CSSM : CONCEPT AND STRUCTURAL SIMILARITY METHOD - A SOLUTION ON ACCORDANCE OF METADATA IN LEGACY AND MODERN DATABASE

[1]NEGAR MAJMA, [2]AHMAD BARAANI-DASTJERDI

[1]M.Sc. Student, Computer Engineering Department, University of Sheykhbahaee, Isfahan, Iran

[2]Assistant Prof., Computer Engineering Department, University of Isfahan, Isfahan, Iran

ABSTRACT

Importance of a software system is due to its inner data and databases and not to its codes. Therefore, with the appearance of new software, converting data of the previous systems to the new ones is significant and plays an important role in organizations. Although this conversion process may have different phases but the first phase is under consideration in this paper. First phase in the process of converting data, which refers to assessing tables and finding equivalent fields in both legacy and modern databases is being called matching metadata. Considering its' importance, this paper provides an automatic and intelligent method for assessing table and equivalent fields in both databases and for extracting the degree of their accordance. Then, the proposed method will be evaluated on sample databases and the consequent results will be presented.

Keywords: *Database, Legacy database, Integration, Schema matching*

1. INTRODUCTION

Experiences have shown that lifetime of a software is too variable and depends on many factors. For instance, many large systems remain in use for more than 10 years [1]. However, due to the changes in demands and business processes in every organization, applying a new generation of software is inevitable. Considering this fact and the reality that no system is valuable for organization without the data of legacy systems, organizations must consider converting data from legacy systems to a new one at the end of developing new software. Nevertheless, most of old databases are constant, stable and unchangeable and their information is needed only for preparing reports. Such systems that are stable and constant against the changes are being called Legacy Information Systems [2].

The first step in converting data is recognition of schema structures and matching them. In ordinary cases this operation can be done by DBA or using expert studies. However, with the aim of simplifying this assessment, this article provides an automatic method for recognition and matching databases' schemas. As a sample scenario, this method will be implemented and evaluated using old and new databases of a real organization.

The rest of the paper is organized as follows. A brief overview of related work will be presented in section 2. This overview will be followed by a method of schema extraction discussed in section 3. In the next step, some matching algorithms will be analyzed in sections 4 and 5. Thereafter, final algorithm of this paper will be introduced in section 6. Finally, section 7 will include assessment and deduction and will present some opportunities for future research.

2. RELATED WORK

Related work could be studied in two general categories: The first category consists of the activities related to studying data conversion and its methods [3, 4, 5, 6, 7]. The second category also consists of the activities related to methods of establishing semantic correlation [8,9,10,11,12,13] for information retrieval.

2.1. Activities related to data conversion

This category provides with different strategies for promotion of legacy systems to the new ones. These strategies can be divided in three major categories. Redevelopment, Wrapping and

Migration. Figure 1 has shown position of these three categories.

In redevelopment, companies change the system or create an entirely new system [7]. Not only Hardware, architecture and tools of the system, but also old and new databases will be analyzed for reengineering or redevelopment. Anyway, reengineering can be achieved using two major methods [6]. First method is design and reengineering that is extravagance method for solving legacy systems' problems. Second method is being called reverse engineering, which divided into two major groups called white and black box methods. Both methods act on perception and understanding of software [5, 4].

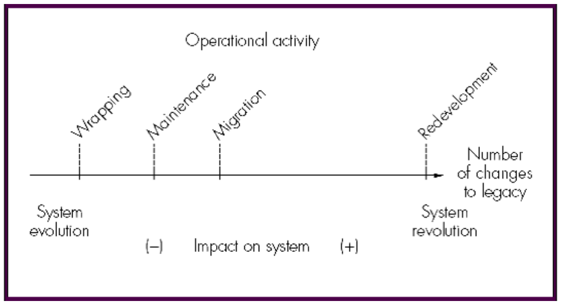

Figure1.Place of different strategy[7]

Wrapping is the second category. In this method a wrapper or an envelope is being set out in legacy system and the new interface is being assigned for a user on the previous structure [7]. This method is useful for replacement and gradual reengineering of legacy system and for avoidance of sudden changes in the system. Because of importance of their data, in this method legacy systems will not get aside; instead, they will be applied as a part of new software. Although no new information is being inserted in these systems, but their activeness and availability always would be necessary for information retrieval purposes. Wrappings are divided into three major groups according to their level of activity. This division consists of user interface wrapping for the first group, data wrapping for the second group and functions wrapping for the third group [14, 4].

