J. Basic. Appl. Sci. Res., 3(1s)580-589, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Two-Step-Clustering Method for Clustering Similar Records in Databases by Using Cluster-Similarity Based on Weighted-Matching

Mohammad-Reza Feizi-Derakhshi[1], Azade Roohany[2*]

[1]Department of Computer, University of Tabriz, Tabriz, Iran
[2]Department of Computer, Shabestar Branch, Islamic Azad University, Shabestar, Iran

Received: June 10 2013
Accepted: July 8 2013

ABSTRACT

Proper access to existing data in databases is one of the main and unavoidable issues. But the existence of similar records that were created as a result of type errors and incomplete entering of data makes trouble in proper access and causes size increment of database. In this paper, the cluster similarity method has been introduced in which base similarity is computed based on Weighted-matching method. This fact has a good effect on proper clustering of fields. This method tries to cluster similar records more reasonable, so it uses cluster similarity of fields instead of direct similarity of fields in order to find similar records. This method uses existing data similarities in the database in order to find record similarities which can obtain more reasonable similarity for incomplete information. Also, it reduces the effect of empty and invalid fields in finding the similarity of records. Finally this method has been compared by single step clustering method. By two-step-clustering method about 89 percent F1 was obtained and by single-step-clustering method about 60 percent F1 was obtained.

KEYWORDS: Duplicate Records, Weighted-matching, Cluster Similarity Method, Two-Step-Clustering Method

1. INTRODUCTION

The aim of creating information databases was to being able to quick and proper access to their information. In more searches and operations of databases we need to find special data. For example, when we need to calculate the value of balance accounts of a person or when we want to find the educational history of a person, it is possible that his/her information is recorded on heterogeneous and different databases or it may be the information has been extracted from one database [1]. Nevertheless, in order to access stored information correctly, correct and accurate information must be entered. Otherwise, they could not be restored easily by usual searches. Usually, the common errors in entering information include type problems, incomplete entering of multi-part words, transposing of words in multi-part words, misspelling, incorrectly entering of letters written in multi shapes and … [2, 3].By incorrectly entering of data into database, that value can not be found in the next search. So, user enters this value again which results in repetitive records. So that increasing of these records results in increasing the size of databases and makes problem in terms of time and memory. Sometimes, integration of multiple databases also has this problem [4].

Due to importance of this issue, a lot of works have been done in order to find repetitive records [5, 6] which find similarity degree among corresponding fields comparing them at first and then by using this obtained similarity, similarity of records is evaluated. Therefore, labeling of records which are possibly the same, but are appeared as different entities due to incomplete entering of information is placed in a cluster and if necessary, we can eliminate repetitive data and reduce the size of databases [7]. All methods which follow the above mentioned procedure are called single step clustering and will be explained further in section 2. Another method called two-step clustering has been introduced by authors and it will be described in section 3. In section 4, selective database will be presented which is the base of done experiments. Two-step clustering and used algorithms and also evaluations and their comparisons will be explained in the later sections and finally a general conclusion and comparison of these methods will be presented.

2. FINDING SIMILAR RECORDS BY SINGLE-STEP-CLUSTERING METHOD

We can identify and combine similar records by using some methods in order to minimize the size of databases [8, 9]. So first, field matching algorithms take two fields as an input and then they return their similarity in the numerical format between one and zero [10, 11]. After that the detection of similar records among records of database is done according to obtained numbers and finally clustering is done based on obtained similarities of records.

In this method as it can be seen it has one stage of clustering and in order to fit field, Jaro algorithm will be used and for clustering, Hierarchical clustering method will be used that we explain in nest sections.

3 TWO-STEP-CLUSTERING METHOD

A method to fine similar records is the cluster similarity method which has been introduced in [12] by Feizi and Roohany. In this method, cluster similarity of fields is used instead of direct similarity of fields. The performed experiments of that article show the better precision of this method over single step clustering. That article shows the main idea and in this article the procedure is continued by extending experiments in different steps of method and studying this method for different applications.

***Corresponding Author:** Azade Roohany, Department of Computer, Shabestar Branch Islamic Azad University, Shabestar, Iran, roohany_azade@yahoo.com

In this article, a database will be studied that is mostly based on people's name. The used method is called two- step clustering which uses cluster similarity [12] and Weighted-Matching method [13, 14]. In a way that in cluster similarity step first the similarity of fields is found and then the obtained similarity amount is used for distance of clusters in which Weighted-Matching method is used to find the similarity of fields that the basic idea of this method has been introduced in [13, 14] by Feizi and Roohany. This method is more useful for names especially multi-part names and Iranian names that Jaro and Jaro-Winkler and Improved -Jaro and Token-Jaro methods will be studied in evaluations in order to select the suitable method. After finding cluster similarity as above mentioned, record similarity finding step is done in which the similarity of records is obtained based on performed clusters and finally the similar records are placed in a cluster. The process steps are shown in (Fig.1)

4 THE FEATURES OF DATABASE

The selected database is a real database that this database is employed in the thorough country that the data in this database have been filled in the offices all over the country and by several users that is more than 1000 users. After some decades this database has been gathered in a central database of organization entirely and now this database is used in all the states, towns and cities of Iran. The given table has 10 fields and more than 13 million records that according to obtained statistic about 8 million records are true and almost it has 5 million duplicate records that it is greatly due to incompletely entering data or incorrectly typing these duplicate records.

