Analysis of String Comparison Methods During De-Duplication Process

Maria del Pilar Angeles, Francisco Javier García-Ugalde,
Ricardo Valencia, Arturo Nava
Facultad de Ingeniería
Universidad Nacional Autónoma de México
México, D.F.
Email: pilarang@unam.mx,fgarciau@unam.mx,
ricardofdk8@hotmail.com, arturoshox@hotmail.com

***Abstract*—This paper presents three comparison algorithms in terms of computational resources utilized during record linkage process. The comparison algorithms are Monge-Elkan, Bag Distance and Edit Distance. The Monge-Elkan method meets all the requirements to be implemented and to obtain reliable results characteristics. Besides, the method falls within the average execution time efficiency.**

Keywords–data matching; de-duplication; record linkage.

I. INTRODUCTION

During the integration process of a number of heterogeneous databases, the identification and resolution of extensional inconsistencies is one of the main problems to deal with [1]. The data matching process, is focused on joining records from disparated data sources describing the same real world entity. This process requires data standardization, indexing of records for reducing the number of records comparison, field and record comparison, the identification of matched records, not matched records and possible matched records, and finally the evaluation of classification of records. Therefore, the data matching process grows exponentially as the databases to be matched get larger. In real-world data matching applications, the true status of two records that are matched across two databases is not known. Thus, accurately assessing data matching quality and completeness is challenging [2].

We focused on the integration of the Freely Extensible Biomedical Record Linkage prototype (FEBRL) system [3] to any database from any Database Management System (DBMS) by querying the native data dictionary; the research proposal is aimed to the enhancement and addition of further standardization, indexing, and classification algorithms for data matching. We have called our prototype Universal Evaluation System Data Quality (SEUCAD), which nowadays supports six DBMS. The present work is related to the open issues on the comparison of algorithms that reduce the quadratic complexity of the naive process of pair-wise comparing each record from one database with all records in the other database.

The present paper is organized as follows: The next section briefly explains the data matching process. Sections III, IV and V explain the string comparison functions Monge-Elkan, Bag-Distance and Levenshtein distance respectively, along with their role within de process of data matching. Section VI presents the experiments carried out. Section VII analyses the results. Finally, the last section concludes the main topics achieved regarding the performance of the comparison functions and the future work to be done.

II. RELATED WORK

The data matching process is mainly concerned with the record comparison among databases in order to determine if a pair of records corresponds to the same entity or not. This process, in general terms, consists of the following tasks:

- A standardization process [4], which refers to the conversion of input data from multiple databases into a format that allows correct and efficient record correspondence between two data sources.
- The indexing process aims to reduce those pairs of records that are unlikely to correspond to the same real world entity and retaining those records that probably would correspond in the same block for comparison; consequently, reducing the number of record comparisons. The record similarity depends on their data types because they can be phonetically, numerically or textually similar. Some of the methods implemented within FEBRL are for instance, Soundex [5], Phonex [2], Phonix [2], the New York State Identification and Intelligence System Phonetic Code (NYSIIS) [6], Double metaphone [7], QGrams.
- Field and record comparison methods provide degrees of similarity and define thresholds depending on their semantics or data types. In the prototype, the algorithms Qgram, Jaro - Winkler Distance [8] [9], Longest common substring comparison are already implemented.
- The classification of pairs of records grouped and compared during previous steps is mainly based on the similarity values were already obtained, since it is assumed that the more similar two records are, there is more probability that these records belong to the same entity of the real world. The records are classified into matches, not matches or possible matches, the classification of records can be unsupervised or supervised.

The unsupervised process classifies pairs or groups of records based on the similarities between them without having access to more information about the characteristics of those records. The supervised process requires training based on data identified as similar or not similar. In this case, comparison vectors with an associated value that determines whether records correspond or not are required. In the case of potentially corresponding records, the duplicates detection may be performed manually. Within FEBRL, there are methods based

 ISBN: 978-1-61208-408-4

on thresholds, probabilistic methods, costs based methods or rule-based methods; The evaluation of data matching refers to how many of the classified matches correspond to true real-world entities, and how many of the real-world entities that appear in both databases were correctly matched [10]. The present research assessed the algorithms in terms of computational resources which are detailed in Section VIII.

