IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 19, 2020, accepted December 28, 2020, date of publication January 1, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048755

Extended E-N-DIST Algorithm for Alias Detection

MOHAMMED HADWAN, (Member, IEEE)
Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
Intelligent Analytics Group (IAG), College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
Department of Computer Science, College of Applied Sciences, Taiz University, Taiz, Yemen

e-mail: m.hadwan@qu.edu.sa

ABSTRACT Nowadays personal names are not the only way to refer to celebrities and experts from different fields, instead, they can be referred to by their aliases on the web. Associated aliases have remarkable importance in retrieving information about the personal name from the websites. Therefore, disclosing aliases can have an important role in overcoming many real-world challenges. In this research, the aim is to explore and propose a reliable algorithm that can detect aliases that occurred due to transliteration of Arabic names into English. An extension to the Enhanced N-gram distance algorithm (E-N-DIST) which was previously published is introduced in this paper. The proposed algorithm is called the Extended Enhanced N-gram distance algorithm (E-E-N-DIST). The differences between E-N-DIST and E-E-N-DIST are two main changes in calculating the cost of substitution and transposition. First, E-E-N-DIST is computed based on $2^{n+1} - 1$ states. The second is the use of an edit operation called the 'Exchange of Vowels' to count the common spelling errors that happen due to the transliteration from one language to another. The idea of exchange of vowels is to search for vowels (viz. $=$ a', $=$ e', $=$ i', $=$ o', and $=$ u') and the non-vowel character $=$ y' that has a vowel sound or a part of it in other languages to estimate the operations cost of insertion and deletion. The proposed algorithm tested using a dataset for the literature; the results obtained are compared with other algorithms from the state of the art. The proposed algorithm outperforms other algorithms; it achieved a better average percentage of similarity than all other compared algorithms.

INDEX TERMS Alias detection, edit distance (ED), Levenshtein distance (LD), E-N-DIST, dynamic programming.

I. INTRODUCTION

Due to the wide use of the internet and the huge data generated by users every day, detection methods play a vital role in many important domains. Several detecting methods have been explored to detect aliases and fake information in the internet, databases, and other storage methods. Researchers explored different methods and algorithms for detection such as detecting name alias [1], malicious domain detection [2], [3], terrorism intentions detection [4], [5], community intelligence [6], fake website detection [7]–[9], fake news detection [10], [11], fake social media use detection [12], [13], fake comments and reviews detection [14], drug name recognition [15], [16].

Detect names, aliases or strings are very important to many real-world applications related to various domains. If any individual has more than one name, these names are called aliases. Alias detection is a widespread method used to detect names in many areas such as the user's behavior monitoring, databases and marketing, social network analysis, intelligence community, and biology [18]. Alias detection is applied for flag malicious intents, clean data, link knowledge, Passenger Prescreening Systems Assisted by Computers, Biopolitics of Terrorist Watch listing etc. [19]. According to [20], there are two main types of aliases: (i) the first type can be through string similarity for instance ''Mohammed Hadwan'' and ''M. Hadwan'' and (ii) the second type is nicknames that have low or no string similarity such as ''Abu Abdulwahab'' is the nickname for ''Mohammed Hadwan'' refer to Figure 1. In this research, the focus is to explore the first type. Arabic names have no single methodology in place for Latin script representation due to confounding transliteration practices. Using the Latin alphabet, the name of Libyan ex-president Muammar Gaddafi for instance can be written over one hundred ways [19]. These combinations include, for instance, Mouammar Kadhafi, Mu'ammar al-Qadafi, Moamar Gaddafi, and Muammar Qaddafi.

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

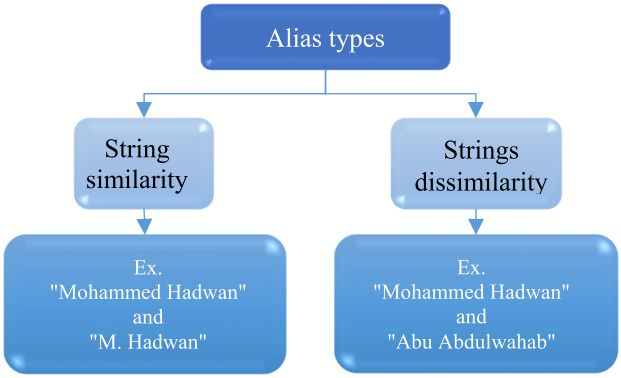


FIGURE 1. Main types of Aliases [20].

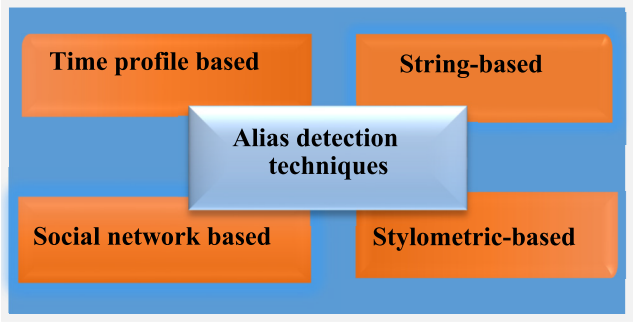


FIGURE 2. Alias detection techniques [13].

For detecting these different name combinations, researchers introduced many algorithms for alias detection. One way to detect alias is to measure the similarity and differences between two strings.

In [13], a classification of alias detection techniques presented based on four main categories: 1) String-based, 2) Stylometric-based, 3), Social network-based and 4) Time profile-based refer to Figure 2.

In this paper, the researcher focuses on investigating techniques for string-based matching focusing on an alias that consists of text strings. The attention is paid to detect aliases variations in Arabic names that occurred due to the transliteration process to English language. Transliteration is the technique of using the words of one language using the alphabets of another language [21].

This paper is organized as follows; Sections II describes the theoretical background. Section III presents the state of the art. Section IV discusses the proposed algorithm. Followed by Section V introduces the experimental study. Section VI and VII devoted for the discussion and conclusion respectively.

