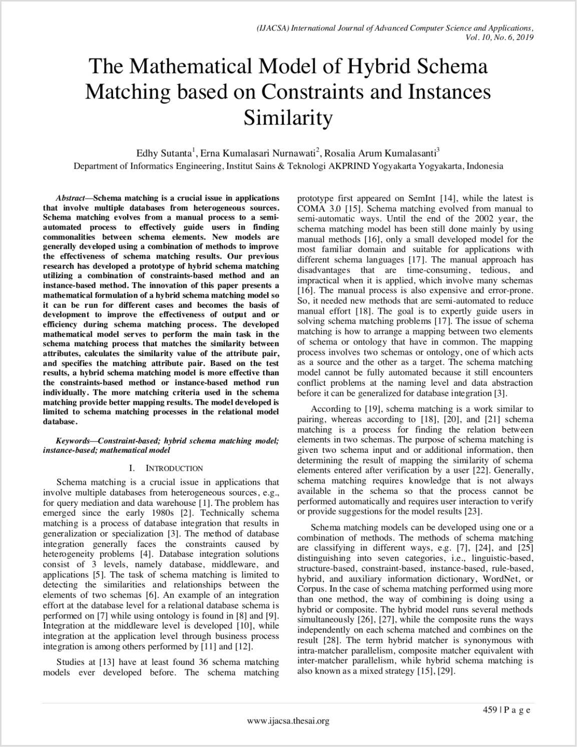
Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Rosalia Arum Kumalasanti
Assignment title:	Periksa similarity
Submission title:	The Mathematical Model of Hybrid Schema Matching based ...
File name:	chema_Matching_based_on_Constraints_and_Instances_Simi...
File size:	43.82M
Page count:	6
Word count:	5,272
Character count:	27,703
Submission date:	02-Aug-2022 09:41AM (UTC+0700)
Submission ID:	1877934787



The Mathematical Model of Hybrid Schema Matching based on Constraints and Instances Similarity

Edhy Sutanta[1], Erna Kumalasari Nurnawati[2], Rosalia Arum Kumalasanti[3]
Department of Informatics Engineering, Institut Sains & Teknologi AKPRIND Yogyakarta Yogyakarta, Indonesia

***Abstract*—Schema matching is a crucial issue in applications that involve multiple databases from heterogeneous sources. Schema matching evolves from a manual process to a semi-automated process to effectively guide users in finding commonalities between schema elements. New models are generally developed using a combination of methods to improve the effectiveness of schema matching results. Our previous research has developed a prototype of hybrid schema matching utilizing a combination of constraints-based method and an instance-based method. The innovation of this paper presents a mathematical formulation of a hybrid schema matching model so it can be run for different cases and becomes the basis of development to improve the effectiveness of output and or efficiency during schema matching process. The developed mathematical model serves to perform the main task in the schema matching process that matches the similarity between attributes, calculates the similarity value of the attribute pair, and specifies the matching attribute pair. Based on the test results, a hybrid schema matching model is more effective than the constraints-based method or instance-based method run individually. The more matching criteria used in the schema matching provide better mapping results. The model developed is limited to schema matching processes in the relational model database.**

***Keywords*—Constraint-based; hybrid schema matching model; instance-based; mathematical model**

I. Introduction

Schema matching is a crucial issue in applications that involve multiple databases from heterogeneous sources, e.g., for query mediation and data warehouse [1]. The problem has emerged since the early 1980s [2]. Technically schema matching is a process of database integration that results in generalization or specialization [3]. The method of database integration generally faces the constraints caused by heterogeneity problems [4]. Database integration solutions consist of 3 levels, namely database, middleware, and applications [5]. The task of schema matching is limited to detecting the similarities and relationships between the elements of two schemas [6]. An example of an integration effort at the database level for a relational database schema is performed on [7] while using ontology is found in [8] and [9]. Integration at the middleware level is developed [10], while integration at the application level through business process integration is among others performed by [11] and [12].

Studies at [13] have at least found 36 schema matching models ever developed before. The schema matching prototype first appeared on SemInt [14], while the latest is COMA 3.0 [15]. Schema matching evolved from manual to semi-automatic ways. Until the end of the 2002 year, the schema matching model has been still done mainly by using manual methods [16], only a small developed model for the most familiar domain and suitable for applications with different schema languages [17]. The manual approach has disadvantages that are time-consuming, tedious, and impractical when it is applied, which involve many schemas [16]. The manual process is also expensive and error-prone. So, it needed new methods that are semi-automated to reduce manual effort [18]. The goal is to expertly guide users in solving schema matching problems [17]. The issue of schema matching is how to arrange a mapping between two elements of schema or ontology that have in common. The mapping process involves two schemas or ontology, one of which acts as a source and the other as a target. The schema matching model cannot be fully automated because it still encounters conflict problems at the naming level and data abstraction before it can be generalized for database integration [3].

According to [19], schema matching is a work similar to pairing, whereas according to [18], [20], and [21] schema matching is a process for finding the relation between elements in two schemas. The purpose of schema matching is given two schema input and or additional information, then determining the result of mapping the similarity of schema elements entered after verification by a user [22]. Generally, schema matching requires knowledge that is not always available in the schema so that the process cannot be performed automatically and requires user interaction to verify or provide suggestions for the model results [23].

