2012 International Conference on Information and Knowledge Management (ICIKM 2012)
IPCSIT vol.45 (2012) © (2012) IACSIT Press, Singapore

Adjusted Edit Distance Algorithm for Alias Detection

Muniba Shaikh[1], Humaira Dar[2], Asadullah Shaikh[3] and Asadullah Shah[4]

[1, 3, 4] Department of Computer Science, Kulliyyah of Information and Communication Technology (KICT), International Islamic University Malaysia (IIUM)

[1, 3] {muniba.shaikh,asadullah.shaikh}@live.iium.edu.my,

[4]asadullah@kict.iium.edu.my,

[2]Department of computer &information engineering Kulliyyah of electrical & electronic engineering, International Islamic University Malaysia (IIUM)

[2]{engg87.iium@gmail.com}

Abstract. 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. Edit distance algorithm (EDA) have find wide applicability in the process of AD when the same is based upon orthographic and typographic variations. Levenshtein approach, a popular EDA works well and fulfill the cause, but at the same time we uncover that Levenshtein EDA (LEDA) 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 edit distance metric measure of LSM 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 LEDA by applying both forms of LEDA on the above data set. As expected we come across that adjusted LEDA 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.

Keywords: alias detection (AD), Edit Distance algorithm (EDA), Levenshtein edit distance algorithm (LEDA), Adjusted Levenshtein (ALev), transliteration, orthographic, typographic.

1. Introduction

Malevolent or malicious use of numerous and fake identity credentials for fraudulent and criminal intent have become a routine today. Due to this associative phenomenon, the process of AD has become a more serious and exigent task. The direct or indirect involvement of AD has seen tremendous increase in fields, such as analysis of social networking sites, database applications, state intelligence, and data mining processes, bio-metrics, e-commerce, law enforcement and counter terrorism [7]. Thus as already stated the problem domain for AD has become quite vast with time and same can be forecasted for future.

Focus of database community presently is enormously on aliases (deduplication) detection for the purpose of data cleaning [10]. By means of baseline study of AD, counter terrorism & other law enforcement agencies [4, 13] focus on detecting terrorists and their activities. The scale of problem featuring AD may range from quite simple ones (data sets contain aliases merely because of accident) to complicated ones where in the multiple and fake identities have been put in place intentionally for fetching malevolent purposes [7, 8]. AD also find wide applicability with accidental occurrence of errors in relational data featuring due to any of the CRUD i.e.; create, read, update and delete operation and data integration phenomenon. These errors are often based upon transcription problems, incomplete information, lack of standard formats or even any combination of these and others.

Moreover, 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 [2]. 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 [1]. A novel enhancement of Levenshtein edits distance (a popular EDA) form the central position in the paper. Results based on analytical modeling and measurement which proves the effectiveness of the enhance LED algorithm over the basic one also find space in the paper.

The rest of the paper's organization is as follows: commonly used EDA and their fields of applicability have been discussed in section 2. Section 3 relates to the formulation of the solution to adjust LEDA. Section 4 provides insight into the dataset that have been used for experimentation purpose. Again the results from the experimentation conducted finds space in this section. Lastly sections 5 presents the conclusion and part of the research lines towards future research and are drawn clearly in the same section.

2. Edit Distance Algorithms

There are a number of metrics available to achieve the string matching tasks but the basic metrics are based on ED metrics. Various ED metrics have been developed so far to decrease the penalty for the most possible transcription errors [4,7]. The main problem is how to select or combine multiple orthographic measures [6] in order to achieve desired results.

The basic EDA's are based on dynamic programming including Smith-Waterman, Levenshtein Distance and Needleman Wunsch.[7] These dynamic programming algorithms needs O (m*n) operations to calculate the edit distance between two strings, where 'm' and 'n' are the lengths of $string_1$ and $string_2$, respectively. Dynamic programming generates the (m + 1)*(n + 1) matrix and compute all values of D (i, j) by using a recursive function and stores the result in a table, where 'i' and 'j' represents all strings from '1 to m/n'.