II. THEORETICAL BACKGROUND

The focus of this research is on alias string-based matching. If names X and Y have a sequence of size n and m respectively, the Edit Distance (ED) denotes the editing operations minimum cost of deletion, substitution and insertion to convert the sequence of X into Y [22].

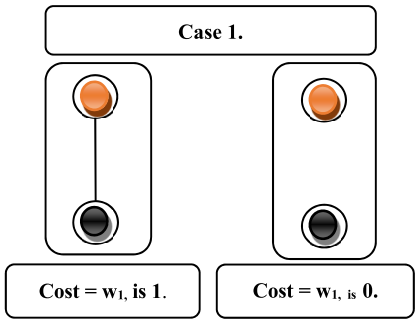


FIGURE 3. Cases of LD algorithm in the two strings.

For instance, the ED of the following medicine name (-Zantac‖ and -Xanax‖) is 3. Because it needs two substitutions (Z → X and c → x) and one deletion (letter t). In this research, editing operations (deletion and insertion) have a cost of one is w_1 and w_2, respectively. Therefore, the edit distance between X and Y is given by Lev(i,j) calculated using equation (1) as follows:

$$\mathrm{Lev}_{s,t}(i,j) = \begin{cases} \mathrm{Max}(i,j) & (i=0 \text{ or } j=0) \\ \mathrm{Min}\begin{cases} \mathrm{Lev}_{s,t}(i,j-1) + \mathrm{w}_1, \mathrm{is1}. \\ \mathrm{Lev}_{s,t}(i-1,j) + \mathrm{w}_2, \mathrm{is1}. \\ \mathrm{Lev}_{s,t}(i-1,j-1) + \mathrm{w}_3 \end{cases} \end{cases} \quad (1)$$

Equation 1 used to compute the substitution cost (replacement, Cr) the cost can be noted Cr. The Cr can have a value within an interval [0.0, 1.0]. Its value is one when the source does not equal the target and it is set to zero otherwise as shown in figure 3 for Case 1 representing match and mismatch characters.

In [23], Damerau–levenshtein distance (DLD) is presented which is relatively similar to the LD algorithm. DLD differs from LD in that it allows for one more transposition edit of two adjacent characters. The DLD algorithm describes the distance between two strings s and t as follows in using recursive relation presented in equation (2).

$$DLev_{s,t}(i,j) = \begin{cases} Max(i,j) & (i=0 \text{ or } j=0) \\ Min\begin{cases} DLev_{s,t}(i,j-1) + \mathrm{w}_1, \mathrm{is1}. \\ DLev_{s,t}(i-1,j) + \mathrm{w}_2, \mathrm{is1}. \\ DLev_{s,t}(i-1,j-1) + \mathrm{w}_3 \\ DLev_{s,t}(i-2,j-2) + \mathrm{w}_4 \end{cases} \end{cases} \quad (2)$$

Equation 1 computes the substitution and transposition cost denoted by (Crt). Crt can assigned interval value between [0.0, 1.0]. The two possible transposition of two adjacent characters representing match and mismatch characters by characters, respectively. Cases are shown in Figure 4.

In [24], a MDLD was introduced and verified using two input strings with multiple characters supported by block transpositions. MDLD applied on the Oracle database with O(N3) of the time complexity. This computes the cost of substitution and transposition denoted by Crt. Crt can be

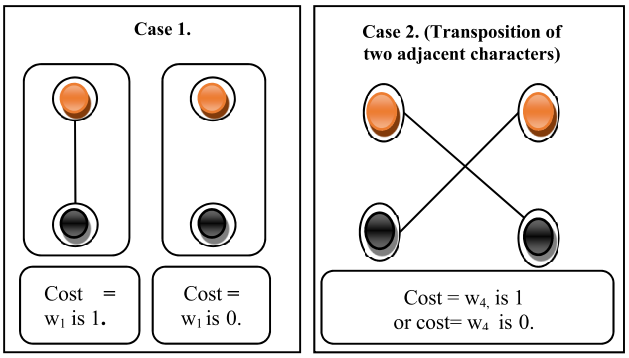


FIGURE 4. Cases of DLD algorithm in the two strings.

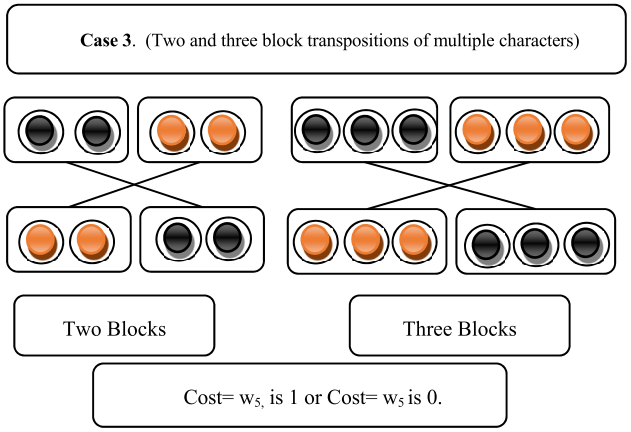


FIGURE 5. Cases of MDLD algorithm in the two 2 strings.

assigned a value within an interval of [0.0, 1.0]. The two possible cases are presented in Figure 5 representing match and mismatch Block by Block respectively.

