Comparison of methods Hamming Distance, Jaro, and Monge-Elkan

Maria del Pilar Angeles, Adrian Espino-Gamez
Facultad de Ingenieria
Universidad Nacional Autonoma de Mexico
Mexico, D.F.
Email: pilarang@unam.mx, adriespingam@gmail.com

***Abstract*—The present paper shows the implementation of a more strict comparison algorithm Hamming Distance, which has been enhanced because it not only determines similarity among substrings, but also takes into consideration their corresponding order. Furthermore, we have carried out an evaluation of quality data matching through the string similarity functions Hamming distance, Jaro distance, and Monge-Elkan distance in terms of precision-recall, f-measure, and execution time.**

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

I. INTRODUCTION

The task of data matching has been approached by a number of disciplines, such as artificial intelligence, statistics, and databases, with a different perspective, and different techniques proposed. In our case, data matching is approached from the perspective of Integration of Heterogeneous Databases, especially on how to improve the accuracy of data matching, and how to scale data matching to very large databases that contain many millions of records. The experiments presented here have only been evaluated on small data sets. However, there is still work to do in publishing surveys that compare the various data matching and deduplication techniques that have been developed in different research fields.

We have developed a prototype named Universal Evaluation System Data Quality (SEUCAD) [1] on the basis of the Freely Extensible Biomedical Record Linkage prototype (FEBRL) [2] for cleaning, deduplication and record linkage. The process of data matching is executed within three stages: Indexing: In order to reduce the number of comparisons of pairs of records, the information is segmented according to certain fields (blocked index) and coded function. Comparison: Is a function that identifies the similarity between pairs of records, which returns a numeric value between 0 for total dissimilarity and 1 in case of two identical strings. Classification: There are methods of supervised and unsupervised classification, considering a weight vector and the comparisons made in the previous step, the classification determines false positives, false negatives, true positives and true negatives.

The present paper is organized as follows: Section 2 is focused on similar approaches and their lack of some edit similarity functions comparison. We briefly explain the three comparison methods and their role within de process of data matching. Section 3 details the process of evaluation of data matching to be executed in order to obtain the performance of the comparison methods. Section 4 shows the experimentation plan and the four scenarios considered. Section 5 analyses the performance of the comparison methods from the experiment results and Section 6 concludes the main topics achieved and the future work to be done.

II. RELATED WORK

The present section briefly explains what has been done in terms of comparison of string similarity functions and why we propose the Hamming-distance, Jaro, and Monge-Elkan comparison methods to be tested during the process of data matching [3]. There has been a small number of comparisons of string metrics for data matching. In [4] there was a comparison between the following distance functions: Jaro, Jaro-Winkler, Smith Waterman, and Monge-Elkan. In such research, the results showed that on average, Monge-Elkan method performed best of the edit-distance-like methods in terms of recall.

Michelle Cheatham et al. [5] compare a number of string similarity functions for Ontology alignment. Among the similarity functions the Jaro Winkler, and Monge-Elkan methods were analysed. The outcomes regarding this two functions were that Jaro Winker performed better than Monge-Elkan in terms of precision and recall. In the case of legal case management systems in [6] the performance of a number of name matching algorithms was evaluated such as Exact-Match, Nsoundex, Palmer, Approximate matching, etc. However, the similarity functions proposed in this research work were not considered. Even though human reading seems to be unimpressed by framed permutations ambiguous cases might arise, such as with/whit and expcet/expect then the hamming distance would determine the interpretation.

As our intention is to determine which similarity function works best in terms of quality of data matching, we have carried out a number of comparisons considering different string edit distances, and token edit distances, but their publication still on process. The present paper is aimed to show the comparison of three already mentioned similarity functions as part of a our work.

A. Hamming distance

The Hamming distance [7] is named after Richard Hamming, who introduced it in his fundamental paper on Hamming codes error detecting and error correcting codes in 1950. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. Therefore, it measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other.