Some important and relatively correct fields on which we can decide are selected from this database that here the fields of "name", "family name" and "father name" have been selected. One small sample is selected from database and tests are done by this sample. This sample mostly includes records which have some similarities with each other. In this chapter the required tests are performed on this data in order to determine the extent of performance and by necessary comparisons on different algorithms, the proper algorithm is selected.

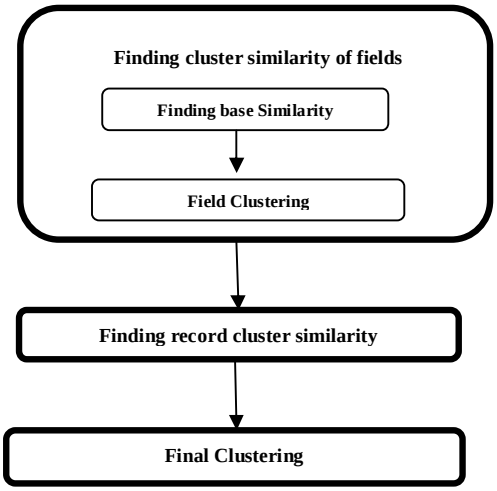

Figure 1. The process steps Two-Step-Clustering Method.

5 FINDING CLUSTER SIMILARITY OF FIELDS

In cluster similarity method, information of database is used to find better similarity degree and the typed cases in multi-part words are found better by this method. Each field is clustered separately. Three types of cluster are considered for this method. * Invalid cluster * Valid clusters * Empty cluster

If the value of each field is false, for instance number 2 has been entered for name, that value is placed in invalid cluster. If the value of a field is empty and nothing has been entered, that value is placed in empty cluster and the remaining with true value are clustered based on base similarity which will be mentioned in the next sub-section and are divided into clusters. As mentioned the cluster similarity is calculated in two steps in which first one base similarity is found that is described in sub-section 1-5 and calculations and evaluations are performed with different algorithms. The second part is clustering which is done by following the above mentioned principles and by using one of the clustering methods. In sub-section 2-5 the methods' explanations and performed evaluations and related results will be presented.

5-1 Base Similarity

Since finding similarity among fields has great value in the beginning and its precision has great influence on other works, in this section the method of Weighted-matching is introduced that the value of bas similarity is calculated by using Weighted-matching. Also since the Jaro method has been introduced to proper name fields [15], Jaro method has been taken as a base and with other algorithms. After measuring the similarity of fields by using these algorithms, cluster similarity will be measured. Now, these algorithms are explained.

- ***-Jaro Distance Metric***

Jaro introduced a string comparison algorithm that was mainly used for comparison of last and first names. The basic algorithm for computing the Jaro metric for two strings S1 and S2 includes the following steps:

1. Compute the string lengths |S1|and |S2|. 2. Find the "common characters" c in the two strings; common are all the characters S1[i] and S2[j] in (1).

$$|i-j| <= 1/2 \min \{ |S1| , |S2| \} \quad (1)$$

3. Find the number of transpositions t; the number of transpositions is computed as follows: We compare the ith common character in S1 with the ith common character in S2. Each non matching character is a transposition. The Jaro comparison value is calculated by equation (2).

$$Jaro(|S1|,|S2|) = \frac{1}{3}\left(\frac{c}{|S1|} + \frac{c}{|S2|} + \frac{c - \frac{t}{2}}{c}\right) \quad (2)$$

From the description of the Jaro algorithm, we can see that the Jaro algorithm requires O(|S1|*|S2|) time for two strings of length |S1| and |S2|, mainly due to Step 2, which computes the "common characters" in the two strings [5,16,17].

- ***-Jaro-Winkler Distance Metric***

Winkler modified the Jaro metric to give higher weight to prefix matches since prefix matches are generally more important for surname matching. The Jaro-Winkler distance metric is best suited for short strings such as person names [16, 18]. Jaro-Winkler is calculated by equation (3).

$$Jaro\_win = Jaro + (D * 0.1 * (1 - Jaro)) \quad (3)$$

In this equation, Jaro is the Jaro distance for two string and D is the length of common prefix at the start of the string up to a maximum of 4 characters. Also is constant scaling factor for how match the score is adjusted upwards for having common prefixes. The standard value for this constant is 0.1 [15].

- ***-Token-Jaro Method***

Always Jaro does his tasks with some limitations so that he checks these characters by putting a window and checking limitation and if it is out of this range, he would not check it that makes some problems for long and multi-section strings which some part of it missed or replaced. In the Token-Jaro method that Feizi and Roohany [19] stated some changes take place on Jaro method in which input string is studied as token, so strings have a logical similarity degree.