The N-DIST in [25] proposed by Kondrak, merges features applied by grams of size n for noncrossing-links constraints. For the drug's name, the Initial letter is repeated at the beginning. To clarify its mechanism let us assume that two strings are given: $X = x_1 \ldots x_k$ and $Y = y_1 \ldots y_k$. Let $T_{i;j} = (x_1 \ldots x_k, y_1 \ldots y_k)$ and $T_{ni;j} = (x_{i+1} \ldots x_{i+n}, y_{j+1} \ldots y_{j+n})$. Strings are divided by aligned and compared by forming all possible N consecutive sub-strings letters. The measure of n-gram distance score is introduced in Equation (3) as follows:

$$d_n(T^n_{i,j}) = \frac{1}{n} \sum_{u=1}^{n} d_1(x_{i+u}, y_{j+u}), \tag{3}$$

The $\mathbf{NDIST}_{s,t}$ is used to show the recursive relation between string s and t as introduced in Equation (4) as follows:

$$NDIST_{s,t}(i,j) = \text{Min} \begin{cases} NDIST_{s,t}(i-1,j) + 1. \\ NDIST_{s,t}(i,j-1) + 1. \\ NDIST_{s,t}(i-1,j-1) + d_n\left(T^n_{i,j}\right). \end{cases} \tag{4}$$

For drug names, 3 is the distance between "Zyrtec" and "Zantac.

Affixing and normalization are included in the compared algorithms to measure the distance. To emphasize initial segments, the affixing method is employed to determine the similarity of the words. A unique symbol is defined for every letter of the original alphabet. A prefix is augmented for each word to compose n-1 copies of special symbols corresponds to the initial letter.

For instance, if n = 2 then "Qassim" is transformed into "-Qassim" or if n = 3 then it transferred into "–Qassim", while if n = 4 then Qassim will be transferred into "—Qassim". For this example, a similar cost for edit operation (insert, delete, and replace) will be given by the compared algorithm as it does not give any attention to the letter's similarity. This affects the quality and accuracy of the compared algorithm when applied to English and other languages as well. N-DIST algorithm needs to find the weights set $W = \{\text{weight (w1); w2;...; wn}\}$, using Equation (5).

$$\text{Number of Stats} = 2^n \tag{5}$$

When $n = 2$, Equation (6) is used to compute a set of four weights, i.e. W = {w1 = 0, w2 = 1, w3 = 0.5 and w4 = 0.5}. Similarly, a modified equation can be used for larger values of n, e.g. W = {w1; w2;... ; w8} when n = 3 and W ={w1; w2;... ; w16} when n = 4 and so on.

Different costs for substitution and transposition of two final letters of the word strings last is given by the N-DIST algorithm, the operations are shown in Equation (6).

$$d_n(T^n_{i,j}) = \begin{cases} w_1 = 0, \text{if } (S_{i-1} = T_{j-1}) \text{ and } (S_i = T_j) & \text{Case 1} \\ w_2 = 1, \text{if } \left(S_{i-1} = T_{j-1}\right) \text{ and } \left(S_i = T_j\right) & \text{Case 2} \\ \quad \text{and } (S_{i-1} \neq T_j) \text{ and } (S_i \neq T_{j-1}) & \\ w_3 = 0.5, \text{if } \left(S_{i-1} = T_j\right) \text{ and } \left(S_i = T_{j-1}\right) & \text{Case 3} \\ w_4 = 0.5, (S_{i-1} \neq T_{j-1}) \text{ and } (S_i = T_j) & \text{Case 4} \end{cases} \tag{6}$$

In [26] E-N-DIST algorithm is proposed based on the idea of N-DIST. The E-N-DIST algorithm improved the accuracy of matching names. E-N-DIST needs to find the set of weights W = {w1; w2;...; wn} of the scale of distance as in Equation (7).

$$\text{Number of multiple states} = 2^{n+1} - 1 \tag{7}$$

When n = 2, Equation (8) is used to compute a set of seven weights, i.e. W = {$w_1 = 0$, $w_2 = 1$, $w_3 = 0.5$, $w_4 = 0.5$, $w_5 = 0.5$, $w_6 = 0.5$, $w_7 = 0.5$}. Similarly, a modified equation can be used for larger values of n, e.g. W = {w_1; w_2; ... ; w_{15}} when n = 3 and W = {w_1; w_2; ... ; w_{31}} when n = 4 and so on. The transposition and substitution cost of E-N-DIST algorithm is computed based on the states weights

as introduced in Equation (8).

$$d_n\left(T^n_{i,j}\right) = \begin{cases} w_1 = 0, & \text{if } (S_{i-1} = T_{j-1}) \text{ and } (S_i = T_j) \text{ Case 1} \\ w_2 = 1, & \text{if } (S_{i-1} = T_{j-1}) \text{ and } (S_i \neq T_j) \text{ Case 2} \\ & \text{and } (S_{i-1} \neq T_j) \text{ and} (S_i \neq T_{j-1}) \\ w_3 = 0.5, & \text{if } (S_{i-1} = T_j) \text{ and } (S_i = T_{j-1}) \text{ Case 3} \\ w_4 = 0.5, & \text{if } (S_{i-1} \neq T_{j-1}) \text{ and } (S_i = T_j) \text{ Case 4} \\ w_5 = 0.5, & \text{if } (S_{i-1} = T_{j-1}) \text{ and} (S_i \neq T_j) \text{ Case 5} \\ w_6 = 0.5, & \text{if } (S_{i-1} = T_j) \text{ and } (S_i \neq T_{j-1}) \text{ Case 6} \\ w_7 = 0.5, & \text{if } (S_{i-1} \neq T_j) \text{ and } (S_i = T_{j-1}) \text{ Case 7} \end{cases} \quad (8)$$

To measure the similarity, the proposed algorithm uses the same measures for algorithms (N-DIST, A-N-DIST, and E-N-DIST) as in Equation (9). Where $[\![\text{distance}]\!]_{(S,T)}(i,j)$ denotes the similarity between the strings S_i and T_j. For strings S_i and T_j, the maximum value of the characters contained is denoted by max ($|s_i|$, $|t_j|$).

$$\text{Similarity}_{s,t}(i,j) = 1 - \frac{\text{Distance}_{s,t}(i,j)}{\text{Max}(|s|, |t|)} \quad (9)$$

III. RELATED WORK

In this section, the related work to the proposed algorithm is presented. Detection methods in general and Alias detection in particular has attracted the attention of researchers to investigate, explore, and introduce effective methods and techniques.