Schema matching models can be developed using one or a combination of methods. The methods of schema matching are classifying in different ways, e.g. [7], [24], and [25] distinguishing into seven categories, i.e., linguistic-based, structure-based, constraint-based, instance-based, rule-based, hybrid, and auxiliary information dictionary, WordNet, or Corpus. In the case of schema matching performed using more than one method, the way of combining is doing using a hybrid or composite. The hybrid model runs several methods simultaneously [26], [27], while the composite runs the ways independently on each schema matched and combines on the result [28]. The term hybrid matcher is synonymous with intra-matcher parallelism, composite matcher equivalent with inter-matcher parallelism, while hybrid schema matching is also known as a mixed strategy [15], [29].

The Mathematical Model of Hybrid Schema Matching based on Constraints and Instances Similarity

by Kumalasanti Rosalia Arum

Submission date: 02-Aug-2022 09:41AM (UTC+0700)
Submission ID: 1877934787
File name: chema_Matching_based_on_Constraints_and_Instances_Similarity.pdf (43.82M)
Word count: 5272
Character count: 27703

The Mathematical Model of Hybrid Schema Matching based on Constraints and Instances Similarity

Edhy Sutanta[1], Erna Kumalasari Nurnawati[2], Rosalia Arum Kumalasanti[3]
Department of Informatics Engineering, Institut Sains & Teknologi AKPRIND Yogyakarta Yogyakarta, Indonesia

***Abstract*—Schema matching is a crucial issue in applications that involve multiple databases from heterogeneous sources. Schema matching evolves from a manual process to a semi-automated process to effectively guide users in finding commonalities between schema elements. New models are generally developed using a combination of methods to improve the effectiveness of schema matching results. Our previous research has developed a prototype of hybrid schema matching utilizing a combination of constraints-based method and an instance-based method. The innovation of this paper presents a mathematical formulation of a hybrid schema matching model so it can be run for different cases and becomes the basis of development to improve the effectiveness of output and or efficiency during schema matching process. The developed mathematical model serves to perform the main task in the schema matching process that matches the similarity between attributes, calculates the similarity value of the attribute pair, and specifies the matching attribute pair. Based on the test results, a hybrid schema matching model is more effective than the constraints-based method or instance-based method run individually. The more matching criteria used in the schema matching provide better mapping results. The model developed is limited to schema matching processes in the relational model database.**

***Keywords*—Constraint-based; hybrid schema matching model; instance-based; mathematical model**

I. Introduction

Schema matching is a crucial issue in applications that involve multiple databases from heterogeneous sources, e.g., for query mediation and data warehouse [1]. The problem has emerged since the early 1980s [2]. Technically schema matching is a process of database integration that results in generalization or specialization [3]. The method of database integration generally faces the constraints caused by heterogeneity problems [4]. Database integration solutions consist of 3 levels, namely database, middleware, and applications [5]. The task of schema matching is limited to detecting the similarities and relationships between the elements of two schemas [6]. An example of an integration effort at the database level for a relational database schema is performed on [7] while using ontology is found in [8] and [9]. Integration at the middleware level is developed [10], while integration at the application level through business process integration is among others performed by [11] and [12].

Studies at [13] have at least found 36 schema matching models ever developed before. The schema matching prototype first appeared on SemInt [14], while the latest is COMA 3.0 [15]. Schema matching evolved from manual to semi-automatic ways. Until the end of the 2002 year, the schema matching model has been still done mainly by using manual methods [16], only a small developed model for the most familiar domain and suitable for applications with different schema languages [17]. The manual approach has disadvantages that are time-consuming, tedious, and impractical when it is applied, which involve many schemas [16]. The manual process is also expensive and error-prone. So, it needed new methods that are semi-automated to reduce manual effort [18]. The goal is to expertly guide users in solving schema matching problems [17]. The issue of schema matching is how to arrange a mapping between two elements of schema or ontology that have in common. The mapping process involves two schemas or ontology, one of which acts as a source and the other as a target. The schema matching model cannot be fully automated because it still encounters conflict problems at the naming level and data abstraction before it can be generalized for database integration [3].

According to [19], schema matching is a work similar to pairing, whereas according to [18], [20], and [21] schema matching is a process for finding the relation between elements in two schemas. The purpose of schema matching is given two schema input and or additional information, then determining the result of mapping the similarity of schema elements entered after verification by a user [22]. Generally, schema matching requires knowledge that is not always available in the schema so that the process cannot be performed automatically and requires user interaction to verify or provide suggestions for the model results [23].

Schema matching models can be developed using one or a combination of methods. The methods of schema matching are classifying in different ways, e.g. [7], [24], and [25] distinguishing into seven categories, i.e., linguistic-based, structure-based, constraint-based, instance-based, rule-based, hybrid, and auxiliary information dictionary, WordNet, or Corpus. In the case of schema matching performed using more than one method, the way of combining is doing using a hybrid or composite. The hybrid model runs several methods simultaneously [26], [27], while the composite runs the ways independently on each schema matched and combines on the result [28]. The term hybrid matcher is synonymous with intra-matcher parallelism, composite matcher equivalent with inter-matcher parallelism, while hybrid schema matching is also known as a mixed strategy [15], [29].