2.1. The Levenshtein Edit Distance Algorithm

The LEDA counts the minimum number of edit operations required to transform one string to another [2, 3, 5, 7, 8, 12, 13, 16, and 19]. It is also referred as basic Levenshtein (BLev) EDA. The LEDA allows three basic edit operations as given below:

1) Insert: D (i-1, j) +1
2) Delete: D (i, j-1) +1
3) Substitute: D (i-1, j-1) + Cost

If $a_i = b_j$ then Cost=0 and if $a_i \neq b_j$ then Cost=1

We have modified the algorithms mentioned in Sections 2.1 and the details of the adjusted algorithms are described in Section 2.2.

2.2. The Proposed Adjusted Levenshtein Edit Distance

In this section, we introduce an additional new edit operation, that is, 'exchange of vowels' (a, e, i, o, u, y). This 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.

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) [14]. 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 section 2.1) of basic Levenshtein (BLev) algorithm and new algorithm named as 'Adjusted Levenshtein (ALev)'. 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 [7]". 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 as shown in Table 1 and Figure 1.

3. Data Set and Experimental Results

This section contains description of the data set used to evaluate the performance of proposed algorithm on the basis of similarity scores and the results obtained by applying the basic and extended EDA on the data set. . In this paper we make use of that is taken from hsuing et al. [13]. It is based on '20 Ground Truth Entities' and contains 919 'alias pairs' and 4088 'names'. This data set is extracted manually from open source web pages and news stories [13]. In Table1, we have shown 10 alias pairs chosen manually from the hsuing et al. [13] data set that contain variations of the vowels and are aliases of each other.

In Figure 1, the Levenshtein EDA is applied on the data set before and after modification/adjustment and it is visible that ALev performance is increased as compare to BLev. It is obvious from the results shown in table1 and the Figure 1 that the similarity score of the Levenshtein is increased from 92 to 96 % in first string pair and 80% to 90% in seventh and eighth string pairs. So, in this case, the overall results are increased from 3 to 10%.

Table 1: Comparison of basic and adjusted Levenshtein EDA

String 1	String 2	Basic Levevshtein (BLev) Similarity Score	Adjusted Levenshtein (ALev) Similarity Score
abu abdallah	abu abdalluh	0.92 = 92%	96 = 96%
Mujahid shaykh	mujahid shaikh	0.93 = 93%	0.96 = 96%
hussein al-sheik	hassan ali-sheik	0.75 = 75%	0.81 = 81%
osama bin laden	usama bin laden	0.93 = 93%	0.97 = 97%
usama bin ladin	usama bin laden	0.93 = 93%	0.97 = 97%
usama bin laden	osama bin ladin	0.87 = 87%	0.93 = 93%
abdel muaz	abdul muiz	0.80 = 80%	0.90 = 90%
abdal muaz	abdel muiz	0.80 = 80%	0.90 = 90%
abu mohammed	abu Muhammad	0.83 = 83%	0.92 = 92%
Ayman al- awahari	ayman al-zawahiri	0.94 = 94%	0.97 = 97%