III. THE MONGE-ELKAN ALGORITHM

Monge and Elkan proposed in [11] a simple but effective method for measuring the similarity between two strings containing multiple tokens, using an internal similarity sim (a, b) capable of measuring the similarity between two individual tokens a and b. Given two texts A, B being their respective number of tokens $|A|$ and $|B|$, the Monge-Elkan algorithm measures the average of the similarity values between pairs of more similar tokens within texts A and B. The Monge-Elkan similarity formula is as follows:

$$simMongeElkan(A,B) = \frac{1}{|A|}\sum_{i=1}^{|A|} max\{sim'(A_i,B_j)\}_{i=1}^{|B|} \quad (1)$$

In order to compare two strings, the Monge Elkan algorithm requires a similarity function; we will utilize the Jaro similarity function, which is detailed as follows: The Jaro similarity function was developed by Matthew Jaro in [8], this function was designed specifically for comparing short length strings, such as names, and is given by the following formula:

$$simjaro(s1,s2) = \frac{1}{3}(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c}) \quad (2)$$

The Jaro similarity function counts the number of characters that match, where c is the number of coincident characters and t is half the number of transpositions (two adyacent characters that are interchanged en both strings, such as 'pe' and 'ep'). For instance, considering two strings S1 = 'mario alfonso' and S2 = 'Marian alonso'. Applying the Jaro similarity function, the results are as follows: Jaro('alfonso','Marian') = 0.6190; Jaro('alfonso','Alonso') = 0.9523; Jaro('mario', 'Marian') = 0.9047; Jaro('mario','Alonso') = 0.5777. Given the similarity between tokens, the two best matches are selected for computing the Monge-Elkan comparison. Considering the Monge-Elkan formula given in (1), $|A| = 2$ as the number of tokens s1, $|B| = 2$ as the number of tokens of s2, and the Jaro similarity function already calculated Sim(0.9523,0.9047). Therefore, the Monge-Elkan comparison is been given by:

$$monge - elkan(s1,s2) = 1/2(0.95 + 0.90) = 0.928 \quad (3)$$

IV. THE BAG DISTANCE ALGORITHM

This method obtains the non-common characters between two strings s1 and s2. The characters of s1 and s2 are divided and ordered to obtain two charsets X and Y, and theri corresponding differences X-Y and Y-X. The largest difference between s1 as s2 is computed by:

$$bagdistance(s1,s2) = max(|x-y|,|y-x|) \quad (4)$$

The bag distance similarity function is given by the following formula:

$$simbag(s1,s2) = 1 - (\frac{bagdistance(S1,S2)}{max(|S1|,|S2|)}) \quad (5)$$

For instance, let be s1 = maria; s2 = mariano, with $|s1| = 5$, $|s2| = 7$, then X = a,a,i,m,r and Y = a,a,i,m,n,o,r. Therefore, $|x-y| = |\{\varnothing\}| = 0$, $|y-x| = |\{n,o\}| = 2$. The difference is identified as $bagdistance(s1,s2) = 1.0 - (\frac{2}{7}) = 0.71$

V. LEVENSHTEIN DISTANCE ALGORITHM

The Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. In order to compare two strings s1 and s2 with lengths $|s1|$ and $|s2|$ requires a matrix of length $|s1|+1, |s2+1|$. The matrix is filled according to the number of deletion, insertion and substitution operations required. Consequently, the levenstain distance between s1 and s2 is obtained as the value of the lower left cell. The Levenshtein similarity is computed by the following formula:

$$levensthein(s1,s2) = 1 - (\frac{levenshteindist(S1,S2)}{max(|S1|,|S2|)}) \quad (6)$$

where the numerator is the value of the lower right corner and the denominator the greater length between strings s1 and s2. For instance, let be strings s1 = raul and s2 = ramon. The intersection of (ra-r) has a value of 1 because it is required to perform one operation to reach the same state. In the case of different characters, a value of 1 indicates, since it is a replacement operation. An example of the Levensteing Matrix when comparing the strings "ramon" and "raul" are shown in Table I In our case, the formula is replaced in the

TABLE I. LEVENSTEING MATRIX

		r	a	m	o	n
	r	0	1	2	3	4
	a	1	0	1	2	3
	u	2	1	1	2	3
	l	3	2	2	2	3

following manner: $levensthein(s1,s2) = 1 - (\frac{3)}{5} = 0.4)$ The Comparative analysis of the Monge-Elkan, Bag Distance and Edit distance methods is presented in the following section.

