

# Aggregation of Similarity Measures in Schema Matching based on Generalized Mean

Faten A. Elshwimy\*, Alsayed Algergawy<sup>\*†</sup>, Amany Sarhan\*, Elsayed A. Sallam\*

<sup>†</sup>Institute for Computer Science, Friedrich Schiller University of Jena, Germany  
{alsayed.algergawy@uni-jena.de}

\*Department of Computer Engineering,  
Tanta University, Egypt

**Abstract**—Schema matching represents a critical step to integrate heterogeneous e-Business and shared-data applications. Most existing schema matching approaches rely heavily on similarity-based techniques, which attempt to discover correspondences based on various element similarity measures, each computed by an individual base matcher. It has been accepted that aggregating results of multiple base matchers is a promising technique to obtain more accurate matching correspondences. A number of current matching systems use experimental weights for aggregation of similarities among different element matchers while others use machine learning approaches to find optimal weights that should be assigned to different matchers. However, both approaches have their own deficiencies. To overcome the limitations of existing aggregation strategies and to achieve better performance, in this paper, we propose a new aggregation strategy, called the AHGM strategy, which aggregates multiple element matchers based on the concept of generalized mean. In particular, we first develop a practical way to obtain optimal weights that will be assigned to each associated matcher for the given aggregation task. We then use these weights in our aggregation method to improve the performance of matcher combining. To validate the performance of the proposed strategy, we conducted a set of experiments, and the obtained results are encouraging.

## I. INTRODUCTION

Schema matching is the task of identifying and discovering semantic correspondences between elements of two or more schemas [1], [2]. It plays important roles for many database applications, such as data integration to identify and characterize inter-schema relationships between multiple (heterogeneous) schemas [3], data warehousing to map data sources to a warehouse schema [4], E-business to help map messages between different XML formats, Semantic Web to establish semantic correspondences between concepts of different web site ontologies, data migration to migrate legacy data from multiple sources into a new one [5], data transformation to map a source object to a target object, and XML data clustering to determine semantic similarities between XML data [6], [7]. Therefore, numerous schema matching approaches have been proposed and many techniques have been developed. For more details, surveys can be founded in [1], [8], [2].

Almost currently promoted matching systems aggregate the results of a number of individual matchers to obtain more accurate matching results (correspondences). In this way,

every possible kind of information about the input schemas to be matched can be utilized and exploited. Therefore, these approaches exploiting multiple matching strategies meet the problem of aggregating multiple similarities. It has been known that these systems utilize different combining methods, such as parallel, sequential, learning-based, and iterative combination. However, these basic methods may also be combined within more sophisticated match strategies [3], [9], [10], [11]. In our work we restrict ourselves to the most common system architecture of parallel combination that was first introduced by COMA/COMA++ [12], [13]. However, our technique can be easily used and adapted by the other combination strategies. In parallel combination systems, all matchers are executed independently, typically on the whole cross product of source and target schema elements as shown in Figure 1. Results of individual matchers, so called similarity matrices, are put into a similarity cube. A similarity combination operation reduces the cube down to a single similarity matrix. A subsequent selection operator tries to select the most promising element pairs, i.e. by using a threshold. Some systems post-process the found mapping with a constraint resolving step to prune out conflicting mappings.

A challenging issue in most current matching systems is that combination strategies are based on pre-defined aggregation functions which may or may not be suitable for the matching task. Based on this observation, we first adapt the harmony measure to reflect the importance and reliability of each element matcher. We then propose a unified aggregation strategy which is based on the concept of generalized mean. The generalized mean can vary from the minimum to the maximum value of a set depending on its internal parameters. Therefore, we could obtain the arithmetic, geometric, and harmonic means by tuning these parameters. In this paper, we aim at finding a practical way to obtain optimal weights in order to assign to each matcher for similarity aggregation task and searching for the most suitable aggregation method for each matching task. To this end, we adopt this combining strategy, called *AHGM* (Adaptive Harmony-based on Generalized Mean), in order to improve the performance of matching combining. The proposed strategy has been evaluated in order to validate its performance.

To summarize, the main contributions of the paper are the following.

