Suffix Array Blocking for Efficient Record Linkage and De-duplication in Sliding Window Fashion

Yamini Warke

Abstract Record linkage is an essential process in information mix, which is utilized as a part of combining, coordinating and copy expulsion from a few databases that allude to the same substances. De-duplication is the procedure of uprooting copy records in a solitary database. Because of multifaceted nature of today's database, coordinating records in single database is an essential one. Indexing strategies are utilized to productively actualize record linkage and De-duplication. Our additional gathering strategy with jaro-winkler similarity measure exploits the ordering used by the list to combine comparative pieces at negligible additional cost, bringing about a much higher exactness while holding the high adaptability of the base suffix array method. We complete an inside and out examination of our system what's more, show results from examinations using Cora, restaurant and real identity data which highlights the significance of utilizing proficient as a part of indexing and hindering in true applications where information sets contain a large number of records. This paper presents suffix array blocking for efficacious record linkage and de- duplication in sliding window fashion.

Keywords Record linkage · Blocking · Suffix array

1 Introduction

As different government offices, business, and examination tasks assemble outstandingly a lot of information, expertise that offers ascend to handling and mining of huge databases have as of late respect with both institute and industry for holding the consideration. In the period of processing information of numerous information mining activities, connecting, or coordinating records which identified with same element from more than two databases get to be grater errands. The point of such

Yamini Warke (✉)
Dr. D.Y. Patil School of Engineering and Technology, Savitribai Phule Pune University, Pune, India
e-mail: warkeyamini85@gmail.com

S.C. Satapathy et al. (eds.), *Proceedings of the International Conference on Data Engineering and Communication Technology*, Advances in Intelligent Systems and Computing 468, DOI 10.1007/978-981-10-1675-2_7

Table 1 Motor gas station example

Associated address	Description
SR.#23/2 Near yash Hotel, dapodi, Pune, Maharashtra	Residential location of business
Patil A Suyash 345 Hallmark avenue Ravet Road No.7	Residential location of holder of business
P A S, Inc C/o Suresh S Mahajan Ravet Road no.4 Pune	Incorporated name of business accountant does books and government form

linkages is to match and make cement of all records identifying with the same element, for example, wiped out individual, a buyer, venture, a customer item, a copyright reference.

Future utilization of existing information for new studies and the expense and decided endeavor in information securing, record linkage and de-duplication is used [1]. Evacuating copy records in a solitary database is additionally essential one. In motor gas station example, this is illustrated in Table 1 [2]. The main name alludes to name of Business and its area of residency. The second is the business holders name with his location. Third is the location of bookkeeper who does the books for the organization. The name 'P A S. Inc' is a contraction of the genuine name of the business 'Patil A Suyash' which is the holder of engine washing focus. It is potential that distinctive rundown partner with the arrangement of organizations may have passages equal to anybody of the recorded types of the element which is the engine overhauling station. For this situation there are such a variety of indistinguishable Entries discovered, that indistinguishable (duplications) are adjusted when that specific individual give back the structure. However, it is extremely monotonous errand in the event that we need [2] that data after a few years, as that individual may be not at the comparing location. Table 1 elucidate this sample. Then again, as the measure of advanced data is quickly expanding everywhere throughout the world and a large portion of the information is unstructured one, for example, picture, sound, feature, and report records. This fast development of information size causes a few issues, for example, stockpiling impediment, expanding expense. We can beat this issue by utilizing de-duplication system.

