Developing Algorithm For Matching Arabic Names Entered by Mobile Phone

Muneer Alsurori and Salah AL-Hagree
Ibb University/faculty of Sciences/Department of Computer Sciences &Information Technology , Ibb ,Yemen
Email: msurory@yahoo.com , s.alhagree@gmail.com

Maher Al-Sanabani
Thamar University /2 Faculty Computer Science and Information Systems , Thamar, Yemen
Email:M.sanabani@gmail.com

Sarah Abdulmalik , Noor Al-huda AlArhabi , Suad Abdu , Khawlah Meqran
Department of Mathematics & Computer Sciences, Faculty of Sciences, Ibb University ,Yemen.
Email: sara56pr@gmail.com, nooralarhabi1@gmail.com, SuadAbdu995@gmail.com , K1hawlah.1M@gmail.com

***Abstract*—**Name matching plays a vital and crucial role in many applications. They are for example used in information retrieval or deduplication systems to do comparisons among names to match them together or to find the names that refer to identical objects, persons, or companies. Since names in each application are subject to variations and errors that are unavoidable in any system and because of the importance of name matching, so far many algorithms have been developed to handle matching of names. These algorithms consider the name variations that may happen because of spelling, pattern or phonetic modifications. However most existing methods were developed for use with the English language and so cover the characteristics of this language. Up to now no specific one has been designed and implemented for the Arabic language. The purpose of this study is to present a name matching algorithm for Arabic language. In this project, after consideration of all major algorithms in this area, we selected one of the basic methods for name matching that we then expanded to make it work particularly well for Arabic names. This proposed new algorithms based on the convergence and spacing between the Arabic characters in the keyboard of the mobile phone in order to give more accurate results for Arabic names. In this study the experiments have been accomplished in order to evaluate the proposed algorithm (LD_F,LD_S and LD_KE). The first experiment has been carried for the proposed algorithms (LD_F,LD_S,LD_KM and LD_KE). This experiment is carried based on F-Dataset which has 15 pairs of names. The result of the experiment showed that the proposed algorithms gave more accurate results than the Levenshtein algorithm. Therefore, it can be used in many applications such as Automatic Spell Correction (ASC), Search Engines (SE), Data Retrieval (DR), Computational Biology "DNA" ,Customer Relation Management (CRM), Customer Data Integration (CDI), AntiMoney Laundering (AML) and Criminal Investigation (CI)..

***Index Terms*—** Arabic Language, Name Matching, Levenshtein Distance, Mobile Phone, Phone Keyboard Arabic .

I. INTRODUCTION

Name matching is the process of determining whether two name strings are instances of the same name. Multiple methods have been developed for matching names, which reflects the large number of errors or transformations that may occur in real-life data [1].The basic goal of all techniques is to match names (or strings) that don't necessarily required to be identical. Instead of exact match, a normalized similarity measure is usually calculated to have a value between 1.0 (when the two names are identical) and 0.0 (when the two names are totally different). There are several well-known methods for estimating similarity between strings, which can be roughly separated into two groups: character-based techniques and token-based techniques.

Levenshtein algorithm and its variants are character based matching techniques based on edit distance metrics. Levenshtein edit distance is defined originally for matching two strings of arbitrary lengths. It counts the minimum differences between strings in terms of the number of insertions, deletions or substitutions required to transform one string into the other. A score of zero represents a perfect match. The basic Levenshtein method has been extended in many directions.. For example, having an extension to consider reversals of order (transposition of characters) directly in the edit distance operation [Hall and Dowling 1980]. Another direction of generalization is to allow different weights at character level. The weights for replacing characters can depend on keyboard layout or phonetic similarities [2]. In other research Bilenko [3], a distance function is produced by a distance function learner and the weights are learned from a training data set to have a combined record-level similarity metric for all fields.

The affine gap distance metric Waterman[4], offers a smaller cost for gap mismatch, and hence it is more suitable for matching names that are truncated or shortened. Smith and Waterman[5] described an extension of edit distance and affine gap distance to find the optimal local alignment at character level. Regions of gaps and mismatches are assigned lower costs. Jaro [6] introduced a string comparison metric, which is dependent on both number of common characters and number of non-matching transposition in the two strings. Therefore, these algorithms give more accurate results than distance based algorithm while comparing different arrangement such as "William Smith" Vs "Smith William". The q-gram algorithm [7], q-gram with tfidf algorithm [8], and cosine similarity with tfidf algorithm [9] are an examples of this category. The last category is the phonetic algorithms, which focus on word pronunciation and do not focus on typographical errors or character arrangements. The soundex algorithm [10–12], and metaphone algorithm [13] are an examples of this category. In general, distance based algorithms give more accurate results compared to phonetic based algorithms but the performance of the phonetic based algorithms is better than distance based algorithm.

In this paper we have developed algorithms to match the Arabic names entered by the mobile phone based on the convergence and spacing of the Arabic characters in the mobile phone keyboard and taking into consideration the different levels of similarity of the Arabic characters (Form similarity, Phonetic similarity and similarity of the mobile phone keyboard). Where these names are entered in the medical field, because errors in this area many roses and is one of the most important areas, especially when making decisions when diagnosis.

II. .LITERATURE REVIEW

In accordance with this search there are a few studies and research in the field of matching names entered through mobile phone Arabic language other languages, there are many studies related to this search our scans summary for business related to the direction of this search.

Levenshtein [14] found that Levenshtein distance-editing algorithm, based on binary symbols (0,1)The distance factor is used to compute the similarity between two strings. The distance is computed by computing the minimum number of operations we need to convert a string to another series. These processes are addition, substitution, and deletion. However, this algorithm did not give accurate results because it did not take into account similarity levels but gave value 1 even if the two words were similar. The algorithm was not suitable for Arabic.