We have enhanced the comparison algorithm Hamming distance, because it not only finds the order of the letters, but also takes into consideration the content of the words regardless the order and size of both strings. For instance, let be to strings S1, S2, the first step is to identify which is the largest string, regardless spaces. Suppose S2 is the longest

 ISBN: 978-1-61208-408-4

string. Then, the distance is the difference between the longest string minus the smallest string.

$$distance = length(S2) - length(S1) \quad (1)$$

S1 (the smallest string) will be scanned letter by letter and in case a letter that does not correspond with S2, the variable distance will be added by one. The next step is to obtain the factor of distance, which is been given in the following formula:

$$distance\_factor = \frac{(length(S2) - distance)}{length(S2string)} \quad (2)$$

The Hamming weight of a string is the number of symbols that are different from the zero-symbol of the alphabet used. It is thus equivalent to the Hamming distance from the all-zero string of the same length.

B. Jaro distance

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:

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

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.

C. Monge-Elkan distance

Monge and Elkan proposed in [9] 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:

$$MonElkan(A, B) = \frac{1}{|A|}\sum_{i=1}^{|A|} max\{sim'(A_i, B_j)\}_{i=1}^{|B|} \quad (4)$$

III. Evaluation of Matching

Matching quality refers to how many of the classified matches correspond to true real-world entities, while matching completeness is concerned with how many of the real-world entities that appear in both databases were correctly matched [10]. Each of the record pair corresponds to one of the following categories according to [3].

- True positives (TP). These are the record pairs that have been classified as matches and that are true matches. These are the pairs where both records refer to the same entity.
- False positives (FP). These are the record pairs that have been classified as matches, but they are not true matches. The two records in these pairs refer to two different entities. The classifier has made a wrong decision with these record pairs. These pairs are also known as false matches.
- True negative (TN). These are the record pairs that have been classified as non-matches, and they are true non-matches. The two records in pairs in this category do refer to two different real-world entities.
- False negatives (FN). These are the record pairs that have been classified as non-matches, but they are actually true matches. The two records in these pairs refer to the same entity. The classifier has made a wrong decision with these record pairs. These pairs are also known as false non-matches.
- Precision calculates the proportion of how many of the classified matches (TP + FP) have been correctly classified as true matches. It thus measures how precise a classifier is in classifying true matches [11]. It is calculated as: precision= TP/(TP+FP)
- Recall measures how many of the actual true matching record pairs have been correctly classified as matches [11]. It is calculated as: recall= TP/(TP+FN).

An ideal outcome of a data matching project is to correctly classify as many of the true matches as true positives, while keeping both the number of false positives and false negatives small. Based on the number of TP, TN, FP and FN, different quality measures can be calculated. However, most classification techniques require one or several parameters that can be modified and depending upon the values of such parameters, a classifier will have a different performance according to the number of false positives and negatives. The outcomes can be visualized in different ways to illustrate the performance of several classification tecniques.

- Precision-recall graph. In this visualisation the values of precision and recall are plotted against each other as generated by a classifier with different parameter settings. Recall is plotted along the horizontal axis (or x-axis) of the graph, while precision is plotted against the vertical axis (or y-axis). As parameter values are changed, the resulting precision and recall values generally change as well. Therefore, in Precision-recall graphs there is often a curve starting in the upper left corner moving down to the lower right corner. Ideally, a classifier should achieve both high recall and high precision and therefore the curve should be as high up in the upper right corner as possible.
- F-measure graph. An alternative is to plot the values of one or several measures with regard to the setting of a certain parameter, such as a single threshold used to classify candidate records according to thieir summed comparison vectors, as the threshold is increased, the number of record pairs classified as non-matches increases (and thus the number of TN and FN increases), while the number of TP and FP decreases.