2 Related Work

Dunn [3] and Marshall [4], and Fellegi and Sunter [5] proposed a hypothesis in light of measurable grouping, in which record linkage is utilized first. Record Linkage can radically expand the data accessible for thing planned, for example, substantial medicinal wellbeing frameworks [6], business investigation, and

misrepresentation identification [7] in subtle element. Indexing strategies, or blocking techniques as they are known in the connection of record linkage, were immediately perceived as a key segment for an opportunity to update or copy edit the Papers, which is not possible due to time constraints. Blocking calculations normally contain additional usefulness over standard indexing, to tackle particular record linkage issues. Blocking arrangements endeavor to diminish the quantity of applicant records for examination however much as could be expected, while as yet holding a precise result by guaranteeing that competitor records that would coordinate the question record are not left out of the hopeful set because of the blocking tenets. There are assortment of blocking routines right now utilized as a part of record linkage systems, with the most surely understood ones including customary blocking, sorted neighborhood [8], Q-gram based blocking [9], Canopy Clustering [10], string guide based blocking [11] and Suffix Array blocking [12].

Every blocking system characterize an arrangement of key fields from the information to be coordinated, that are utilized to figure out which piece (or obstructs) every record is to be put into. Huge numbers of these methodologies oblige a solitary string to be utilized as the key on which to locate the right square. In this manner, the estimations of the key fields are commonly connected together into one long string. This string is known as the blocking key value (BKV) [13]. The choice of key fields to incorporate in the BKV and additionally the requesting of these fields is vital to consider.

A suitable BKV ought to be the quality or mix of properties which are as recognizing as could be expected under the circumstances, consistently dispersed, and having low lapse likelihood. Initiate [14] thought about and assessed these blocking methods, changed two of them to make them more vigorous with respect to parameter settings, a critical thought for any calculation that is to be considered for genuine applications. The trial results demonstrated that there are expansive Contrasts in the quantity of genuine coordinated applicant record sets produced by the diverse systems, when tried utilizing the same information sets.

As different vast associations, organizations have on the whole extensive measure of information. With a specific end goal to prepare and investigate that information, coordinating of records that identify with the same substances from a few databases is important.

There are a few distinctive indexing methodologies are accessible, which includes, Conventional blocking, q-gram base indexing, covering grouping, string-guide based indexing, postfix exhibit indexing. The time intricacy of conventional blocking is O (dn log n) where n is the quantity of records in each of the two information sets that are being coordinated and d is the quantity of key fields picked [15].

Essential thought behind postfix cluster indexing is to embed the BKVs and their Additions into a postfix exhibit based modified record. In this indexing procedure, Postfixes down to a Minimum length, lm, are embedded into the addition cluster.

For instance, for a BKV “bannana” and lm = 5, the qualities ‘bannana’, ‘annana’, “nnana” will be created, and the identifiers of all records that have this BKV will be embedded into the relating four transformed Index records.

3 Methodology

3.1 Proposed System

In proposed system, Suffix array blocking in sliding window fashion is used with Grouping function. In this grouping function, suffixes are compared using different similarity measures including edit based Jaro and Jaro-Winkler [16] Similarity measures. This will give more improved results. Time complexity of algorithm is $o\ (n)^2$.

Figure 1 shows system architecture of proposed system.

In proposed system, as shown in Fig. 1, in first step one dataset is taken as input. Dataset include any Japanese and bibliographic data.

In second step, suffix array indexing is applied on that data. While applying suffix array indexing firstly blocking key value (BKV) is generated by concatenating key fields. Then suffixes are generated of that key field. All that suffixes are stored in index structure.

In third step, maximum block size is set. then for every record corresponding to that suffix is checked with block size. If no of record corresponding to that suffix is greater than maximum block size, all suffix-reference pair of that corresponding suffix are removed.

In fourth step, grouping of suffixes is done. In this for each unique suffix in inverted index comparison is done (compare *sf* to previous suffix *sg*).using comparison function jaro Winkler. Threshold is set.(if Jaro Winkler(*sf*, *sg* > *jt*)) all suffix reference pairs are grouped together corresponding to sf and sg using set join.