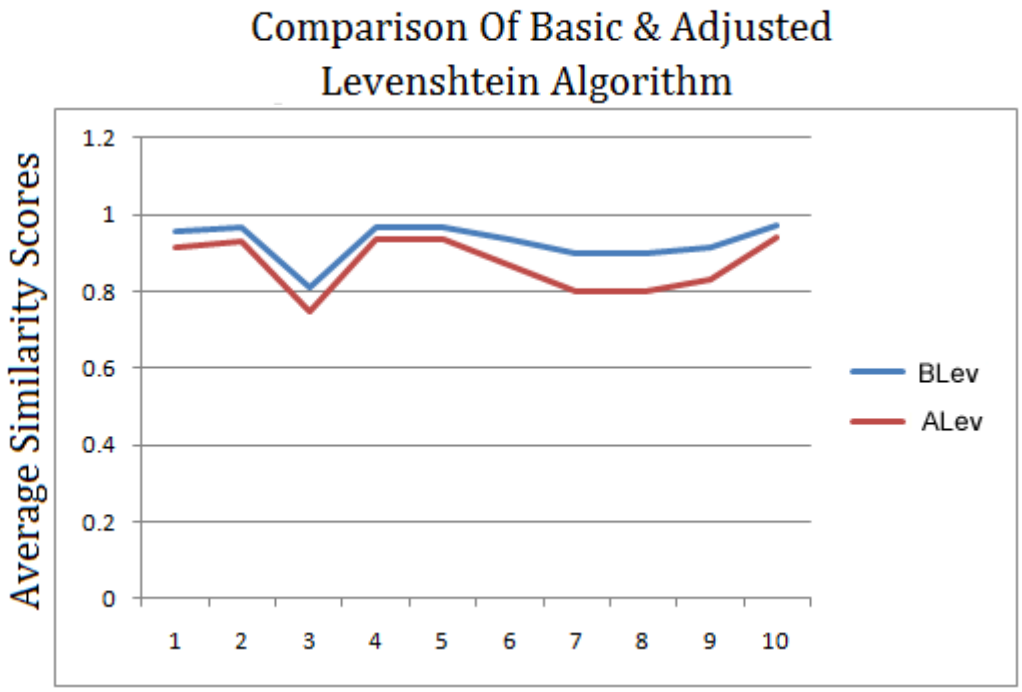

Figure 1: Performance of Basic and Adjusted Levenshtein edit distance algorithms on the Dataset

4. Conclusion and Future Work

This paper presents the proposed 'adjusted Levenshtein (ALev)' EDA that is the adjusted version of the basic Levenstein (BLev) EDA. 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 scores 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 swapped with less different penalty scores such as 'i', 'e' and 'y' in one category, 'o' and 'u' in other.

5. References

[1] Duplicate record detection: A survey. IEEE Trans. on Knowl. and Data Eng., 19(1):1-16, 2007. Senior Member-Elmagarmid, Ahmed K. and Member-Ipeirotis, Panagiotis G. and Member-Verykios, Vassilios S.

[2] 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.

[3] T. Boongoen and Q. Shen. Intelligent hybrid approach to false identity detection. In ICAIL '09: Proceedings of the 12th International Conference on Artifcial Intelligence and Law, pages 147-156, New York, NY, USA, 2009. ACM.

[4] 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.

[5] W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), August 9-10, 2003, Acapulco, Mexico, pages 73-78, 2003.

[6] 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.

[7] Muniba Shaikh, U. K. Wiil, Nasrullah Memon. Extended approximate string matching algorithms to detect name aliases. In Proceedings of the IEEISI, 2011.

[8] P. Pantel. Modeling observation importance for alias detection. In Proceedings of the DHS Conference on Partnerships in Homeland Security, 2005.

[9] D. P. Papamichail and G. P. Papamichail. Improved algorithms for approximate string matching (extended abstract). CoRR, abs/0807.4368, 2008.

[10] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '02, pages 269-278, New York, NY, USA, 2002. ACM.

[11] M. M. Taye. State-of-the-art: Ontology matching techniques and ontology mapping systems. International Journal of ACM Jordan, 1:48-55, 2010.

[12] W. E. Yancey. Evaluating string comparator performance for record linkage. Technical report, Statistical Research Division, U.S. Census Bureau, 2005.

[13] N. Zamin. Information extraction for counter-terrorism: A survey. Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, Computation World, 0:520-526, 2009.

[14] B.Pouliiquen, R.Steinberger, C.Ignat, I.Temnikova, A.Widiger, W.Zaghouani, and J.Zizka. Multilingual person recognition and transliteration .CoRR, abs/CS/0609051,2006.