The present work has evaluated the data matching outcomes using synthetically generated data. Consequently, true match status of record pairs was already known. Therefore,

we have developed a set of experiments in order to compare the performance of the following distance algorithms: Hamming, Jaro and Monge-Elkan through the evaluation of the data matching process by computing Precision, Recall and F-measure metrics varying the comparison method only. The set of experiments will be detailed in the following section.

IV. EXPERIMENTATION

The set of experiments correspond to four scenarios, one data file per scenario, which has been indexed, compared and classified three times each. In the case of indexing and classification processes, the corresponding methods and parameters remained the same. However, in the case of comparison process the method has been changed to Hamming-distance, Jaro distance, and Monge-Elkan method.

1) Common configuration for Indexing: Data de-duplication can generally operate at the file or block level. The process of de-duplication by scanning the entire file is not a very efficient means of deduplication. In the case of Block de-duplication, it looks within a file and saves unique iterations of each block. Each chunk of data sorted according to an specific index key, and the comparison process is executed on each block.

- Indexing method: Rather than deduplicate the entire file we have selected Blocking index.
- Blocking key: We have chosen three fields as blocking key or index key ”surname”, given_name and suburb, on each case, we did not set a maximum number of characters for the definition of indexing, otherwise large values will be truncated, and 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” .

The configuration for indexing is presented in Figure 1.

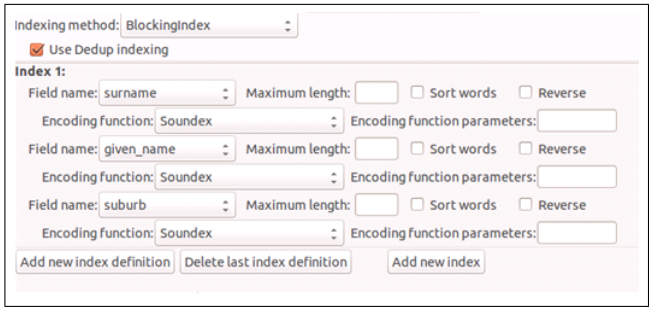

Figure 1. Indexing configuration

2) Common configuration for Classification: This section is aimed to present the configuration parameters established for the execution of the classification method.

- Weight vector classification method: The Fellegi Sunter method has been selected. We have enhaced our prototype in order to compute the data matching metrics even in the case of using a non exact classification method. Otherwise we would have to use for instance, String-exact classification method. It is assumed that the true match status for all compared record pairs is known in the case of supervised classification.

Figure 2 shows the classification settings for the experiments.

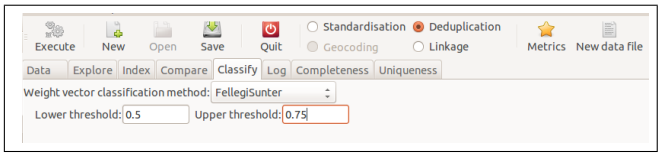

Figure 2. Configuration settings for classification of records.

3) Common configuration for Comparison: The following parameters were chosen for all the comparison cases above described.

- Field name A and Field B: Correspond to the name the fields to be compared against each other. We have chosen ”given_name” field, since the given name is relevant to identify people, and the comparison field is recommended to be of datatype String.
- 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. 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. In the case of the second comparison Dist-Hamming, Jaro or Monge-Elkan, a parameter threshold was required.
- 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. Therefore, we have chosen 0.25.

A. First Scenario

The file called data_comparacion1.csv was generated with a total length of 1000 records, 500 original records, 500 duplicated records, and only one changed field per record. In the case of classification with Fellegi-Sunter, the lower threshold was 0.5 and the upper threshold was set to 0.75.

B. Analysis of Results of the first scenario

As the number of duplicated records was 500, a total of 418 pairs of records were detected and evaluated on each comparison method.