VI. ANALYSIS OF THE MONGE-ELKAN METHOD WITH BAG-DISTANCE AND EDIT-DISTANCE

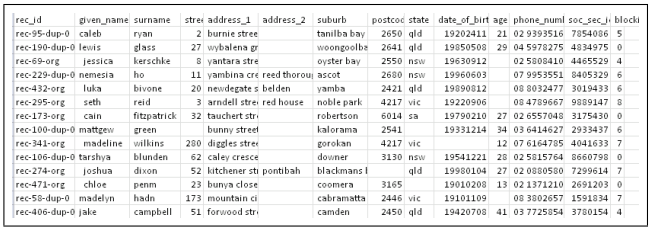
This section shows a comparative analysis of the Monge-Elkan, Bag-Distance and Edit Distance methods in terms of pairs records comparison time, memory used, etc. We have tested the three already mentioned methods with a number of input files generated by the SEUCAD prototype, one of them will be presented as follows:

A. Exploring the file

The de-duplication process initiates with the exploration of the input file. The file called data1.csv was generated with a total length of 1000 records, where 500 records were original and 500 duplicated records.

 ISBN: 978-1-61208-408-4

From statistics we can analyse the file in terms of the number of unique data values, most frequent field values, minimun and maximum value lenghts. Such information allow us to identify which fields are suitable for indexing or the method to apply that corresponds to the data type and length the data source contains. The structure of the data file is presented in Figure 1.



rec_id	given_name	surname	stree	address_1	address_2	suburb	postcod	state	date_of_birt	age	phone_numl	soc_sec_i	blocki
rec-95-dup-0	caleb	ryan	2	burnie stree		tanilba bay	2650	qld	19202411	21	02 9393516	7854086	5
rec-190-dup-0	lewis	glass	27	wybalena gr		woongoolba	2641	qld	19850508	29	04 5978275	4834975	0
rec-69-org	jessica	kerschke	8	yantara stre		oyster bay	2550	nsw	19630912		02 5808410	4465529	4
rec-229-dup-0	nemesia	ho	11	yambina cre	reed thorou	ascot	2680	nsw	19960603		07 9953551	8405329	6
rec-432-org	luka	bivone	20	newdegate s	belden	yamba	2421	qld	19890812		08 8032477	3019433	6
rec-295-org	seth	reid	3	arndell stre	red house	noble park	4217	vic	19220906		08 4789667	9889147	8
rec-173-org	cain	fitzpatrick	32	tauchert str		robertson	6014	sa	19790210	27	02 6557048	3175430	0
rec-100-dup-0	mattgew	green		bunny street		kalorama	2541		19331214	34	03 6414627	2933437	6
rec-341-org	madeline	wilkins	280	diggles stre		gorokan	4217	vic		12	07 6164785	4041633	7
rec-106-dup-0	tarshya	blunden	62	caley cresce		downer	3130	nsw	19541221	28	02 5815764	8660798	0
rec-274-org	joshua	dixon	52	kitchener sti	pontibah	blackmans l		qld	19980104	27	02 0880580	7299614	7
rec-471-org	chloe	penm	23	bunya close		coomera	3165		19010208	13	02 1371210	2691203	0
rec-58-dup-0	madelyn	hadn	173	mountain ci		cabramatta	2446	vic	19101109		08 3002657	1591834	7
rec-406-dup-0	jake	campbell	51	forwood str		camden	2450	qld	19420708	41	03 7725854	3780154	4

