I.J. Information Engineering and Electronic Business, **2018, 3, 36-44**
Published Online May 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2018.03.05

Design an Accurate Algorithm for Alias Detection

Muneer Alsurori
Ibb University/Faculty of Sciences /Department of Computer Sciences & Information Technology, Ibb, Yemen
Email: msurory@yahoo.com.

Maher Al-Sanabani and Salah AL-Hagree
Thamar University/2Faculty Computer Science and Information Systemst, Thamar,Yemen
Email: M.sanabani@gmail.com, s.alhagree@gmail.com.

Received: 01 November 2017; Accepted: 08 January 2018; Published: 08 May 2018

***Abstract*—**An improvement in detection of alias names of an entity is an important factor in many cases like terrorist and criminal network. Accurately detecting these aliases plays a vital role in various applications. In particular, it is critical to detect the aliases that are intentionally hidden from the real identities, such as those of terrorists and frauds. Alias Detection (AD) as the name suggests, a process undertaken in order to quantify and identify different variants of single name showing up in multiple domains. This process is mainly performed by the inversion of one-to-many and many-to-one mapping. Aliases mainly occur when entities try to hide their actual names or real identities from other entities i.e.; when an object has multiple names and more than one name is used to address a single object. N-gram distance algorithm (N-DIST) have find wide applicability in the process of AD when the same is based upon orthographic and typographic variations. Kondrak approach, a popular N-DIST works well and fulfill the cause, but at the same time we uncover that (N-DIST) suffers from serious inabilities when applied to detect aliases occurring due to the transliteration of Arabic name into English. This is the area were we have tried to hammer in this paper. Effort in the paper has been streamlined in extending the N-gram distance metric measure of the approximate string matching (ASM) algorithm to make the same evolve in order to detect aliases which have their basing on typographic error. Data for our research is of the string form (names & activities from open source web pages). A comparison has been made to show the effectiveness of our adjustment to (N-DIST) by applying both forms of (N-DIST) on the above data set. As expected we come across that adjusted (A-N-DIST) works well in terms of both performance & functional efficiency when it comes to matching names based on transliteration of Arabic into English language from one domain to another.

***Index Terms*—**Alias Detection (AD), N-gram Distance, Transliteration, Name Matching.

I. INTRODUCTION

Monitoring and analysis of web forums is becoming important for intelligence analysts around the globe since terrorists and extremists are using forums for spreading propaganda and communicating with each other. Due to this associative phenomenon, the Alias analysis is perhaps one of the most crucial and widely used analyses, and has attracted tremendous research efforts over the years but a problem related to this is that individuals can make use of several aliases. A problem with such a content analysis is that it is not unusual that individuals make use of several aliases on a single web forum or on different social media sites, making it harder to make correct assessments. As an example, the Norwegian right-wing extremist and lone-wolf terrorist Anders Behring Breivik made use of several aliases on various social media sites before his attacks in Norway 2011 [1].The use of several aliases can be perfectly normal, but can become a problematic issue when utilizing content-based analysis. To overcome this problem, we propose a number of matching techniques that can be used to identify users with multiple aliases. The obtained experimental results suggest that the combination of matching techniques can give significantly better results than if the techniques are applied individually. We also show that the achieved accuracy is largely dependent upon the number of aliases under consideration.

The problem of alias detection is very broad. In another variant of this problem, one name corresponds to many entities. For example the name Michael Jordan represents a statistician and a sports figure as well as many others who share that name. Various methods that address this problem are discussed in Neill (2002) and Jurafsky and Martin (2000) [2 &3].

The important workers who contributed in natural language processing focuses on entities include those of Muhammad Ghafoor et al 2017 who presented a good literature review about Kurdish Script Languages (TSL).

In their review, they introduced a new system for plagiarism detection for Kurdish Language, based on n-gram algorithm, that can detect the word, phrases, and paragraphs. Moreover, this system effectiveness for detect plagiarist texts in local host and online especially in Google search engine. This system is more useful for the academic organizations such as schools, institutes, and universities for finding copied texts from another document the plagiarism detection techniques [4]. Prianka Mandal and B M Mainul Hossain presented a systematic literature review on checking and correcting spelling errors in Bangla language. Their investigate the current methods used for spell checking and find out what challenges are addressed by those methods. We also report the limitations of those methods. Recent relevant studies are selected based on a set of significant criteria. Their results indicate that there are research gaps in this research topic and has a potential for further investigation [5]. Ibrahim et al., 2017 presented a good literature review about Arabic Script Languages (ASL). In their review, they introduced the plagiarism detection techniques per year. They reviewed all publications from 2009 to 2017, as their results plagiarism detection techniques widely used for that language. Moreover, Ibrahim et al. presented the techniques that used for ASLs based on their review, most techniques used for Arabic language, then Persia and Urdu languages, but there is no publication exist for the Kurdish language [6].

Moreover wide range of research in natural language processing focuses on entities. These range from basic language tasks like coreference resolution to broader aggregation applications like sentiment analysis and information extraction. Building an accurate picture of an entity (e.g., aggregate sentiment toward the entity, entity tracking across websites, database population) requires an understanding of all the varying ways people refer to that entity. Tracking “Facebook” is not enough to know how people feel about it, as mentions of “fbook”, “FB”, and “the book” also need to be understood. Although many applications exist for tracking known mentions of entities, less research exists for detecting nicknames and aliases.