- ***-Improved-Jaro-winkler Method and Improved-Jaro Method***

As we know in Jaro method, two strings are compared character by character and the result of this comparison is zero or one so that one means that they are the same and zero means that they are not same. In other words, Jaro method considers zero similarity for two "n" and "m" characters, and also it considers zero similarity for two "n" and "q" characters. In this method Feizi and Roohany [19] have considered similarity degree more than zero like 0.5 for similar characters which might be entered incorrectly due to type errors that these changes were performed on Jaro and Jaro-Winkler method.

- ***-Weighted-Matching Method***

In a Weighted-Matching method which is introduced by Feizi and Roohany [13], in order to compare two fields with each other and find their similarity degree, two parameters Sim and Dist were introduced. The value of Sim shows the similarity degree of characters and the value which Sim can assign to itself is a number between zero and one. $Sim \in [0,1]$ if in two strings, two given characters are completely same like "k" and "k", the assigned value to Sim will be one. But if they are quite different like characters "a" and "k", the assigned value will be zero. Sometimes while typing, some characters are typed incorrectly, for example instead of word "Database" the word "Databace" is entered that the letter "c" was entered instead of "s" or errors like "END" that was entered into database as "FND" where "F" was entered instead of "E".In previous methods the value of zero is considered as similarity degree for this kind of errors in comparison of "E" and "F". In our method a degree in a range of zero and one like 0.8 is considered for these similar letters. These errors are divided into four groups and similarity degree is considered for each of them.

* Typographical errors (keyboard)
* Nominal similarity like "l" and "I"
* Multi shape letter like "y", "i" and "e "*
* Phonetic errors

In this case, similarity degree of 0.6 for type errors, 0.8 for nominal similarity, 0.9 for multi shape letters and 0.5 for vocal errors has been considered. In this method in addition to Sim which was the similarity value of both characters, one weight Dist has been considered that this weight has been assigned according to local value of each character. We can say that the value of this weight for each letter is calculated as equation (4) So that the distance of each letter from first string is measured with distance of a letter from second string.

$$Dist_{i,j} = 1 - 0.1 * |i - j| \quad (4)$$

Where, |i-j| shows its distance from basic location. Also similarity degree between two strings is calculated by using equation(5) and for multi section fields it acts as recursive.

$$Sf = \frac{1}{2}\left(\frac{1}{m}\sum_{i=1}^{m} \max_{j=1}^{n} Sim_{i,j} Dist_{i,j} + \frac{1}{n}\sum_{j=1}^{n} \max_{i=1}^{m} Sim_{i,j} Dist_{i,j}\right) \quad (5)$$

* The case of multi shape letters is not so common in Latin language, but it is common in Persian language like ph and f or ث, ص and س all of which have s phonetic or ذ, ز, ظ and ض all of which have z phonetic.

Where n=|S1| and m= |S2| in which |S1| is the length of first string and |S2| is the length of second string. Also Disti,j is calculated by equation (4) and Simi,j shows the value of similarity. In a particular manner where the length of two strings is equal, similarity equation is also shortened and it is converted into equation (6).

$$Sf = \sum_{i=1}^{n} \frac{1}{n} Sim_i \, Dist_i \quad (6)$$

Also if there is not any displacement, then weight equation changed into Dist =1 [13, 14,19].

5-1-1 Metrics for Evaluation

Five comparisons metric have been applied to evaluation: False Reject, False Accept, True Accept, True Reject and Correct Answer. These metric have been obtained based on scale, personal opinion and results of algorithm.First in comparison of these two cases, four number sets were extracted from database that the first number is Dtt which shows the number of cases that both algorithm and person introduced similar to each there. The second number is Dff that shows the number of cases that both algorithm and person agree on their opposition. The third number is Dtf that shows the number of cases where the algorithm has introduced it same but the person has introduced it dissimilar. The fourth number is Dft that shows the number of cases where the algorithm has introduced it dissimilar but the person has introduced it same.

The value of false reject was calculated by equation (7) which shows the number of cases that algorithm has introduced it dissimilar incorrectly.

$$\text{False Reject} = \frac{D_{ft}}{sn} \quad (7)$$

Where sn shows total number of comparisons. The value of false accept calculated by equation (8) in fact shows the number of cases where algorithm introduced it same incorrectly. The value of true accept was calculated by equation (9) that algorithm introduced true correctly. The value of true reject was calculated by equation (10) that algorithm introduced false correctly. After calculation of these four equations, the value of correct answer like equation (11) is calculated by addition of two equations 9 and 10.

$$\text{False Accept} = \frac{D_{tf}}{sn} \quad (8)$$

$$\text{True Accept} = \frac{D_{tt}}{sn} \quad (9)$$

$$\text{True Reject} = \frac{D_{ff}}{sn} \quad (10)$$

$$\text{Correct Answer} = \text{True Accept} + \text{True Reject} \quad (11)$$

Correct Answer of algorithm is calculated by correct answer criterion, so that we can know that how much these cases have been introduced similar and dissimilar correctly. Since the obtained value of similarities is in a range of zero and one, now we should determine a Threshold so that the value more than this Threshold is considered as the similar and the lower value are considered as dissimilar. The way of determining threshold has been completely described in the previous work of authors in journal [14].