Ghafour [15] found devise a new approach to match the Arabic name and personal name specifically. The matching algorithms have been classified into three algorithms based on distance, character-based, and audio algorithms .The AEDA algorithm, which is classified within distance-based algorithms to measure the similarity between two Arabic series, is based on the levenshtien algorithm, based on dynamic programming, and takes into account the unique features of Arabic language, similarity of voice, similarity, and similarity of the keyboard .This algorithm is compared to the Levenshtien algorithm and found to produce more accurate results from levenshtien as it gives different weights to the editing processes and takes into account different levels of similarity of Arabic characters. levenshten gives the same weight to all edits and does not give any attention to similarity.

Salah [19] study and discussed the algorithms of measuring string similarities for the Arabic language. The proposed framework in the study took advantage of unique features of the Arabic language and the different levels of similarity of the Arabic characters, the similarity of sound, the similarity of the form and the similarity of the keyboard distance. The proposed framework was also considered a transfer and enhanced case for insertions and deletions .Experiments in the study showed that the proposed framework gives more accurate results. The proposed framework includes a combination of unigram to bigram and split bigram to unigram and the unique characteristics of Arabic.

Alsurori [17] found that the A-N-DIST, which combines N-DIST &LD matching techniques to produce much better and more accurate results .The technique proposed by the study as well as the editing process is a new process of exchanging vowels (y, u, o, i, e, a) to compute the most common misspellings in vowels due to the conversion of names into another language .

Gueddah [18] proposed a new approach aimed at improving the process of correcting spelling mistakes when writing documents in Arabic by linking the costs to exchange errors involving the proximity of the keyboard characters, the computer and the linear similarity in the Arabic alphabet This idea is the introduction of costs in the levenshtien algorithm regardless of the estimates associated with the results obtained from the training test group. The results show the advantage of introducing the proximity matrix and the similarity between the Arabic alphabet within the cross-error correction However, this study was conducted on cross-errors, which account for 65% of the total errors.

Al-sanabani [16] designed an algorithm to match the Arabic names. The algorithm designed in this study has two steps in the computed of similarity in the first step. It computes the cost and the output. The second step is used to compute the similarity ratio. The cost is computed using the n-gram method in computing the similarity ratio taking into consideration the characteristics Which is characterized by the Arabic language as well as the similarity of voice and formality and the similarity of the keyboard, the algorithm suggested in this study gave more accurate results to compare the Arabic names and better results in the retrieval of information from large databases and other processes with On the corresponding period so as to taking into account the features of Arabic.

Hamza [20] used morphological analysis in the spelling process. The study relied on dividing the word into prefix, suffix, root and then computing the distance of Levenshtein for all the prefix, sequential and root possibilities and taking the smaller result after the comparison. The study concluded that the correction process using this method is better than using the comparison The classic two lexical forms using the Levenshtein distance are either at the lexicon level or at the run-time level or compared to the correction rate. The

results of this study indicated that their proposed system for correcting wrong words was 84% compared with the mean of 50.3% in Levenshtein distance, and the average time to correct a faster word in the proposed method is 0.10 ms vs. 0.19 ms in Levenshtein. The lexicon used is much smaller Because the tables of matches between suffixes, prefixes and roots reduce the number of words in the lexicon compared to the Levenshtein dictionary.

Aljameel [21] study and discussed the methods of matching strings and constraints imposed by the Arabic language by dividing them into three lexical similarities, semantic similarities and hybrid similarities. The study discussed four algorithms in lexical similarities, two algorithms,In addition, the Arabic Word Net discussed semantic similarities as well as discussing the difficulties posed by the Arabic language. It also discussed the hybrid similarity that combines the previous lexical and semantic standards, The study concluded that the hybrid similarity, which combines semantic and lexical measures, gives better results in matching Arabic texts than using lexical or semantic similarities due to difficulties and challenges posed by the Arabic language.

Lhoussain [22] suggested a new approach that will make some solutions relatively easy for others by comparing the wrong words and words of a given dictionary in the language and suggesting the closest words to the wrong word based on the similarity and distance between words. The n-gram model is used to improve proposed solutions and improve Schedule their locations. As well as the use of the Levenshtien algorithm Metric method to measure the similarity between two words by computing the distance that is defined as the minimum number of basic edits to convert the wrong word to a word from the dictionary. The challenge in using the algorithm is that the spelling correction using the space editing system does not provide a good suggested scheduling of all words Candidate. Therefore, the bi-gram model is entered into the levenshtien algorithm where bi-gram relies on the preceding word of the misspelled word. The main drawback of this approach is that the size of the basic part is not specified, as are solutions with the same probability.

Hicham.[23] found a new approach taken from the Levenshtien algorithm. This approach helps to make comparisons between two words. It allows for better scheduling of proposed solutions where comparison results are a number rather than an integer as in the levenshtien algorithm. This approach was tested in comparison to Levenshtien's approach to scheduling proposed solutions in the new approach. The disadvantages of this approach were that only 90% of errors were treated and when statistical work 62-63% of the correct words were proposed versus 10% in levenshtein calculation of distance in this approach Was distributed to 21.05% while in levenshtein at 89.7%. The rate of corrections for this approach was 11.05% while in levenshtein 63.2%. The wrong words in editing the distance were estimated at 26.5% and levenshtein at 57.1%.