In other hand, aliases can also be formulated intentionally with a malicious or mischief plan in mind. This brand of aliases is most wicked and tough in terms of detecting them completely as they are created deliberately by playing with names and personal information. To find a quantifiable mapping criterion is still found to be an uphill task. This class of aliases is referred to semantic errors. Aliases can be based upon various underlying phenomenon such as typographic variations, semantic variations or orthographic & other resulting from their combined existence in the data set (Bilenko et al., 2003; Ning et al., 2014) [7&8]. The process of aliases detection must be inherently automated to utmost degree possible in order to improve the efficiency with regard to functionality & performance of the system. The detection of aliases is still an open area for research which inherits till date, many issues which have not been addressed completely. For detection of aliases occurring due to the typographic variations, edit distance metrics scale well enough but suffer from serious inabilities to detect aliases when based upon other types of variations (Branting et al., 2005) [9]. A enhancement of N-gram Distance (a popular N-DIST) form the central position in the paper. Results based on analytical modeling and measurement which proves the effectiveness of the enhance. Hybrid method product from component Edit Distance and N–gram algorithms for Matching Names but using languages other than Arabic such as English etc.. (N-DIST) algorithm over the basic one also find space in the paper.

In this paper, we propose an extension to widely used (N-DIST) algorithms to detect vowel variations of Arabic names including other types of typographic variations. This paper is organized as follows. Section 2 illustrates some challenges of Alias detection. Section 3 describes the related work in field of study. Section 4 demonstrates the proposed algorithm for Alias Detection. Section 5 presents the experimental results and discussions. Finally conclusions and future work are presented in Section 6..

II. The face of Alias Detection

In this paper, we aim to detect aliases that occur due to transliteration variations in Arabic names. "Transliteration is the process of representing words from one language using the alphabet or writing system of another language" (Shaikh et al., 2012) [10]. Exact transliteration of the Arabic names to English (Latin alphabets) is a challenging task due to the fact that short vowels are not written in Arabic (Shaikh et al. ,2012)[10]. Branting (Branting et al., 2005) discussed reasons that make Alias Detection a challenging task due to various types of spelling variations. Branting describes eight types of aliases based on orthographic variations: cross-lingual transliterations, misspelling, phonetic similarities, nicknames, titles, name changes, identifying phrases, name permutations, and omissions [9]. The Alias Detection poses several issues for English and Arabic language in following aspect (Alhagree et al., 2016;, Ahagree, Master’s thesis,2017) [11&12]:

The first issue, the reasons for appearing different include typing and OCR errors such as “Usama” is misspelled as “Usarna”.

The second issue, the cause of these errors to come as result of the keyboard adjacencies such as “Osama” is misspelled as “Usama”.

The third issue, is that if duplicate letter, repeated just in pronunciation such as “Barack obama” is misspelled as “Barrack Obama” and “Rajinikanth ” is misspelled as “. Raajinikanth”

The fourth issue, is that if deletion letter, removed just in pronunciation such as “Sylvester stallone” is misspelled as “Sylvester stalone” and “embarrass” is misspelled as “embarrass”.

The fifth issue, the insertion or deletion cost of a blank has been defined to be equals to zero beneficial to segmentations which might occur in names. Thus if a blank appears accidentally inside a name such as “the letter” is misspelled as “the letter”, “sylvesterstellone” is misspelled as “Sylvester stallone”, “ابو بكر” is misspelled as “ابوبكر” and “معمر القذافي” is misspelled as “معمرالقذافي ”.This enhancement is based on the observation that typographical variations occur more commonly due to transliteration or cultural difference. For example, "Osama bin Laden" and "Usama bin Ladan" are two strings, S1 and S2, respectively.

(Cross-Lingual Transliterations)There are certain names that can be changed phonetically and or structurally when transferred to a different language, e.g. “Joseph” in English is equivalent to the Italian name “Giuseppe” and is equivalent to the Arabic name “يوسف” and “father” in English is equivalent to the German “vater” and “far” in Norwegian. (Nicknames)There are some people who have more than one name such as pet name and nick name. For instance, one person can be called by his pet name or nick name instead if his first name or last name. In some cultures, the last name of a woman is changed to her husband's last name. Moreover, there are some people who can change their names during their lives.

(Phonetic Similarities) Phonetic error can considered as subcategory of cognitive errors. This can be occur when the writer confused between how the word is pronounced and how it is written , the writer substitutes letters into a word because he mistakenly mispronounced the word that lead him to misspelling the word. On the other hand, there are words which have the same pronunciation but different spelling.

In other cases , Punctuation can be used as a way to separate the parts of the names e.g., “ Owens Corning ” vs. “ Owens - Corning ” ; “ IBM ” vs. “ I.B.M. ”.

III. Related Work

The problem of entity alias detection has a close connection with the data matching problem (Christen et al., 2012) [13]. A brief survey of the related work in this research direction is presented below.

(Levenshtein, 1966) Levenshtein Distance (LD) introduces Edit distance algorithm which is used for (Pattern Matching) string processing [14]. This algorithm measures the difference between two string sequences. Levenshtein Distance counts the minimum number of single-character edits (Insertion, Substitution and Deletion) required to change name into the alias, where the cost of substitution is the same as the cost of insertion or deletion, depends on binary codes. This work does not consider the transposition operation of two adjacent characters.

(Shaikh et al., 2012) study aims to enhance Levenshtein Distance algorithm for Alias Detection, they introduce an additional new edit operation, that is, ‘exchange of vowels’ (a, e, i, o, u, y) [10]. This edit operation they proposed to find the most commonly occurring orthographic and typographical errors especially in person names. The ‘exchange of vowels’ edit operation they introduced to account for the most commonly occurring spelling mistakes of vowels due to the converting names from one language to another, and cannot the allows in transposition errors as mentioned in Levenshtein algorithm. Furthermore, there are several studies that used Levenshtein Distance for different applications such as: Plagiarism Detection (Zhan et al., 2008) and Email Hoax Detection (Chen et al., 2014) [15&16].