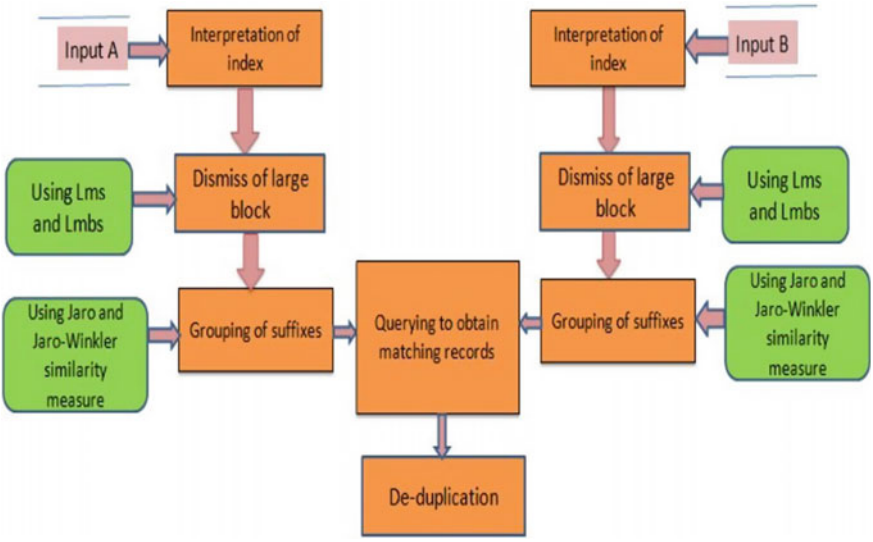

Fig. 1 System architecture

In last step, for calculating matching records, all first three steps are applied on another (Second) data set. And duplicated records are removed (De-duplication).

3.2 Algorithm Pseudo Code

Input:

1. *A* and *B*, the sets of records to find matches between
2. The suffixes comparison function similarity threshold ts
3. The minimum suffix length lms and the maximum block size lmbs

Let **I** be the inverted index structure used.
Let *Ci* be the resulting set of candidates to be used when matching with a record *A*

Interpretation of Index structure

1. For record ri1ε *A*
2. Construct BKV by concatenating Key fields
3. Generate suffixes in sliding window fashion
4. Insert suffixes and reference records to suffixes into I

Dismiss Large Block

5. For every unique suffix *Sf* in *I*
6. If the number of record reference paired with *Sf* > Lmbs
7. Remove all suffix reference pairs where the suffix is *Sf*

Grouping of suffixes

8. For each, unique suffix *Sf* in *I*
9. Compare all suffixes *Sf* with previous suffix *Sg*
10. Using chosen comparison function (e.g.,Jaro-Winkler)
11. If Jaro-Winkler (*Sf*, *Sg*) > *ts*
12. Group together the suffix reference pairs
13. Corresponding to *Sf* and *sg*.

Querying to gather candidate sets for matching:

14. For record ri1ε *B*
15. Construct BKV by concatenating key fields
16. Generate suffixes of BKV
17. Match suffixes of *A* and *B*
18. Ci resulting set of records with no duplication

4 Results and Comparison

This section briefly describes the results of the existing system, proposed system along with experiments carried out. The brief comparison of same also provided.

4.1 Description of System Execution and Results

Our experiments are designed to compare improved suffix Array blocking against proposed system primarily using the measurements of pair's completeness and pair's quality [17]. We run the experiments on Cora, Restaurant and Real identity data. As mentioned in the Sect. 2, Record linkage is done on the basis of similarity between two string records. Thus for calculating similarity between string records, some similarity measure is used like Jaro and Jaro-Winkler similarity measure [16], as described in Sect. 3.1. Following Figs. 2 and 3 shows results obtained by existing jaro and proposed jaro_winkler.In this case, all suffixes with their corresponding records are compared using Jaro and Jaro-Winkler similarity measure by observing both the Figs. 2 and 3, it can be seen that proposed (Jaro-Winkler) similarity measure gives maximum similarity than existing (jaro) similarity measure. Also time required by Jaro and Jaro-Winkler on Cora, restaurant and real identity dataset is shown in Figs. 4, 5, and6, respectively. All experiment results shown use Jaro-Winkler for the grouping similarity function, and the threshold for determining Jaro-Winkler similarity between two strings is set at 0.85 for all experiments.