Figure 1. File structure

B. Selecting the indexing method

- Indexing method: The QGramIndex method was selected, with a length of parameter Q of 2, to generate bigrams.
- Threshold: This parameter value should be set in the range of 0.0 and 1.0, if the value is 1.0 only records that have the same definition of indexing will be compared. Therefore, in order to establish a more accurate indexing and comparison of records, the threshold value we have defined for this experiment was 0.75.
- Padded: As this parameter sets whether the input strings are set to (q-1) grams of characters or not. It was enabled for a better analysis and more accurate the indexing.
- Blocking key: We have chosen the field ”surname”. We did not set a maximum number of characters for the definition of indexing, otherwise large values will be truncated. As the fields to be compared contain more than one word.
- Sort words: This option was not enable in order to avoid the division and ordering of each word.
- Reverse: The reverse parameter was disabled because otherwise the input string will be reversed and in the case of surname field would not be a representative indexing definition.
- Encoding function: The encoding function selected was ”Soundex”. As we required the whole word to be encoded, this value was not set.

The previous configuration for indexing is presented in Figure 2.

C. Selecting the comparison method

The following parameters correspond to the comparison method. As we focused on the execution of three comparison methods, the parameters are set equally in the three of them.

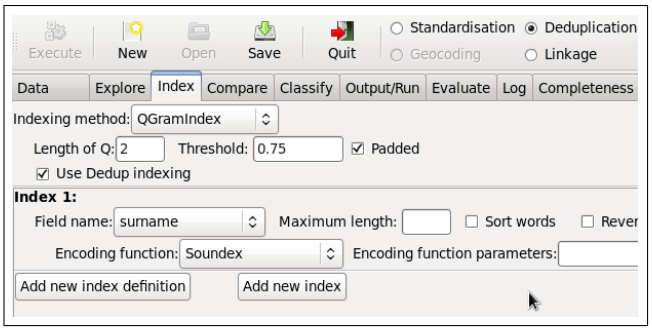


Figure 2. Indexing configuration

- Field name A: The name of the first field of comparison. Since the given name is a relevant field to identify people, and the comparison field is recommended to be of datatype String, the field ”given_name” was selected as the first option for comparison purposes.
- Field name B: The name of the second field of comparison. Once again the field ”given_name” was selected in order to compare those records according to the given name.
- Cache comparisons: Indicate whether the calculation of the similarity of values can not take place on memory. It is recommended when data values are large, complex, or there is limited number of fields. As ”given_name” field is not complex nor large, the option of ”Cache comparisons” will be disabled. Thus the calculations will not be performed in memory.
- Maximum cache size: This value is limited to a certain number of pairs of fields. If not selected (None), then all the comparisons will be made. Since it is desired that the comparisons of all pairs of fields are made, the option to ”Maximum cache size” will default to ”None”.
- Missing value weight: The value to be given in the event that one or both fields have no value. Its default value is 0.0 and its value must be within the range of ”value Disagreeing weight” and ”weight Agreeing value.” For comparison operations are more accurate when one or both fields have no value, the value of ”Missing value weight” will be ”0.0”.
- Agreeing value weight: The value to be given when the similarity is entirely accurate. By default the value is 1.0.
- Disagreeing value weight: The value to be given when the similarity is entirely different. By default the value is 0.0. This value must be less than ”Agreeing value weight”. Like the previous value, the value of ”Disagreeing value weight” as ”0.0” will be defined when the similarity of two strings is totally different.
- Threshold: This value should be set in the range of 0.0 and 1.0, and will determine a better level of accuracy. If the calculation of approximate similarity method is higher than indicated in this field (threshold), then the similarity value will be calculated. If the approximate

 ISBN: 978-1-61208-408-4

similarity is less than that indicated in this field (threshold), then the similarity value will correspond to the ”Disagreeing value weight” parameter.
The selected attributes per comparison method are shown in Table II.

TABLE II. CONFIGURATION MATRIX

Test	*Process*	*Method*
1	Comparison	Monge-Elkan (given_name)
2	Comparison	Bag-Distance (given_name)
3	Comparison	Edit-distance (given_name)
	Coding	soundex(surname)

D. Selecting the method of classification

This section is aimed to present the configuration parameters established for the execution of the classification method.

