the value returned by the corresponding component.

$$\text{Precision} = \frac{\sum_{i \in M_p \cap M_r} \frac{1}{1 + e^{g(\sum_{u=0}^{c_w} w_{iu} c_{iu} - \tau)}}}{\sum_{j \in M_p} \frac{1}{1 + e^{g(\sum_{v=0}^{c_w} w_{jv} c_{jv} - \tau)}}} \quad (3)$$

$$\text{Recall} = \frac{\sum_{i \in M_p \cap M_r} \frac{1}{1 + e^{g(\sum_{u=0}^{c_w} w_{iu} c_{iu} - \tau)}}}{\sum_{j \in M_r} \frac{1}{1 + e^{g(\sum_{v=0}^{c_w} w_{jv} c_{jv} - \tau)}}} \quad (4)$$

The F-measure is simply the harmonic mean of the precision and the recall. The trade-off of the threshold function is that the correctly classified entity pairs residing in the vicinity of threshold will not be scored as full match, either. Since we have employed the threshold function also in the denominator of the accuracy measures, the maximal accuracy value can be attained in case of perfect matching.

5 Accuracy Measure Maximization Using the Threshold Function

As already mentioned in the introduction (*Prob. II*), the accuracy measure maximization is our schema matcher optimization technique to define the optimal parameter set of schema matcher by means of the derivation of accuracy measure formulas [22]. Unfortunately, the objective function of the accuracy measure maximization problem is non-continuous. We will now present the revised accuracy measure maximization technique using the F-measure as accuracy measure.

First of all, we introduced weights to the F-measure. The reason is that we found that the exact harmonic mean of the precision and recall may not be appropriate for all schema matching scenarios. Consequently, we have developed the generalization of the F-measure formula based on the concept of weighted harmonic mean: the weighted F-measure – see Eq. 5. Let π denote the weight of the precision, while ρ should represent the weight of the recall. P is precision value, while R is recall value and F is the F-measure. The modified formula of the F-measure is the following:

$$F_{\pi, \rho} = (\pi + \rho) \frac{P \cdot R}{\rho \cdot P + \pi \cdot R} \quad (5)$$

Using the threshold function and the proposed weighted F-measure, the reformulation of our original accuracy measure maximization problem setting [22] became possible. Eq. 6 describes the objective function for the weighted f-measure using the already presented notation.

$$\max_{0 < w_{ij} < 1} \frac{\sum_{i \in M_p \cup M_r} \frac{\pi + \rho}{1 + e^{g(\sum_{u=0}^{c_w} w_{iu} c_{iu} - \tau)}}}{\sum_{j \in M_p} \frac{\rho}{1 + e^{g(\sum_{v=0}^{c_w} w_{iv} c_{iv} - \tau)}} + \sum_{k \in M_r} \frac{\pi}{1 + e^{g(\sum_{z=0}^{c_w} w_{kz} c_{kz} - \tau)}}} \quad (6)$$

If we use the classic f-measure, the formula will be less complex (see Eq. 7) as the denominator will not contain weights:

$$\max_{0 < w_{ij} < 1} \frac{\sum_{i \in M_p \cap M_r} \frac{1}{1 + e^{g(\sum_{u=0}^{c_w} w_{iu} c_{iu} - \tau)}}}{\sum_{j \in M_p \cup M_r} \frac{1}{1 + e^{g(\sum_{v=0}^{c_w} w_{jv} c_{jv} - \tau)}}} \quad (7)$$

Furthermore, the schema matching reference values should also be specified, with the help of Eq. 8:

$$R_j = \frac{1}{1 + e^{g(\sum_{v=0}^{c_w} w_{jv} c_{jv} - \tau)}} \quad (8)$$

The expression above declares that the weighted sum of the component relatedness values for a given entity pair (j^{th} entity pair in the list) should be equal to the reference.

The benefit of this accuracy measure maximization problem formulation is that all contained functions are continuous, thus it can be easily solved using optimization tools. We shall note that this reformulation of the original accuracy measure maximization problem has retained its original characteristics as the sigmoid threshold function returns closely the same values outside of the transition as the original threshold value based accuracy evaluation.

6 Experimental Results

The goal of our experiments was to validate the benefits of the threshold function compared to the classic threshold value based approach. We have also performed several experiments to reveal the possible favorable impact of the employment of the revised accuracy measure maximization.

Our experiments had the objective of directly comparing the results of the classic accuracy measure approach with the revised method. We have tested our approach using three schema matchers: NTA [6], Similarity Flooding [10] and a WordNet based complex matcher [4]. We found these three approaches adequate to assess the possible benefits of using our proposed threshold function and refined accuracy measures. We have executed these techniques on three different test scenarios. These test scenarios are defined to contain the key challenges found in schema matching: different naming conventions, structures, etc.

Our comparison approach is as follows. We have calibrated these three techniques to scenarios, then evaluated their accuracy using the classic accuracy measures. As next, we have defined the ideal threshold function for each and every schema matching scenario and schema matcher. Lastly, we have evaluated the accuracy of every schema matcher in every scenario by means of the revised accuracy measures employing the threshold function. We have calculated the average precision, recall and F-measure in the test scenarios. The results can be seen on Fig. 3 (precision), Fig. 4 (recall) and Fig. 5 (F-measure).