- We propose a new matcher combining strategy, called *AGHM*,
- we adapt the current COMA++ matching system in order to embed the proposed strategy, and
- we conduct an extensive set of experiments to validate the proposed strategy.

The remainder of this paper is organized as follows: Section 2 introduces basic concepts and definitions used through the paper, while Section 3 is devoted to provide an overview of combination strategies. We present our new combining approach in Section 4. We then report the experiments conducted and analyze the results in Section 5. Section 6 concludes the paper.

## II. PRELIMINARIES

In order to keep the paper self-contained, in this section we introduce some basics and definitions that will be used through the paper.

### A. Definitions

A *schema* is a finite set of elements  $S = \{e_1, e_2, \dots, e_n\}$ . Each element has a set of *features* that describe both the internal and external element properties. By internal properties, we mean that the set of features an element has independent on other elements in the schema. Element *name*, *type*, and *constraint* are examples of internal element features. While the structural properties represent the external features of the element and are characterized by the element location w.r.t. other elements in the schema [14].

An *element matcher* is a similarity measure that quantifies the similarity between two elements from two different schemas. It accepts two elements from two schemas and returns a value between 1 and 0, i.e.  $sim : S1.e_i, S2.e_j \rightarrow [0, 1]$ , where  $sim$  is a similarity measure,  $S1.e_i$  is the element  $e_i \in S1$  and  $S2.e_j$  is the element  $e_j \in S2$ . Schema matching is a difficult process, and exploiting only one feature to quantify the similarity between schema elements is not sufficient and results in a poor matching quality. Therefore, combining multiple matchers has been shown to achieve superior results to those obtained by single matchers [14], [13]. This can be represented in the following equation:

$$sim(S1.e_i, S2.e_j) = AGG_{f \in \text{featureset}}(sim_f(S1.e_i, S2.e_j)) \quad (1)$$

where  $S1.e_i$  and  $S2.e_j$  are two elements belonging to schema  $S1$  and schema  $S2$ , respectively,  $sim_f$  is the similarity measure that quantifies the element similarity based on the feature  $f$ , and  $AGG_{f \in \text{featureset}}$  is an aggregation function. In the following we present a definition of the aggregation function [15], [14].

**Definition 1:** An aggregation function  $AGG$  is a function of  $N > 1$  arguments that maps the ( $N$ -dimensional) unit cube onto the unit interval  $AGG : [0, 1]^N \rightarrow [0, 1]$ , with the following properties

- 1)  $AGG(\underbrace{0, 0, \dots, 0}_{N\text{-times}}) = 0$  and  $AGG(\underbrace{1, 1, \dots, 1}_{N\text{-times}}) = 1$ .
- 2)  $x \leq y$   $\implies AGG(x) \leq AGG(y)$  for all  $x, y \in [0, 1]^N$

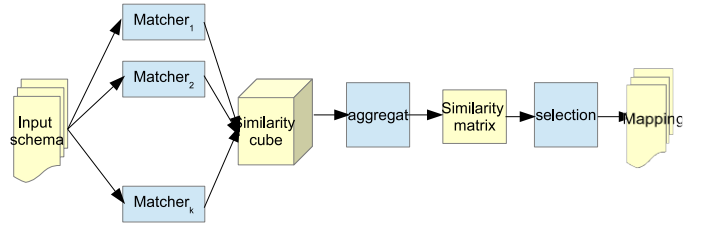


Fig. 1. General matching process.

The input value of 0 is interpreted as no membership, or no evidence, and naturally, an aggregation of  $N$  0s yields 0. Similarly, the value of 1 is interpreted as full membership, and an aggregation of  $N$  1s yields 1. This is the first fundamental property of aggregation functions, the preservation of bounds. The second property is the monotonicity condition. The condition can be stated that for every  $x_1, x_2, \dots, x_N \in [0, 1]$  and  $y_1, y_2, \dots, y_N \in [0, 1]$  such that  $x_i \leq y_i$ ,  $AGG(x_1, x_2, \dots, x_N) \leq AGG(y_1, y_2, \dots, y_N)$ .