Mohammed [24] study and discussed the development of a new approach aimed at correcting derivative words, taking into account that most of the Arabic words are derived by modifying the levenshtein algorithm that helps to reduce the number of words. Number of comparisons when viewing from dictionary. In this study, algorithms were developed to identify misspelled words. The lexical correction was independent of the context and depended on dictionary research. Several methods were also proposed to reduce the number of suggestions made by the spelling tools of these methods: Hidden markor and n- gram. Levenshtein, adapted to Arabic, was used to identify the nearest words to the wrong word

Jasper [25] found a new method for computing the distances between the sequences of different lengths by computing the distance of the Levenshtein series of matrices based on the Levenshtein scale. The distances between the sequences are computed by computing the number of edits. The sequence alignment method used in the Levenshtein distance scale is used to align the sequence with elements Which can not be divided. The comparison of patterns of different lengths is represented by the calculation of differentiation and qualitative integration in the form of sequential matrix.

Rani [26] study and discussed the use of the Levenshtein space-editing algorithm as a measure to compare text documents and compute the voltage required to convert one document to another. It is used to detect plagiarism. The algorithm is optimized by removing the stop words which are, they, are, etc., where they are ignored by the search engine processing. The array method is used to compute the Levenshtein space, which depends on the number of edits needed to modify a single document to obtain a document else. The entries from two documents and the number of words are displayed after removing the stop words from all documents using the Levenshtein edit formula and computing the time it takes for the account to compare the documents in milliseconds. Several studies have used the Levenshtein algorithm in their applications of these image processing applications [27]. The application of voice control in the application of calories by [28] which uses the Levenshtein algorithm to improve the accuracy of the two audio inputs by the user and corrects the audio input errors by checking the input Voice using the words of the database. This app has been made on mobile devices with the Android operating system.

Almost everyone has a portable device and their daily activities are also connected to their mobile devices. They always bring their mobile devices and use them anywhere and anytime.

The algorithm was also used in other applications, including image processing applications and application for document theft detection [29-31]..

The Ghafour [15] AED A and Al-sanabani [16] NDIST-A algorithm have been elaborated in above. These algorithms have two problems such as :

These algorithms depend on the similarity and distance between the characters on a keyboard and a computer.

These algorithms used the Euclid scale and did not use the Manhattan scale in computing the distance.

III. THE PROPOSED ALGORITHM

In this section, introduces the proposed algorithm that is used for the performance evaluation of this paper. A new algorithm that is Levenshtein distance Form, Levenshtein distance Sound, Levenshtein distance _Keyboard Eqleeds and Levenshtein distance_Keyboard

Manhattan (LD_F,LD_S,LD_KE and LD_KM) algorithms respectively .

The (LD_F,LD_S,LD_KM and LD_KE) algorithms unlike the original Levenshtein distance (LD) as show Figure 1.Several algorithms have been developed to measure chain similarity. The researchers used the Levenshtein algorithm and applied it to the Arabic language, which is one of the matching algorithms that use the distance factor to compute the similarity between two series. We have proposed new algorithms to match Arabic strings that take into account the unique features of Arabic language and similar levels of similarity For Arabic characters such as phonetic similarities and character forms as well as the distance of the keyboard. The application of string matching algorithms to the Arabic context is a difficult task because of its unique and unique characteristics One of these studies is [15, 16]. The disadvantages of these studies: The costs were not included in the Levenshtein algorithm. The researchers applied this algorithm to a computer keyboard and were not used for a keyboard Especially for mobile phones focused on the slang especially Egyptian, the Manhattan scale was not used to measure the distance between characters.

To address this problem, we recommend Architectural Form show in Figure 1.

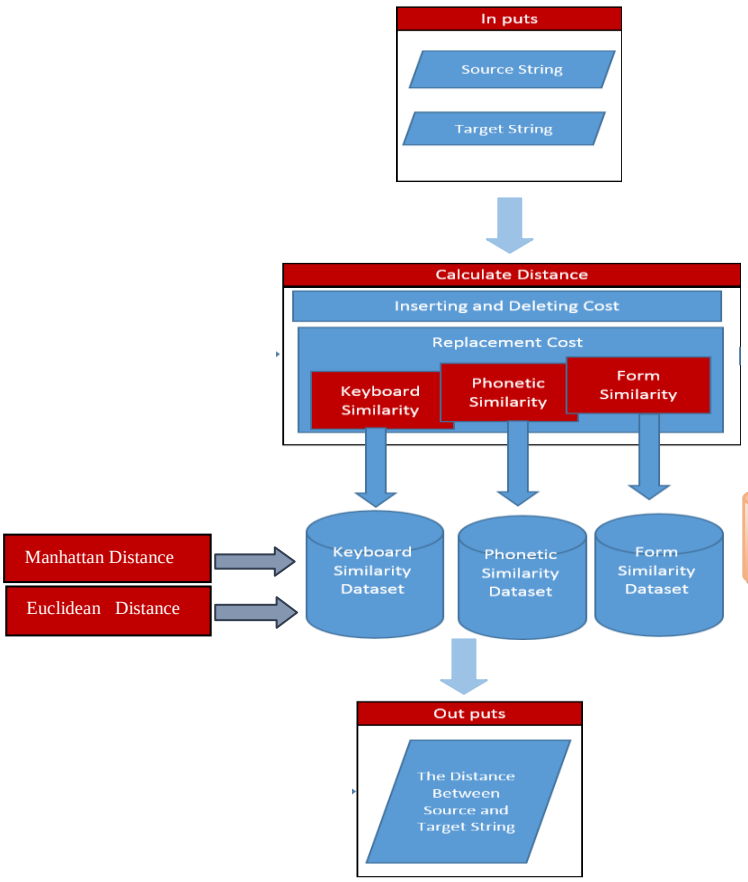

Fig. 1. Architectural Form Of The Developed Algorithm

The architectural design of the developed algorithms, which explains how the Arabic words and cost computed are matched, is the following steps:

- The characters of the source word and the target word are converted to a character matrix Compare each character of the source matrix with each letter of the target matrix, if it matches the distance 0 and the similarity ratio is 100% and if it does not match we compute the cost(insert , delete or replacement).