Based on survey results [13], research [7] developed a hybrid schema matching model by combining simultaneously constraints-based method approaches and instance-based methods. The model was tested 36 times and yielded the interval parameter values Precision (P) between 71.43% - 100.00%, Recall I at 75.00% -100.00% intervals, and F-measure (F) at intervals 81.48% -100.00 %. The focus on [7] is to develop a logical model and operational architecture for the hybrid schema matching model. The novelty of this paper is to present a mathematical formulation for the hybrid schema matching model [7], so it can be run for different cases and the basis of development to improve the effectiveness of output and or efficiency of schema matching process. The developed mathematical model serves to perform the primary task in the schema matching process that matches the similarity between attributes, calculates the similarity value of the attribute pair, and specifies the matching attribute pair.

The remainder of this contribution is structured as follows: Section 2 describes the material and method which is already applied in several real-world relational databases. Section 3 discusses the results and discussion of the results of the experiment, and finally, Section 4 contains the conclusion and opportunities for further research.

II. The Material and Method

The process in schema matching requires two types of input, i.e., database pairs to match and user verification of the mapping output generated by the model. In this research, the test data consist of a simulation database and test database. Both were developed using a relational database model. The input database is called DBSource (database matched) and DBTarget (database for matching reference).

The simulation database is prepared specially by the researcher for logical model validity testing. Constraints and instances in the simulation database are arranged to vary in a controlled manner so that when used for testing, it can display possible errors. The simulation database consists of 4 db_loc1, db_loc2, db_loc3, and db_loc4, each containing the master data of the provincial, district, and sub-district codes used in e-government applications in Indonesia and developed using MySQL. Each of the simulation databases is composed of 3 tables, eight attributes, and 9.953 instances. The test database is the result of a survey that meets the various criteria on aspects of DBMS, application domain, and size, used as many as 30 pieces. Based on the DBMS used, the test database consists of 22 databases developed using MySQL and eight others developed using MS Access. Based on the application domain, it consists of 8 college educational applications, 12 high school academic applications, eight e-government applications, and two e-commerce applications. By size, the most significant test database composed of 204 tables, the largest attribute count is 1,851, the largest instance count is 232,893 items, and the largest capacity is 79,769 Kb while the smallest test database contains 1 table, with 16 attributes, 480 instances, measuring 115 Kb.

The process sequence in our model is:

1) accept DBSource and DBTarget input;

2) extracting constraints and instances;

3) perform matching and computation of similarity values (SIM) in the attribute pair in DBSource and DBTarget;

4) specify the matching attribute pairs;

5) displaying the initial result of mapping the similarity of the attribute pair for the DBSource and DBTarget pair;

6) receive user verification of the initial finding of mapping the similarity of the attribute pair;

7) displaying the final result of mapping the similarity of the attribute pair that has been verified by the user; and calculate and display the values of effectiveness parameters, i.e., P, R, and F.

The output generated by the model includes:

1) general information about DBSource and DBTarget, including constraint and instance;

2) attribute values (SIM) of each pair of attributes in DBSource and DBTarget;

3) initial results mapping the similarity of the attribute pair as an output model;

4) the result of the mapping of the similarity of attribute pair which has been verified by the user; and

5) the effectiveness parameter values P, R, and F.

The values of Precision (P), Recall (R), and F-measure (F) are calculated using the following formula [30]:

$$P = \frac{TP}{TP+FP} \quad (1)$$

$$R = \frac{TP}{TP+FN} \quad (2)$$

$$F = \frac{2*P*R}{P+R} \quad (3)$$

The TP represents the relevant model output, and the user accepts it. TN is outward of the relevant model, and the user does not accept it. The FP is an outsource of the model that is irrelevant and accepted by the user, and the FN is the outward model that is irrelevant and is not accepted by the user.

Precision reflects the share of real correspondences among all found ones, Recall specifies the share of real correspondences that found, and F-measure represents the harmonic mean of Precision and Recall and is the most common variant of F-measure in Information Retrieval.

Overall this research consists of 7 stages, namely, 1) development of logical model, 2) development of model architecture, 3) development of procedures, 4) development of prototype model, 5) testing the validity of the model using the database simulation, 6) the development of mathematical models, and 7) testing model using the test database. Stage 1-5 has been performed on [7], and the focus of this paper is to present the mathematical model (steps 6) and show the comparison of the effectiveness of the hybrid schema matching model over the constraint-based method and the instance-based method performed separately (stage 7). The matching mechanism and the similarity value calculation of the attribute pair in the hybrid schema matching model shown in Fig. 1.