(Jaro, 1989) introduced a string matching algorithm to find the similarity between two strings. This algorithm is based on three basic steps: (1) Compute the string length, (2) Find the number of common characters between two strings and (3) Calculate the number of transpositions (t). (Winkler et al.,, 1990) modified the Jaro algorithm stating that if the prefix is common in two strings then the similarity score is increased [17]. This enhancement of Winkler is based on the observation that most common typographic variations occur towards the end of a string. (Shaikh et al., 2011) they proposed an extension to basic ASM algorithms (Jaro and Jaro-Winkler) to enhance the efficiency of the basic algorithms to detect person name aliases [10]. This enhancement is based on the observation that typographical variations occur more commonly due to transliteration or cultural difference. For example, "Osama bin Laden" and "Usama bin Ladan" are two strings, S1 and S2, respectively. They introduced a new operation called "exchange of vowels" to increase the similarity scores of the basic algorithms. This operation can also be applied to extend some other ASM algorithm such as edit distance algorithm.

(Hsiung et al.2005) used link data sets to extract string variant and semantic aliases. He used orthographic features such as string edit distance and semantic features like friends information to to traing a classifier which classifies between an alias or not [18]. (L. Jiang et al., 2012; Ning et al., 2014) they proposed a classifier that is based on active learning for detecting this type of aliasing. To minimize the cost of pair-wise comparison, a subset-based method is designed to restrict the selection within entity subsets. An active learning classifier is then employed in each entity subset to find the probability of whether a candidate is the alias of a given entity within the subset [19&7]. (P.Selvaperumal et al., 2016) they proposed , string variant aliases are first extracted from the web and then using seven different string similarity metrics as features, candidate aliases are validated using ensemble classifier random fores [20].(Kondrak, 2005) Kondrak suggests a hybrid method (N-DIST) that is mixing the components of LD and N–Gram algorithms and proposed a new similarity measurement [21].This measurement has been evaluated depend on Genetic cognates words of the same origin that belong to distinct languages. For example, German “vater”, English “father” and Norwegian “far” constitute a set of cognates, Confusable drug names, and Translational cognates. This algorithm takes the advantage of LD and n-grams algorithms. Therefore, this algorithm has been increased the time complexity to O (N3). This work does not consider the transposition operation of two adjacent characters. In (Sanabani et al., 2015 and Abdulhayoglu et al., 2016) have been used N-DIST algorithm in different applications based on Arabic and English language

[22&23]. We have modified the N-DIST algorithm mentioned in Sections 3 and the details of the adjusted algorithm are described in Section

IV. The Proposed Technique

In this section, we presented an additional new edit operation, that is, 'exchange of vowels' (a, e, i, o, u, y). This new edit operation is proposed to find the most commonly occurring orthographic and typographical errors especially in person names. The 'exchange of vowels' edit operation is introduced to account for the most commonly occurring spelling mistakes of vowels due to the converting names from one language to another.

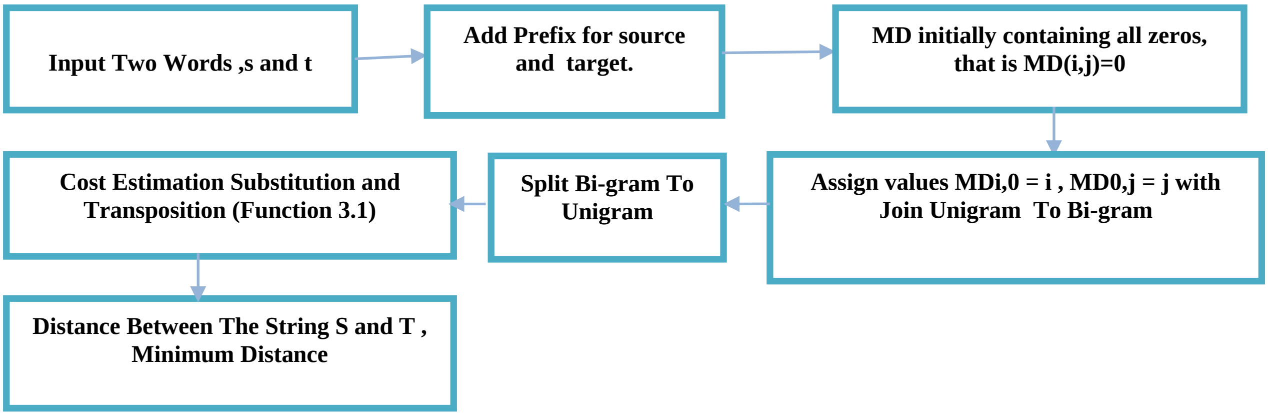
Input Two Words ,s and t
Add Prefix for source and target.
MD initially containing all zeros, that is MD(i,j)=0
Assign values MDi,0 = i , MD0,j = j with Join Unigram To Bi-gram
Split Bi-gram To Unigram
Cost Estimation Substitution and Transposition (Function 3.1)
Distance Between The String S and T , Minimum Distance