III.I. Compute cost of insert and delete

Compute the cost of insertion and deletion and in the following rules:

- If the character equals a blank, the cost is zero
- If the letter is not equal to the letter before it and this letter is a Hamza and before it one of the vowels delete Hamza without affecting the meaning of the word and the cost is zero.
- If the letter is equal to the letter before it, the cost is and its value is greater than zero and less than one
- If the character is not equal to the character that preceded it and belongs to a vowel, the cost is μ and that is 1 Compute the cost of insertion and deletion is shown in Figure2.

Example of deletion the word 'كالسوم' in true form, it have to write in this form'كالسيوم'.

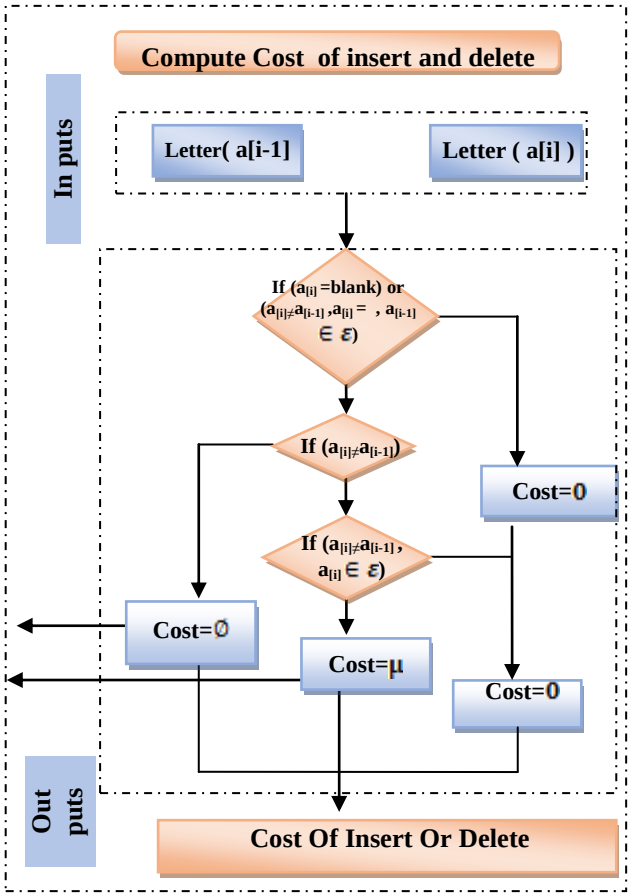

Fig. 2. Compute Cost of insert and delete.

III.II. Cost of Replacement

The cost of the replacement, which is on three levels: the formality, the voice and the keyboard and we compute each level separately

1-Form Similarity

The Arabic alphabet consists of 29 characters, seven of which have two forms and twenty-two of them have four forms depending on their position within the

word. These forms are initial, intermediate, final, isolated and "separated". [16, 18].
Compute the cost of the character shape = 1 - the similarity indicator of the character shape, called this algorithm Depending on the form similarity (LD_F)as shown in Figure 3.

Function for computed cost between characters in cellphon's keyboard by form similarity

1:input:a and b
2:output:costReplacement
3:computed the cost between character a and b by costReplacement←1-form_similarity(a,b)Where a and b are Arabic character

Fig3. Proposed function for replacement cost depending on Form similarity

2- Phonetic Similarity

Another aspect of the similarities can be in the pronunciation of the letter since the Arabic language is very rich in terms of its vocal abilities where the Arabic alphabet consists of 29 characters that appear from about 17 different points so any deviation in the pronunciation of the letter may lead to a change of meaning depends on the pronunciation of Arabic language as classic or standard, modern or public [16, 18].
Compute the acoustic cost element = 1 - the acoustic similarity index, called this algorithm Depending On The Phonetic Similarity (LD_S)as shown in Figure 4.

Function for computed cost between characters in cellphon's keyboard by phonetic similarity

1:Input:a and b
2:Output:costReplacement
3:Computed the cost between character a and b by CostReplacement←1-phountic_similarity(a,b)Where a and b are Arabic character

Fig 4. Proposed function for Replacement cost depending on phonetic similarity

3-Keyboard Similarity

The keyboard distance is the distance between the location of two letters on the keyboard, where the proportion of similarity is significant if the characters are close to each other and are small if they are far apart [16]. In this subsection we discuss how to compute the similarity of some keyboards used in mobile phones. Because of the multiplicity of paintings and their differences from one device to another.
Compute the acoustic cost element = 1 – Keyboard similarity (computed by Euclid's distance) index, called this algorithm Depending On The Keyboard similarity (computed by Euclid's distance) (LD_KE)as shown in Figure ,also Compute the acoustic cost element by 1 – Keyboard similarity (computed by Manhattan distance) index, called this algorithm Depending On The Keyboard similarity (computed by Euclid's distance) (LD_KM) as shown in Figure 5.