Suffix1	Record1	Suffix2	Record2	Jaro Max-Si..
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1997	R75	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1997	R75	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1994	R55	0.81
a-1993	R22	a-1997	R75	0.81
a-1994	R55	a-1997	R75	0.81
a-1994	R55	a-1997	R75	0.81
a-1994	R55	a-1997	R75	0.81
cesa-1993	R21	cesa-1994	R56	0.88
cesa-1993	R21	cesa-1994	R56	0.88
cesa-1993	R21	cesa-1994	R56	0.88
cesa-1993	R21	cesa-1997	R75	0.88
cesa-1993	R21	cesa-1994	R50	0.88
cesa-1993	R21	cesa-1994	R56	0.88
cesa-1993	R21	cesa-1994	R56	0.88

Fig. 2 Suffixes with Jaro similarity (Existing system)

Suffix1	Record1	Suffix2	Record2	Jaro-Winkl...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1997	R75	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1997	R75	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1994	R55	0.8480000...
a-1993	R22	a-1997	R75	0.8480000...
a-1994	R55	a-1997	R75	0.8480000...
a-1994	R55	a-1997	R75	0.8480000...
a-1994	R55	a-1997	R75	0.8480000...
cesa-1993	R21	cesa-1994	R56	0.904
cesa-1993	R21	cesa-1994	R56	0.904
cesa-1993	R21	cesa-1994	R56	0.904
cesa-1993	R21	cesa-1997	R75	0.904
cesa-1993	R21	cesa-1994	R56	0.904
cesa-1993	R21	cesa-1994	R56	0.904
cesa-1993	R21	cesa-1994	R56	0.904

Fig. 3 Suffixes with Jaro-Winkler similarity (Proposed system)

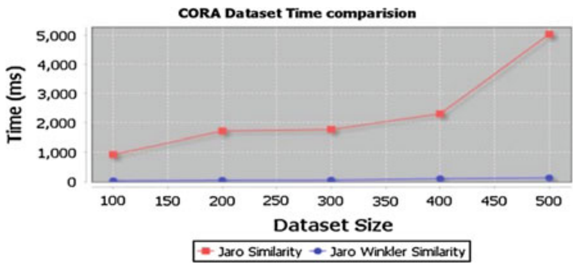

Fig. 4 Time obtained by Jaro and Jaro-winkler similarity on Cora dataset

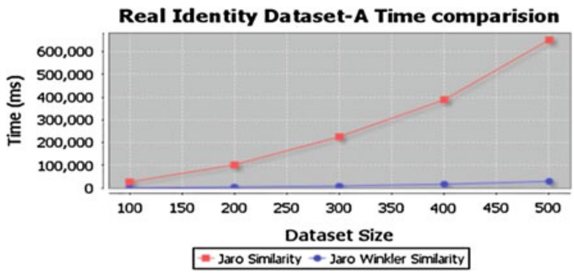

Fig. 5 Time obtained by Jaro and Jaro-winkler similarity on real identity dataset

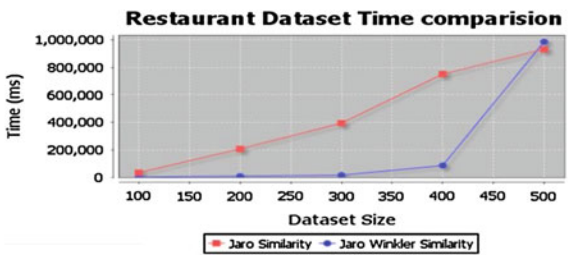

Fig. 6 Time obtained by Jaro and Jaro-winkler similarity on restaurant dataset

5 Conclusion and Future Scope

Suffix array blocking in sliding window fashion is highly capable and relevant to outperform traditional methods in scalability, at the cost of indicative amount of accuracy, depending on the attributes of the data used. Proposed improvement derives these qualities, but significantly improves the accuracy at the cost of very small amount of extra processing. The qualities of suffix array blocking in sliding window fashion make it well suited for large-scale applications of record linkage. We have also shown that the accuracy or pair completeness of proposed Suffix Array blocking is much higher than improved Suffix Array blocking for the data sets we used in our experiments. For example, identity matching of Rina and Tina, proposed approach gives more accuracy as compared to improved suffix array blocking. Because proposed approach generates suffixes in sliding window fashion. As in many industries; it is common situation that many large data sets exist including archival and current. It is necessary to keep that data together, in order to increase knowledge that is available to inform and derive decisions.