5-1-2 Experiments and Their Results

As introduced in section of evaluation criteria, at first a threshold must be specified so that the upper number is assumed as true and the lower number is considered as false. In (Fig. 2) one sample of finding optimum threshold is studied that this diagram was drawn by using the values of (Table 1) that 0.65 was selected for this case of optimum threshold that has zero False Reject and 0.97 Correct Answer. In order to find other threshold some tests were done that it is tried to find correct answer in a reasonable threshold and false reject reaches the minimum value, with this criterion, a threshold is considered for algorithms that in (Table 2) the related results of name threshold from database was mentioned and also as far this value was calculated for other fields and about each database that is presented in (Tables 2-4). After finding threshold for each field with each algorithm, by using field matching algorithms described, the values of each field is studied and in each field, similarity degree is obtained for each value pair that after comparing each two values of every field, evaluation is done on these obtained values. So that the value of correct answer, false reject and false accept is calculated. In the name fields, Jaro method has high accuracy, so tests have been started by Jaro method and this method has been run on databases and the degree of its correct answer has been calculated and these operations are repeated by Jaro-Whinkler algorithm and also it is repeated by Token-Jaro, improved Jaro, improved Jaro-Whinkler and weighted-Matching algorithms which are previous works of authors that the obtained results are shown in (Table 5). These operations have dome on other fields that the obtained results from family name field has been shown in (Table 6) and the obtained results from father name field is shown on (Table 7).

As it can be seen from (Tables 5-7) both Jaro and Weighted-Matching methods have proper correct answer and this value is about 0.9 that is a good value and the value of false accept and false reject are in their least value. For name field the value of correct answer of Weighted-Matching method is 0.06 more than that in Jaro method and in the family name and father name fields it is 0.07 and 0.04 more respectively. In the name field the value of false reject of Weighted-

Matching is 0.06 less than Jaro which is desirable. But false reject is about 0.02 more but in other fields this value has became less.

In general, the weighted matching method has performed well and it could increase correct answer about 0.06 in the database. Also, in the case of false reject and false accept it has less percentage compared with other algorithms. So we continue remained steps by obtained results of this method.

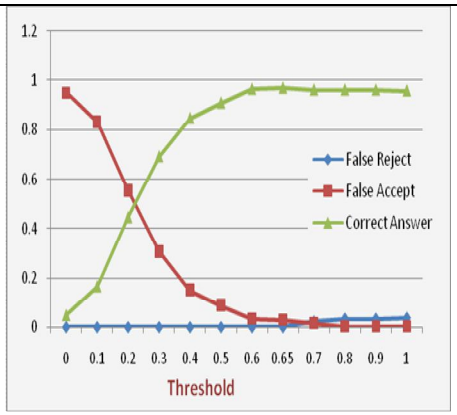

Figure 2. The value of false reject, false accept and correct answer in terms of threshold for Finding optimum threshold for name field by Weighted-matching method This figure represents the values of table 1

Table 1. The values of Correct Answer, False Accept and False Reject for name field by Weighted-matching method, it is usable for finding threshold

Threshold	Correct Answer	False Accept	False Reject
0	4.92E-02	0.9507937	0
0.1	0.1666667	0.8333333	0
0.2	0.4460317	0.5539683	0
0.3	0.6936508	0.3063492	0
0.4	0.8492063	0.1507937	0
0.5	0.9111111	8.89E-02	0
0.6	0.9666667	3.33E-02	0
0.65	0.9714286	2.86E-02	0
0.7	0.9634921	1.43E-02	2.22E-02
0.8	0.9634921	3.17E-03	3.33E-02
0.9	0.9634921	3.17E-03	3.33E-02
1	0.9603175	3.17E-03	3.65E-02

Table 2. Optimum threshold of algorithms for name field

	Threshold
Jaro Distance Metric	0.7
Jaro-winkler Distance Metric	0.7
Token-Jaro	0.7
Improved-Jaro	0.7
Improved-Jaro-Winkler	07
Weighted-matching Method	0.65

Table 3. compared results of algorithms' running over name

	Correct Answer	False Accept	False Reject
Jaro Distance Metric	0.9	0.02	0.069
Jaro-winkler Distance Metric	0.89	0.04	0.049
Token-Jaro	0.91	0.08	0.0003
Improved-Jaro	0.9	0.09	0.0017
Improved-Jaro-Winkler	0.92	0.079	0.0019
Weighted-matching Method	0.96	0.04	0.0016

Table 4. Optimum threshold of algorithms for last name field

	Threshold
Jaro Distance Metric	0.7
Jaro-winkler Distance Metric	0.7
Token-Jaro	0.7
Improved-Jaro	0.7
Improved-Jaro-Winkler	07
Weighted-matching Method	0.65

Table 5. Compared results of last name field

	Correct Answer	False Accept	False Reject
Jaro Distance Metric	0.91	0.08	0.0016
Jaro-winkler Distance Metric	0.93	0.06	0.016
Token-Jaro	0.89	0.1	0.02
Improved-Jaro	0.93	0.65	0.02
Improved-Jaro-Winkler	0.91	0.84	0.02
Weighted-Matching Method	0.98	0.02	0.016