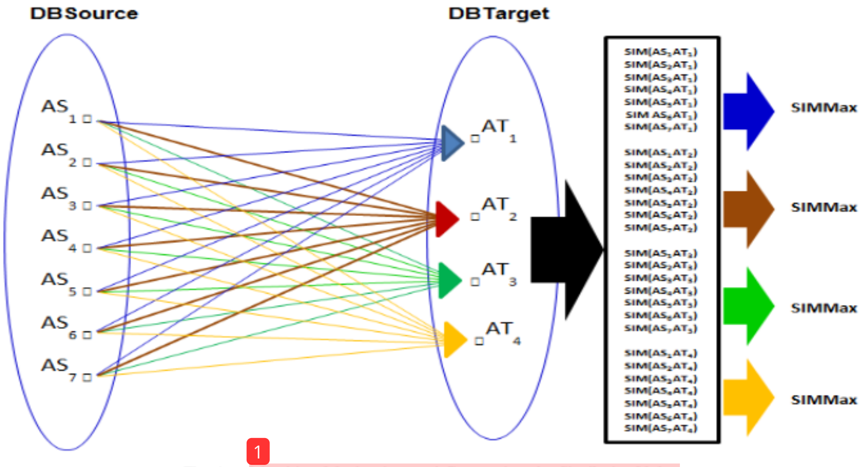


Fig. 1. Matching Mechanism and Computing the Similarity Value.

III. Result and Discussion

A. A Sample of Hybrid Schema Matching

To provide an overview process of the hybrid schema matching model, the following given a simple example of schema matching process for the employee (Table I and Table II) and religion (Table III and Table IV).

Table I to Table IV show the constraints and instances of employee and religion. Each possible attribute pair on employee and religion is then matched and calculated by the value of the similarity to determine the matching pair. Then it is verified to ascertain whether the pair that is declared to be matched by the model strictly matches, or must be revised, or otherwise unpaired. In this example, the calculation of the similarity value of each pair of attributes is based on the assumption that each matching criterion has the same weight so that each value is 0.166. The matching criteria on the constraint are of type, width, nullable, unique, and domain, whereas in an instance the attribute pair is the same if found at least one value of the same value. In each matching process using a specific criterion, if found similarity, then the value of similarity on the criterion is 0.166, otherwise the value of the similarity is 0.000. The attribute pair SIM value calculates by summing the similarity values across all criteria. Based on the SIM attribute pair values, then determined the attribute pair that is declared suitable that has the highest value or 1.00.

TABLE I. The Table Structure of Employee

Column	Type	Width	Null	Unique	Domain
employee_id	char	3	no	yes	-
employee_name	varchar	100	no	no	-
religion_id	char	1	no	no	-

TABLE II. The Instance of Employee

employee_id	employee_name	religion_id
emp_01	Edhy Sutanta	M
emp_02	Erna Kumalasari N	M
emp_03	Rosalia Arum K	C

TABLE III. The Table Structure of Religion

Column	Type	Width	Null	Unique	Domain
religion_id	Char	1	no	yes	-
religion_name	varchar	10	no	yes	-

TABLE IV. Instance of Religion

employee_id	religion_name
emp_01	Moslem
emp_02	Christian

There are three attributes on employee and two attributes in religion, so the matching process will doing six times. The matching pair of attributes is employee_id in employee and religion_id in religion, employee_name in employee and religion_id in religion, religion_id in employee and religion_id in religion, employee_id in employee and religion_name in religion, employee_name in employee and religion_name in religion. The result obtained three pairs of attributes that are otherwise suitable, is religion_id in employee match with religion_id in religion on SIM = 0.664, employee_id in employee match with religion_name in religion on SIM = 0.332, and employee_name in employee match with religion_name in religion on SIM = 0.332. Then, when verification, only one pair of attributes are found to be suitable is religion_id in employee match with religion_id in religion. The effectiveness parameter values obtained are P = 1.00, R = 0.25, and F = 0.40.

B. The Hybrid Schema Matching Logical Test

Testing the validity of the model is done to ensure that the logical model, model architecture, procedures, and prototypes developed are valid logically. Testing is done in 2 ways, manually and using a software of model prototype. The test was conducted 16 times using a combination of 4 databases simulated as source and target. The values of TP, FP, FN, and TN, and P, R, and F in each test using the developed and manually model are the same as the results obtained in the test using the model prototype so that the developed model is validly valid logically. The end result obtained the average value of P = 94.42%, R = 100.00%, and F = 97.02%.

C. The Hybrid Schema Matching Logical Test

The description of the mathematical model for our hybrid matching schema is as follows:

$\mathrm{DS} = \{\mathrm{RS_1}, \mathrm{RS_2}, .. \mathrm{RS_m}\}$, $\mathrm{RS_i}$ is a table in DS

$\mathrm{RS_i} = (\mathrm{AS_{i1}}, \mathrm{AS_{i2}}, .., \mathrm{AS_{ik_I}})$, $1 \le I \le n$, representation of Rsi as attribute pair

$\mathrm{tS_{ij}} = (\mathrm{vS_{i1j}}, \mathrm{vS_{i2j}}, .., \mathrm{vS_{ik_ij}})$, $1 \le j \le l_i$, tuple to-j in Rsi

where

$\mathrm{tS_{ij}} \in r(\mathrm{RS_i}) \subset \mathrm{DOM}(\mathrm{AS_{i1}})_x \mathrm{DOM}(\mathrm{AS_{i2}})_x .. _x\mathrm{DOM}(\mathrm{AS_{ik_I}})$

$\mathrm{DT} = \{\mathrm{RT_1}, \mathrm{RT_2}, .. \mathrm{RT_n}\}$, $\mathrm{RT_i}$ is a table in DT

$\mathrm{RT_p} = (\mathrm{AT_{p1}}, \mathrm{AT_{p2}}, .., \mathrm{AT_{pr_p}})$, $1 \le p \le m$, representation of Rti as attribute pair