Migration is the third strategy. All legacy system's data will be transferred while the process of converting to a new system [7]. Migration method is the long term solution for promoting systems and indicates more flexibility against any kinds of systems and databases. Though there is no comprehensive and precise method right now that could be used for modernizing any legacy systems[6].

2.2. Activities related to establishing semantic relation

In this group of activities the procedure is based on schema integration. In other words, groups of schemas merge into a general schema [13]. In this process semantic correspondence between members of schema is achieved by an interface for transmitting data. This interface should also solve the problem of inconsistency between schemas or should be able to convert data and queries among them. [12] Studies the merging problems and [10] deals with semantic correspondence between schemas. [9] Suggests an architecture which could consider complex correspondence as well. Considering inserted values in the fields, [8] provides a method for schemas' semantic correspondence which does not decide for semantic correspondence just because of correspondence of fields' names.

Other methods are being recognized as data integration systems. Examples of this method have been discussed at [15, 16, 17, 18]. Accessing to a data source in this method is achievable using a transparent search interface that helps the users to be unaware of complexity of searching process [13]. Presented method of this article lies in this category. Subsequently, accordance algorithms for making data integration between fields will be studied. However, before that extracting fields and tables from the database schema should be studied.

3. SCHEMA EXTRACTION

In this section we will study how to extract schema structure of legacy and modern databases. Like whatever presented at [19,20,21,13,22], numbers and names of tables, names of fields, data types of every field, length of each field, each fields' descriptions, existent constraint on tables, primary key of each field, fields' foreign key and indicating their relations are the least necessary characteristics for correspondence and making decision on queries conversion. For extracting such data, studying and recognizing schemas' structures of both databases are mandatory. Recognizing process is divided into two major parts: recognizing tables with their relations and recognizing fields. This action is performable by SQL commands.

After recognizing tables and fields of each database, a schema tree will be depicted according to achieved characteristics. An example of such a tree has been used at [14] for illustrating matching process between schemas. In this tree, root presented database and first level nodes presented tables. Tree's leafs also correspond to fields of every table. Using the gained schema tree, matching between fields' and tables' names will be analyzed. Section 4 describes algorithms of this matching.

4. MATCHING ALGORITHMS

We will study matching algorithms of tables' and fields' name at this section. Accordance topic introduces two points of view of similarity, structural and conceptual similarity [24]. In structural similarity, extent of similarity is recognized and characterized in accordance with applied words. The closer the applied words and letters are the more the degree of similarity would be. Degree of similarity in conceptual similarity is recognized in accordance with extent of existent of synonymous words. Conceptual similarity of two fields is studied and characterized by their existent data, thus the closer the existent data of two databases fields comparing together are, the more the degree of similarity would be.

Subsequently, two models of conceptual similarity algorithms, one used for numerical fields and the other for text fields, and structural similarity of their fields will be studied. Then, three kinds of popular algorithms and their obtained results will be discussed. Finally, a method has been presented at the final section of this paper which merges two methods of correlation analysis and similarity algorithms of string.

4.1. Algorithms of correlation analysis

Correlation analysis is used for the subjects of data mining [22]. This analysis could be used for specifying extent of fields' relation. The correlation between two attributes, A and B, can be evaluated by computing correlation coefficient. This formula presented on figure2 [22].