1) Hamming Distance: In the case of Hamming distance, 406 record pairs were classified as matches with 391 record pairs with a sum weight of 0.9, 11 record pairs with a sum weight of 0.8 (above of threshold, which was set to 0.75), 7 record pairs with sum weight of 0.7, 4 record pairs were classified as possible matches and 8 record pairs were classified as non matches, being completely discarded with a sum weight of 0.0,1 record pair obtained a sum weight of 0.60. The 406 record pairs were true positives. The total time taken for the process was of 337 miliseconds.

2) Jaro: In the case of Jaro, 410 record pairs were classified as matches, 403 pairs obtained a sum weight of 0.9,7 record pairs obtained a sum weight of 0.8 (above of threshold, which was set to 0.75), 8 record pairs were completely discarded with a sum weight of 0.0. The 410 record pairs were true positives, and 8 record pairs were false negatives, no true negatives, no false positives. Regarding the quality properties calculated, recall was 0.9808, F-measure 0.9903 and Precision 1. The total time taken during the process was of 357 miliseconds.

3) Monge-Elkan: In the case of Monge-Elkan method, 410 were classified as matches with 403 with a sum weight of 0.9 and 7 record pairs with a sum weight of 0.8 (above of threshold, which was set to 0.75) 8 record pairs were completely discarded with a sum weight of 0.0. 410 record pairs were true positives, and 8 record pairs were false negatives, no true negatives, no false positives. We can observe from the experiments results that Monge-Elkan and Jaro had the same quality of data matching. However, regarding the Quality properties calculated, recall was 0.9808, F-measure 0.9903 and Precision 1. The total time taken for the process was of 336 miliseconds. Thus, the Monge-Elkan method was the fastest.

We observed that the three methods obtained practically the same scores for the quality metrics, Hamming distance presented a more specific values on the sum weights of the comparisons, compared to Jaro and Monge-Elkan, because the Hamming distance comparison was the strictest in order to assign a high value, but this higher level of precision is a disadvantage because is more sensible to the threshold value. Hamming distance was the only comparison method that classified 4 record pairs as possible matches, Jaro and Monge-Elkan classified the same 4 record pairs as matched and they were true possitives. On one hand, the Monge-Elkan method presented better results than Hamming distance taking practically the same time. On the other hand, Monge-Elkan method had the same results as Jaro, but with a difference of 21 miliseconds.

C. Second Scenario

The file called "data_comparacion2.csv" was generated with a total length of 500 records, 300 original records, 200 duplicated records, only one record duplicated of original record as maximum, and four field changes per record. In the case of classification with Fellegi-Sunter, the lower threshold was 0.4 and the upper threshold was set to 0.75.

D. Analysis of Results of second scenario

As the number of duplicated records was 200, a total of 114 pairs of records were detected and evaluated on each comparison method. During the execution of the second scenario, there were classified 114 records as matches, The 114 record pairs were true positives. The data matching quality metrics Precision, Recall and F-measure obtained a score of 1.

1) Hamming Distance: In the case of Hamming function string, 6 record pairs obtained a sum weight of 0.80, 2 record pairs obtained a sum weight of 1.40, 8 record pairs obtained a sum weight of 1.6, and 98 record pairs obtained a sum weight of 1.80. the total time taken for the process was of 289 miliseconds. Figure 3 shows the results.

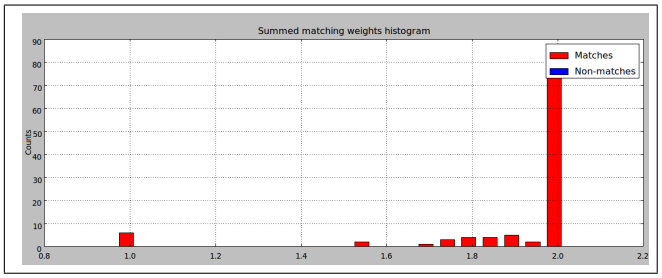

Figure 3. Classification of records with Hamming as comparison method

2) Jaro: In the case of the Jaro function string, from the 114 record pairs classified as true matches, 6 record pairs obtained a sum weight of 0.80, and 108 record pairs obtained a sum weight of 1.8. The total time taken for the process was of 286 miliseconds, being the fastest on the second scenario. Figure 4 shows the results.