Fig.1. The Proposed Alias Detection System

The substitution of vowels in names is ignorable as compare to dictionary words. For example, (usama, osama) and (same, some) are two pairs of strings with only difference of vowels 'o' and 'a' in each pair of the string but you can see that in pair1 (usama, osama) in this case, the difference of vowels ('o' and 'a') is ignorable because it's not influencing the meaning but in case of pair2 (same, some) the difference of 'o' and 'a' change the meaning of the two strings of pair2. So, on the basis of this observation that if the two names only have the difference of vowels then it can be assumed to be aliases, therefore we have introduce the 'exchange of vowels' edit operation to detect the name aliases more efficiently. According to our observation and analysis, these kind of aliases mainly occur because of the vowel variations because short vowels cannot be written in Arabic that's why the vowelisation process is required, that is insertion of short vowels in target language (English in our case). For that reason, the new edit operation known as 'exchange of vowels' is proposed to detect these types of name variations (errors). The proposed new edit operation is added in the list of edit operations as stated in (Sanabani et al., 2015) of basic [21]. N-DIST algorithm and new algorithm named as 'Adjusted N-gram Distance (A-N-DIST)' see Figure 1. This operation listing vowels as 'a, e, i, o, u, and y', "character 'y' is particularly not a vowel but it sounds like a vowel and also a part of vowels in different languages such as Danish, Swedish, etc. Therefore, 'y' is included in the vowel's list in order to detect the most commonly occurred typographic errors efficiently and accurately (Shaikh et al., 2012) [10]. This operation allows swapping and substitution of vowels from the list of vowels at reduced penalty cost that is 0.5. As a result of reducing penalty cost of the vowels in names especially in Arabic names the similarity scores of the name-alias pairs are described in Section 5.

Table 1. The proposed function for substitution operation.

```
Function. To Compute the Cost of Substitution of N-gram
Input: N-gram Letters (Letter1, Letter2)
Output: cost Substitution Distance (cost)
Decimal  cost- Substitution-N-gram-Distance(a[i -1+ni], t[ni])
 if (a[i -1+ni] ≠'a') or (a[i -1+ni] ≠'e') or (a[i -1+ni] ≠'i') or (a[i -1+ni] ≠'o') or (a[i -1+ni] ≠'u')  or (a[i -1+ni] ≠'y')  then
   cost ← 0.5
   else
   cost ← cost +1     // cost++
end if
return cost
```

V. Results and Discussion

This section gives details of experiments that have been carried out in this work to illustrate the proposed algorithm that is called (A-N-DIST) and compare it against the compared algorithm.

A. Dataset

This section describes the names of Arabic and English that is used to test the proposed algorithm for Alias detection. For more investigation a collection of datasets have been used in this experiment for testing the proposed (A-N-DIST) and compared algorithms. Because no standard collection of Alias Detection exists, therefore, two datasets have been extracted manually form that are named Dataset 1 (Shaikh et al., 2012), Dataset 2 [24&10]. Each dataset contains some of Alias Detection 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.

B. Performance Measure

In the subsection, a comparative study is carried out to evaluate the performance of the proposed A-N-DIST algorithm. The first experiment has been carried for the proposed A-N-DIST algorithm and original N-DIST algorithm as compared algorithm with N equal to Bi and Tri (Bi=2, Tri=3) respectively. This experiment is carried based on Dataset 1 which has 10 pairs of names. The result of this experiment is shown in Table 1. The A-N-DIST Algorithm gives better results than the LD and N-DIST algorithms especially when comparing names transposition as shown in Table 1. For example, the names in 1 and 4 rows. Unlike the LD and N-DIST algorithms the A-N-DIST algorithm is sensitive to replacement as shown in 2, 3, 5 and 6 rows.

The A-N-DIST Algorithm handles a repeated letters, deletion and dictation errors more efficiently than the LD and N-DIST algorithms as shown in 7, 8,9 and 10 rows. The A-N-DIST algorithm shows many advantages over the LD, and N-DIST algorithms as aforementioned. Therefore, the A-N-DIST algorithm gives more accurate results than the LD and N-DIST algorithms with BI and TRI for all pairs in Dataset 1 as shown in Table 1. And Figure 2.In order to understand how the editing operations in the A-N-DIST algorithm works with variation of names, it will be elaborate of in detail in the following examples and as shown in Figure 3 shows how the N-DIST-A algorithm measures the distance between the name1 “abu abdallah” and name2 “abu abdalluh” as first step. The distance is 0.33, therefore, the similarity between them is 0.97 %. It is obvious that A-N-DIST algorithm gives a very low cost for replacing ‘a’ with “u” 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 A-N-DIST algorithm ability. Datasets are selected and applied on the LD, N-DIST and A-N-DIST algorithms with N=BI as shown in Table 2. That gives the evidence of A-N-DIST algorithm ability in Alias Detection. Table 2 shows the accuracy of the percentage similarity as an mean for Dataset 2. In this Table, the A-N-DIST algorithm gets 74.0% while LD and N-DIST algorithms get 45%, 40%, respectively. Therefore, the A-N-DIST algorithm gives more accurate results than the LD and N-DIST algorithms for dataset , because LD and N-DIST algorithms has not taken into account the characteristics and unique features Alias Detection. More details about the result can be found in Appendix A

Table 2.Comparison between algorithms in Alias Detection Dataset.

NO	String.		Compared Algorithm						Proposed Algorithm			
			LD		N-DIST							
					BI		TRI		BI		TRI	
	S	T	Dist.	Sim %	Dist.	Sim %	Dist.	Sim %	Dist.	Sim %	Dist.	Sim %
1	abu abdallah	abu abdalluh	1.00	0.92	1.00	0.92	0.67	0.94	0.50	0.96	0.33	0.97
2	mujahid shaykh	mujahid shaikh	1.00	0.93	1.00	0.93	1.00	0.93	0.50	0.96	0.50	0.96
3	hussein al-sheik	hassan ali-sheik	4.00	0.75	4.50	0.72	5.17	0.68	2.25	0.86	1.92	0.88
4	osama bin laden	usama bin laden	1.00	0.93	1.50	0.90	1.83	0.88	0.75	0.95	0.92	0.94
5	usama bin ladin	usama bin laden	1.00	0.93	1.00	0.93	0.67	0.96	0.50	0.97	0.33	0.98
6	usama bin laden	osama bin ladin	2.00	0.87	2.50	0.83	2.50	0.83	1.25	0.92	1.25	0.92
7	abdel muaz	abdul muiz	2.00	0.80	2.00	0.80	1.67	0.83	1.00	0.90	0.83	0.92
8	abdal muaz	abdel muiz	2.00	0.80	2.00	0.80	1.67	0.83	1.00	0.90	0.83	0.92
9	abu mohammed	abu muhammad	2.00	0.83	2.00	0.83	1.67	0.86	1.00	0.92	0.83	0.93
10	ayman al- awahari	ayman al-zawahiri	2.00	0.88	2.00	0.88	2.00	0.88	1.00	0.94	1.00	0.94
Average (percentage similarity)				0.86		0.85		0.86		**0.93**		**0.94**