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \overline{A})(b_i - \overline{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i b_i) - N\overline{AB}}{N\sigma_A\sigma_B}$$

Figure2. Correlation Coefficient formula

In this formula N is the number of tuples, a_i and b_i are the respective values of A and B in tuple i, A and B are the respective average values of A and B, σ_A and σ_B are the respective standard deviations of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

The result for above-mentioned formula would be -1< $r_{A,B}$< +1 and could be analyzed as follows:

- If $r_{A,B}$>0 ($r_{A,B}$ larger than 0): A,B is positively correlated and the larger this amount is the more the correlation would be.

- If $r_{A,B}$=0 ($r_{A,B}$ equal 0): A,B are not dependent to each other and there is no correlation between them.

- If $r_{A,B}$<0 ($r_{A,B}$ is less than 0): A,B are negatively correlated and increasing one of them leads to decreasing the other one[22].

For instance 10 values tested on this algorithm for three fields of CITYCODE, MARK and AVERAGE. Values of the fields have been shown at table1 and the results have been illustrated at table2.

Table1. Example values for 3 fields

CICTYCODE	MARK	AVERAGE
100	16	16.5
101	17	17.75
100	17.5	18.5
200	19.75	19.33
202	12.25	19.80
113	14	14.1
200	17	17.55
119	19	19.1
124	13	12.4
118	16	15.1

Table2.Result of Correlation Coefficient Algorithm

	CITYCODE	MARK	AVERAGE
CITYCODE	1	-0.15	0.43
MARK	-0.15	1	0.5
AVERAGE	0.43	0.5	1

As you see at table2, the relation between CITYCODE and MARK fields that have different numerical ranges is **-0.15.** This matter shows the fact that these two fields' relation is too low and negatively correlated. For the two fields of MARK and AVERAGE with similar numerical ranges the value is **0.5** which demonstrate their close correlation. The value of one field in relation to itself, for instance about CITYCODE, is **1** which demonstrates the high correlation.

This evaluation demonstrates the point that extent of correlation extensively depends on values' ranges. If values' ranges are spread, the extent of correlation will be a small value. Number of samples assigning for evaluation also have determining role at the extent of correlation.

Therefore, in legacy and modern databases especially in the case that legacy database has many data records and modern database has little data record at its start up, this point could result in problems at correlation evaluation. Considering this fact, a solution has been presented at section 5 for this problem.

4.2. Modification of correlation analysis' algorithm

Instead of using values themselves, abundant ratios of them were used in accidental samples of records for solving this problem. Thus, it has prevented from the high dependency of some values to each other and a more comprehensive outcome has obtained. This algorithm has been shown in figure3.

```
1-Execute this query for first field

SELECT    Mark, Count
FROM      (SELECT     Mark, COUNT(Mark) AS
Count  FROM   Registration
GROUP BY Mark) AS derivedtbl_1

2- Execute same query for second field
3- Set 0 for values that does not exist in the other field
4- Execute Correlation algorithm same as figure2 for
values
```

Figure3.Correlation Algorithms of two text fields considering inserted values

Applying the algorithm of figure3, abundant ratio for the example of table1 is observable in table3.

Table3. abundant ratios

Values	f_i CityCode	Values	f_i Mark	f_i Average
100	5839	5	5	11
101	15	21	40	12
102	219	5	6	12.5
103	87	19	17	13
104	41	5	6	13.5
105	21	40	34	14
106	0	2	14.25	6
107	12	20	15.5	6
108	1	3	15.75	3
109	69	48	16	3
110	16	23	16.5	19
111	94	72	17	28
112	20	35	17.5	38
113	87	73	18	65
114	38	29	18.5	68
115	68	70	19	56
116	2	3	19.25	17
117	17	24	19.5	14
118	4	3	19.75	13
119	37	47	20	64

Applying the algorithm of correlation analysis, results of table 4 have been achieved for the data at table3. As you perceived these outcomes and the acquired relation for the values is much closer to reality.