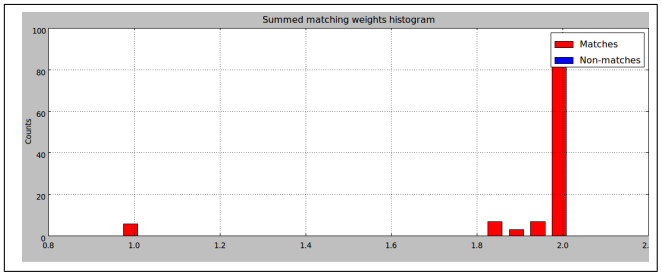

Figure 4. Classification of records with Jaro as comparison method

3) Monge-Elkan : In the case of the Monge-Elkan method from the 114 record pairs classified as true matches, 6 record pairs obtained a sum weight of 0.80, and 108 record pairs obtained a sum weight of 1.80. The time was the longest with 294 miliseconds. Figure 5 shows the results. The three methods obtained the same results. There was practically no difference.

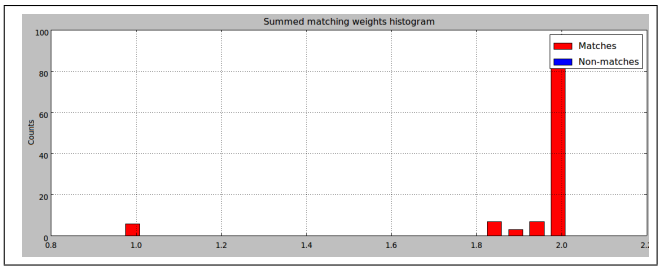

Figure 5. Classification of records with Monge-Elkan as comparison method

E. Third Scenario

The file called "data_comparacion3.csv" was generated with a total length of 1000 records, 800 original records, 200 duplicated records, with two records duplicated of original record as maximum, and three changes per record. In the case of classification with Fellegi-Sunter, the lower threshold was 0.5 and the upper threshold was set to 0.85.

F. Analysis of Results of third scenario

As the number of duplicated records was 200, a total of 141 pairs of records were detected and evaluated on each comparison method.

1) Hamming Distance: In the case of Hamming distance, 122 were classified as matches, 8 record pairs were completely discarded as non matches with 0.0 as a summed weight, 11 record pairs were classified as possible matched. There were 3 record pairs with a sum weight of 0.60, 4 record pairs with a sum weight of 0.70, 5 record pairs with a sum weight of .8 and 121 record pairs with a sum weight of .90. The 122 record pairs classified as matches were true positives. However, there were 8 record pairs classified as false negatives. Precision was 1, recall was of .9384 and f-measure was of .9682. The total time taken for the process was of 328 miliseconds. Fig. 6 shows the results.

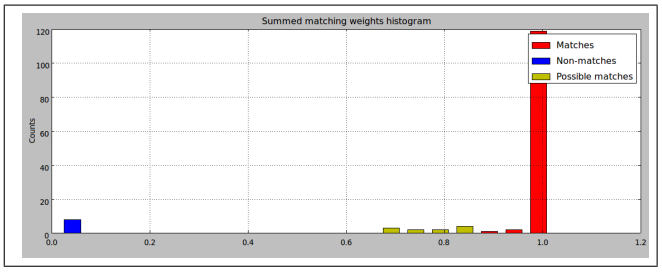

Figure 6. Classification of records with Hamming as comparison method

2) Jaro: In the case of Jaro, 130 records were classified as matches, 8 record pairs were completely discarded with 0.0 as a summed weight as non matches, and 3 record pairs were classified as possible matches. 1 record pair obtained a sum weight of 0.70, there were 5 record pairs with a sum weight of 0.80, and 127 record pairs with a sum weight of .90. The 130 record pairs classified as matches were true positives. There were 8 false negatives and 3 possible matched record pairs. However, there were 8 record pairs classified as false negatives. Precision was 1, recall was of .9420 and f-measure was of .9701. The total time taken for the process was of 336 miliseconds, taking longer than Jaro and Monge-Elkan. Figure 7 shows the results.