In [27], A detection method was developed for aliases in online systems by analyzing the feedback of the users'. Another extension is introduced by [9], [28], to detect the similarity to identify malicious and fraudulent websites. Alias detection combined algorithm is proposed in [29] by taking the advantages of orthographic and semantic information where a method for Multilingual person name recognition and transliteration is introduced. Alias detection for Arabic names is studied in [21] by improving Approximate String Matching (ASM) algorithm, which measures the similarity between two strings (the name and alias). For analysis of intelligence data, a Qualitative Alias Detection was introduced by [30] using Fuzzy Order-of-Magnitude Based on Link Analysis for terrorism-related datasets. Exploration of the name aliases using web mining techniques and the semantic web is presented in [31]. For web and social media, [32] proposed a context-based text mining approach to determine alias names sharing a common name. According to [33], the entity alias detection problem has a closer linking to the problem of data matching. For detecting users that use multiple aliases for the non-concealed case, the similarity of two aliases is used. Based on the state of the art, Edit Distance (ED) measures have been suggested by several researchers. For instance, Damerau-Levenshtein distance (DLD) [23], Levenshtein distance (LD) (also called Edit distance (ED) [22], Modified Damerau-Levenshtein distance (MDLD) algorithm [24], N-gram distance algorithm (N-DIST) by [25] and adjusted N-gram distance [34]. The idea of ED is utilized in this research to calculate the similarity between two strings.

This paper focuses on detect aliases variations in Arabic names that happen because of transliteration to English language. The Arabic language does not use short vowels, which makes it a hard task for exact transliteration to English (Latin alphabets) [21]. One of string edit operation methods called 'exchange of vowels' was introduced by[35] to identify errors occurs due to the transliterations The idea of exchange of vowels' is to list vowels (viz. 'a', 'e', 'i', 'o', and 'u') in addition to the non-vowel character 'y' that sounds like a vowel or a part of vowels in some languages as in Danish and Swedish languages. This helps to detect the most commonly occurred typographic errors effectively.

Branting in [36] deliberated the effect of various types of spelling variations that make alias detection an extremely difficult task by enumerating several types of orthographic variations for aliases. These variations are: name permutations, cross-lingual transliterations, misspelling, titles, phonetic similarities, name changes, nicknames, identifying phrases, and omissions. Alias detection poses several issues for the English and Arabic language in the A-N-DIST algorithm [34]. Therefore, a novel algorithm based on the idea of E-N-DIST is introduced which enhanced the accuracy of alias detection is introduced by [26]. The proposed algorithm called the Extended Enhanced N-gram distance algorithm E-E-N-DIST.

Kondark in [25] proposed the orthographic N-DIST algorithm, which have been successfully used in several English language-based applications. An enhancement for original N-DIST algorithm focusing on Arabic language called DIST-A is introduced in [34]. For alias detection, adjusted N-DIST algorithm (A-N-DIST) is introduced based on the idea of N-gram Distance for detecting Arabic names aliases that occurred due to transliteration variations [37]. A public dataset was used to test A-N-DIST, the results showed that, A-N-DIST outperformed other methods from the state of the art. Further investigation to improve the weakness found in A-N-DIST, Enhanced N-gram Distance (E-N-DIST) were developed by [26], the obtained results using E-N-DIST was better than the compared algorithms using the same dataset. Therefore, the research presented in this paper aims to further extend the E-N-DIST algorithm in [26] by considering multiple states of transposition operation in order to deal with different errors. In addition, to use the idea of 'exchange of vowels' (a, e, i, o, u, y) that helps in increase the detection accuracy. The proposed algorithm is discussed in section IV.

IV. EXTENDED E-N-DIST ALGORITHM

Extended E-N-DIST algorithm, which is called the E-E-N-DIST algorithm, is introduced in this section. The main difference between the proposed algorithm and the E-N-DIST algorithm [26] is twofold. First, the cost of

substitution and transposition in E-E-N-DIST is computed according to $2^{n+1} - 1$ states according to Equation (7) in addition to using 'exchange of vowels' (a, e, i, o, u, y). Second, E-E-N-DIST depends on the number of states divided by n and 'exchange of vowels' to estimate the operations cost of insertion and deletion. The ultimate goal of E-E-N-DIST is to furtherly enhancing the accuracy of detecting alias names. This was done by considering different types of spelling errors in English language. The 'exchange of vowels' (a, e, i, o, u, y) is a new edit operation added to E-E-N-DIST for computing the cost of the transposition and substitution operations.

This edit operation helps in finding the most common orthographic and typographical errors in personal names. It counts the most frequent spelling errors of vowels that occur when names are converted from one language to another.

The substitution and transposition of vowels in names are ignorable compared to dictionary words. For instance, (osama, usama) and (some, same) are two string pairs with one vowel difference only. However, it can be noticed that the difference in pair 1 (osama, usama) can be ignored due to the fact that the exchange in vowels does not change the meaning. While in the other case, the (some, same) the meaning is change completely due to changing the vowels. Based on this observation, the 'exchange of vowels' edit operation is introduced to detect the name aliases efficiently. This led us to consider any two names that differ only in vowels to be assumed as aliases. Due to the case of the Arabic language, where no possibility of writing short vowels, such aliases may occur in the vowel variations. This makes the vowel action process essential to insert short vowels in the target language, which is the English language.