As it has been known that many aggregation functions have been proposed. The question is how to choose the most suitable aggregation function for a specific application. In the context of schema matching, a single (fixed) aggregation function is always applied to combine similarities from different schemas without considering the special features of each schema pair [13], [16]. At this stage we can formulate our problem as follows:

*Given two schemas, schema  $S1$  having  $n$  elements and schema  $S2$  having  $m$  elements, and  $K$  element matchers. After the matching step (matcher execution), we get a similarity cube with  $K \times n \times m$ . The problem now is how to reduce this dimension to be a similarity matrix of  $n \times m$  elements.*

### B. General Matching Process

To better understand where aggregation methods are needed in schema matching, in the following, we briefly present the general matching process as shown in Fig. 1. The figure shows that a general matching process contains three main steps: element matcher, aggregation, and selection.

- 1) **Element matcher.** It computes a similarity degree between every pair of schema elements using multiple heuristics. The output of this step is called a *similarity cube*. According to element features element matchers exploit, they can be either individual or combining matchers. Individual matchers exploit only one element feature. However, combining matchers exploit multiple types of clues. They can either be hybrid matchers that integrate multiple criteria, or composite matchers that combine results of independently executed matchers.
- 2) **Aggregation.** The available information about semantics of schema elements may vary; at the same time, the relationships between them are fuzzy. Therefore, most of the existing matching systems make use of multiple element matchers. Every element matcher computes a similarity degree between schema element pairs exploiting a certain element property (feature). Therefore, a matching system with  $K$  element matchers produces  $K$  similarity values for every

schema element pair. To combine these values into a single one, the similarity aggregation is used.

- 3) **Selection.** At this stage, schema elements are corresponding to one or more elements from the other schema(s). The similarity selector step is concerned with selecting the most suitable mapping(s) for each element. However, even having a good ensemble of complementary element matchers cannot guarantee that an optimal mapping will always be identified as the top choice of the ensemble. To address such situations to the last possible degree, one can adapt the approach in which  $\kappa$  top-ranked schema mappings are generated and examined [16]. These mappings can be ranked based on different criteria.

### III. RELATED WORK

Similarity aggregation is one of the important components of schema matching systems to aggregate results of multiple base matchers. Many schema matching approaches that have been developed, such as COMA/COMA++ [12], [13], HADPT [17], RiMOM [10], Falcon-AO [18] use different similarity aggregation strategies. In the following we discuss shortly some of these similarity aggregation strategies which are most related to our work.

- **Min.** The Min combination strategy always chooses the minimum value of a set of values that have been computed by different matchers. This approach is very pessimistic, since it requires all matchers to return high similarity values to later "survive" a selection.
- **Max.** This strategy returns the maximal similarity value of a set of values computed by different matchers.
- **Weighted sum.** The weighted sum strategy computes a weighted sum of matcher input similarities. It needs relative weights which should correspond to the expected importance of the matchers [13].
- **Average.** This strategy represents a special case of Weighted and returns the average similarity over all matchers; it assumes equal weights for every matcher [13].
- **Sigmoid.** This strategy prepares the matcher results for the weighted sum by setting the adjustment function to:

$$adj(x) = \frac{1}{1 + \exp(-t(x - s))} \quad (2)$$

where  $t$  sets the slope and the  $s$  describes a shifting factor for the sigmoid function. These parameters of the sigmoid functions were assigned manually with respect to the underlying similarity method [19].

- **HADAPT.** The harmony-based adaptive aggregation strategy presents a harmony adaptive similarity aggregation method to calculate weights of each matcher based on the harmony of different similarities as their weights for each pair of input ontologies, separately. This method is homogeneous for all pairs of entities of schemas [17].

TABLE I. GENERALIZED MEAN OF INPUT

$p$	Mean
$\rightarrow +\infty$	maximum of $x = (x_1, x_2, \dots, x_n)$
2	quadratic mean
1	arithmetic mean
$\rightarrow 0$	geometric mean
-1	harmonic mean
$\rightarrow -\infty$	minimum of $x = (x_1, x_2, \dots, x_n)$