Table4.Result of Correlation analysis' algorithm

	CITYCODE	MARK	AVERAGE
CITYCODE	1	-0.152	-0.153
MARK	-0.152	1	0.94
AVERAGE	-0.153	0.94	1

4.2. Algorithm of correlation analysis for text fields

Previous algorithm could not compare fields with text type. For example, the extent of correlation of two S_BIRTHLOC and NAME fields will be studied in this part. Number of records studied from each field is 100. Table5 shows the acquired results of executing figure 3's query for 100 records on S_BIRTHLOC field and table 6 shows the acquired results on S_NAME field.

Table5. Result of execution of figure3 of 100 records of S_BirthLoc field

S_BirthLoc	Count
ABADEH	1
ESFAHAN	86
TABRIZ	1
TEHRAN	5
DAMANE	1
SHAHREZA	2
SHAHREKORD	2
SHIRAZ	2

Table6.Some result of figure3 query execution of 100 record of S_Name field

S_Name	Count
EBRAHIM	3
EHSAN	8
AHMADREZA	5
ASGHAR	4
AFSHIN	2

AKRAM	2
ZAHRA	27
ALI	17
MOHAMADREZA	8
MEHDI	24

Studying values of tables 5 and 6 demonstrate that there are no common values between two fields. Thus, for getting correlation analysis, their correspondence value in other field is **0**. Using algorithm of correlation analysis will result in getting value of **0.16**. For further examples we will study the situation in which there are many common values for the fields. Consider two fields S_BIRTHLOC and S_EXPORTLOC which one is for keeping place of birth and the other is for keeping place of ID card's issue. Obviously names of cities have many common values. Table7 depicts achieved values of executing query of figure3.

Table7.Result of figure3 query execution on 100 records

Title	Values of S_BirthLoc	Values of S_ExportLoc
ABAHEH	1	1
ESFAHAN	83	73
TABRIZ	1	0
TEHRAN	5	9
DAMANE	1	0
SHAHREZA	2	2
SHAHREKORD	2	0
SHIRAZ	2	5
FELAVARJAN	0	1
AHVAZ	0	3
FARSAN	0	1
FEREYDON	1	2
MASJED SOLYMAN	2	3

0.9 is the acquired value from the algorithm of correlation analysis which demonstrates the high extent dependency of these two fields. This algorithm also has been applied for studying other fields its outcomes are accessible at table 8. Amplitude of obtained results is between 1 to -1 which shows the extent of positively or negatively correlation.

Table8.Analysis of correlation algorithm for text fields

Lagacy table/field name	Current table/field name	Result of correlation algorithm
Lessons	Course	1
BirthPlace	BirthLoc	0.9
LastName	Family	0.6
FirstName	Name	0.9
CUST_ID	Cust_id	1
StudentName	STDName	0.9
IDNo	IdentCode	-0.6
Product_Id	User_Status	-1
Name	Address	-0.6
Phone	Tel	-0.7

As it is noticeable using table 8, correlation value is acquired **1** (high correlation) for the two fields LESSONS and COURSE which was the expected extent of correlation in two databases due to data values for these two fields. Extent of correlation for the two fields TEL and PHONE is **-0.7** and this extent is acceptable and expectable because of changes in amplitude of numbers available in both databases. Some of the reasons which could make such differences at the extent of fields' correlation in databases are listed as follow:

-Change in numbers' amplitude: for example, phone numbers in legacy databases consist of 5 digits and in new databases consist of 6 digits.

-Variety in acronyms of applied words for names: for example, three letter acronyms have been used for showing cities' names in legacy databases (ISF, THE, SHI), though their full names have been used in new databases.

-Values are unique and there are no similar values in legacy and modern databases or they are very rare. For example, national code is a kind of data which its repetition in legacy and new databases occurs only in the case of further requirement to that value.

Considering these conditions, the low extent of correlation between some fields is justifiable. Later in section 5, we will study the algorithms which work based on structure similarity.

5. SIMILARITY ALGORITHMS OF STRING

Using similarity algorithms of string is another way to characterizing extent of similarity and correlation of tables and fields [19]. In this method, algorithms of string comparison are being analyzed that can characterize their similarity by comparing string of words. Later similarity algorithms of string will be studied by an example.