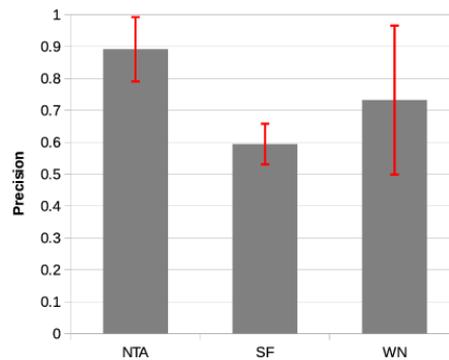


Figure 3

Average precision values with standard deviation of the analyzed schema matchers in the test scenarios

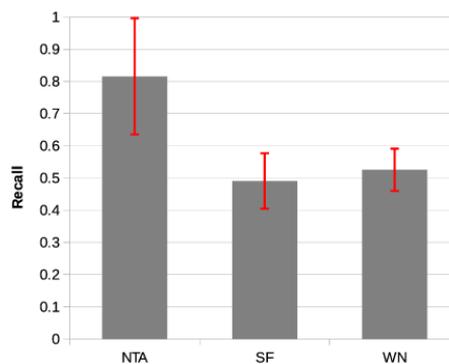


Figure 4

Average recall values with standard deviation of the analyzed schema matchers in the test scenarios

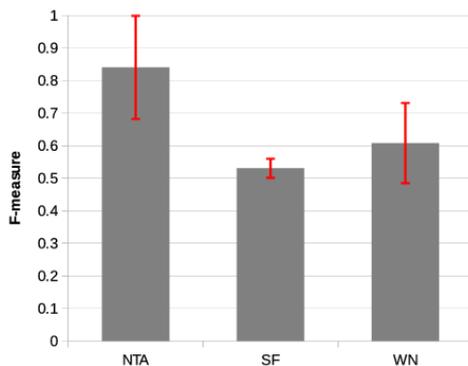


Figure 5

Average F-measure values with standard deviation of the analyzed schema matchers in the test scenarios

Fig. 3 (precision), Fig. 4 (recall) and Fig. 5 (F-measure) shows us that the standard deviation for the accuracy values are limited. We could observe less deviation in case of the recall, but precision error was also in the expected zone. The aggregated accuracy measure – the f-measure – has a less than expected deviation. In fact, the standard deviation became less after the application of the threshold function as it can be seen on Fig. 6.

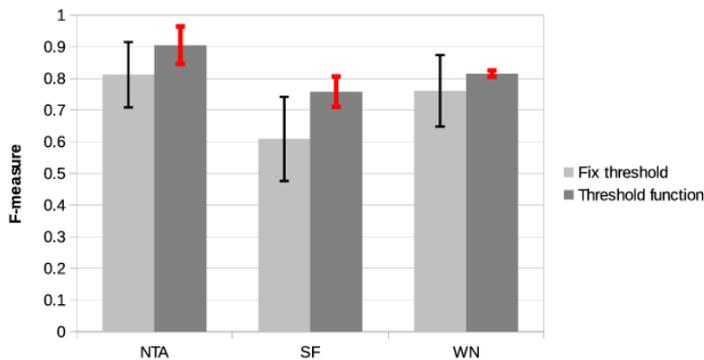


Figure 6

Average F-measure value comparison of the threshold value and the threshold function application

As it can be seen on Fig. 6, we saw an increase in the f-measure compared to the classic approach in almost every test scenario after applying adequate threshold function. We argued that this increase occurred due to the more justified evaluation of the schema matcher performance. This outcome has further strengthened our conviction that the cutting threshold problem does not just occur under very specific circumstances, but it should be treated as a ubiquitous phenomenon.

The deviation of the obtained f-measure values from the means can also be seen on Fig. 6. As we can observe it, the deviation is not substantial even in the case of the fix threshold value application, but we managed to further reduce it by means of threshold function application. In our interpretation the deviation from the average F-measure shows how much the schema matcher performance varies in different scenarios. The lesser this values is the more balanced the schema matcher performance is. This characteristic is an especially important notion in a highly variable application context, i.e. when the schema matchers are applied to significantly different test scenarios. For further details, consult Table 1 and Table 2. All in all, the application of threshold function entailed a 9.85% accuracy improvement.

Table 1

Average accuracy using fix threshold value and the threshold function (expressed with f-measure)

	NTA	SF	WN
Threshold function	0.905	0.758	0.815
Threshold value	0.812	0.609	0.761

Table 2

Standard deviation of the accuracy using fix threshold value and the threshold function (expressed with f-measure)

	NTA	SF	WN
Threshold function	0.059	0.048	0.08
Threshold value	0.103	0.133	0.113

Conclusion

In some schema matching scenarios the result sets overlap. This phenomenon hinders the specification of an optimal threshold and also significantly distorts the accuracy evaluation of schema matchers leading to false conclusions on the real capabilities of a tested schema matcher. This problem — the cutting threshold problem — was presented in detail herein.