- **OWA.** Ordered Weighted Average (OWA) is used to determine the weights for a weighted combination. In this strategy each set of similarity values computed for an element pair is ordered and each position in the ordered list gets a weight assigned. A weight used by the OWA operator is associated not with a specific similarity measure, but instead with a specific ordered position [20].
- **Nonlinear.** The nonlinear combination strategy relies on weights but follows an extended approach. It considers the interdependencies between element measures by using nonlinear technique [14].
- **CMC.** The Credibility-based Matcher Combiner (CMC) prediction combining strategy predicts the accuracy of each matcher on the current matching task, and then calculates the matcher's credibility. Results from different element matchers are then aggregated based on their credibilities [21].
- **Meta-learner.** LSD and its extension GLUE make use of a multi-strategy learning approach. The approach has two basic learners: *Name* and *Content* learners. The predictions of basic learners are combined using the meta-learner that assigns to each base learner a learner weight that indicates how much it trusts that learner's predictions. Then it combines the base learners' predictions via a weighted sum [3].

### IV. THE PROPOSED APPROACH

In this section, we aim to introduce our new aggregation strategy for combining similarities of multiple matchers. First we present the generalized mean that the proposed method is based on. Then, to reflect the importance of each matcher, we adopt the harmony strategy. Finally, we show how to use the generalized mean to develop our new aggregation strategy.

#### A. Generalized Mean

In mathematics, generalized means are a family of functions for aggregating sets of positive numbers. The generalized mean can vary from the minimum to the maximum value of the set depending on its parameter. We can obtain the arithmetic, the geometric, and the harmonic means by tuning the parameter.

The generalized mean (GM) of  $N$  numbers can be defined as in Equation 3, where  $p$  is a real number and the value of  $p$  gives the various means depending upon its values [22], [23], as shown in Table I.

$$GM = \left( \frac{1}{N} \sum_{i=1}^N x_i^p \right)^{\frac{1}{p}} \quad (3)$$

### B. Average Harmony

Similarity aggregation is an important and difficult issue in building schema matching systems that contain multiple individual matchers. Many aggregation strategies, such as Max, Weighted, Average and HADAPT, have been proposed to aggregate different similarities. However, the weighted strategy needs to manually set aggregation weights using experience numbers. Alternatively, machine learning based parameter optimization is solution to the problem of manual parameter setting. However, learning based approach needs extra information, i.e., ground truth, which is usually unavailable in real world mapping tasks. HADAPT is a harmony adaptive similarity aggregation method to assign a higher weight to reliable and important similarity measure and a lower weight to those fail to map similar ontologies [17]. The harmony weight for a similarity measure is calculated according to the number of the highest values in the corresponding similarity matrix. However, the harmony measure has drawbacks when there exist other similarity measures that are important as the ones with the highest similarity value. Hence, we extend the HADAPT strategy such that we take into account the values of highest similarity measures rather than using the number of these values. In our method that is called average harmony (ah), we compute the average of the highest similarity measures as in Equation 4.

$$ah = \frac{\sum(max\_sim)}{\#max\_sim} \quad (4)$$

where  $max\_sim$  are the highest similarity measures computed by individual element matchers and  $\#max\_sim$  is the number of the pair of elements that has the highest and the only highest similarity in its corresponding row and column in the similarity matrix.

*Example 1:* [17] Table II is the name similarity matrix obtained after applying name similarity measures on the associated elements. The left part of the table lists the original similarity score between each pair of elements, while the right part of the table illustrates how the harmony is calculated in this case, where  $\times$  denotes the cell that has the highest similarity score in each row,  $\circ$  denotes the cell that has the highest similarity score in each column, and  $\otimes$  denotes the overlapped cell that has the highest similarity in both the row and the column. Therefore the number of max similarities is 4 (the number of  $\otimes$  in Table II) and hence the normal harmony is 0.8. To calculate the average harmony, the summation of max similarities is 3.26 (1+0.36+.9+1), and hence the average harmony equals 0.815.

This example shows that the original harmony considers only the number of the pair of elements that has the highest and the only highest similarity ignoring how high is this similarity? Therefore, the average harmony remedies this issue by considering the values of these similarities instead of their count.