5.1 Name Similarity Algorithm Based On Edit Distance

Another name for Edit Distance algorithm is based on the name of Russian scientist VLADIMIR LEVENSHTEIN who invented it in 1965.

This algorithm acts in a way that if two words TEST and TEST compared together, it will return LD=0. Thus, nor a change neither a modification is necessary for assimilation of these two words. The value LD=1 is acquired while two words TENT and TEST compared together which demonstrate existence of one distinguished letter. Assimilation between capital and small letters of words must be done before using this algorithm. Otherwise, capital and small letters will not be recognized as the same.

The obtained result of Edit distance algorithm will be converted to a comparable value, using figure 4's formula which is being named "named similarity algorithm".

$$\mathrm{NaSimLD}(\mathrm{s},\mathrm{t}) = 1 - \frac{\mathrm{LD}(\mathrm{s},\mathrm{t})}{\mathrm{MaxLength}(\mathrm{s},\mathrm{t})}$$

Figure4. Name Similarity algorithm based on Edit Distance

In this formula S is the source string, T is the target string and LD(s,t) is the acquired value from edit distance algorithm.

Using proposed implementation and its test and analysis, results of table9 will be attained.

Table9. Some sample for name Similarity algorithm based on Edit Distance

Legacy table/field name	Current table/field name	Name similarity algorithm
Lessons	Course	0
BirthPlace	BirthLoc	0.7
LastName	Family	0.125
FirstName	Name	0.44
CUST_ID	Cust_id	1
StudentName	STDName	0.6
IDNo	IdentCode	0.4
Product_Id	User_Status	0.09
Name	Address	0.14
Phone	Tel	0

As it also has been mentioned at [19], applying this algorithm won't lead to appropriate consequence in finding similar words. Since for the two words LESSONS and COURSE that have very closed meaning and concept, as you see at table9, acquired value by this algorithm is 0; it means they have no relation to each other. Hence, this algorithm is not useful lonely and other algorithms must be studied too.

5.2. Name Similarity Algorithm Based On Humming distance

Numbers of different characters in two words with the same length are acquired by humming distance. In other words it is the number of characters needed to be changed (inserted, omitted or modified) for creating second string in the first string. Using humming distance algorithm which could be seen on figure5, one could calculate the extent of equality and similarity of two strings.

$$NaSimHD(m,o) =$$

$$1 - \frac{(\sum_{i=1}^{MinLength(m,o)} f(i)) + |m.length() - o.length()|}{MaxLength(m,o)}$$

That:

$$f(i) \begin{cases} 0 & m[i] = o[i] \\ 1 & otherwise \end{cases}$$

Figure5. name Similarity algorithm based on Humming distance

By implementing it and comparing sample strings, results of table10 have been found.

Table10. some Sample for name Similarity algorithm based on Humming distance

Legacy table/field name	Current table/field name	Humming distance algorithm
Lessons	Course	0
BirthPlace	BirthLoc	0.5
LastName	Family	0.125
FirstName	Name	0
CUST_ID	Cust_id	1
StudentName	STDName	0.18
IDNo	IdentCode	0.2
Product_Id	User_Status	0.09
Name	Address	0
Phone	Tel	0

5.3. Name Similarity Algorithm Based On LCS

LCS could acquire the largest common substring between two strings. Number of common letters between two strings could be found by this algorithm. The Formula for calculating extent of name similarity by LCS algorithm is like figure6.

$$NaSimLCS(m,o) = \frac{LCS(m,o)}{MaxLength(m,o)}$$

Figure6. name Similarity algorithm based on LCS algorithm

Acquired results for sample strings from this algorithm are observable at table11.