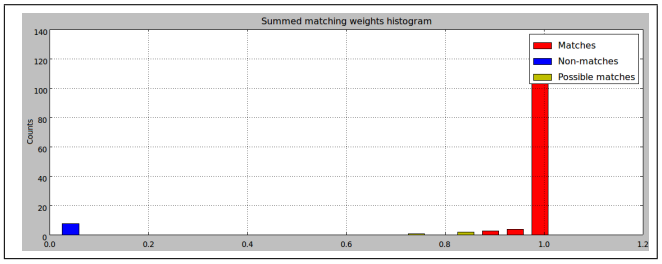

Figure 7. Classification of records with Jaro as comparison method

3) Monge-Elkan : In the case of Monge-Elkan method, 130 records were classified as matches, 8 record pairs were non matches, and 3 record pairs classified as possible matches. 4 record pairs were coompletely discarded with 0.0 as a summed weight, only 2 record pairs with a sum weight of 0.60, 53 record pairs with a sum weight of 0.90, and 82 record pairs with a sum weight of 1.80. The 130 record pairs classified as matches were true positives. However, there were 8 record pairs classified as false negatives. Precision was 1, recall was of .9420 and f-measure was of .9701. The total time taken for the comparison process was of 316 miliseconds, the Monge-Elkan method was the fastest.

As we can observe from the experiment results Jaro and Monge-Elkan obtained the same results and the same quality of data matching, both with better results than Hamming distance. Monge-Elkan distance was the fastest method, and Jaro distance was the slowest. We can observe that the Hamming distance method was more restrictive when making comparisons.

G. Fourth Scenario

The file called "data_comparacion4.csv" was generated with a total length of 800 records, 700 original records, 100 duplicated records. In the case of classification with Fellegi-Sunter, the lower threshold was 0.5 and the upper threshold was set to 0.85.

H. Analysis of Results of fourth scenario

As the number of duplicated records were 100, the number of records pairs evaluated for each method was 93.

1) Hamming Distance: In the case of Hamming distance, 68 record pairs were classified as matches, 20 record pairs classified as non matches, and 5 record pairs classified as possible matches. The 68 record pairs were true positives, 19 record pairs were true negatives, 1 record pair was false negative and no false positives. Regarding the data matching quality metrics, the scores corresponding to precision, recall, and f-measure were 1, 98.55, and 99.27 respectively. The total time taken for the process was of 305 miliseconds. Figure 8 shows the results.

2) Jaro: In the case of Jaro distance, 73 record pairs were classified as matches, 9 record pairs classified as non matches, and 11 record pairs classified as possible matches. The 73 record pairs were true positives, 8 record pairs were true negatives, 1 record paris was false negative and no false positives. Regarding the data matching quality metrics, the

 ISBN: 978-1-61208-408-4

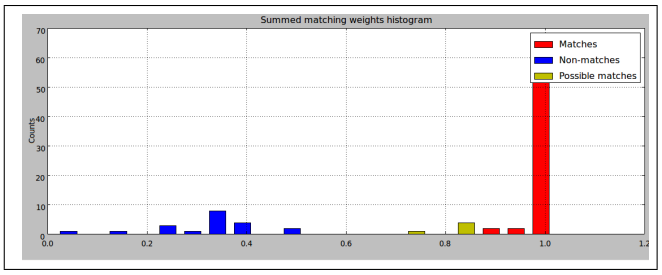

Figure 8. Classification of records with Hamming Distance

scores corresponding to precision, recall, and f-measure are 1, 98.64, and 99.31 accordingly. The total time taken for the process was of 327 miliseconds. Figure 9 shows the results.