### C. Average Harmony based Generalized Mean aggregation strategy (AHGM)

Once settling on the importance of each element matcher, the question now is how to combine these matchers. Inspired by the importance of flexibility of generalized mean, we propose a similarity aggregation strategy for aggregating various element matchers based on the concept of generalized

mean, called average harmony based generalized mean *AHGM*. To address manual weight assignment, we use the average harmony (ah) as a weight assigned to each matcher in the proposed aggregation strategy. To this end, we rewrite Eq. 1 utilizing Eqs. 3 & 4 as follows:

$$sim(S1.e_i, S2.e_j) = \left( \sum_{f=1}^k ah_f \times sim_f(S1.e_i, S2.e_j)^p \right)^{\frac{1}{p}} \quad (5)$$

where  $k$  is the number of element matchers. From this equation, we get the following special aggregation strategies:

- when  $p = 1$ , we get

$$sim(S1.e_i, S2.e_j) = \sum_{f=1}^k ah_f \times sim_f(S1.e_i, S2.e_j)$$

which represents the weighted sum aggregation method.

- When  $p = 2$ , Eq. 5 becomes

$$sim(S1.e_i, S2.e_j) = \sqrt{\sum_{f=1}^k ah_f \times sim_f(S1.e_i, S2.e_j)^2}$$

which represent the quadratic mean of the similarity measures.

- When  $p = -1$ , Eq. 5 becomes

$$sim(S1.e_i, S2.e_j) = \frac{1}{\sum_{f=1}^k \frac{ah_f}{sim_f(S1.e_i, S2.e_j)}}$$

which stands for the harmonic mean of the similarity measures.

Since  $p$  has a wide range of values extending from  $+\infty$  (to represent the max. strategy) to  $-\infty$  (to represent the min. strategy), in the current implementation we consider values of  $p$  between 0 and 1. This can be explained as follows: it has been known that the average aggregation strategy performs well in almost matching tasks and selecting values of  $p$  between 0 and 1 ensures the generalized mean behaves similar to the average mean.

## V. EXPERIMENTAL EVALUATION

In order to evaluate the performance of the generalized mean-based aggregation strategy, we conducted a set of experiments utilizing real-world schemas of different characteristics. We ran all our experiments on a 2.66 GHz Intel(R) Xeon(R) processor with 4 GB RAM running Windows 7.

### A. Data sets

In our evaluations we used datasets that are five real-world XML schemas on purchase order (CIDX, Excel, Apertum, Noris, Paragon), which were first used in [12], four real-world XML schemas on Spicy [24] (deptDB, orgDB, S3\_source, S3\_target), and two real-world XML schemas on University (CsDeptUs, CsDeptAust) as shown in Table III.

TABLE II. HARMONY COMPUTATION [17]

	composite	book	proc	monography	collection						
Reference	.11	0	.22	0.1	.1				×		
Book	0.22	1	.2	.2	.2			⊗			
Proceedings	.18	.09	.36	.09	.1				⊗		
Monograph	.11	.22	.11	.9	.1					⊗	
Collection	.3	.2	.1	.1	1		○				⊗

TABLE III. DATA SET CHARACTERISTICS

Domain	No. schemas	No. matching task	Min/Max
Spice	4	2	19/123
PO	5	10	40/147
University	2	1	11/11

### B. Experimental methodology

To evaluate the performance of the *AHGM* aggregation strategy, we conduct a set of experiments comparing with seven other aggregation methods selected from both schema matching and information retrieval area. We used the *COMA++*'s *Allcontext* (a combination of *Name*, *Path*, *Leaves*, and *Parents* matchers) matching strategy. To measure the matching quality, we use the same criteria used in the literature, including precision, recall and F-measure as following:

- Precision is a value in the range [0, 1]; the higher the value, the fewer the wrong mappings (false positives) computed and it can be given as:

$$precision = \frac{No\_of\_correct\_found\_matches(bytools)}{No\_of\_found\_matches(bytools)}$$

- Recall is a value in the range [0, 1]; the higher this value, the smaller the set of correct mappings which are not found (also called true positives) and it can be defined as:

$$precision = \frac{No\_of\_correct\_found\_matches}{No\_of\_existing\_matches(byexperts)}$$

- F-measure is a value in the range [0, 1], which is a global measure of the matching quality and it can be defined as:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

### C. Evaluation result

We present results for two sets of experiments. The first set is conducted to validate the performance of the *AHGM* strategy. We run the *COMA++ AllContext* (i.e. Name similarity, Path similarity, Leaves similarity, and Parents similarity) with the proposed strategy across data sets illustrated in Table III. Quality measures for each matching task have been computed and results are reported in Fig. 2. The figure shows that *COMA++* using the *AHGM* strategy produces F-measures of 0.89, 0.92, and 0.7 for Spicy, university, and PO domains, respectively.