Therefore, E-E-N-DIST employs the edit operation known as 'exchange of vowels' to detect these types of name variations (errors). This operation is more tolerant of swapping and substitution of vowels rather than considering the list by giving less penalty cost (0.5 in this work). Such cost penalty reduction (especially for names in Arabic) resulted in scores similarity of the name-alias pairs as described in section 2.1. The function for computing the cost of substitution and transposition operations of exchange of vowels is present in figure 6. The function is initialized to n then it depends upon the state's condition where the cost is decreased by 1. It is noted that the conditions if (ni< n - 1) and if (ni >= n - 1) are both considered to achieve the symmetry property, e.g. the distance between 'abdal muaz' and 'abdal muiz' is the same as the distance between 'abdal muiz' and 'abdal muaz'. Figure 6. Presents the function to compute the cost of substitution and transposition operations for the exchange of vowels'.

The Function to compute the Cost of exchange of vowels

```
Input: N-gram Letters (Letters1 and Letters2)
Output: Cost c n-gram Distance (cost_s_t)
Decimal Substitution and Transposition Op (so_name [ ] , tar_name[ ])
 if (ni< n - 1)
 {
   if (so_name[i - 1 + ni] == tar_name[ni])
        {cost_s_t --;}
   else if (((so_name[i - 1 + ni] == 'a') || (so_name[i - 1 + ni] == 'e') || (so_name[i -
        1 + ni] == 'i') || (so_name[i - 1 + ni] == 'o') || (so_name[i - 1 + ni] == 'u')
        || (so_name[i - 1 + ni] == 'y')) || ((tar_name[ni] == 'a') || (tar_name[ni] ==
        'e') || (tar_name[ni] == 'i') || (tar_name[ni] == 'o') || (tar_name[ni] == 'u')
        || (tar_name[ni] == 'y')))
        {cost_s_t = ncost;}
   else if (so_name[i - 1 + ni] == tar_name[ni + 1])
        {cost_s_t --;}
   else if (((so_name[i - 1 + ni] == 'a') || (so_name[i - 1 + ni] == 'e') || (so_name[i -
        1 + ni] == 'i') || (so_name[i - 1 + ni] == 'o') || (so_name[i - 1 + ni] == 'u')
        || (so_name[i - 1 + ni] == 'y')) || ((tar_name[ni + 1] == 'a') || (tar_name[ni
        + 1] == 'e') || (tar_name[ni + 1] == 'i') || (tar_name[ni + 1] == 'o') ||
        (tar_name[ni + 1] == 'u') || (tar_name[ni + 1] == 'y'))) {cost_s_t = ncost;}
   else if (so_name[i - 1 + ni+1] == tar_name[ni])
        {cost_s_t --;}
   else if (((so_name[i - 1 + ni + 1] == 'a') || (so_name[i - 1 + ni + 1] == 'e') ||
        (so_name[i - 1 + ni + 1] == 'i') || (so_name[i - 1 + ni + 1] == 'o') ||
        (so_name[i - 1 + ni + 1] == 'u') || (so_name[i - 1 + ni +1] == 'y')) ||
        ((tar_name[ni] == 'a') || (tar_name[ni] == 'e') || (tar_name[ni] == 'i') ||
        (tar_name[ni] == 'o') || (tar_name[ni] == 'u') || (tar_name[ni] == 'y')))
        {cost_s_t = ncost;}
 }
 else if (ni>= n - 1)
 {
   if (so_name[i - 1 + ni] == tar_name[ni])
        {cost_s_t --; }
   else if (((so_name[i - 1 + ni] == 'a') || (so_name[i - 1 + ni] == 'e') || (so_name[i -
        1 + ni] == 'i') || (so_name[i - 1 + ni] == 'o') || (so_name[i - 1 + ni] == 'u')
        || (so_name[i - 1 + ni] == 'y')) || ((tar_name[ni] == 'a') || (tar_name[ni] ==
        'e') || (tar_name[ni] == 'i') || (tar_name[ni] == 'o') || (tar_name[ni] == 'u')
        || (tar_name[ni] == 'y')))
        {cost_s_t = ncost;}
   else if (so_name i - 1 + ni] == tar_name[ni - 1])
        { cost--;}
   else if (((so_name[i - 1 + ni] == 'a') || (so_name[i - 1 + ni] == 'e') || (so_name[i -
        1 + ni] == 'i') || (so_name[i - 1 + ni] == 'o') || (so_name[i - 1 + ni] == 'u')
        || (so_name[i - 1 + ni] == 'y')) || ((tar_name[ni - 1] == 'a') || (tar_name[ni
        - 1] == 'e') || (tar_name[ni - 1] == 'i') || (tar_name[ni - 1] == 'o') ||
        (tar_name[ni - 1] == 'u') || (tar_name[ni - 1] == 'y')))
        {cost_s_t = ncost;}
   else if (so_name [i - 1 + ni-1] == tar_name [ni] )
        {cost_s_t --;}
   else if (((so_name[i - 1 + ni-1] == 'a') || (so_name[i - 1 + ni-1] == 'e') || (so_name[i
        - 1 + ni-1] == 'i') || (so_name[i - 1 + ni-1] == 'o') || (so_name[i - 1 + ni-1]
        == 'u') || (so_name[i - 1 + ni-1] == 'y')) || ((tar_name[ni] == 'a') ||
        (tar_name[ni] == 'e') || (tar_name[ni] == 'i') || (tar_name[ni] == 'o')
        ||(tar_name[ni] == 'u') || (tar_name[ni] == 'y')))
        {cost_s_t = ncost;}
 }
 Return Cost
```

FIGURE 6. The function to compute the cost of substitution and transposition operations for exchange of vowels'.

A. DATASETS

A collection of datasets of Arabic names transliterate into English are used to evaluate E-E-N-DIST for Alias detection. As there is no available standard dataset for alias detection, two datasets have been used in this research where it extracted from open source web-page based on '20 Ground Truth Entities', refer to [38].

Each dataset includes different possible variations for aliases with typographical and spelling errors of the same names. All kinds of variation have been collected and considered.

For a comparison purpose, we use the same datasets that include 10 pairs that were used in [34], [35] to do the experiments and comparison for E-E-N-DIST.