- Weight vector classification method: The Fellegi and Sunter method will be selected since it has been broadly used and tested.
- Lower threshold: The lowest threshold value to be considered for the classification of records. Since only the comparison of ”given_name” field, place the sum of similarities remain in the range of 0.0 to 1.0. Thus, an acceptable and considered for the lower threshold value is ”0.5”.
- Upper threshold: The higher threshold value to be considered for the classification of records. Since only the comparison of ”given_name” field, place the sum of similarities remain in the range of 0.0 to 1.0. Thus, an acceptable and considered for the higher threshold value is ”0.98”.

Therefore the values of the minor similarities to 0.5 will be classified as ”Non match”, which are greater than 0.9 will be classified as ”Match” and remaining in the range of 0.5 and 0.9, will be classified as ”Potential Match”. Figure 3 shows the classification settings.

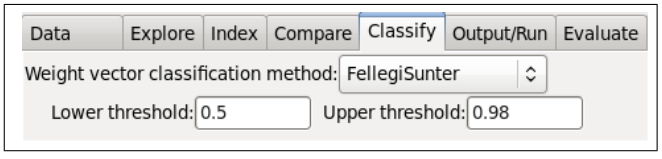


Figure 3. Configuration settings for classification of records.

E. Select the output characteristics and execution

The SEUCAD prototype is able to store the configuration selected for the de-duplication process in a python file. Figure 4 shows the output execution settings. The prototype allows the user to save the weight vector, histogram, match status and match data files.

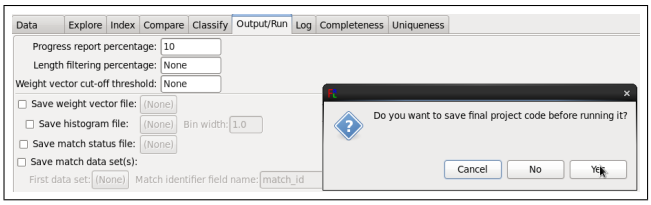


Figure 4. Configuration settings for execution of de-duplication process.

VII. RESULTS

The results of the data matching process are presented in this section. The test file named data1.csv contained 1000 total number of records, considering rec_id as the record identifier, given_name as the fields of comparison, surname as indexing field for Qgram as an indexing method. The number of pairs of records compared were of 13598 and Fellegi and Sunter as a classifier method with lower threshold of 0.5 and an upper threshold of 0.98. The use of memory and time resources per comparison method are shown in Table III.

TABLE III. TIME AND MEMORY UTILIZED PER METHOD

Comparison method	*Total time*	*avg time per pair*	*total memory*	*resident memory*
[1ex] Monge-Elkan	1.64s	0.12ms	11264KB	7.72MB
Bag-Distance	1.22s	0.09ms	11264KB	7.968MB
Edit-Distance	2.38s	0.18ms	11264KB	7.751MB

The quality metrics per method are shown in Table IV.

TABLE IV. QUALITY METRICS PER METHOD

Comparison method	Matches vectors	True positives	no matches	possible matches
Monge-Elkan	474	474	8036	5088
Bag-Distance	437	415	11518	1643
Edit-Distance	432	413	12875	291

A. Monge-Elkan

In the case of Monge-Elkan, the total time taken for the comparison process was 1.64 seconds with a comparison average time per pair of records of 0.12 milliseconds. The total memory usage was of 11264 Kbytes, 7.71875 MB of resident memory, 474 vectors classified as ”match”, and 474 true positives, 8036 vectors classified as ”no match” and 5088 vectors classified as ”possible-match”: 5088. The amount of weight partial acceptance value of 0.0 was of 13166 and the amount of weight partial acceptance of 1.0 were of 432. Figure 5 shows the outcome results considering Monge-Elkan as comparison method.