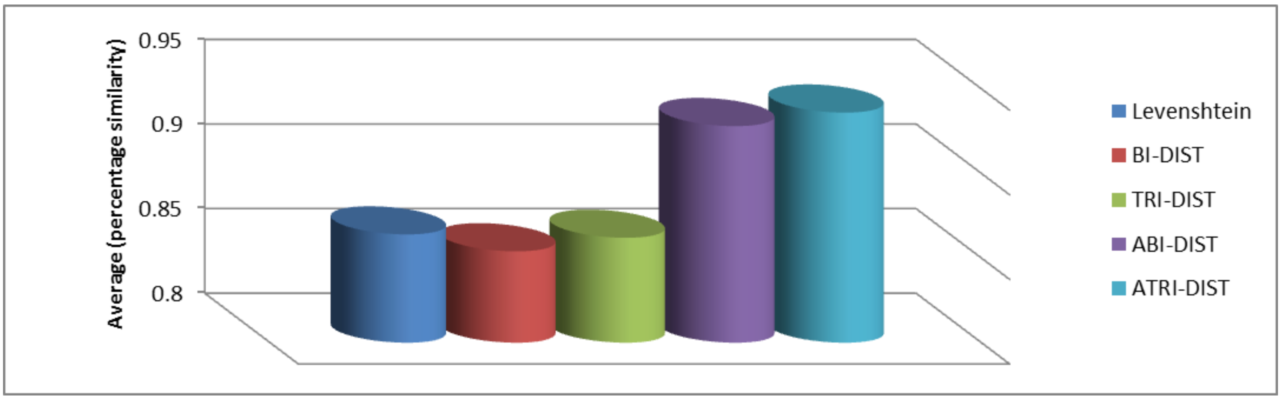

Fig.2. Average (percentage similarity

Table 3. The Distance Between “abu abdalluh” → “abu abdallah” in the A-N-DIST Algorithm with BI

		-a	ab	bu	u	a	ab	bd	da	al	ll	la	ah
	0	1	2	3	4	5	6	7	8	9	10	11	12
-a	1	0.00	1.00	2.00	3.00	4.00	4.50	5.50	6.00	6.50	7.50	8.00	8.50
Ab	2	1.00	0.00										
bu	3	2.00		0.00									
U	4	3.00			0.00								
a	5	3.50				0.00							
Ab	6	4.00					0.00						
Bd	7	5.00						0.00					
Da	8	5.50							0.00				
Al	9	6.00								0.00			
Ll	10	7.00									0.00		
Lu	11	8.00										0.00	
Uh	12	9.00											**0.33**

Table 4. The Average similarity of LD, N-DIST and A-N-DIST algorithms.

Dataset 2 (100 pairs)	Compared Algorithms		Proposed Algorithm
	LD	N-DIST	A-N-DIST
Average (percentage similarity)	0.45	0.40	0.740

VI. Conclusion

This paper presents the proposed ‘adjusted N-gram Distance A-N-DIST that is the adjusted version of the basic N-DIST and LD. The adjustment is proposed to encounter the problem of aliases generated because of transliteration of Arabic names. Therefore, we have proposed the ‘exchange of vowel’ edit operation to deal this problem. This operation reduces the penalty cost for exchanging the vowels with each other in two strings (name and alias pair) to increase the similarity percentage between the true alias pairs as shown in the experimental results. In our future work we intend to apply our proposed algorithm to larger data set and to calculate the effects on precession and recall measures. Furthermore, we intend to categorize the ‘exchange of vowel’ operation as the vowels that sound like same can be transposition with less different penalty scores such as ‘i’, ‘e’ and ‘y’ in one category, ‘o’ and ‘u’ in other , and includes extracting other kinds of alias names like semantic aliases. Working with non-English language for extracting string variant alias has its own challenges.

References

[1] J. Brynielsson, A. Horndahl, F. Johansson, L. Kaati, C. M°artenson, and P. Svenson, “Harvesting and analysis of weak signals for detecting lone wolf terrorists,” Submitted to Security Informatics, 2013.

[2] D. B. Neill 2002. Fully Automatic Word Sense Induction by Semantic Clustering. M.Phil Thesis. Cambridge University.

[3] D. Jurafsky and J. H. Martin 2000. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall.

[4] Muhammad Ghafoor, Mehyeddin Abdulrahman and Shvan Tariq: Plagiarism Detection System for the Kurdish Language" I.J. Information Technology and Computer Science journal, V.12, no. 64-71,2017.

[5] Prianka Mandal and B M Mainul Hossain: A Systematic Literature Review on Spell Checkers for Bangla Language " I.J. Information Technology and Computer Science journal, V.6, no. 40-47,2017.

[6] R. Ibrahim, S. Saeed, and K. Wakil, "Plagiarism Detection Techniques for Arabic Script Languages: A Literature Review," Kurdistan Journal of Applied Research, vol. 2, no. 3, 2017.