Function For Computed Cost Between Characters In Cellphon's Keyboard By Keyboard Similarity

1:input:a and b
2:output:costReplacement
3:computed the cost between character a and b by costReplacement←1-keyboard_similarity(a,b)Where a and b are Arabic character

Fig5. proposerd function for Replacement cost depending on Keyboard similarity

Example of replacement the word 'تقلسات ' It has to write as form 'تقلصات"

III.III. MEASURE KEYBOARD SIMILARITY

The similarity between each two letters in the keyboard is measured using the Euclid and Manhattan distance calculation rules expressed in the following equations:

1. Euclidean Distance

Euclid's law is used to compute the distance between two points at the coordinate level where the two-point dimension law is one of the laws of mathematics [18] which can be computed equation (1).

$$\text{Eqleds's Distance}(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (1).$$

This results can be shown in Table 1 that shows the Euclidean Distancematrix that has been built for some Arabicletters by repeated the previous steps or by function as shows Figure 5.

Table 1.ComputeEuclidean Distance for some Arabic letters in Mobile's keyboard

	ا	ب	ت	ث	ج	ح	خ	د
ا	1							
ب	0.83	1						
ت	0.92	0.75	1					
ث	0.75	0.58	0.83	1				
ج	0.58	0.42	0.68	0.67	1			
ح	0.67	0.5	0.75	0.75	0.92	1		
خ	0.75	0.58	0.83	0.83	0.83	0.92	1	
د	0.75	0.92	0.67	0.67	0.33	0.42	0.5	1

1. Manhattan Distance

The Manhattan Code is used to compute the distance between two points at the coordinate level where the distance between two points is known as the length of the straight line between these two points (x, y) which can be computed by the equation (2) law:

$$\text{Manhattan's Distance}(a,b) = \text{abs}(x_a - x_b) + \text{abs}(y_a - y_b) \quad (2)$$

The values of x, y are the coordinate axes, xa, b, ya, b are the character locations on the coordinates, corresponding to different mobile phone keyboards.
This result can be shown in Table 2 that shows the Manhattan Distance matrix that has been built for some Arabic letters by repeated the previous steps or by functions shows Figure 5.

Table 2.ComputeManhattan Distance for some Arabic letters in Mobile's keyboard

	ا	ب	ت	ث	ج	ح	خ	د
ا	1							
ب	0.83	1						
ت	0.92	0.75	1					
ث	0.81	0.66	0.88	1				
ج	0.66	0.49	0.74	0.76	1			
ح	0.74	0.58	0.81	0.81	0.92	1		
خ	0.81	0.66	0.88	0.83	0.83	0.92	1	
د	0.81	0.92	0.74	0.67	0.47	0.55	0.63	1

III.IV. Measure Similarities In Different Keyboards For Mobile Phones

Mobile phones have a permanent presence in most people's lives, but there are situations where we can not easily access their services. For example, when walking in a busy place, the focus should be on what is happening around us, rather than writing.

Most keyboards are automatically corrected because different text input devices lead to different types of typing errors. Automatic error correction can result in greatly improved results. An effective way to correct errors automatically increases keyboard ease and write speed, because users will not have to stop typing to correct errors.

Traditionally, the discovery and correction of text errors focuses on the character level, which can be classified into three categories: deletion, character deletion, insertion, when an extra character is inserted, and substitution, when a character is replaced by another character. It is inevitable to modify some words written correctly. This happens if another word is more likely for the written word and thus correcting the error will only increase the scalability of these keyboards.

Computed by using similarity distance in keyboard, as in equation (1), Computed by using similarity, as in equation (2).

ض	ص	ق	ف	غ	ع	ه	خ	ح	ج
ش	س	ي	ب	ل	ا	ت	ن	م	ك
ظ	ط	ذ	د	ز	ر	و	ث	ة	⌫
؟١٢٣	→↑←	،	<العربيـة				•		↵

Fig 6. Keyboard For Mobile Phone.

IV. The Experiment and Results

This section shows the experiments that have been carried in this paper to illustrate the proposed algorithms (LD_F,LD_S,LD_KM , LD_KE) and compare it against the Levenshtein algorithm.

IV.I. Data Preparation of Dataset and Experimental Results

This subsection describes the names of Arabic that is used to test the proposed algorithm for matching Arabic Names entered by phones. For more investigation a collection of datasets have been used in this experiment for testing the proposed (LD_F,LD_S,LD_KM,LD_KE) and levenshtein algorithm .because no standard collection of Arabic names exists, therefore, Three datasets have been extracted manually form. Sample of the text of three the Facebook, Whatsapp and Telegrams datasets after extract errors word can be shown in Figure7. They have been obtained from Facebook , Whatsapp and Telegram Applications that are called(FDataset,WDataset and TDataset) respectively,Table 3 shows this datasets , also one dataset has been extracted manually form that is called Dataset 1 [16]. Each dataset contains some of Health words with different possible variation (such as typographical and spelling errors) of same names. A collection of all kind of variation have been considered in the variation of datasets.

Fig 7. Original Text

Table 3.Datasets statistic

N0	Dataset	Application	Words No
1	FDataset	Facebook	15
2	WDataset	Whatsapp	30
3	TDataset	Telegram	115

IV.II. Experimental Results

In the subsection, a comparative study is carried out to evaluate the performance of the proposed (LD_F,LD_S and LD_KE) algorithms, The first experiment has been carried for the proposed algorithms (LD_F,LD_S,LD_KM and LD_KE). This experiment is carried based on F Dataset which has 15 pairs of names. The result of this experiments shown in Table 4. The (LD_F,LD_S and LD_KE) Algorithms gives better results than the LD algorithm. The LD_F,LD_S and LD_KE algorithms give more accurate results than the LD algorithm which has shown in Table 4 and Figure 8.