B. BagDistance

In the case of BagDistance, the total time taken for the comparison process was of 1.22 seconds with a comparison average time per pair of records of 0.09 milliseconds. The total memory usage was of 11264 KB and 7968 of resident memory, 437 vectors classified as ”match”, and 415 true positives, 11518 vectors classified as ”no match”, and 1643 vectors classified as ”possible-match”. The amount of weight

 ISBN: 978-1-61208-408-4

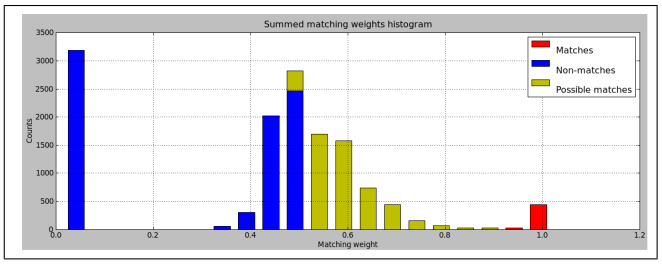


Figure 5. Classification of records with Monge-Elkan as comparison method.

partial acceptance value of 0.0 was of 11518 and the amount of weight partial acceptance of 1.0 were 437. Figure 6 shows the outcome results with Bag-Distance as comparison method.

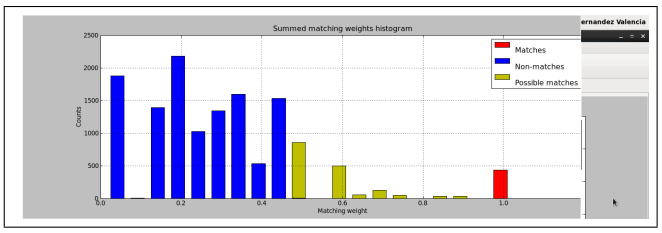


Figure 6. Classification of records with Bag-Distance as comparison method

C. Edit-Distance

In the case of Edit-Distance, the total time taken for the comparison process was of 2.38 seconds with a comparison average time per pair of records of 0.18 milliseconds. The total memory usage was of 11264 KB and 7.7578125 MB of resident memory resident, 432 vectors classified as ”match”, , and 413 true positives, 12875 vectors classified as ”no match”, and 291 vectors classified as ”possible-match”. The amount of weight partial acceptance value of 0.0 was of 13166 and the amount of weight partial acceptance of 1.0 were 432. Figure 2 shows the results. Figure 7 shows the outcome results with Edit-Distance as comparison method.

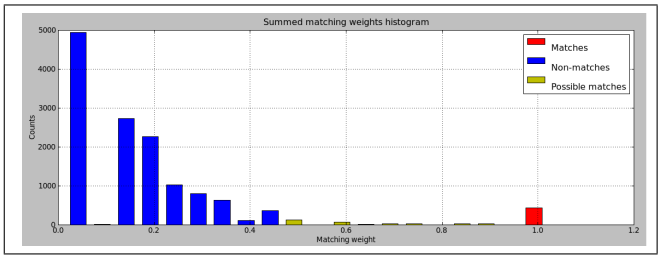


Figure 7. Classification of records with Edit-Distance as comparison method

According to the experiment results, we can make some conclusions about the resources used by Monge-Elkan, Edit Distance and Bag Distance methods. Considering the execution time, the Bag-Distance method is a comparison that performs operations quickly as is shown in Figure 8. Regarding the amount of memory used by the Monge-Elkan method, it has

Figure 8. Pair records comparison time (seconds)

been in the same range as in the rest of the comparison methods. There is a small variation when the number of records and comparisons increases, as the Monge-Elkan method is more efficient by using less memory, as is shown in Figure 9. After performing the relevant comparisons, the classification of