Table 6. Optimum threshold of algorithms for father name field

	Threshold
Jaro Distance Metric	0.7
Jaro-winkler Distance Metric	0.7
Token-Jaro	0.7
Improved-Jaro	0.7
Improved-Jaro-Winkler	07
Weighted-matching Method	0.65

Table 7. Compared results of father name field

	Correct Answer	False Accept	False Reject
Jaro Distance Metric	0.93	0.04	0.02
Jaro-winkler Distance Metric	0.93	0.04	0.02
Token-Jaro	0.9	0.07	0.02
Improved-Jaro	0.93	0.04	0.02
Improved-Jaro-Winkler	0.93	0.04	0.02
Weighted-Matching Method	0.97	0.02	0

5-2 Field Clustering

After finding a base similarity among fields, in this step fields are clustered. The related explanations are mentioned in section 5; cluster similarity that based on which and by using one of the clustering methods this process is performed. At first clustering methods are presented and then evaluation criteria are introduced and in the last sub-section the performed evaluation results of this section is given.

-Hierarchical Clustering Method

Hierarchical clustering method is one of the oldest and simplest clustering methods and also is one of individual clustering methods.

Single-Linkage is one of the Hierarchical clustering methods. It is also called Nearest Neighbor and either Connectedness method or Minimum method. For Single Linkage, assuming that A and B are two clusters, distance d(A, B) is equal to minimum distance between correspondent patterns of A and B that is calculated by equation (12).

$$d(A,B) = \min_{i \in A,\, j \in B} d(i.j) \tag{12}$$

Complete-linkage is one of the Hierarchical clustering methods. It is also called Furthest Neighbor and either Diameter method or Maximum method. For Complete-Linkage, assuming that A and B are two clusters, distance d(A, B) is equal to maximum distance between correspondent patterns of A and B that is calculated by equation (13).

$$d(A,B) = \max_{i \in A,\, j \in B} d(i.j) \tag{13}$$

Average-linkage is one of the Hierarchical clustering methods. For Average-linkage, assuming that B and A are two clusters, distance d(A, B) is equal to average distance between correspondent patterns of A and B that is calculated by equation (14).

$$d(A,B) = \frac{\sum_{i \in A,\, j \in B} d(i.j)}{|A| . |B|} \tag{14}$$

In these methods distance between two clusters is considered. Clustering based on this distance is one of the most common methods in clustering. Since these algorithms are hierarchical, when clusters are combined in order to form new clusters, it erases correspondent rows and columns in the adjacency matrix [20].

-K-means Clustering Method

K-means is a well-known partitioning method [21]. It is a popular clustering algorithm and is widely used in various domains. K-means clustering was developed around 3 decades ago [22]. It starts by initializing the k cluster centers. The data is then assigned to these centers depending upon their proximity to the initial centers. The proximity is measured in terms of square of the Euclidean distance. centroids are recomputed using points assigned to each cluster center. This process of reassigning points to clusters is carried out till there is no change in cluster centers. The original k-means algorithm is computationally expensive and the quality of the resulting clusters from k-means algorithm depends on the selection of initial centroids [23].

5-2-1 Metrics for Evaluation

We introduce 3 metrics for evaluation that include R (Recall), P (Precision) and F1 (F-measure). Since final evaluation has been done on clustering, metric also discuss both on the number of proper clusters and improper clusters, so that P and R are given by equations (15) and (16), respectively. And finally the value of F1 is calculated based on P and R like equation (17) [1, 24, 25].

$$\mathrm{P} = \frac{|\mathrm{M} \cap \mathrm{H}|}{|H|} \tag{15}$$

$|M \cap H|$ is the number of common clusters, means how many clusters are there to make them same either manually or programmatic. $|H|$ is the number manual clusters, shows that these records were clustered in how many clusters manually.

$$\mathrm{R} = \frac{|\mathrm{M} \cap \mathrm{H}|}{|M|} \tag{16}$$

The number which is considered to the number of program clusters ($|M|$) equals to the number of clusters that program form after final clustering.

$$\mathrm{F1} = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{17}$$

In Hierarchical clustering, initially each data is put into one cluster and in each step the near clusters are combined in order to reach one unit cluster. Here we want to combine clusters to some extent that this amount shows that to somewhat similar clusters will be combined. In other words, it shows the maximum extent of similarity that clusters should have in order to combine. The proper selection of extent value has great influence on results, so one other parameter that is introduced here is Te and it shows the stop condition for Hierarchical clustering algorithm. The range of this value is between zero and one. We can obtain the proper value of Te for clustering by giving different values to Te and final testing of R, P and F1.