$\mathrm{tT_{pq}} = (\mathrm{vT_{p1q}}, \mathrm{vT_{p2q}}, .., \mathrm{vT_{pr_pq}})$, $1 \le q \le s_p$, tuple to-q in Rti

where

$\mathrm{tT_{pq}} \in r(\mathrm{RT_p}) \subset \mathrm{DOM}(\mathrm{AT_{p1}})_x \mathrm{DOM}(\mathrm{AT_{p2}})_x .. _x\mathrm{DOM}(\mathrm{AT_{pr_p}})$

DS declares the source database to be matched (DBSource). RS is a relation in DS, and AS is an attribute in the RS. tS is a tuple in RS. vS is the data value in tS. M is the relation cache in DS. DT is the reference target database for matching (DBTarget). RT is a relation in the DT, and AT is an attribute in the RT. tT is a tuple in RT. vT is the value of data in tT, and n is a relation count in DT. While a DMATCH is representing the result of schema matching for the pair of DS and DT, where x is a cache attribute in DS, and y is a cache attribute in DT, then;

$\mathrm{DMATCH} = \{(\mathrm{AS_1}, \mathrm{AT_1}, \mathrm{AS_1}, \mathrm{AT_2}),.. (\mathrm{AS_x}, \mathrm{AT_y})\}$

The C represents the set of the match criteria;

$C = \{T, W, N, U, D, I\}$

where the T, W, N, U, D, and I represent the type, the width, the nullable, the unique, the domain, and the instance, respectively.

The similarity value for any pair of attributes in the DS and the b attribute in the DT value is calculated as follows:

$$\mathrm{SIMT}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \mathrm{T}(\mathrm{AS_a}) = \mathrm{T}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (4)$$

$$\mathrm{SIMW}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \mathrm{W}(\mathrm{AS_a}) = \mathrm{W}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (5)$$

$$\mathrm{SIMN}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \mathrm{N}(\mathrm{AS_a}) = \mathrm{N}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (6)$$

$$\mathrm{SIMU}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \mathrm{U}(\mathrm{AS_a}) = \mathrm{U}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (7)$$

$$\mathrm{SIMD}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \mathrm{D}(\mathrm{AS_a}) = \mathrm{D}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (8)$$

$$\mathrm{SIMI}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \exists\, \mathrm{I}(\mathrm{AS_a}) = \mathrm{I}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (9)$$

Where the SIMT is the value of similarity for the T criteria. The SIMI is the value of similarity for I criteria. The SIMW is the value of similarity for W criteria. The SIMN is the value of similarity for N criteria. The SIMU is the value of similarity for U criteria. The SIMD is the value of similarity for D criteria, and the SIMI is the value of similarity for I criteria. In general, the calculation of the similarity value of the attribute pair for any C member is:

$$\mathrm{SIM_c}(\mathrm{AS_a}, \mathrm{AT_b}) = \begin{cases} 1, \mathrm{x}(\mathrm{AS_a}) = \mathrm{x}(\mathrm{AT_b}) \\ 0, \text{others} \end{cases} \quad (10)$$

where x(A)= criteria x for A.

While WI is the weight of the instance, WT is the weight of type, WW is the width weight, WN is the nullable weight, WU is a specific weight, and WD is the domain weight. Thus, the $\mathrm{AS_a}$ and $\mathrm{AT_b}$ pair similarity values are calculated as follows:

$$\mathrm{SIM}(\mathrm{AS_a}, \mathrm{AT_b}) = \sum\nolimits_{y \in C} \mathrm{SIM_y}\,(\mathrm{AS_a}, \mathrm{AT_b})\, \mathrm{W_y} \quad (11)$$

Paired attributes that are otherwise matched by the model, if the $\mathrm{AT_b}$ matched to $\mathrm{AS_a}$ then $\mathrm{AT_b}$ are taken which meets the following conditions:

$$\mathrm{SIM}(\mathrm{AS_a}, \mathrm{AT_b}) = \underset{z=1}{\overset{m}{\mathrm{Max}}}\ \mathrm{SIM}(\mathrm{AS_a}, \mathrm{AT_z}) \quad (12)$$

where

z = 1, 2, .., m.

D. The Hybrid Schema Matching Test Result

A valid hybrid schema matching model, then tested 32 times using a combination of randomly chosen test database pairs, and the results displayed in Table V.

Based on the test results in Table V and Table VI, it is known the highest P-value is 93.04% in the hybrid model, the lowest P-value is 33.28% in the constraint-based method, while the instance-based method is between constraint-based and hybrid. These results show that hybrid models provide the best results from the precision. Compared to the instance-based method, there was an increase of 31.94%, while compared to the constraint-based approach, there was an increase of 59.76%. This condition occurs because the hybrid model matches by involving more criteria (6 criteria at once), compared to the constraint-based method using five criteria or the instance-based method using one criterion. The more criteria applied for matching between attributes, and the results increase the value of the parameter P because the pair declared matched by the model have a higher chance of being accepted by the user as the right pair. These results indicate that the similarity of instances can be the basis for finding the matching attribute pair with an average P of 61.10%. Lowest P occurs in the constraint-based method, which is 33.28%. These results show that, even if the matching attribute pair has the same constraint, the result is still lower than the hybrid model and the instance-based method individually.