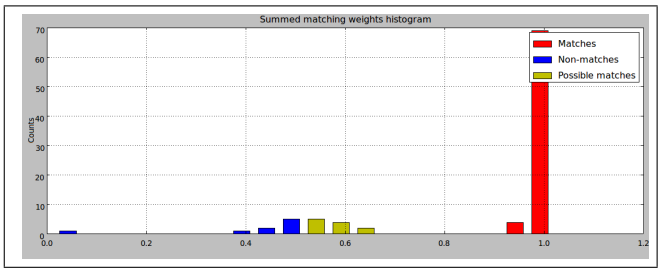

Figure 9. Classification of records with Jaro as comparison method

3) Monge-Elkan: In the case of Monge-Elkan method, 73 record pairs were classified as matches, 17 record pairs classified as non matches, and 3 record pairs classified as possible matches. The 73 record pairs were true positives, 16 record pairs were true negatives, 1 record paris was false negative and no false positives. Regarding the data matching quality metrics, the scores corresponding to precision, recall, and f-measure were 1, 98.64, and 99.31 respectively, presenting the same behaviour as Jaro. The total time taken for the comparison process was of 297 miliseconds, being the fastest method. Figure 10 shows the results.

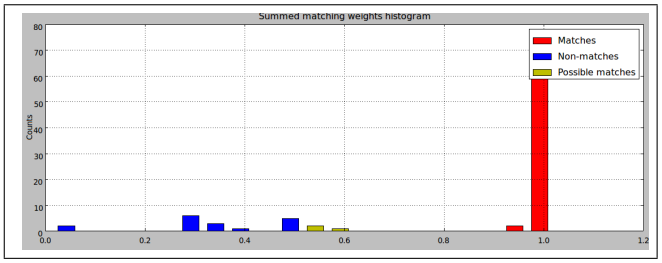

Figure 10. Classification of records with Hamming-Distance as comparison method

V. Performance of Hamming-distance, Jaro and Monge-Elkan

Table I shows data matching quality metrics obtained from the four scenarios. The metrics are: true positives, false positives, true negatives, false negatives, the comparison time, and the number of weights obtained per string function. We can observe that the Monge-Elkan distance function was the fastest method three times out of four scenarios. Hamming distance obtained more true negatives than Monge-Elkan and Jaro because is more strict, but obtained less matched pairs of records for the same reason. Figure 11 shows the corresponding precision, recall and f-measure graph for Hamming Distance. Almost the same performances of data matching were obtained for the three string metrics.

TABLE I. Data Matching Quality Metrics

Scenario	Function	TP	FP	TN	FN	(ms)	no. weigths
1	Hamming	406	0	0	8	337	5
1	Jaro	410	0	0	8	357	3
1	Monge-Elkan	410	0	0	8	336	3
2	Hamming	114	0	0	0	289	4
2	Jaro	114	0	0	0	286	2
2	Monge-Elkan	114	0	0	0	294	2
3	Hamming	122	0	0	8	328	5
3	Jaro	130	0	0	8	336	4
3	Monge-Elkan	130	0	0	8	316	4
4	Hamming	68	0	19	1	305	8
4	Jaro	73	0	8	1	327	6
4	Monge-Elkan	73	0	16	1	297	6

During the first scenario, the Hamming distance was the less effective in terms of f-measure and recall, classifying less pairs of records from the sample than Jaro and Monge-Elkan methods. Furthermore, Monge-Elkan method was the fastest.