5-2-2 Experiments and Their Results

In the next step after calculating similarity value of each value pair per field, one similarity clustering is measured that in this step three clustering algorithms namely Hierarchical clustering, Single-linkage and Complete-linkage and Average-linkage and k-means clustering algorithms have been used and some evaluations take place over obtained clusters. As above stated, one Stop-point is needed for Hierarchical clustering that the Stop-point of 0.1 to 0.98 will be studied. For example, this value for name field is calculated based on (Fig. 3) that this figure has been drawn according to (Table 8) which is the result of Single-linkage clustering. As figure shows as the value of Te increases up to 0.09 the value of F1 increases so that in Te=0.9 the value of F1 reaches its highest value and by increasing the value of Te the value of F1

decreases gradually, so the value of 0.9 is an optimal limit. Other Stop-points have been calculated as mentioned sample and in general the obtained Stop-points have been presented in (Table 9) for database fields by Single-linkage clustering method. Then according to obtained Stop-points the values of optimal F1, R and P are specified and obtained results of similarity evaluation are presented in (Table 10) These operations were performed by other methods of clustering that by running Complete-linkage method, the obtained optimal Stop-points and also optimal values of F1, R and P are presented in (Tables 10 and 12) respectively. By running Average-linkage method, the obtained optimal Stop-points and also the values of optimal F1, R and P are presented in (Tables 13 and 14) respectively. In these mentioned results where the base similarity was evaluated by weighted matching method again it has performed by base similarity of Jaro algorithm and the obtained results of optimal Stop-points and also optimal values of F1, R and P are presented in (Tables 15 and 16) respectively. In order to more study we also have used K-means clustering method and for another time this clustering of fields has been tested by this method and both Weighted-Matching and Jaro methods have been considered as a base that the obtained results of F1, R and P and these bases are presented in (Tables 17 and 18).

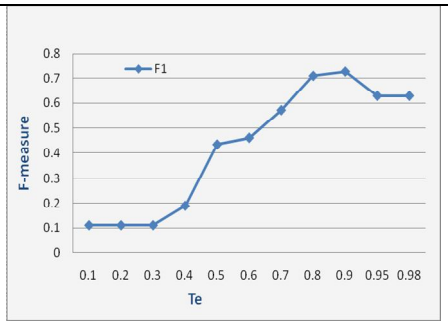

Figure 3. The value of F1 in terms of termination value of name field in clustering similarity step, this figure represents the values of table

Table 8. Values of F1, R and P for name field in clustering similarity step. It is usable for finding Te

Te	F1	P	R
0.1	0.111111111	0.5	0.0625
0.2	0.111111111	0.5	0.0625
0.3	0.111111111	0.5	0.0625
0.4	0.19047619	0.4	0.125
0.5	0.434782609	0.714285714	0.3125
0.6	0.461538462	0.6	0.375
0.7	0.571428571	0.666666667	0.5
0.8	0.709677419	0.733333333	0.6875
0.9	0.727272727	0.705882353	0.75
0.95	0.628571429	0.578947368	0.6875
0.98	0.628571429	0.578947368	0.6875

Table 9. Optimum Te for fields of database whit single-linkage clustering

	Name field	Last name field	Father name field
Te	0.9	0.7	0.7

Table 10. The results of field clustering similarity on fields of database whit single-linkage clustering

	F1	P	R
Name field	0.72	0.70	0.75
Last name field	0.6	0.58	0.72
Father name field	0.8	0.8	0.8

Table 11. Optimum Te for fields of database whit Complete-linkage clustering

	Name field	Last name field	Father name field
Te	0.7	0.6	0.7

Table 12. The results of field clustering similarity on fields of database whit Complete-linkage clustering

	F1	P	R
Name field	0.647	0.6	0.68
Last name field	0.7	0.68	0.72
Father name field	0.68	0.61	0.76

Table 13. Optimum Te for fields of database whit Average-linkage clustering

	Name field	Last name field	Father name field
Te	0.7	0.6	0.7

Table 14. The results of field clustering similarity on fields of database whit Average-linkage clustering

	F1	P	R
Name field	0.72	0.7	0.75
Last name field	0.77	0.77	0.77
Father name field	0.685	0.61	0.764

Table 15. Optimum Te for fields of database whit single-linkage clustering and base Jaro

	Name field	Last name field	Father name field
Te	0.9	0.8	0.8

Table 16. The results of field clustering similarity on fields of database whit single-linkage clustering and base Jaro

	F1	P	R
Name field	0.7	0.72	0.68
Last name field	0.83	0.83	0.83
Father name field	0.57	0.55	0.58

Table 17. The results of field clustering similarity on fields of database whit K-means clustering and base Weighted-matching

	F1	P	R
Name field	0.4	0.4	0.4
Last name field	0.42	0.42	0.42
Father name field	0.38	0.38	0.38

Table 18. The results of field clustering similarity on fields of database whit K-means clustering and base Jaro

	F1	P	R
Name field	0.2	0.2	0.2
Last name field	0.21	0.21	0.21
Father name field	0.2	0.2	0.2

According to performed evaluations the K-means clustering method does not act well but it has logical F1 Hierarchical clustering methods, so it is better to continue by created clusters via Single-linkage clustering.