Based on the experiments on each model, effectiveness obtained, as shown in Table VI.

The highest R-value is 100.00% that is in the constraint-based method and the instance-based method, and both are equal. The lowest R-value is 99.83%, in the hybrid model. The value of R = 100.00% indicates that all attribute pairs declared matched by the model (in the initial result) are perfectly matched and accepted by the user (TP = N), and the attribute pair stated unmatched by the model (in the initial result) is received by the user (FN = 0).

TABLE V. THE RESULT OF HYBRID SCHEMA MATCHING

DBSource	DBTarget	Test Result (%)		
		P	R	F
sipt_admision	sipt_admision	89.90	100.00	94.68
sipt_admision	sipt_academic	85.11	100.00	91.95
sipt_academic	sipt_payroll	85.42	98.80	91.62
sipt_academic	sipt_employee	83.08	96.43	89.26
sipt_academic	sipt_tax_pph	91.37	99.45	95.24
sipt_academic	sipt_workshop	82.95	100.00	90.68
sipt_academic	sipt_library	88.46	100.00	93.88
sipt_academic	sipt_user	92.98	100.00	96.36
egov_dptkp	license	94.64	100.00	97.25
egov_dptkp	license_oln	88.48	100.00	93.89
egov_dptkp	egov_dptbgcpt	100.00	100.00	100.00
egov_dptkp	quickcount_bgcpt	94.01	100.00	96.91
egov_dptkp	egov_dptbtl	100.00	100.00	100.00
egov_dptkp	egov_dptkp	100.00	100.00	100.00
egov_dptkdy	ecomm_rsmitra	94.12	100.00	96.97
egov_dptkdy	ecomm_motorcred	90.55	100.00	95.04
nuptk	nuptk	99.41	100.00	99.71
nuptk	hs_sinisa	89.37	100.00	94.39
nuptk	hs_sipp	93.15	100.00	96.45
nuptk	hs_psb	98.01	100.00	99.00
hs_sipp	hs_sinisa	99.69	99.97	99.83
hs_sipp	hs_sipp	99.35	100.00	99.68
hs_sipp	hs_psb	93.73	100.00	96.76
hs_sipp	hs_grade	93.96	100.00	96.89
hs_sipp	hsgrade_ol	90.45	100.00	94.99
hs_sipp	hs_report	98.76	100.00	99.38
hs_sipp	hs_hspwt	94.49	99.98	97.16
hs_sipp	hs_forum	90.06	100.00	94.77
hs_sipp	hs_announcement	87.52	100.00	93.34
hs_sipp	hs_webinfo	98.24	100.00	99.11
hs_sipp	hs_osis	91.05	100.00	95.32
hs_sipp	hs_elearning	98.99	100.00	99.49
Average:		93.04	99.83	96.25

TABLE VI. THE RESULT OF HYBRID SCHEMA MATCHING

Method	Test Result Average (%)		
	P	R	F
Hybrid (constraint-based and instance-based)	93.04	99.83	96.25
Constraint-based method	33.28	100.00	38.60
Instance-based method	61.10	100.00	69.87

The highest F value is 96.25%, i.e. in the hybrid model, while the lowest is 38.60% in the constraint-based method. Value F = 96.25% indicates that the estimated level of effort required to add FN and eliminate FP has reached the best condition. The value of F in the hybrid model, compared with the instance-based method has increased by 26.38%, and when compared with the constraint-based method, there is an increase of 57.65%. This increase occurs because hybrid models perform matching by involving more criteria that are using six criteria at once, so the attribute pair declared fit by the model has a higher chance of being accepted by the user as a matching pair.

The experimental results in this study indicate that the mathematical model for hybrid schema matching has functioned as expected and provided better effectiveness than the original constraint-based method and instance-based method. The model that we have developed is still likely to be further investigated, at least to improve the effectiveness of output, process efficiency, and modification so that the model can be applied to non-relational databases.

This hybrid schema matching model still contains problems related to the effectiveness of the outcome because the constraint matching criteria and instance are assumed to have equal weight when calculating the similarity values (SIM) of each pair of attributes. Each criterion on the constraint is also considered to have the same weight. These criteria can have different weights in determining the value of similarity (SIM); one of the weighting ideas ever done [31]. The results of the survey at the time of collecting the test database also found the fact that the database designer has the freedom to make the database schema definition, including determining the data size (width) in the string data type. In this test, the string attribute pair is declared the same if it has the same size; it has not yet accommodated the freedom of the database designer.

Another problem is related to process efficiency. The developed model requires repetition of matching steps and simulated (SIM) value calculations on all possible attribute pairs. When schema matching performed on a database pair it involves many foreign keys, the model still encounters efficiency problems in matching and simulated values (SIM) and user verification steps that can only be done manually.

IV. CONCLUSIONS

The main contribution of this paper is to present the proposed mathematical model for hybrid schema matching. The model is developed based on a combination of two methods, namely the constraints-based method and an instance-based method. The model has been tested using a relational database, and the results are more effective than the original constraint-based method or instance-based method. Our research further refines the model by adding features to improve output effectiveness and process efficiency. Also, the model developed to run in non-relational database formats.