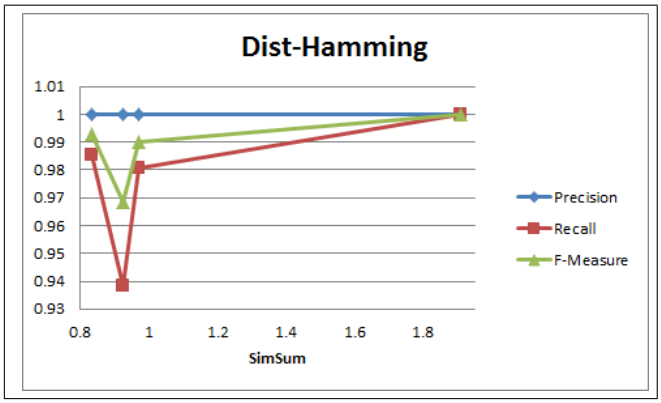

Figure 11. Quality of Data Matching of Hamming Distance

In the second scenario The three distance methods performed equal, but the Jaro method was the fastest.

In the third scenario, Jaro distance was slower than Hamming and Monge-Elkan. In the case of Hamming method, it obtained just one more weight, its comparison allows to classify less pairs of records as matches, presenting lower values of recall and f-measure than Jaro and Monge-Elkan methods.

Finally, in the fourth scenario, Hamming allow the classification of less pairs of records as matches. The three comparison methods obtained just one false negative.

VI. Conclusion and Future Work

We have compared in this research paper three string similarity metrics algorithms: the Hamming distance, the Jaro distance, and the Monge-Elkan distance through our prototype SEUCAD. [1].

We have enhanced the comparison algorithm Hamming distance, because it not only finds the order of the letters,

 ISBN: 978-1-61208-408-4

but also takes into consideration the content of the words regardless the order and size of both strings. We have improved our prototype in order to utilized any classification method, such as Fellegi Sunter, compute the quality metrics, and be able to assess the data matching with no consideration of exact classification methods such as String exact comparison method.

After a number of experiments we have been carried out we can conclude that as the Hamming-distance is stricter during comparison, it obtains a higher number of weights. Therefore, it is more sensible to the thresholds assigned. However, its performance was lower than Jaro and Monge-Elkan methods in terms of recall, and f-measure.

Since, there has not been a significant difference in comparing the string distance methods Jaro, Monge-Elkan and Hamming. There is still work to be done in publishing surveys that compare the various data matching and deduplication techniques that have been developed in different research fields. Furthermore, we will be focused on the enhancement of the already implemented methods and the test on large volume of data as part of our future work.

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] P. Angeles, F. Garcia-Ugalde, C. Ortiz, R. Valencia, E. Reyes, A. Nava, and J. Pelcastre, "Universal evaluation system data quality," DBKDA 2014 : The Sixth International Conference on Advances in Databases, Knowledge, and Data Applicationst, vol. 32, 2014, pp. 13–19.

[2] 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.

[3] P. Christen and K. Goiser, "Quality and complexity measures for data linkage and deduplication," F. Guillet, H. Hamilton (eds.) Quality Measures in Data Mining, Studies in Computational Intelligence,Spriger, vol. 43, 2007, p. 127151.

[4] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," 2003, pp. 73–78.

[5] M. Cheatham and P. Hitzler, "String similarity metrics for ontology alignment," in The Semantic Web ISWC 2013, ser. Lecture Notes in Computer Science, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, Eds. Springer Berlin Heidelberg, 2013, vol. 8219, pp. 294–309. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41338-4_19

[6] K. Branting, "A comparative evaluation of name-matching algorithms." in ICAIL, 2003, pp. 224–232. [Online]. Available: http://dblp.uni-trier.de/db/conf/icail/icail2003.html#Branting03

[7] U. Pfeifer, T. Poersch, and N. Fuhr, "Retrieval effectiveness of proper name search methods," Information Processing and Management, vol. 32, no. 6, 1996, pp. 667–679.

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

[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. M. I. H. Witten and T. C. Bell, Managing Gigabytes, 2nd ed. Morgan Kaufmann, 1999.

 ISBN: 978-1-61208-408-4