6. FINDING RECORD CLUSTER SIMILARITY

In this section the similarity of records is evaluated. This similarity is calculated based on cluster similarity of fields explained in the previous step. In addition, one importance degree is also calculated for field which is given in formula 18.

$$Df_i = k_i v_i \quad (18)$$

In this formula, wi is a weight given to each field that this weight performs a control, so that the more important and correct fields' results have more influence. The value of ki which is obtained from formula 19 refers to the type of cluster obtained in the previous step.

$$k_i = \begin{cases} 0 & \text{Invalid} \\ 0 & \text{Empty} \\ 1 & \text{Valid} \end{cases} \quad (19)$$

Formula 20 is used to find the similarity of records which produces a normal number between 0 and 1.

$$\text{Sr} = \sum_{i=1}^{f} \frac{1}{K} Sc_i Df_i \quad (20)$$

In the above formula, f is the number of fields of a record and Sc is the similarity of fields' clusters. In other words, if two fields are in the same cluster, it takes value one and otherwise it shows the distance of clusters. The value of Dfi is the importance of fields which is described in formula 18. K determines the validation of fields according to formula 21.

$$K = \sum_{i=1}^{f} k_i v_i \quad (21)$$

Where, f is the number of fields of one record, wi is the weight of each field as mentioned and ki is the cluster type which is calculated according to formula 18.

6-1 Experiments and results related to find record cluster similarity

The results of this section are evaluated by using evaluation criteria mentioned in section 5-1-1. In the next step record similarity is calculated based on clustering similarity by mentioned methods in previous section and optimum threshold was determined for them Where for each set of runs the optimal threshold has been calculated that it is presented in (Table 19) in which three clustering methods of Single-linkage, Complete-linkage and Average-linkage were the inputs of this part based on two similarity bases of Weighted-Matching and Jaro methods. Based on optimal threshold, the correct answer has been calculated and the obtained result based on Jaro base has been given in (Table 20) and the obtained result based on weighted-matching base has been given in (Table 21).

Table 19. Optimum threshold of clustering methods for base weighted-matching and Jaro

	Optimum threshold for base weighted-matching	Optimum threshold for base Jaro
Record similarity whit single-linkage Cluster Similarity	0.8	0.8
Record similarity whit complete-linkage Cluster Similarity	0.5	0.3
Record similarity whit Average-linkage Cluster Similarity	0.8	0.5

Table 20. Obtained Correct Answer from record similarity step whit weighted-matching base

	False Reject	False Accept	Correct Answer
Record similarity whit single-linkage Cluster Similarity	0	0.0016	0.99
Record similarity whit complete-linkage Cluster Similarity	0.016	0.008	0.99
Record similarity whit Average-linkage Cluster Similarity	0	0.0047	0.995

Table 21. Obtained Correct Answer from record similarity step whit Jaro base

	False Reject	False Accept	Correct Answer
Record similarity whit single-linkage Cluster Similarity	0	0.0016	0.998
Record similarity whit complete-linkage Cluster Similarity	0	0.023	0.97
Record similarity whit Average-linkage Cluster Similarity	0	0.007	0.992

7 FINAL CLUSTERING

After finding the degree of similarity among records, clustering on records is done, so that similar records are located in one cluster. Selective clustering are Hierarchical clustering and K-means clustering, that have better accuracy compared with other methods that we describe in 5-2 sections.

7-1 Experiments and Their Results

Evaluation criterion which is used in this section is the criterion has been introduced in section 5-2-1. In this section one final clustering is done on record similarity and one proper Stop-point whose selection method was mentioned before the optimal Stop-points and the obtained results of these clustering methods have been presented in (Tables 22 and 23) respectively. Clustering algorithms used in this step include Single-linkage clustering, Complete-linkage, Average-linkage

and K-means. In (Table 23) the final result of single step clustering method is presented rather than the results of Two-step clustering method by several algorithms. As (Table 23) shows Two-step-clustering method based on weighted-matching method, could increase F1 30 percent compared with Single-step clustering.

Table 22. Optimum Te of final clustering for clustering algorithms

	Te
Single-Linkage Clustering whit Weighted-Matching base	0.8
Complete-Linkage Clustering whit Weighted-Matching base	0.7
Average-linkage Clustering whit Weighted-Matching base	0.7
Single-Linkage Clustering whit Jaro base	0.8

Table 23. Results F1, R and P of final clustering for clustering algorithms

	F1	R	P
Two-step-clustering method: Single-Linkage Clustering whit Weighted-Matching base	0.89	0.89	0.89
Two-step-clustering method: Complete-Linkage Clustering whit Weighted-Matching base	0.52	0.54	0.5
Two-step-clustering method: Average-linkage Clustering whit Weighted-Matching base	0.84	0.8	0.88
Two-step-clustering method: Single-Linkage Clustering whit Jaro base	0.84	0.8	088
Two-step-clustering method: K-means clustering whit Weighted-Matching base	0.22	0.22	0.22
Two-step-clustering method: K-means clustering whit Jaro base	0.1	0.1	0.1
Single-step-clustering method	0.56	0.56	0.56

8 CONCLUSIONS

Since proper finding and clustering of similar records has great importance, we have tried to increase precision by implementing new methods and studying the existing methods.