Table4.Comparison between algorithms in

NO	String.		Compared Algorithm				Proposed Algorithm					
			LD		LD_S		LD_F		LD_KE		LD_KM	
	S	T	Dist.	Sim %	Dist.	Sim %	Dist.	Sim %	Dist.	Sim %	Dist.	Sim %
1	كالسوم	كالسيوم	1	86%	0.15	98%	0.15	98%	0.15	98%	0.15	98%
2	تقلسات	تقلصات	1	83%	0.2	97%	1	83%	0.08	99%	0.08	99%
3	يؤادي	يؤدي	1	80%	1.15	77%	1.15	77%	1.15	77%	1.15	77%
4	التقيوء	التقيؤ	2	71%	1	86%	1	86%	1	86%	1	86%
5	التنغسي	التنفسي	1	86%	1	86%	1	86%	0.08	99%	0.08	99%
6	دا	داء	1	67%	0	100%	0	100%	0	100%	0	100%
7	فخص	فحص	1	67%	1	67%	0.4	87%	0.08	97%	0.08	97%
8	نديف	نزيف	1	75%	1	75%	1	75%	0.08	98%	0.08	98%
9	نفاخ	انتفاخ	2	67%	2	67%	2	67%	1.59	73%	1.73	71%
10	نسيك	نسيج	1	75%	1	75%	1	75%	0.08	98%	0.08	98%
11	الوليد	الوريد	1	%83	1	%83	1	%83	0.12	98%	0.17	97%
12	البو	الربو	1	%80	1	%80	1	%80	0.59	88%	0.73	85%
13	حساسيه	حساسية	٢	%71	1.15	%84	0.15	%98	0.39	94%	0.48	93%
14	الهيموجلوبين	الهيموغلوبين	٢	%85	1	%92	1	%92	0.42	97%	0.42	97%
15	عين	عين	١	%67	1	%67	0.4	87%	97%	97%	0.08	97%
Average (percentage similarity)				76.15%		82.14%		84.87%		93.26%		92.79%

In order to understand how the editing operations in the algorithm works with variation of names, it will be elaborate of in detail in the following examples and as shown in Figure 9 shows how the LD_KE algorithm measures the distance between the name1 "نضريه" and name2 "نظرية" as first step. The distance is 0.4, therefore, the similarity between them is 92%. It is obvious that LD_KE algorithm gives a very low cost for replacing 'ه' with "ة" from name2 into name1 due to their form similarity.

Furthermore, more experiments have been carried with a variety of datasets to get the evidence of (LD_F,LD_S and LD_KE) algorithms ability. Datasets are selected and applied on the (LD_F,LD_S and LD_KE) algorithms as shown in Table5 and Figure 10. That gives the evidence of (LD_F,LD_S and LD_KE) algorithm ability in match names are entered by phone. Table 5 shows the accuracy of the percentage similarity as an mean for W Dataset , T Dataset and Dataset1. In Table 5, the LD algorithm gets 74.44% while LD_S,LD_F ,LD_KM and LD_KE algorithms get 82.45% , 88.83%, 95.55% , 95.96%,respectively. Therefore, the LD_KE algorithm gives more accurate results than the LD,LD_S,LD_F and LD_KM algorithms for W Dataset .

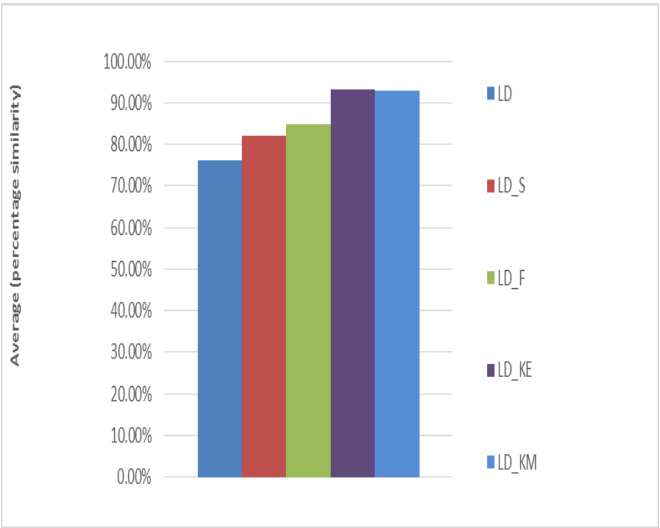

Fig 8. Average (percentage similarity)

		ن	ظ	ر	ي	ة
	0	1	2	3	4	5
ن	1	0	1	2	2.15	3.15
ض	2	1	0.17	1.17	1.32	2.32
ر	3	2	1.17	0.17	0.32	1.32
ي	4	2.15	1.32	0.32	0.17	0.82
ه	5	3.15	2.32	1.32	0.66	0.4

Fig 9. The Distance Between"نظرية" →"نضريه" in the LD_KE Algorithm.