In future work link list can be used instead of using suffix array. As by using suffix array we have limitation in size this will not occur in case of link list. So it will be challenging and different to implement system by using link list.

Acknowledgmens The author would like to thank colleagues, friends, all researchers and everyone supported to and associated with the research work.

References

1. P. Christen. “A survey of indexing techniques for scalable record linkage and de- duplication”, IEEE Transactions on Knowledge and Data Engineering, Vol. 24.9, pp. 1537–1555, 2012.
2. Winkler, William E. “Overview of record linkage and current research directions.” *Bureau of the Census*. 2006.
3. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. “Duplicate record detection: A survey”, IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 1, pp. 116, 2007.
4. Vladu, Adrian, and Cosmin Negrueri.“Suffix arrays programming contest approach”,2005.
5. C. Xiao, W. Wang, and X. Lin. “Ed-join: an efficient algorithm for similarity joins with edit distance constraints”, Proceedings of the VLDB Endowment, vol. 1, no. 1, pp. 933–944, 2008.
6. A. Behm, S. Ji, C. Li, and J. Lu. “Space-constrained gram-based indexing for efficient approximate string search”, IEEE ICDE09, Shanghai vol. 2,pp. 604–615, 2009.
7. U. Draisbach and F. Naumann. “A comparison and generalization of blocking and windowing algorithms for duplicate detection”, Workshop on Quality in Databases, held at VLDB09, Lyon vol. 3,pp. 274–283, 2009.
8. N. Adly. “Efficient record linkage using a double embedding scheme”, DMIN09, Las Vegas vol. 2,pp. 274–281, 2009.
9. T. Bernecker, H.-P. Kriegel, N. Mamoulis, M. Renz, and A. Zuefle. “Scalable probabilistic similarity ranking in uncertain databases”, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 9, pp. 1234–1246, 2010.4.

10. Gog, Simon, Alistair Moffat, J. Culpepper, Andrew Turpin, and Anthony Wirth. “Largescale pattern search using reduced-space on-disk suffix arrays”, IEEE Transactions on Knowledge and Data Engineering, VOL. 26, NO. 8, AUGUST 2014.
11. Winkler, William E.. “Overview of record linkage and current research directions”, US Bureau of the Census., Tech. Rep. vol. 2, 2006.
12. M. Weis, F. Naumann, U. Jehle, J. Lufter, and H. Schuster. “Industry-scale duplicate detection ”, Proceedings of the VLDB En- dowment, vol. 1, no. 2, pp. 1253–1264, 2008.
13. G. V. Moustakides and V. S. Verykios. “Optimal stopping: A record-linkage approach”, Journal Data and Information Quality vol. 1, pp. 9:19:34, 2009.
14. P. Christen and A. Pudjijono “Accurate synthetic generation of realistic personal information”, IEEE Transactions on Knowledge and Data Engineering, vol. 5476, pp. 507–514,20095.
15. P. Christen. “Automatic record linkage using seeded nearest neighbour and support vector machine classification”, ACM SIGKDD08, Las Vegas, pp. 151–159, 2008.
16. van der Loo, M., van der Laan, J., Team, R. C. & Logan, N, “Package stringdist”, 2013.
17. T. de Vries, H. Ke, S. Chawla, and P. Christen, “Robust record linkage blocking using suffix arrays,” *ACM CIKM’09*, pp. 305–314, 2009.