The second set of experiments has been conducted to evaluate the performance of the *AHGM* strategy by comparing it with different aggregation strategies. To this end, we compared our strategy to *max*, *min*, *average*, *sigmoid*, *nonlinear*, *OWA*, and *HADAPT* aggregation strategies. Results are reported in Figure 3. Each line represents an aggregation strategy. It is evident that the *max* strategy performs the worst between other

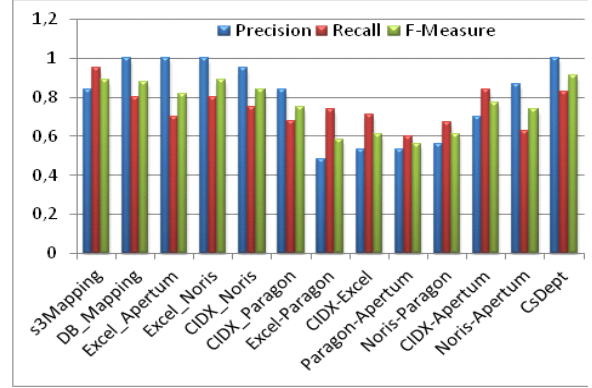
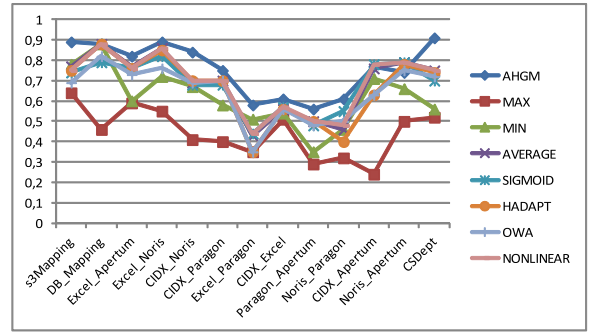
Fig. 2. Quality measure result for *AHGM*.

Fig. 3. Comparison of F-Measure of different aggregation methods.

aggregation strategies (this is similar to results represented in [25]). What surprising is that the *min* strategy performs well compared to some other aggregation strategies. The figure also shows that our proposed strategy, *AHGM*, performs the best compared to all other aggregation strategies. It achieves an average F-measure of 0.76 across all matching tasks compared to 0.68, 0.68, 0.67, 0.66 F-measures produced by nonlinear, average, sigmoid, HADAPT strategies, respectively.

## VI. CONCLUSIONS

It has been proved that, in most cases, combining the results of multiple matching approaches or matchers is a promising technique to get better results than just using one matcher at a time. In this paper, we investigated the problem of combining multiple element matchers in the context of schema matching showing shortcomings in existing aggregation strategy. To cope with this problem, we introduced a similarity aggregation strategy, which can be almost available for each pair of schemas to combine multiple schema-matchers. We first adapted the harmony measure to reflect the importance and reliability of each element matcher. Then, we utilized the generalized mean to develop a new aggregation strategy. The proposed

strategy has been evaluated and validated by comparing it to a well-known set of aggregation strategies in the schema matching community. The experimental results have shown the effectiveness of our method and achieving higher performance than other aggregation methods. In future, we need to find out from the matching task, which aggregation strategy should be applied. Also a combination of automatically setting the weights and automatically adjusting similarity values could be promising.