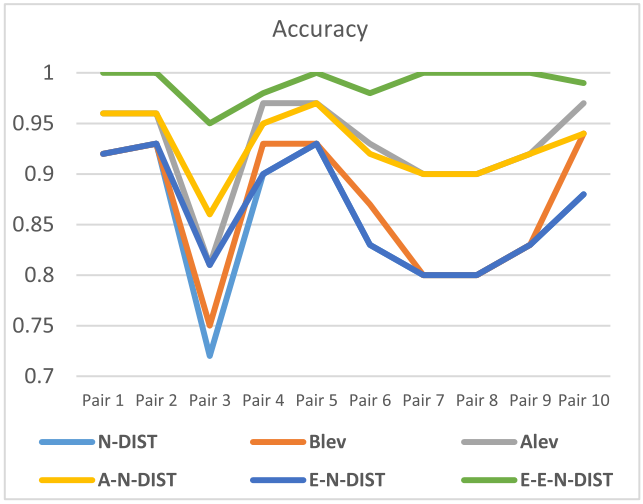


FIGURE 7. The Comparison between proposed algorithm and compared algorithms.

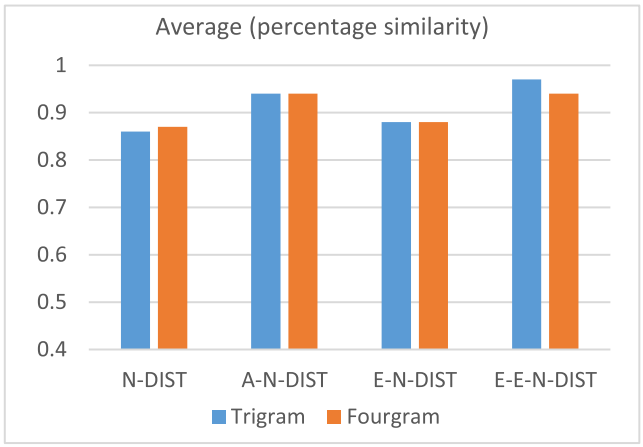


FIGURE 8. Comparison between algorithms with trigram (tri = 3) and fourgram (four = 4).

V. THE EXPERIMENTAL STUDY

As this research is an extension of previous work, the proposed algorithm is compared against the original N-DIST [25], A-N-DIST algorithm [34], and E-N-DIST [26] in addition to Basic Levevshtein (BLev) and Adjusted Levenshtein (ALev) in [35].

Table 1 and figure 7 present 10 pairs of names that were used to evaluate the performance of E-E-N-DIST and other algorithms. Based on the obtained results, it is clear that the E-E-N-DIST algorithm gives the best results compared to other algorithms, especially when comparing names transposition as for the name in rows 1 and 4.

Datasets preparation and experimental results obtained by E-E-N-DIST are introduced and analyzed along with a comparison against different algorithms from the literature.

Table 2 and figure 8 present the accuracy comparison of the percentage similarity between E-E-N-DIST and other algorithms using trigram (tri = 3) and fourgram (four = 4).

TABLE 1. Comparison between algorithms.

	String		Compared Algorithms					Proposed Algorithm
			N-DIST	BLev	ALev	A-N-DIST	E-N-DIST	E-E-N-DIST
1	abu abdallah	abu abdalluh	0.92	0.92	0.96	0.96	0.92	**1.00**
2	mujahid shaykh	mujahid shaikh	0.93	0.93	0.96	0.96	0.93	**1.00**
3	hussein al-sheik	hassan ali-sheik	0.72	0.75	0.81	0.86	0.81	**0.95**
4	osama bin laden	usama bin laden	0.90	0.93	0.97	0.95	0.90	**0.98**
5	usama bin ladin	usama bin laden	0.93	0.93	0.97	0.97	0.93	**1.00**
6	usama bin laden	osama bin ladin	0.83	0.87	0.93	0.92	0.83	**0.98**
7	abdel muaz	abdul muiz	0.80	0.80	0.90	0.90	0.80	**1.00**
8	abdal muaz	abdel muiz	0.80	0.80	0.90	0.90	0.80	**1.00**
9	abu mohammed	abu muhammad	0.83	0.83	0.92	0.92	0.83	**1.00**
10	ayman al-awahari	ayman al-zawahiri	0.88	0.94	0.97	0.94	0.88	**0.99**
AVERAGE of SIMILARITY PERCENTAGE			0.85	0.87	0.93	0.93	0.86	**0.99**

TABLE 2. Comparison between algorithms with trigram (tri = 3) and fourgram (four = 4).

Algorithms	Compared Algorithms						Proposed Algorithm	
	N-DIST		A-N-DIST		E-N-DIST		E-E-N-DIST	
Trigram and fourgram	n=3	n=4	n=3	n=4	n=3	n=4	n=3	n=4
AVERAGE Of SIMILARITY PERCENTAGE	0.86	0.87	0.94	0.94	0.88	0.88	**0.97**	**0.94**

VI. DISCUSSION

According to the obtained results in table 1, the E-E-N-DIST algorithm is shown to be sensitive to replacement as presented in rows 2, 3, 5, and 6. In addition, for repeated letters, deletion and dictation errors, E-E-N-DIST handles these situations in a perfect manner compared to other algorithms used for this evaluation as displayed in table 1, rows 7, 8, 9, and 10. Therefore, when comparing the E-E-N-DIST to other compared algorithms, it provides accurate results for all instances in table1 for all tested pairs.

The accuracy of the percentage similarity is also used for a comparison between E-E-N-DIST and other algorithms using trigram (tri = 3) and fourgram (four = 4) as shown in table 2. According to table 2, E-E-N-DIST gets an accuracy of 97.0% when (Tri = 3) and 94.0% when (Four = 4) respectively. While N-DIST and A-N-DIST and E-N-DIST algorithms get 86%, 94%, 88%, when (Tri = 3) and 87%, 94%, 88%, when (Four = 4) respectively. This because E-E-N-DIST was taken into account the characteristics and unique features of 'exchange of vowels' (a, e, i, o, u, y) for alias detection while other A-N-DIST and E-N-DIST algorithms not. Even

Blev and Alev used the exchange of vowel idea, E-E-N-DIST shows to perform much better due to considering multiple states of transposition operation in order to deal with different errors.