Table 5. The Average similarity of LD, LD_S, LD_F, LD_KE and D_KM algorithms

Dataset	Compared Algorithm	Proposed Algorithms			
	LD	LD_S	LD_F	LD_KM	LD_KE
WDatabase(30 pairs)	74.44%	82.45 %	88.83 %	95.55 %	95.96 %
TDataset (115 pairs)	74.64%	84.36 %	86.73 %	93.62 %	93.85 %
Dataset1 (10 pairs)	65.75%	90.15 %	87.65 %	93.62 %	93.81 %

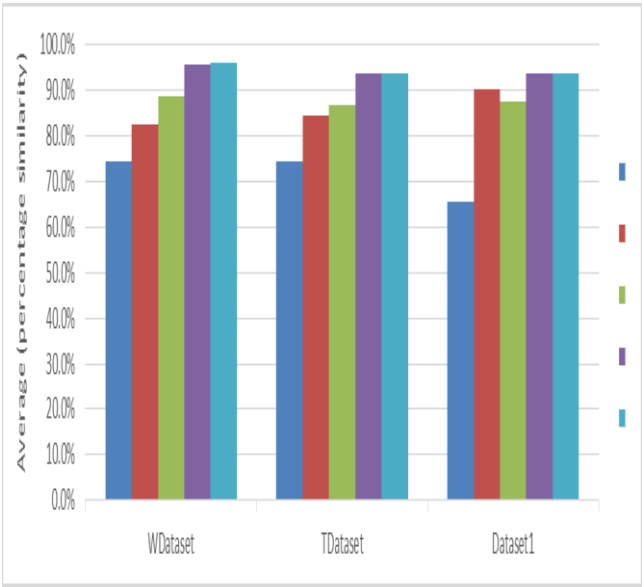

Fig 10. Compared algorithm and proposed algorithm with different datasets.

V. CONCLUSION AND FUTURE WORK:

As mentioned earlier there are different name variations, all algorithms developed up to now focused on some of them but not all. Because they are different from each other and in some cases need totally different solutions. Thus one algorithm cannot be a suitable solution to deal with all different name variations.

This work proposed a new Name Matching algorithm based Arabic language which entered by Mobile Phone. The accurate of this method is evaluated from three aspects: similarity of sound, the similarity of the form and the similarity of the keyboard For Matching Arabic Names Entered by Mobile Phone in the Face book, Telegrams and the whats app Dataset. The proposed method has been compared with levenshtein-distance algorithm. The experimental results show that the proposed LD, LD_S, LD_F, LD_KE and LD_KM algorithms significantly outperforms the other methods in most considered cases. For future work, more, we intend to apply our proposed algorithms to another database, such as The patients 'data. Moreover ,we intend use rules of language or rules that bind a set of letters , one or two or more with a different set of letters . Therefore, it can be used in Search Engines (SE), Data Retrieval (DR), Computational Biology "DNA" ,Customer Relation Management (CRM), Customer Data Integration (CDI), AntiMoney Laundering (AML), Criminal Investigation (CI) and Automatic Spell Correction (ASC) for Arabic language or any languages written in Arabic letters like Persian, Pashto and Urdu the official language of Iran, Afghanistan and Pakistan respectively . Also ,rules of distinction build similar letters , such as (ا ، ى)and (ه ، ة) in the last word. As well as the rules of distinctionض between ظ and. We apply the proposed algorithms to other languages such as Persian.

VI. ACKNOWLEDGEMENTS

The authors would like to thank all those who contributed toward making this research successful. Also we would like to thank to all the reviewers for their insightful comment.

VII. .REFERENCES

[1] Ahmed,k,Elmagarimd (2007). Duplication Record Detection: A survey . IEEE transactionon knowledge and dataenginering, , 19(1).
[2] P. A. Hall and G. R. Dowling, "Approximate string matching. ACM Computing Surveys (CSUR)," ACM Computing Surveys (CSUR), vol. 12, no. 4, pp. 381–402, 1980.
[3] T. El-Shishtawy, "Linking Databases using Matched Arabic Names," Computational Linguistics and Chinese Language Processing, vol. 19, no. 1, pp. 33–54, 2014.
[4] Branting, L. K. (2003). A comparative evaluation of name-matching algorithms. ICAIL '03 Proceeding of the 9th international conference on Artificial intelligence and law (pp. 224-232). New York: ACM.
[5] M.S. Waterman, T.F. Smith, and W.A. Beyer, "Some Biological Sequence Metrics," Advances in Math., vol. 20, no. 4, pp. 367-387, 1976.
[6] Smith, T. F., & Waterman, M. S. (1981). Identification of Common Molecular Subsequences. Journal of Molecular Biology , 147, 195-197.
[7] Jaro, M. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American statistical Association , 89, 414-420.