Single-step-clustering method acts well and has a good precision. But according to performed evaluations, Two-step-clustering method has better accuracy. It uses existing information of database in clustering similarity method in order to fins the similarity of fields and it does not depend on direct similarity of fields that leads to cluster more reasonable similar records. Clustering-similarity method was also evaluated based on obtained similarity from Weighted-matching method. According to obtained results, it could increase F1 up to 30 percent compared with Single-step clustering.

REFERENCES

1. M.Michelson, SA.Macskassy,(2009): "Record Linkage Measures in an Entity Centric World" Fetch Technologies, 841 Apollo St, Ste. 400, El Segundo, CA 90245 USA, 2010.
2. N.Koudas, S.Sarawagi, and D.Srivastava,(2009): "Record Linkage: Similarity Measures and Algorithms", in Proc, ACM SIGMOD Intl, Conf, on Management, pp. 802–803 of Data.
3. Arasu, S.Chaudhuri, R.Kaushik,(2008) "Transformation-Based Framework for Record Matching" In ICDE.
4. Nora M_eray, Johannes B. Reitsma, Anita C.J. Ravelli, Gouke J.Bonsel, (2007): "Probabilistic Record Linkage Is a Valid and Transparent Tool to Combine Databases Without a Patient Identification Number", Elsevier, Journal of Clinical Epidemiology 60.
5. K.Elmagarmid, P.G.Ipeirotis, VS.Verykios, (2007): "Duplicate Record Detection: a Survey" IEEE Trans, on Knowledge and Data Engg, vol. 19, no. 1, pp. 1–16.
6. Wenfei Fan, Xibei Jia, Jianzhong Li, Shuai Ma1, (2009): "Reasoning About Record Matching Rules", ACM, in VLDB '09, Lyon, France, August 2428.
7. F.Maggi(2008): , "A Survey of Probabilistic Record Matching Models, Techniques and Tools", Sciatic Report TR-2008.
8. A.Philip, S.Abimbola, A.Babajide,(2011): "Modeling and Implementing Record Linkage in Health Information Systems", The Journal on Information Technology, 169-181, sic_00597942, version1- 2jun 2011.
9. I.Bhattacharya, L.Getoor,(2004): "Iterative Record Linkage for Cleaning and Integration", ACM 1-58113-908-X/04/06, DMKD'04, Paris, France, June 13, 2004.
10. C.E.Varghese,(2011): "Record Matching: Improving Performance in Classification" International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 3.
11. A.E.Monge,(2000): "Matching Algorithms within a Duplicate Detection System", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol.23 No.4.
12. M. Feizi-Derakshi, A.roohany, (2011): "Proposing Cluster_Similarity Method in Order to Find as Much Better Similarities in Databases", International Journal of Computer Science Issues (IJCSI), Vol. 8, Issue 5, September 2011.

13. M.Feizi-Derakshi, A.Roohany, M.Sabbagh-Nobarian,(2011): "Proposing Weighted Matching Method for Fields and Records Matching in Database", 16th CSI Computer Conference (CSI2011), Tehran, Iran, (Persian).

14. M.Feizi-Derakshi, A.Roohany, "Evaluation the Precision of Weighted-Matching Method on Two Real Databases", submit to international Journal of Data Warehousing and Mining.

15. W.W.Cohen P.Ravikumar S.E.Fienberg , (2003): "A Comparison of String Distance Metrics for Name-Matching Tasks", American Association for Artificial Intelligence.

16. A.Furer,(2007): "Combining Runtime and Static Universe Type Inference", Master Project Report, Software Component Technology Group Department of Computer Science ETH Zurich.

17. M.Feizi-Derakshi, A.Roohany,(2010): "Review of Field Matching Methods in Database Integration", 2nd Information Technology, Present, Future, Mashhad- Iran. (Persian).

18. Winkler, W.E, (2006): "Overview of Record Linkage and Current Research Directions", Research Report Series, RRS.

19. M. Feizi-Derakshi, A.Roohany,(2011): "Proposing some Methods to Increase Precision of Finding Similarity Records in large Databases", 3rd National Conference on Computer Engineering & Information Technology, Hamedan, Iran (Persian).

20. Mali.F, Mitra.S,(2002): "Clustering of symbolic data and its validation", Advances in Soft Computing.

21. O.Abu-Abbas,(2008): "Comparisons between Data Clustering algorithms", The International Arab Journal of Information Technology, Vol. 5, No. 3.

22. A.Thombr, (2011): "New Initialization Algorithm for k-means Clustering", IEEE, 3rd International Conference on Machine Learning and Computing Singapore.

23. K.A.Abdul-Nazeer, M.P.Sebastian, (2009): "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm", Proceedings of the World Congress on Engineering 2009 Vol I WCE, London, U.K.

24. Yves R.Jean-Mary, E.Patrick Shironoshita, Mansur R.Kabuka, (2009): "Ontology Matching with Semantic Verification", Elsevier, journal Web Semantics: Science, Services and Agents on the World Wide Web 7.

25. J.B.D.Santos, C.A.Heuser, V.P.Moreira, Leandro K.Wives,(2010): "Automatic Threshold Estimation for Data Matching Applications", Elsevier, information sciences