VII. CONCLUSION

Aliases detection gain the researcher's attention to introduce reliable and accurate algorithms. In this research, an attempt to increase the detection accuracy of aliases that occurs due to the transliteration errors, happen when using English letters to write Arabic names. This research presented a novel alias detection algorithm called E-E-N-DIST. E-E-N-DIST evaluated using a public dataset from the literature for alias detection. The proposed algorithm outperformed other comparing algorithms by increasing the percentage similarity. This proposed algorithm can play a big role in increasing the accuracy of information and data retrieval. Especially when it comes to detecting the aliases of common strings or names, such as the names of experts and celebrities in various fields that may have been referred to by their personal names or aliases on the web and social media.

Other benefits of the proposed algorithm can be gained by local and global communities in different areas, such as the intelligence community, databases search, social network analysis, biology, medical, marketing. In addition to data mining, extraction of new features or new items to find people's interests, tendencies, and problems based on the real names or aliases. Besides, the proposed algorithm can be employed in the data warehouse development, especially in the data cleaning process, the unification of data items a step forward to support the data integration in data warehouses. Further investigation of other methods to detect aliases is highly suggested. For instance, monarch butterfly, and moth search (MS) algorithm.

ACKNOWLEDGMENT

The author would like to thank the Deanship of Scientific Research, Qassim University for funding publication of this project.

REFERENCES

[1] D. Tam, N. Monath, A. Kobren, A. Traylor, R. Das, and A. McCallum, "Optimal transport-based alignment of learned character representations for string similarity," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5907–5917, doi: 10.18653/v1/p19-1592.

[2] C. Peng, X. Yun, Y. Zhang, S. Li, and J. Xiao, "Discovering malicious domains through alias-canonical graph," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 225–232, doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.241.

[3] R. Perdisci, I. Corona, and G. Giacinto, "Early detection of malicious flux networks via large-scale passive DNS traffic analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 5, pp. 714–726, Sep./Oct. 2012, doi: 10.1109/TDSC.2012.35.

[4] S. A. Azizan and I. A. Aziz, "Terrorism detection based on sentiment analysis using machine learning," *J. Eng. Appl. Sci.*, vol. 12, no. 3, pp. 691–698, 2017.

[5] E. Rohn and G. Erez, "A framework for agro-terrorism intentions detection using overt data sources," *Technol. Forecasting Social Change*, vol. 80, no. 9, pp. 1877–1884, Nov. 2013, doi: 10.1016/j.techfore.2013.06.008.

[6] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, Jan. 2010, doi: 10.1016/j.asoc.2009.06.019.

[7] N. Abdelhamid, F. Thabtah, and H. Abdel-Jaber, "Phishing detection: A recent intelligent machine learning comparison based on models content and features," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 72–77, doi: 10.1109/ISI.2017.8004877.

[8] M. S. Sadi, M. M. R. Khan, M. M. Islam, S. B. Srijon, and M. M. H. Mia, "Towards detecting phishing Web contents for secure Internet surfing," in *Proc. Int. Conf. Informat., Electron. Vis. (ICIEV)*, May 2012, pp. 237–241, doi: 10.1109/ICIEV.2012.6317372.

[9] A. Abbasi, Z. Zhang, D. Zimbra, H. Chen, and J. F. Nunamaker, "Detecting fake websites: The contribution of statistical learning theory," *MIS Quart.*, vol. 34, no. 3, p. 435, 2010, doi: 10.2307/25750686.

[10] Á. Veszelszki, "Linguistic and non-linguistic elements in detecting (Hungarian) fake news," *Acta Univ. Sapientiae Commun.*, vol. 4, no. 1, pp. 7–35, 2018, doi: 10.1515/auscom-2017-0001.

[11] A. Školkay and J. Filin, "A comparison of fake news detecting and fact-checking AI based solutions," *Studia Medioznawcze*, vol. 20, no. 4, pp. 365–383, Dec. 2019, doi: 10.33077/uw.24511617.ms.2019.4.187.

[12] T. Agrawal, R. Gupta, and S. Narayanan, "Multimodal detection of fake social media use through a fusion of classification and pairwise ranking systems," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 1045–1049, doi: 10.23919/EUSIPCO.2017.8081367.

[13] F. Johansson, L. Kaati, and A. Shrestha, "Detecting multiple aliases in social media," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2013, pp. 1004–1011, doi: 10.1145/2492517.2500261.

[14] J. N. Nandimath, B. S. Katkar, V. U. Ghadge, and A. N. Garad, "Efficiently detecting and analyzing spam reviews using live data feed," *Int. Res. J. Eng. Technol.*, vol. 4, no. 2, pp. 1421–1424, 2017.

[15] I. Segura-Bedmar, P. Martínez, and M. Segura-Bedmar, "Drug name recognition and classification in biomedical texts," *Drug Discovery Today*, vol. 13, nos. 17–18, pp. 816–823, Sep. 2008, doi: 10.1016/j.drudis.2008.06.001.

[16] C. Eduardo, "Soft bigram similarity to identify confusable drug names," in *Proc. Mex. Conf. Pattern Recognit.*, 2019, pp. 433–442, doi: 10.1007/978-3-030-21077-9.

[17] F. Johansson, L. Kaati, and A. Shrestha, "Timeprints for identifying social media users with multiple aliases," *Secur. Informat.*, vol. 4, no. 1, p. 7, Dec. 2015, doi: 10.1186/s13388-015-0022-z.