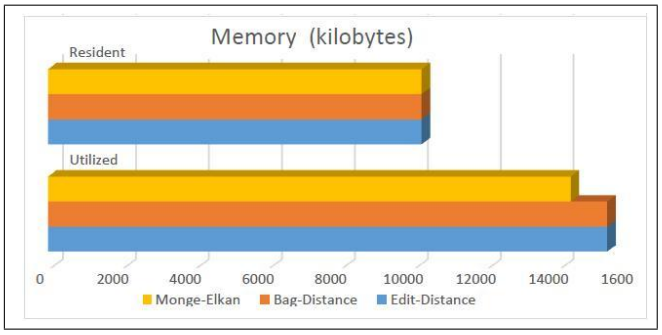


Figure 9. Used Memory and Resident Memory

data has been carried out. Thus, the given records classified as ”match” we can observe that comparisons of methods Bag-distance, Edit-Distance and Monge-Elkan result in similar classification. As for the records classified as ”non-match” and ”possible-match”, it can be concluded that there are a large number of variations in the results, as each algorithm comparison is based on different characteristics, such as: string length, number of similar tokens, etc. Finally, measuring the number of equivalent weights of 0.0 and 1.0 acceptance were practically the same, only small variations were 3 to 4 records. Thus, the level of certainty of the new method of comparison is very similar to those already implemented methods. The Monge-Elkan method meets all the requirements to be implemented and to obtain reliable results characteristics. Besides, the method falls within the average execution time efficiency.

VIII. Conclusion and Future Work

The Monge-Elkan method meets all the requirements to be implemented and to obtain reliable results characteristics. Besides, the method falls within the average execution time efficiency. However, there was no consideration of the quality

 ISBN: 978-1-61208-408-4

of classification, we have utilized synthetic test data sets from which we are able to obtain the number of candidate record pairs generated, or the measures of reduction ratio, pairs quality, and pairs completeness. The corresponding measurements allow to compute the effectiveness of the data matching system, which is a part of our future work.

As we have used synthetic test data sets to evaluate the comparison methods, then it is important to be aware of the limitations of such data, and the results achieved with them should not be generalised, because the performance of a method is dependent on the type and characteristics of the data that are matched.

ACKNOWLEDGMENT

This work is being supported by a grant from Research Projects and Technology Innovation Support Program (Programa de Apoyo a Proyectos de Investigacin e Innovacin Tecnolgica, PAPIIT, UNAM Project IN114413 named Universal Evaluation System Data Quality (Sistema Evaluador Universal de Calidad de Datos).

REFERENCES

[1] A. Motro and P. Anokhin, "Fusionplex: Resolution of data inconsistencies in the integration of heterogeneous information sources," Information Fusion, vol. 7, no. 2, 2006, pp. 176 – 196. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253504000867

[2] P. Christen, Data Matching: Concepts and TEchniques for Record Linkage, Entity Resolution and Duplicate Detection. Springer Data-Centric Systems and Applications, 2012.

[3] P.Christen, "Febrl a freely available record linkage system with a graphical user interface," Second Australasian Workshop on Health Data and Knowledge Management (HDKM 2008), vol. 80, 2008, pp. 17–25.

[4] T. Churches, P. Christen, K. Lim, and J. X. Zhu, "Preparation of name and address data for record linkage using hidden markov models." BMC Medical Informatics and Decision Making, vol. 2, no. 1, 2002, p. 9.

[5] M. Odell and R. Russell, "The soundex coding system," American Patent 1 261 167, 1918.

[6] C. L. Borgman and S. L. Siegfried, "Gettys synonametm and its cousins: A survey of applications of personal name-matching algorithms," Journal of the American Society for Information Science, vol. 43, no. (7), 1992, pp. 459–476.

[7] L. Philips, "The double metaphone search algorithm," C/C++ Users J, vol. 18, no. 6, 2000, pp. 38–43.

[8] M. A. Jaro, "Advances in record-linkage methodology applied to matching the 1985 census of tampa, florida," Journal of the American Statistical Association, vol. 84, 1989, pp. 414–420.

[9] W. Winkler, "String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage," Proceedings of the Section on Survey Research Methods, American Statistical Association., 1990, pp. 354–359.

[10] D. Barone, A. Maurino, F. Stella, and C. Batini, "A privacy-preserving framework for accuracy and completeness quality assessment," Emerging Paradigms in Informatics, Systems and Communication, 2009, p. 83.

[11] A. Monge and C. Elkan, "An efficient domain-independent algorithm for detecting approximately duplicate datadata records." 1997.

 ISBN: 978-1-61208-408-4