[7] Ning An , Lili Jiang , Jianyong Wang , Ping Luo , Min Wang, Bing Nan Li , Toward detection of aliases without string similarity, Information Sciences 261,89–100 ,2014.

[8] M. Bilenko and R. J. Mooney. On evaluation and training-set construction for duplicate detection. In Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, pages 7-12, 2003.

[9] L. K. Branting. Name matching in law enforcement and counter-terrorism. In Proceedings of ICAIL 2005 Workshop on Data Mining, Information Extraction, and Evidentiary Reasoning for Law Enforcement and Counter-Terrorism.

[10] Shaikh, M. , Dar, H., Shaikh, A., and Shah, A. "Adjusted Edit Distance Algorithm for Alias Detection", International Conference on Information and Knowledge Management , 2012 .

[11] Salah Alhagree and Maher A. Al-Sanabani, " A Framework For Name Matching In Arabic Language", 1st Scientific Conference on Information Technology and Networks, 2016.

[12] Salah Alhagree, "Design Algorithms for Matching English and Arabic Names" Master"s thesis, Thamar University, Department of Computer Science. 2017.

[13] P. Christen, Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Springer, 2012. ISBN 978-3-642- 31163-5

[14] Levenshtein, Vladimir I. "Binary codes capable of correcting deletions, insertions, and reversals." In Soviet physics doklady, vol. 10, no. 8, pp. 707-710. 1966.

[15] Zhan Su, Byung-Ryul Ahn, Ki-yol Eom, Min-koo Kang, Jin-Pyung Kim, Moon-Kyun Kim" Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm"The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC'08), 2008.

[16] yoke yie chen , suet-peng yong and adzlan ishak , "Email Hoax Detection System Using Levenshtein Distance Method", journal of computers, vol. 9, no. 2, february 2014.

[17] W. E. Winkler and Y. Thibaudeau. An application of the fellegi-sunter model of record linkage. Technical report, U.S. Decennial Census, Bureau of the Census, 1990.

[18] P. Hsiung, D. Andrew, W. Moore, and J. Schneider. Alias detection in link data sets. In Proceedings of the International Conference on Intelligence Analysis, 2005.

[19] L. Jiang, J. Wang, P. Luo, N. An, M. Wang, Towards alias detection without string similarity: an active learning based approach, in: Proceedings of the 35th Annual International ACM SIGIR Conference, 2012. Computer Science, 3772, Springer, Heidelberg, Germany, 115–126, 2005.

[20] P.Selvaperumal and A.Suruliandi , "String Variant Alias Extraction Method using Ensemble Learner",2016,

[21] Kondrak, G, "N-gram similarity and distance", In M. Consens and G. Navarro (eds.), Proceedings of the String Processing and Information Retrieval 12th International Conference, Buenos Aires, Lecture Notes in

[22] Maher Sanabani, Salah Al-Hagree. ,"Improved An Algorithm For Arabic Name Matching". Open Transactions On Information Processing ISSN(Print): 2374-3786 ISSN(Online): 2374-3778.2015.

[23] Abdulhayoglu, M. A , Bart Thijs , Wouter Jeuris , "Using character n-grams to match a list of publications to references in bibliographic databases" , DOI 10.1007/s11192-016-2066-3,2016.

[24] www.kalmasoft.com/KLEX/dbfamnm.htm.

No.	FAM_ARABIC	string1	string2	Compared Algorithm			Proposed Algorithm	
				LV	BI	TRI	BI	TRI
1	سيلفستر ستالوني	Sylvester Stallone	sylfstr stAlwny	0.53	0.47	0.45	0.66	0.70
2	جون غارانغ	John Garang	jwn gArAng	0.50	0.46	0.42	0.75	0.77
3	فيدل كاسترو	Fidel Castro	fydl kAstrw	0.54	0.46	0.44	0.75	0.78
4	كيزو أوبوتشي	Kizo Obutchi	kyzw !wbwt$y	0.31	0.27	0.23	0.73	0.80
5	توم هانكس	Tom Hanks	twm hAnks	0.60	0.55	0.50	0.83	0.83
6	ديانا لوبيز	Diana L ópez	dyAnA lwbyz	0.33	0.29	0.25	0.75	0.80
7	بيل غيتس	Bill Gates	byl gyts	0.45	0.36	0.30	0.66	0.68
8	ونستون تشرشل	Winston Churchill	wnstwn t$r$l	0.44	0.42	0.37	0.60	0.61
9	أياكا هيراهارا	Ayaka Hirahara	!yAkA hyrAhArA	0.47	0.43	0.43	0.73	0.81
10	ديفيد بيكهام	David Beckham	dyfyd bykhAm	0.43	0.39	0.36	0.73	0.77
11	دونالد ترامب	Donald Trump	dwnAld trAmb	0.54	0.50	0.49	0.77	0.81
12	ألبرت آينشتاين	Albert Einstein	!lbrt *yn$tAyn	0.56	0.50	0.47	0.75	0.79
13	تييري هينري	Thierry Henry	tyyry hynry	0.50	0.46	0.43	0.73	0.75
14	بوب مارلي	Bob Marley	bwb mArly	0.55	0.45	0.39	0.73	0.75
15	إدولف هتلر	Adolf Hitler	Edwlf htlr	0.62	0.54	0.47	0.69	0.71
16	مارلين مونرو	Marilyn Monroe	mArlyn mwnrw	0.53	0.47	0.42	0.72	0.74
17	تايغر وودز	Tiger Woods	tAygr wwdz	0.33	0.29	0.28	0.69	0.76
18	نيلسون مانديلا	Nelson Mandela	nylswn mAndylA	0.53	0.50	0.49	0.78	0.81
19	ناديا كومانتشي	Nadia Com?neci	nAdyA kwmAnt$y	0.33	0.30	0.29	0.73	0.81