[18] P. Pantel, "Alias detection in malicious environments," in *Proc. AAAI Fall Symp., Capturing Using Patterns Evidence Detection*, 2006, pp. 14–20.

[19] G. Kafer, "Big data biopolitics," *Digit. Culture Soc.*, vol. 5, no. 1, pp. 23–42, Dec. 2019, doi: 10.14361/dcs-2019-0103.

[20] N. An, L. Jiang, J. Wang, P. Luo, M. Wang, and B. N. Li, "Toward detection of aliases without string similarity," *Inf. Sci.*, vol. 261, pp. 89–100, Mar. 2014, doi: 10.1016/j.ins.2013.11.010.

[21] M. Shaikh, N. Memon, and U. K. Wiil, "Extended approximate string matching algorithms to detect name aliases," in *Proc. IEEE Int. Conf. Intell. Secur. Informat.*, Jul. 2011, pp. 216–219, doi: 10.1109/ISI.2011.5984085.

[22] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Phys. Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[23] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Commun. ACM*, vol. 7, no. 3, pp. 171–176, Mar. 1964, doi: 10.1145/363958.363994.

[24] T. Rees, "Taxamatch, an algorithm for near ('fuzzy') matching of scientific names in taxonomic databases," *PLoS ONE*, vol. 9, no. 9, Sep. 2014, Art. no. e107510, doi: 10.1371/journal.pone.0107510.

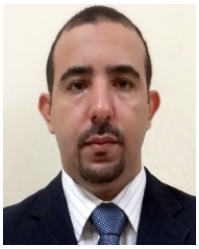
[25] G. Kondrak, "N-gram similarity and distance," in *Proc. Int. Symp. String Process. Inf. Retr.*, 2005, pp. 115–126, doi: 10.1007/11575832_13.

[26] S. Al-Hagree, M. Al-Sanabani, M. Hadwan, and M. A. Al-Hagery, "An improved N-gram distance for names matching," in *Proc. 1st Int. Conf. Intell. Comput. Eng. (ICOICE)*, Dec. 2019, pp. 1–7, doi: 10.1109/ICOICE48418.2019.9035154.

[27] A. Abbasi, H. Chen, and J. F. Nunamaker, "Stylometric identification in electronic markets: Scalability and robustness," *J. Manage. Inf. Syst.*, vol. 25, no. 1, pp. 49–78, Jul. 2008.

[28] A. Abbasi and H. Chen, "A comparison of tools for detecting fake websites," *Computer*, vol. 42, no. 10, pp. 78–86, 2009, doi: 10.1109/MC.2009.306.

[29] B. Pouliquen, R. Steinberger, C. Ignat, I. Temnikova, A. Widiger, W. Zaghouani, and J. Žižka, "Multilingual person name recognition and transliteration," *CORELA-Cognition, Represent., Lang.*, vol. 3, no. 3, pp. 1–25, 2005.
[30] Q. Shen and T. Boongoen, "Fuzzy orders-of-magnitude-based link analysis for qualitative alias detection," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 649–664, Apr. 2012, doi: 10.1109/TKDE.2010.255.
[31] D. Bollegala, T. Honma, Y. Matsuo, and M. Ishizuka, "Automatically extracting personal name aliases from the Web," in *Proc. Int. Conf. Natural Lang. Process.*, 2008, pp. 77–88, doi: 10.1007/978-3-540-85287-2_8.
[32] T. Anwar and M. Abulaish, "Namesake alias mining on the Web and its role towards suspect tracking," *Inf. Sci.*, vol. 276, pp. 123–145, Aug. 2014, doi: 10.1016/j.ins.2014.02.050.
[33] P. Christen, "A comparison of personal name matching: Techniques and practical issues," in *Proc. 6th IEEE Int. Conf. Data Mining-Workshops (ICDMW)*, Hong Kong, 2006, pp. 290–294, doi: 10.1109/ICDMW.2006.2.
[34] M. Alsurori, M. Al-Sanabani, and S. Al-Hagree, "Design an accurate algorithm for alias detection," *Int. J. Inf. Eng. Electron. Bus.*, vol. 10, no. 3, pp. 36–44, May 2018, doi: 10.5815/ijieeb.2018.03.05.
[35] M. Shaikh, H. Dar, A. Shaikh, and A. Shah, "Adjusted edit distance algorithm for alias detection," in *Proc. IPCSIT*, 2012, pp. 1–5.
[36] L. K. Branting, "A comparative evaluation of name-matching algorithms," in *Proc. 9th Int. Conf. Artif. Intell. Law*, 2003, pp. 224–232, doi: 10.1145/1047788.1047837.
[37] S. Al-Hagree, M. Al-Sanabani, K. M. A. Alalayah, and M. Hadwan, "Designing an accurate and efficient algorithm for matching arabic names," in *Proc. 1st Int. Conf. Intell. Comput. Eng. (ICOICE)*, Dec. 2019, pp. 1–12, doi: 10.1109/ICOICE48418.2019.9035184.
[38] P. Hsiung, A. Moore, D. Neill, and J. Schneider, "Alias detection in link data sets," in *Proc. Int. Conf. Intell. Anal.*, 2005, pp. 1–8.



MOHAMMED HADWAN (Member, IEEE) received the B.Sc. degree in computer science from National University, Yemen, in 2003, the M.Sc. degree in computer science from the University of Science, Malaysia, in 2006, and the Ph.D. degree in computer science, (artificial intelligence AI) from The National University of Malaysia, in 2011. He was the Head of the Department of Computer Science, College of Applied Sciences, Taiz University, Yemen, and was the Dean of the College of Engineering and Information Technology, Al Saeed University, Taiz, Yemen. He is currently working as a Senior Lecturer with the Department of Information Technology, College of Computer, Qassim University. He is also a Yemeni Researcher. He is passionate about how technology can help in solving complex optimization problems, such as timetabling problems. His current research interests include image processing, big data, the Internet of Things, smart cities, and machine learning.