For the solution of cutting threshold problems, we propose the application of a threshold function and we also propose a formula which we found to be optimal. Furthermore, the revised schema matching accuracy evaluation approach also reflects the classification uncertainty — as opposed to the classic schema matching accuracy evaluation approach. We also presented revised formulas of the classic accuracy measures (precision, recall, F-measure) incorporating the proposed threshold function.

A related problem is the accuracy measure maximization, which has a non-continuous objective function. The introduction of a threshold function also resolves this problem, by presenting an approximate objective function for the original problem. Thus a revised accuracy measure maximization technique was also presented here.

Using the threshold function, we managed to attain an average accuracy improvement of 9.85%.

References

- [1] Hong-Hai D, Rahm E. Matching Large Schemas: Approaches and Evaluation. *Information Systems* 2007; 32(6): pp. 857-885
- [2] WordNet, <http://wordnet.princeton.edu/>, 2006
- [3] Villanyi B, Martinek P, Szikora B. A Framework for Schema Matcher Composition. *WSEAS Transactions on Computers* 2010; 9(10): pp. 1235-1244
- [4] Boukottaya A, Vanoirbeek C. Schema Matching for Transforming Structured Documents. In: *Proceedings of the 2005 ACM symposium on Document engineering*, Bristol, UK; 2005; pp. 101-110
- [5] Hung NQV, Tam NT, Miklós Z, Aberer K. On Leveraging Crowdsourcing Techniques for Schema Matching Networks. In: *Database Systems for Advanced Applications*, Heidelberg, DE; 2012; pp. 139-154
- [6] Martinek P, Szikora B. Detecting Semantically-related Concepts in a SOA Integration Scenario. *Periodica Polytechnica* 2008; 52(1-2): pp. 117-125
- [7] Wan J, Xiaokun D. A Genetic Schema Matching Algorithm based on Partial Functional Dependencies. *Journal of Computational Information Systems* 2013; 9(12): pp. 4803-4811
- [8] Vejber VV, Kudinov, AV, Markov NG. Algoritm sopostavlenija shem dannyh informacionnyh sistem neftegazodobyvajushhego predpriyatija. *Izvestija TPU* 2011; (5): pp. 75-79
- [9] Zhao C, Shen D, Kou Y, Nie T, Yu G. A Multilayer Method of Schema Matching based on Semantic and Functional Dependencies. In: *Proceedings of Web Information Systems and Applications Conference (WISA)*, Haikou, China; 2012; pp. 223-228
- [10] Melnik S, Garcia-Molina H, Rahm E. Similarity Flooding: a Versatile Graph Matching Algorithm and its Application to Schema Matching. In: *Proceedings of the 18th International Conference on Data Engineering*, San Jose, CA, USA; 2002; pp. 117-128
- [11] Puttonen J, Lobov A, Lastra JLM. Semantics-based Composition of Factory Automation Processes Encapsulated by Web Services, *IEEE Transactions on Industrial Informatics* 2013; 9(4): pp. 2349-2359
- [12] Furdík K, Tomášek M, Hreňo J. A WSMO-based Framework Enabling Semantic Interoperability in e-Government Solutions, *Acta Polytechnica Hungarica* 2011; 8(2): pp. 61-79
- [13] Jordan V, Cicortas A. Ontologies used for Competence Management, *Acta Polytechnica Hungarica* 2008, 5(2): pp. 133-144

- [14] Jameson M, Fei XX, Chun DS. Intelligence Benevolent Tools: A Global System Automating Integration of Structured and Semistructured Sources in One Process. *International journal of intelligent systems* 2004; 19(6): pp. 543-563
- [15] Acampora G, Loia V, Salerno S, Vitiello A. A Hybrid Evolutionary Approach for Solving the Ontology Alignment Problem. *International Journal of Intelligent Systems* 2012; 27(3): pp. 189-216
- [16] Blanco IJ, Vila MA, Martinez-Cruz C. The Use of Ontologies for Representing Database Schemas of Fuzzy Information. *International Journal of Intelligent Systems* 2008; 23(4): pp. 419-445
- [17] Bíla J, Tlapák M. Ontologies and Formation Spaces for Conceptual ReDesign of Systems. *Acta Polytechnica* 2005; 45(4): pp. 33-38
- [18] Zohra B, Bonifati A, Rahm E. *Schema Matching and Mapping*. Springer 2011
- [19] Anicic N, Ivezic N. Semantic Web Technologies for Enterprise Application Integration. *Computer Science and Information Systems* 2005; 2(1): pp. 119-144
- [20] Hong-Hai D, Rahm E. COMA - A System for Flexible Combination of Schema Matching Approaches. In: *Proceedings of the 28th International Conference on Very Large Databases, Hong Kong, China; 2002*; pp. 610-621
- [21] Villanyi B, Martinek P. A Comparison of Schema Matching Threshold Function and ANFIS-generated Membership Function. In: *Proceedings of IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary; 2013*; pp. 195-200
- [22] Martinek P, Villanyi B, Szikora B. Calibration and Comparison of Schema Matchers. *WSEAS Transactions on Mathematics* 2009; 8(9): pp. 489-499