20	لويس مورينو أوكامبو	Luis Morinho Okampo	lwys mwrynw !wkAmbw	0.35	0.33	0.32	0.75	0.81
21	نوانكو كانو	Nuanko Kano	nwAnkw kAnw	0.42	0.38	0.36	0.75	0.80
22	أسامة بن لادن	Usama bin Ladin	!sAm: bn lAdn	0.56	0.47	0.43	0.69	0.73
23	خافيير سولانا	Xavier Solana	KAfyyr swlAnA	0.36	0.32	0.31	0.75	0.80
24	ري تشارلز	Ray Charles	ry t$Arlz	0.42	0.38	0.33	0.67	0.69
25	ليونيد بريجنيف	Leonid Brezhnev	lywnyd bryjnyf	0.38	0.34	0.33	0.70	0.78
26	ليوناردو دافينشي	Leonardo daVinci	lywnArdw dA fyn$y	0.39	0.36	0.35	0.72	0.78
27	مايكل جاكسون	Michael Jackson	mAykl jAkswn	0.38	0.34	0.31	0.66	0.68
28	هوك هوغان	Hulk Hogan	hwk hwgAn	0.45	0.41	0.36	0.73	0.75
29	روبيرتو كارلوس	Roberto Carlos	rwbyrtw kArlws	0.53	0.50	0.47	0.78	0.81
30	هو جنتاو	Hu Jintau	hw jntAw	0.40	0.35	0.30	0.70	0.74
31	معمر القذافي	Muamar Gadhafi	momr AlqcAfy	0.33	0.27	0.24	0.67	0.72
32	دييغو مارادونا	Diego Maradona	dyygw mArAdwnA	0.40	0.37	0.36	0.73	0.81
33	بربارا سترايسند	Barbra Streisand	brbArA strAysnd	0.53	0.44	0.39	0.75	0.79
34	ألبيرتو فوجيموري	Alberto Fujimori	!lbyrtw fwjymwry	0.53	0.50	0.50	0.76	0.81
35	عمر البشير	Umar Albashir	omr Alb$yr	0.57	0.50	0.46	0.68	0.68
36	ويليام شكسبير	William Shakespeare	wylyAm $ksbyr	0.35	0.28	0.23	0.54	0.58
37	فيرديناند ماركوس	Ferdinand Marcos	fyrdynAnd mArkws	0.53	0.50	0.47	0.79	0.82
38	أوبرا وينفري	Oprah Winfrey	!wbrA wynfry	0.43	0.36	0.31	0.73	0.77
39	فولفغانغ أماديوس موزارت	Wolfgang Amadeus Mozart	wlfgAng !mAdyws mwzArt	0.58	0.56	0.54	0.79	0.82
40	جيمس كاميرون	James Cameron	jyms kAmyrwn	0.50	0.43	0.38	0.73	0.77
41	فريدريك هينري	Frederick Henry	frdryk hۂry	0.56	0.47	0.41	0.67	0.70
42	الطيب صالح	Altayib Salih	AlTyb SAlH	0.57	0.54	0.50	0.66	0.69
43	ياكوف سميرنوف	Yakov Smirnoff	yAkwf smyrnwf	0.47	0.43	0.40	0.72	0.77
44	ماريا كيري	Maria Cary	mAryA kyry	0.45	0.41	0.36	0.77	0.81
45	مارك توين	Mark Twain	mArk twyn	0.55	0.50	0.42	0.73	0.75
46	ألتون جون	Alton John	!ltwn jwn	0.55	0.50	0.47	0.73	0.75
47	الفريد هيتشكوك	Alfred Hitchcock	!lfryd hyt$kwk	0.47	0.44	0.42	0.71	0.74
48	هيديكي يوكاوا	Hedeki Yukawa	hydyky ywkAwA	0.43	0.39	0.38	0.73	0.80
49	جون سينا	John Cena	jwn synA	0.40	0.35	0.33	0.70	0.74
50	خوزيه كورتيز	Jose Cortez	Kwzyh kwrtyz	0.38	0.35	0.31	0.73	0.76
51	بطرس غالي	Butrus Ghali	bTrs gAly	0.38	0.31	0.28	0.60	0.64
52	مارتينا نافراتيلوفا	Martina Navr átilov á	mArtynA nAfrAtylwfA	0.40	0.38	0.37	0.75	0.81
53	كمال الشناوي	Kamal Alshennawi	kmAl Al$nAwy	0.47	0.44	0.41	0.62	0.65
54	نجيب محفوظ	Najib Mahfudh	njyb mHfwZ	0.36	0.32	0.31	0.61	0.65
55	مانويل نورييغا	Manuel Noriega	mAnwyl nwryygA	0.40	0.37	0.36	0.75	0.81
56	أيمن الظواهري	Ayman Alzawahri	!ymn AlZwAhry	0.63	0.56	0.53	0.72	0.74
57	جمال بعد الناصر	Jamal Abdel Nassir	jmAl obd AlnASr	0.42	0.37	0.33	0.67	0.70
58	أوليفر كان	Oliver Kahn	!wlyfr kAn	0.33	0.29	0.25	0.71	0.76
59	مارغريت ثاتشر	Margaret Thatcher	mArgryt xAt$r	0.44	0.39	0.33	0.61	0.65
60	لاري كينغ	Larry King	lAry kyng	0.55	0.50	0.45	0.77	0.78
61	خوليو إيغلاسياس	Julio Iglesias	Kwlyw EglAsyAs	0.47	0.43	0.40	0.77	0.81
62	ناستازيا كينسكي	Nastassja Kinski	nAstAzyA kynsky	0.41	0.38	0.37	0.74	0.79
63	إيريك كانتونا	Eric Cantona	Eryk kAntwnA	0.54	0.54	0.56	0.81	0.86
64	ياني خريسوماليس	Yanni Christomalis	yAny KryswmAlys	0.42	0.39	0.33	0.64	0.70
65	ماري أنطوانيت	Marie Antoinette	mAry !nTwAnyt	0.35	0.29	0.25	0.62	0.69
66	ماريا كلاس	Maria Klaas	mAryA klAs	0.42	0.38	0.31	0.71	0.76
67	صوفي مارسو	Sophie Marceau	Swfy mArsw	0.27	0.27	0.28	0.57	0.62
68	مايك أولدفيلد	Mike Oldfield	mAyk !wldfyld	0.43	0.39	0.33	0.77	0.82
69	كيني غورليك	Kenny Gorlick	kyny gwrlyk	0.50	0.46	0.40	0.71	0.72
70	جاك كوستو	Jacques Cousteau	jAk kwstw	0.24	0.21	0.18	0.46	0.49
71	تومي لي	Tommy Lee	twmy ly	0.40	0.35	0.33	0.65	0.68
72	جوزيبي فيردي	Giuseppe Verdi	jwzyby fyrdy	0.27	0.23	0.22	0.67	0.72
73	مصطفى العقاد	Mustafa Alaqad	mSTfY AloqAd	0.47	0.43	0.40	0.70	0.73
74	جاك شيراك	Jacques Chirac	jAk $yrAk	0.20	0.17	0.16	0.50	0.56
75	سلمى حايك	Salma Hayek	slmY HAyk	0.58	0.50	0.44	0.69	0.69
76	مارتن لوثر	Martin Luther	mArtn lwxr	0.43	0.36	0.31	0.63	0.65
77	أميتاب باتشان	Amitab Batchan	!mytAb bAt$An	0.47	0.43	0.41	0.72	0.77
78	بروس لي	Bruce Lee	brws ly	0.30	0.25	0.25	0.60	0.67
79	عدنان خاشقجي	Adnan Khashuqji	odnAn KA$qjy	0.50	0.47	0.47	0.66	0.68