ACKNOWLEDGMENT

The authors would like to thanks the Directorate of Research and Community Service, Directorate General of Research and Development Strengthening at the Ministry of

Research, Technology and Higher Education of the Republic of Indonesia under Fundamental Research Grant Contract No.: 109/SP2H/LT/DRPM/2018.

REFERENCES

[1] L. A. P. P Leme, M. A. Casanova, K. K. Breitman, and A. L Furtado, “OWL Schema Matching,” Journal of the Brazilian Computer Society, 16(5), pp. 21-34, April 2010, DOI: 10.1007/s13173-010-0005-3.

[2] I. F. Cruz, F. P. Antonelli, and C. Stroe, “AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies (demo paper),” in International Conference on Very Large Data Bases (VLDB), 2009, DOI: 10.14778/1687553.1687598.

[3] C. Kavitha, G. S. Sadasivam, and S. N. Shenoy, “Ontology-Based Semantic Integration of Heterogeneous Databases,” European Journal of Scientific Research, 64(1), pp. 115-122, November 2011.

[4] S. Hamill, M. Dixon, B. J. Read, and J. R. Kalmus, “Interoperating Database Systems: Issues and Architectures,” Council for the Central Laboratory of The Research Councils, United Kingdom, Technical Report 1997.

[5] Ministry of Communication and Information Republic of Indonesia, “SIFONAS Sebagai Tulang Punggung e-Governance (SIFONAS as The Backbone of e-Governance),” 2002.

[6] P. Martinek, “Schema Matching Methodologies and Runtime Solutions in SOA Based Enterprise Application Integration,” Department of Electronics Technology, Faculty of Electrical Engineering & Informatics, Budapest University of Technology and Economics, Hungary, Ph.D. Thesis, 2009.

[7] E. Sutanta, R. Wardoyo, K. Mustofa, and E. Winarko, “A Hybrid Model Schema Matching using Constraint-Based and Instance-Based,” International Journal of Electrical and Computer Engineering (IJECE), 6(3), pp.1048-1058, June 2016, DOI: 10.11591/ijece.v6i3.pp 1048-1058.

[8] H. Alani and S. Saad, "Schema Matching for Large-Scale Data Based on Ontology Clustering Method," International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), 7(5), pp. 1790-1797, 2017, DOI: 10.18517/ijaseit.7.5.2133.

[9] K. H. Shafa’amri and J. O. Atoum, “A Framework for Improving the Performance of Ontology Matching Techniques in Semantic Web,” International Journal of Advanced Computer Science and Applications (IJACSA), 3(1), pp. 8-14, 2012.

[10] N. H. Azizul, A. M. Zin, and E. Sundararajan, "The Design and Implementation of Middleware for Application Development within Honeybee Computing Environment," International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), 6(6), pp. 937-943, 2016, DOI: 10.18517/ijaseit.6.6.1415.

[11] D. Sulisworo, Tawar, and U. Ahdiani, "ICT Based Information Flows and Supply Chain in Integrating Academic Business Process," International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), 2(6), pp. 44-48, 2012, DOI: 10.18517/ijaseit.2.6.243.

[12] M. Mohammadi and Muriati Mukhtar, "Business Process Modelling Languages in Designing Integrated Information Systems for Supply Chain Management," International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), 2(6), pp. 464-467, 2012, DOI: 10.18517/ijaseit.2.6.245.

[13] E. Sutanta, R. Wardoyo, K. Mustofa, and E. Winarko, "Survey: Models and Prototypes of Schema Matching," International Journal of Electrical and Computer Engineering (IJECE), 6(3), pp. 1011-1022, June 2016, DOI: 10.11591/ijece.v6i3.pp1011-1022.

[14] W. S. Li and C. Clifton, "Semantic Integration in Heterogeneous Databases Using Neural Networks," in The 20th International Conference on Very Large Data Bases (VLDB), Santiago de Chile, Chile, 1994, pp. 1-12.

[15] E. Rahm, "Schema Matching and Mapping: Towards Large-Scale Schema and Ontology Matching," in Data Centric Systems and Applications, Z. Bellahsene, A. Bonifati, and E. Rahm, Eds. New York, USA: Springer, 2011, pp. 3-27, DOI: 10.1007/978-3-642-16518-4_1.

[16] H. H. Do, S. Melnik, and E. Rahm, "Comparison of Schema Matching Evaluations," in Proceedings of The 2nd International Workshop Web and Databases, In Lecture Notes In Computer Science (LNCS) 2593, Springer-Verlag, Germany, 2002, pp. 221-237, DOI: 10.1007/3-540-36560-5_17.

[17] H. H. Do, "Schema Matching and Mapping-Based Data Integration," Interdisciplinary Center for Bioinformatics and Department of Computer Science, University of Leipzig, Leipzig, Germany, Ph.D. Thesis, 2005.

[18] D. Engmann and S. Massmann, "Instance Matching with COMA++," in Datenbank Systeme in Business, Technologie und Web (BTW Workshop): Model Management and Metadata, Aachen, Germany, 2007, pp. 28-37, https://dbs.uni-leipzig.de/file/BTW-Workshop_2007_EngmannMassmann.pdf.