Table11. some sample for name Similarity algorithm based on LCS

Legacy table/field name	Current table/field name	LCS algorithm
Lessons	Course	0.29
BirthPlace	BirthLoc	0.7
LastName	Family	0.25
FirstName	Name	0.4
CUST_ID	Cust_id	1
StudentName	STDName	0.63
IDNo	IdentCode	0.4
Product_Id	User_Status	0.18
Name	Address	0.29
Phone	Tel	0.2

5.4. Incorporating name similarity algorithm

Three groups of algorithms work on name of fields were studied in three previous sections. These three algorithms will be incorporated at this section and a unit result will get acquired from them. This incorporation is just like the method of [19]. Provided that acquired results from one of these algorithms is larger than 0.9 in this corporation (shown in the following formula), ultimate result will be 1; while the acquired result from each algorithm is less than 0.2, ultimate result would be 0, and otherwise their average is the applied method for getting result. This incorporation has been shown on figure 7.

$$NaSim(m,o) = \begin{cases} 1.0 & (one\ result\ of\ algorithms \geq 0.9 \\ \frac{NaSimLD(m,o) + NaSimHD(m,o) + NaSimLCS(m,o) + Correlation\ (m,o)}{3} & \\ 0.0 & (all\ result\ of\ algorithms \leq 0.2 \end{cases}$$

Figure7. Incorporation of string algorithm

Compared results of three algorithms and ultimate results are observable at table 12.

Table12. Compared result of algorithm

Legacy table/field name	Current table/field name	LD	HD	LCS	Incorporation of 3 algorithm	artificial judgment
Lessons	Course	0	0	0.29	0.09	1
BirthPlace	BirthLoc	0.7	0.5	0.7	0.6	1
LastName	Family	0.125	0.125	0.25	0.1	1
FirstName	Name	0.44	0	0.4	0.2	1
CUST_ID	Cust_id	1	1	1	1	1
StudentName	STDName	0.6	0.18	0.63	0.47	1
IDNo	IdentCode	0.4	0.2	0.4	0.33	1
Product_Id	User_Status	0.09	0.09	0.18	0	0
Name	Address	0.14	0	0.29	0.14	0
Phone	Tel	0	0	0.2	0	1

Comparable diagram of acquired results from tables12 could be observed on figure8.

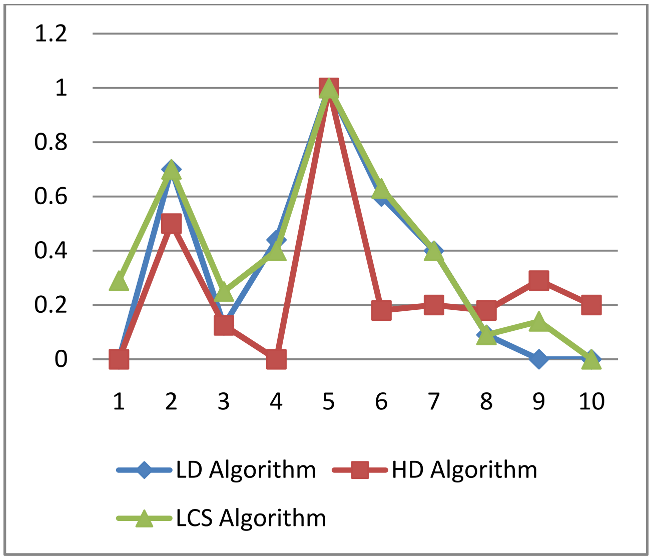

Figure8. Compared result of three algorithm

Comparable diagram of incorporating similarity algorithm of string and the expectable results are observable in figure 9.