80	فرح عيديد	Farah Eided	frH oydyd	0.42	0.38	0.31	0.65	0.69
81	أبراهام لينكولن	Abraham Lincoln	!brAhAm lynkwln	0.56	0.53	0.51	0.80	0.82
82	جوزف كوني	Joseph Koni	jwzf kwny	0.25	0.21	0.19	0.63	0.69
83	عمر الشريف	Umar Alsharif	omr Al$ryf	0.57	0.50	0.44	0.66	0.67
84	يوري أندروبوف	Yuri Andropov	ywry !ndrwbwf	0.43	0.39	0.38	0.77	0.82
85	شاهر عبد الحق	Shaher Abdulhak	$Ahr Obd AlhHq	0.44	0.38	0.33	0.72	0.78
86	مايكل جوردان	Michael Jordan	mAykl jwrdAn	0.40	0.37	0.33	0.70	0.72
87	إبراهيم روغوفا	Ibrahim Rugova	EbrAhym rwgwfA	0.47	0.43	0.43	0.77	0.81
88	نبيل أرمسترونغ	Neil Armstrong	nyyl !rmstrwng	0.67	0.63	0.60	0.85	0.85
89	صدام حسين	Sadam Husein	SdAm Hsyn	0.62	0.54	0.47	0.65	0.67
90	الشاب خالد	Cheb Khalid	Al$Ab KAld	0.42	0.33	0.25	0.71	0.76
91	يوجين كاسبرسكي	Eugene Kaspersky	ywjyn kAsbrsky	0.47	0.41	0.35	0.72	0.76
92	ماجدة الرومي	Majda Alrumi	mAjd: Alrwmy	0.62	0.58	0.56	0.81	0.83
93	باريس هيلتون	Paris Hilton	bArys hyltwn	0.54	0.50	0.46	0.79	0.80
94	أنديرا غاندي	Andira Ghandi	!ndyrA gAndy	0.50	0.46	0.46	0.71	0.77
95	روبرت ميردوخ	Robert Murdock	rwbrt myrdwK	0.47	0.40	0.38	0.70	0.73
96	برفيز مشرف	Pervez Musharraf	brfyz m$rf	0.35	0.29	0.25	0.51	0.54
97	كيم داي جونغ	Kim Dae-jong	kym dAy jwng	0.46	0.42	0.38	0.79	0.81
98	جاكي شان	Jackie Chan	jAky $An	0.33	0.29	0.22	0.58	0.63
99	إيرين جوليو كوري	Ir ène Joliot-Curie	Eryn jwlyw kwry	0.32	0.26	0.24	0.63	0.70
100	هيفاء وهبي	Haifa Wahbe	hyfA' whby	0.33	0.29	0.28	0.71	0.76
Average (percentage similarity)				0.45	0.40	0.37	**0.70**	**0.74**

Authors' Profiles

Assist.Prof. 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 University Kebangsaan Malaysia in 2013. Research interests include Information system,SIS, Artificial Intelligence , Data Mining, , already published several journal papers.

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.

Salah Abdu Al-hagree. Currently a teacher with M.sc degree 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 had 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.

How to cite this paper: Muneer Alsurori, Maher Al-Sanabani, Salah AL-Hagree," Design an Accurate Algorithm for Alias Detection", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.10, No.3, pp. 36-44, 2018. DOI: 10.5815/ijieeb.2018.03.05