[19] J. Li, J. Tang, Y. Li, and Q. Luo, "RiMOM: A Dynamic Multistrategy Ontology Alignment Framework," IEEE Transaction Knowledge Data Engineering, 21(8), pp. 1218-1232, August 2009, DOI: 1109/TKDE .2008.202.

[20] P. A. Bernstein, B. Harry, P. Sanders, D. Shutt, and J. Zander, "The Microsoft Repository," in Proceedings of The 23rd International Conference Very Large Data Bases (VLDB), Athens, Greece, 1997, pp. 3-12.

[21] A. Stabenau, G. McVicker, C. Melsopp, G. Proctor, M. Clamp, and E. Birney, "An Overview of ENSEMBL," Genome Research Journal, 14(5), pp. 929-933, May 2004, DOI: 10.1101/gr.1860604.

[22] J. Madhavan, P. A. Bernstein, and E. Rahm, "Generic Schema Matching with CUPID," in Proceedings of The 27th International Conference on Very Large Data Bases (VLDB), Roma, Italy, 2001, pp. 49-58, http://dl.acm.org/citation.cfm?id=645927.67219.

[23] S. Melnik, E. Rahm, and P. Bernstein, "RONDO: A Programming Platform for Generic Model Management," in Proceedings of The ACM-SIGMOD Conference on Management of Data (SIGMOD), San Diego, California, USA, 2003, pp. 193-204, DOI: 10.1145/872757.872782.

[24] P. A. Bernstein, M. Jayant, and E. Rahm, "Generic Schema Matching, Ten Years Later," in The 33 International Conference on VLDB Endowment, Seattle, Washington, 2011, pp. 695-701, http://www.vldb.org/pvldb/vol4/p695-bernstein_madhavan_rahm.pdf.

[25] A. A. Alwan, A. Nordin, M. Alzeber, and A. Z. Abualkishik, “A Survey of Schema Matching Research using Database Schemas and Instances,” International Journal of Advanced Computer Science and Applications (IJACSA), 8(10), pp. 102-111, 2017.

[26] S. Bergamaschi, S. Castano, and M. Vincini, "Semantic Integration of Semistructured and Structured Data Sources," SIGMOD Record, 28(1), pp. 54-59, March 1999, DOI: 10.1145/309844.309897.

[27] T. Milo and S. Zohar, "Using Schema Matching to Simplify Heterogeneous Data Translation," in Proceedings of The 24th International Conference on Very Large Data bases (VLDB), New York, USA,1998, pp.122-133, http://dl.acm.org/citation.cfm?id=645924.671326.

[28] A. H. Doan, P. Domingos, and A. Y. Halevy, "Reconciling Schemas of Disparate Data Sources-A Machine-Learning Approach," in Proceedings of The ACM SIGMOD International Conference Management of Data, Santa Barbara, California, 2001, pp. 509-520, DOI: 10.1145/376284.375731.

[29] A. Gross, M. Hartung, T. Kirsten, and E. Rahm, "On Matching Large Life Science Ontologies in Parallel," in Proceedings of The 7th International Conference Data Integration in the Life Sciences (DILS), Gothenburg, Sweden, 2010, pp. 35-49, http://dl.acm.org/citation.cfm?id= 1884477.1884483.

[30] Y. Karasneh, H. Ibrahim, M. Othman, and R. Yaakob, "An Approach for Matching Relational Database Schemas," Journal of Digital Information Management, 8(4), pp. 260-269, 2010, https://dblp.org/rec/bib/journals/jdim/KarasnehIOY10.

[31] M. B. Shuaibu, "Determining an Appropriate Weight Attribute in Fraud Call Rate Data Using Case-Based Reasoning," International Journal on Advanced Science, Engineering and Information Technology (IJASEIT), 4(1), pp. 34-36, 2014, DOI: 10.18517/ijaseit.4.1.357.

The Mathematical Model of Hybrid Schema Matching based on Constraints and Instances Similarity

ORIGINALITY REPORT

10%	9%	9%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijece.iaescore.com Internet Source	3%
2	Submitted to University of Philadelphia - Jordan Student Paper	3%
3	E Sutanta, E K Nurnawati. "The Design of Relational Database for Multipurpose WebGIS Applications", Journal of Physics: Conference Series, 2019 Publication	1%
4	ecommons.luc.edu Internet Source	1%
5	Jie He. "A dynamic schema matching approach for multi-version web feature service retrieve", 2009 17th International Conference on Geoinformatics, 08/2009 Publication	<1%
6	ul.qucosa.de Internet Source	<1%

7	iisl.postech.ac.kr Internet Source	<1 %
8	R. Santodomingo, S. Rohjans, M. Uslar, J.A. Rodríguez-Mondéjar, M.A. Sanz-Bobi. "Ontology matching system for future energy smart grids", Engineering Applications of Artificial Intelligence, 2014 Publication	<1 %
9	insightsociety.org Internet Source	<1 %
10	nozdr.ru Internet Source	<1 %
11	epdf.pub Internet Source	<1 %
12	Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis et al. "Valentine: Evaluating Matching Techniques for Dataset Discovery", 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021 Publication	<1 %
13	Anton Yudhana, Abdul Fadlil, Muhamad Rosidin. "Indonesian Words Error Detection System using Nazief Adriani Stemmer Algorithm", International Journal of Advanced Computer Science and Applications, 2019 Publication	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		