## REFERENCES

- [1] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.
- [2] P. Shvaiko and J. Euzenat, "Ontology matching: State of the art and future challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 158–176, 2013.
- [3] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, *Ontology matching: A machine learning approach*. International Handbooks on Information Systems: Handbook on Ontologies, 2004.
- [4] M. Banek, B. Vrdoljak, A. M. Tjoa, and Z. Skocir, "Automating the schema matching process for heterogeneous data warehouses," in *9th International Conference on Data Warehousing and Knowledge Discovery*, 2007, pp. 45–54.
- [5] C. Drumm, M. Schmitt, H.-H. Do, and E. Rahm, "Quickmig - automatic schema matching for data migration projects," in *the 16th ACM Conference on Information and Knowledge Management (CIKM)*. Portugal, 2007, pp. 107–116.
- [6] A. Algergawy, E. Schallehn, and G. Saake, "A schema matching-based approach to XML schema clustering," in *The Tenth International Conference on Information Integration and Web-based Applications Services (iiWAS)*, Linz, Austria. ACM, 2008, pp. 131–136.
- [7] M. L. Lee, L. H. Yang, W. Hsu, and X. Yang, "Xclust: Clustering XML schemas for effective integration," in *CIKM'02*, 2002, pp. 63–74.
- [8] Z. Bellahsene, A. Bonifati, and E. Rahm, *Schema Matching and Mapping*. Springer, 2011.
- [9] E. Peukert, H. Berthold, and E. Rahm, "Rewrite techniques for performance optimization of schema matching processes," in *13th International Conference on Extending Database Technology*, 2010, pp. 553–464.
- [10] J. Li, J. Tang, Y. Li, and Q. Luo, "RiMOM: A dynamic multistrategy ontology alignment framework," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 8, pp. 1218–1232, 2009.
- [11] A. Gal and T. Sagi, "Tuning the ensemble selection process of schema matchers," *Information Systems*, vol. 35, no. 8, pp. 845–859, 2010.
- [12] H. H. Do and E. Rahm, "COMA- a system for flexible combination of schema matching approaches," in *VLDB 2002*, 2002, pp. 610–621.
- [13] —, "Matching large schemas: Approaches and evaluation," *Information Systems*, vol. 32, no. 6, pp. 857–885, 2007.
- [14] A. Algergawy, R. Nayak, and G. Saake, "Element similarity measures in XML schema matching," *Information Sciences*, vol. 180, no. 24, pp. 4975–4998, 2010.
- [15] G. Beliakov, A. Pradera, and T. Calvo, "Aggregation functions: A guide for practitioners," in *Studies in Fuzziness and Soft Computing*. Springer, 2007.
- [16] C. Domshlak, A. Gal, and H. Roitman, "Rank aggregation for automatic schema matching," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 538–553, April 2007.
- [17] M. Mao, Y. Peng, and M. Spring, "An adaptive ontology mapping approach with neural network based constraint satisfaction," *Journal of Web Semantics*, vol. 8, no. 1, pp. 14–25, 2010.
- [18] W. Hu, Y. Qu, and G. Cheng, "Matching large ontologies: A divide-and-conquer approach," *Data & Knowledge Engineering*, vol. 67, no. 1, pp. 140–160, 2008.
- [19] M. Ehrig and S. Staab, "QOM- quick ontology mapping," in *3rd International Semantic Web Conference*, 2004, pp. 683–697.
- [20] Q. Ji, P. Haase, and G. Qi, "Combination of similarity measures in ontology matching using the owa operator," in *Recent Developments in the Ordered Weighted Averaging Operators*. Springer, 2011, pp. 281–295.
- [21] K. Tu and Y. Yu, "Cmc: Combining multiple schema-matching strategies based on credibility prediction," in *10th International Conference on Database Systems for Advanced Applications (DASFAA 2005)*, 2005, pp. 888–893.
- [22] P. Bullen, *Handbook of Means and Their Inequalities*. Springer, 2003.
- [23] J. Oh, N. Kwak, M. Lee, and C.-H. Choi, "Generalized mean for feature extraction in one-class classification problems," *Pattern Recognition*, vol. 46, pp. 3328–3340, 2013.
- [24] A. Bonifati, G. Mecca, A. Pappalardo, S. Raunich, and G. Summa, "Schema mapping verification: the spicy way," in *11th International Conference on Extending Database Technology (EDBT)*. France, 2008, pp. 85–96.
- [25] E. Peukert, S. Massmann, and K. König, "Comparing similarity combination methods for schema matching," in *GI-Workshop*, 2010, pp. 692–701.