[8] F. Ahmed and A. N¨urnberger, "N-grams Conflation Approach for Arabic," in ACM SIGIR Conference, Amsterdam, 2007.
[9] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg, "Adaptive name matching in information integration," IEEE Intelligent Systems, vol. 18, no. 5, pp. 16–23, 2003.
[10] W. W. Cohen, "Integration of heterogeneous databases without common domains using queries based on textual similarity," in ACM SIGMOD Record, vol. 27, pp. 201–212, ACM, 1998.
[11] S. U. Aqeel, S. Beitzel, E. Jensen, D. Grossman, and O. Frieder, "On the Development of Name Search Techniques for Arabic," Journal of the American Society for Information Science and Technology, vol. 57, no. 6, pp. 728–739, 2006.
[12] H. A. Shedeed and H. Abdel, "A New Intelligent Methodology For Computer Based Assessment Of Short Answer Question Based On A New Enhanced Soundex Phonetic Algorithm For Arabic Language," International Journal of Computer Applications, vol. 34, no. 10, 2011.
[13] "Understanding classic soundex algorithms," http://www.creativyst.com/Doc/Articles/ SoundExl/SoundExl.htm.
[14] L. Philips, "Hanging on the Metaphone," Computer Language, vol. 7, no. 12, pp. 39–44, 1990. Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710).
[15] H. H. A. Ghafour, A. El-Bastawissy, and A. F. A. Heggazy, "AEDA: Arabic Edit Distance Algorithm Towards A New Approach for Arabic Name Matching," in IEEE International Conference, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, pp. 926–932, 2011.
[16] Al-Sanabani, M., & Al-Hagree, S. (2015) Improved An Algorithm For Arabic Name Matching. Open Transactions On Information Processing ISSN (Print): 2374–3786 ISSN (Online): 2374–3778.
[17] Alsurori, M., Al-Sanabani, M., & Salah, A. H. (2018). Design an Accurate Algorithm for Alias Detection, ISSN: 2074-9023 (Print), ISSN: 2074-9031 (Online).
[18] Gueddah, H., & Yousfi, A. (2013, May). The impact of arabic inter-character proximity and similarity on spell-checking. In Intelligent Systems: Theories and Applications (SITA), 2013 8th International Conference on (pp. 1-4). IEEE.
[19] Salah, A. H ,&Al-Sanabani, M.. (2016). A Framework For Name Matching In Arabic Language, 1st Scientific Conference on Information Technology and Networks.
[20] Hamza, B., Abdellah, Y., Hicham, G., & Mostafa, B. (2014). For an independent spell-checking system from the Arabic language vocabulary. International Journal of Advanced Computer Science and Applications.
[21] Aljameel, S. S., O'Shea, J. D., Crockett, K. A., & Latham, A. (2016, December). Survey of string similarity approaches and the challenging faced by the Arabic language. In Computer Engineering & Systems (ICCES), 2016 11th International Conference on (pp. 241-247). IEEE.
[22] Lhoussain, A. S., Hicham, G. U. E. D. D. A. H., & Abdellah, Y. O. U. S. F. I. (2015). Adaptating the levenshtein distance to contextual spelling correction. International Journal Of Computer Science And Application.(12), 1, 127-133.
[23] Hicham, G. (2012). Introduction of the weight edition errors in the Levenshtein distance. arXiv preprint arXiv:1208.4503
[24] Mohammed, N., & Abdellah, Y. (2018). The vocabulary and the morphology in spell checker. Procedia Computer Science, 127, 76-81.
[25] Beernaerts, Jasper., Debever, E., Lenoir, M., De Baets, B., & Van de Weghe, N. (2019). A method based on the Levenshtein distance metric for the comparison of multiple movement patterns described by matrix sequences of different length. Expert Systems with Applications, 115, 373-385.
[26] Rani, S., & Singh, J. (2017, October). Enhancing Levenshtein's Edit Distance Algorithm for Evaluating Document Similarity. In International Conference on Computing, Analytics and Networks(pp. 72-80). Springer, Singapore.
[27] Ichimura, T., & Kamada, S. (2013, October). A Clonal Selection Algorithm with Levenshtein Distance based Image Similarity in Multidimensional Subjective Tourist Information and Discovery of Cryptic Spots by Interactive GHSOM. In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on (pp. 2085-2090). IEEE.
[28] Halim, D., & Hansun, S. (2018). Voice Control in Calorie Tracker Application using Levenshtein Distance Algorithm. Aceh International Journal of Science and Technology, 7(1), 1-10.
[29] Nurhayati., & Busman . (2017). Development of document plagiarism detection on Android smartphone. IEEE . https://ieeexplore.ieee.org/document/8089249 .
[30] Wakil, K., Ghafoor, M., Abdulrahman, M., & Tariq, S. (2017). Plagiarism Detection System for the Kurdish.
[31] Lodhi, A., Razzaq, S., & Gull, M. Detecting Urdu Text Plagiarism Using Similarity Matching Techniques.

VIII. BIOGRAPHY

Assistance Professor. Muneer Alsurori is Department Head of Computer Science and Information Technology in Faculty of Science; Ibb University, Yemen. He received the B.sc and M.sc degree in computer science from Sindh University in Pakistan in 1993 and 1995, and Ph.D. in the field of Strategic Information Systems at Univers ity Kebangsaan Malaysia in 2013. Research interests include Information system, SIS, Artificial Intelligence , Data Mining, and Computer Science , Already published several journal papers.

Salah Abdu Al-hagree. Currently Assistant Teacher in Ibb University, Faculty of Science, Department of Computer Science and Information Technology, Received MSc From Department of Computer science, Thamar University in May 2017 and received the BSc degree in computer science from Ibb University in 2002, Ibb, Yemen. His research interests include Artificial Intelligence , Data Mining and Pattern Recognition.

Assoc. Prof. Maher Alsanabani is Deputy Dean for Graduate Studies, Faculty of Computer Science and Information Systems; Thamar University, Yemen. He received his B.Sc. 1996 in Computer Science from Yarmouk University; Jordan, M.Sc. 2002 from University of Technology; Iraq, Ph.D. 2008 from University Putra Malaysia; Malaysia. His research interests are multimedia wireless and mobile networks, resource and mobility managements in mobile radio systems and String Matching …etc. He already published several journal papers.

Sarah Abdulmalik Ali, Obtained a bachelor's degree from the Department of Mathematics and Computer Sciences Excellence in 2018, Faculty of Science, Ibb University ,Yemen.

Noor Alhuda AWdh Alarhabi , Obtained a bachelor's degree from the Department of Mathematics and Computer Sciences Excellence in 2018, Faculty of Science, Ibb University ,Yemen.

Suad Abdu Ali , Obtained a bachelor's degree from the Department of Mathematics and Computer Sciences Excellence in 2018, Faculty of Science, Ibb University ,Yemen,

Khawlah Abdulwahab Meqran. Obtained a bachelor's degree from the Department of Mathematics and Computer Computer Excellence in 2018, Faculty of Science, Ibb University,Yemen.