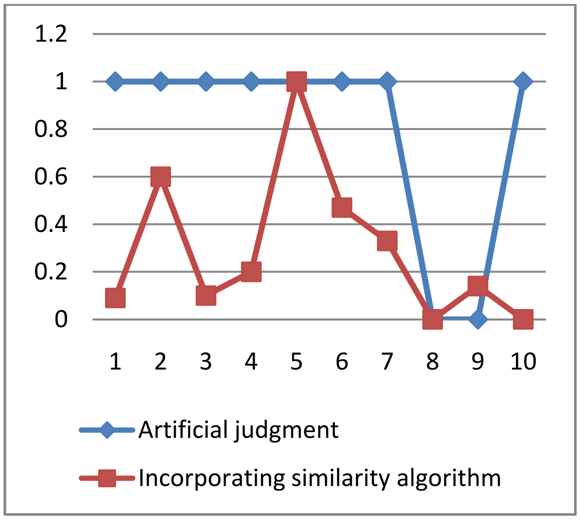

Figure9. Comparable diagram

6. ULTIMATE ALGORITHM

Two groups of similarity algorithms were studied in the previous sections which one of them worked on name of fields and the other assessed similarity according to existent values in fields. Acquiring an ultimate and unit algorithm, we will use the maximum of these two groups of algorithms. The ultimate result could be observed at table13.

Table13. Final Algorithm

Legacy table/field name	Current table/field name	similarity algorithms of string	Correlation analysis	Final Algorithm (MAX)	artificial judgment
Lessons	Course	0.09	1	1	1
BirthPlace	BirthLoc	0.6	0.9	0.9	1
LastName	Family	0.1	0.6	0.6	1
FirstName	Name	0.2	0.9	0.9	1
CUST_ID	Cust_id	1	1	1	1
StudentNam e	STDNam e	0.47	0.9	0.9	1
IDNo	IdentCode	0.33	-0.6	0.33	1
Product_Id	User_Stat us	0	-1	0	0
Name	Address	0.14	-0.6	0.14	0
Phone	Tel	0	-0.7	0	1

You could see the Compared ultimate algorithm and the the expectable results are observable in figure 10.

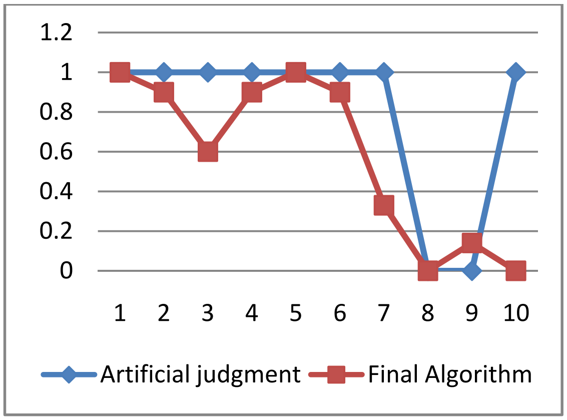

Figure10. Compared Final algorithm and Artificial judgment

Some cases are in distance from the expected results in this diagram each of which has an acceptable reason and justification. These differences will be studied at below.

Case 7 (IDNO, IDENTCODE): due to change of ID number in legacy database to identification code in modern database, no similar information were found and consequently the acquired extent from the algorithms of correlation analysis was low.

Case 10 (TELL, PHONE): the reason of diversity of this result was similar to that of mentioned reason for case 7. In this case, 6 digits phone numbers of legacy database has changed to 7 digits numbers.

7. CONCLUSION AND FUTURE WORKS

This article studied methods of acquiring fields' and tables' names automatically and proved them by presenting appropriate algorithms and studying them on the real data of legacy and new databases. And different algorithms and methods were applied and studied. Consequent results from executing these algorithms were compared to each other and ultimately obtained an appropriate algorithm from their incorporation. This newly obtained algorithm, then, was executed on real information of legacy and new databases and approved the acquired result.

Semantic relationship between tables and fields has been characterized by using this algorithm and the performed accordance has been presented to user for the final control. Final Confirming by user will insert these results into META DATA database as the result of schema structure matching .

The future work includes: the accuracy and generality of similarity algorithm can be further improved and to find a solution for fields that have various range of data but are similar.

REFRENCES:

[1] I. Sommerville, “Software Engineering”, 6th edition ,*ADDISON WESLEY* ,2000 ,chapter26

[2] M. Tamer Ozsu and P. Valduriez, “Distributed DBMS Architecture”, *Principles of Distributed Database System*, Prentice-Hall, 1991.

[3] H. SooKim and J. M. Bieman, “Migrating Legacy Software Systems to CORBA based Distributed Environments through an Automatic Wrapper Generation Technique”, *Korea Science & Engineering Foundation*

[4] S. Comella-Dora, K.Wallnau, R. C. Seacord, J.Robert, “A Survey of Legacy System Modernization Approaches”, *CMU/SEI-2000-TN-003*, April 2000

[5] Memon Asim, “Reverse Engineering” , *Blekinge Institute of Technology,School of Engineering,Department of Computer Science*, Campus SoftCentre, Ronneby,Sweden,

[6]M. AbasiFard, “Database Reverse Engineering in Legacy Systems”, *Tehran University,Iran*, 2005

[7] J. Bisbal, D. Lawless, B. Wu and J. Grimson, “Legacy Information Systems: Issues and Directions”, *IEEE Software*, 1999

[8] W. Wu, A. Doan and C. Yu, ”WebIQ:Learning from the web to match Deep-web Query Interfaces” , *ICDE Conference*, 2006

[9] R. Dhamankar,Y. Lee, A. Doan, A. Halevy and P. Domingos ,” Discovering Complex Semantic Matches between Database Schemas”, *SIGMOD,* 2004

[10] E. Rahm and P. Bernstein,”On matching schemas automatically”, *VLDB Journal*, 2001

[11] C. Batini and M. Lenzerini and S. Navathe,” A comparative analysis of methodologies for database schema integration”, *ACM computing Survey*, 1986

[12] R. A. Pottinger and P. A. Bernstein, “Merging models based on given correspondences”, *Int. Conf. on Very Large Database (VLDB)* , 2003

[13] A. Doan and A. Halevy, “Semantic Integration Research in the Database Community: A Brief Survey” ,*American Association for Artificial Intelligence*, 2004.

[14] Maseud Rahgozar and Farhad Oroumchian, “effective Strategy for legacy systems' evolution”, Journal of Software Maintenance: Research and Practice,V15, 2003, pp325-344

[15] M. Friedman , S. Weld Daniel, “Efficiently Executing Information-Gathering Plans” , *IjCAI*, 1997

[16] C. A. Knoblock , et al, “Modeling Web Source for Information Integration”, *American Association for Artificial Intelligence*, 1997

[17] E. Lambrecht and S. Kambhampati, “Optimizing Recursive Information Gathering Plans”, *Int. Joint Conf. on AI (IJCAI),* 1999

[18] A. Y. Levy,A. Rajaraman, J. Ordille, “Quering hetrogenous information source using source description”, *VLDB*, 1996

[19] ZHANG Lei, YANG Xiaoying, “An Approach to Semantic Annotation for Metadata in Relational Databases”, *International Symposiums on Information Processing*, 2008

[20] Gregory T. Buehrer, Bruce W. Weide, and Paolo A. G. Sivilotti, “Using Parse Tree Validation to Prevent SQL Injection Attacks” , *ACM 1-59593-204*,2009

[21] M. R. Abbasifard, M. Rahgozar, A. Bayati and P. Pournemati, “Using Automated Database Reverse Engineering for Database Integration” , *proceedings of world academy of science, engineering and technology* ,V13, MAY 2006

[22] J. Han, M. Kamber,” Data Mining” ,second edition, *Morgarb Kaufmann,*2006, pp 67-70

[23]E. Dragut, W. Wu, P. Sistla, C. Yu, and W. Meng, "Merging Source Query Interfaces on Web Databases", *22nd International Conference on Data Engineering (ICDE'06),* pp.679-690, Atlanta, Georgia, April 2006.

[24]M. Vargas-Vera, E. Motta, “AQUA - Ontology-based Question Answering System”. *In Proceedings of Third International Mexican Conference on Artificial Intelligence (MICAI-2004)*, Mexico City, Mexico.