Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 1 – No.6, April 2015 – www.caeaccess.org

Contraption of Suffix Array Blocking for Efficacious Record Linkage and De-duplication

Yamini Warke
Savitrbai Phule Pune University
Dr.D.Y.Patil S.O.E.T Pune

Arti Mohanpurkar
Savitribai Phule Pune University
Dr.D.Y.Patil S.O.E.T. Pune

ABSTRACT

Information is united for common purpose from many sidedness computerized files is referred as record linkage. The basic methods compare name and address information across pairs of files to determine those pairs of records that are associated with the same entity. An entity might be a business, a person, or some other type of unit that is listed.

De-duplication is a scold of identifying one or more records in receptacle which represents same object or entity. The same data may be depicting in different way in all possible database causing problem. Diverse indexing techniques have been elaborated for record linkage and de-duplication, in modern time. They are intended to reducing the number of record pairs to be compared in similarity matching process, while at the same time maintaining high matching quality. This paper presents, contraption of suffix array blocking for efficacious record linkage and de-duplication based on different similarity measures.

General Terms

Indexing methods, Record classification

Keywords

Record linkage, suffix array, blocking.

1. INTRODUCTION

As various government agencies, business, and research projects gather together exceptionally large amounts of data, skill that gives rise to processing, examining and mining of large databases have in recent years admire both academy and industry for holding the attention. In the phase of computing data of many data mining projects, linking or matching records which related to same entity from more than two databases become grater tasks. The aim of such linkages is to match and make concrete of all records relating to the same entity, such as sick person, a purchaser, enterprise, a client product, a copyright citation. To allow future use of existing data sources for new studies and minimize the cost and determined attempt in data acquisition, record linkage and de-duplication [16] can be used. Removing duplicate records in a single database is also important one In motor servicing station, refer the example given in table 1. The first name refers to name of Business and its location of residency. The second is the business holders name with his address. Third is the address of accountant who does the books for the company. The name ‘ P A S.Inc’ is an abbreviation of the actual name of the business ‘Patil A Suyash’ which is the holder of motor servicing station. It is potential that different list companion with the set of businesses may have entries equivalent to anyone of the listed forms of the entity which is the motor servicing station. In this case there are so many identical Entries found, that identical (duplications) are corrected when that particular individual return the form. but it is very tedious task if we want [1]that information after some years, as that person may be not at the corresponding address. Table 1.elucidate this example. We can consider another example of banking system; one person may have more than one account in different banks. And that person may use certain different name in each bank. For example, Suppose In IDBI bank he has kept name like Mahajan Yash A and in CANARA bank has kept name as M Yash A and in HDFC like Mahajan Y Ashok All these names are referred to same entity that is (Mahajan Yash A).Here if we use link these records and made concrete of that all records, then it become easy find details of that particular custmor.In order to find out that whether that all names are referred to same person, record linkage is used. On the other hand, as the amount of digital information is rapidly increasing all over the world and most of the data is unstructured one such as image, audio, video & document files. This rapid growth of data size causes several problems such as storage limitation, increasing cost. We can overcome this problem by using de-duplication technique.

Table.1

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 forms.

2. RELETED WORK

In record linkage procedure, a diversity of blocking methods currently used, with the most well-known ones including traditional blocking, sorted neighborhood [11], Q-gram based blocking [3], Canopy Clustering [15], string map based blocking [13] and Suffix Array blocking [5].

To determine which block (or blocks) each record is to be placed into. All blocking methods define a set of key fields from the data to be matched with .To find the correct block; many of these approaches require a single string to be used as the key. Therefore, the values of the key fields are typically concatenated together into one long string. This string is called the Blocking Key Value (BKV) [10].Ordering and selection of key fields to include in the BKV of these fields is important to consider.

A suitable BKV should be the attribute or combination of attributes which are as identifying as possible, uniformly distributed, and having a low error probability. Christen [4] compared and evaluated these blocking techniques, and modified two of them to make them more robust with regards to parameter settings, an important consideration for any algorithm that is to be considered for real-world applications. The experimental results showed that there are large differences in the number of true matched candidate record pairs generated by the different techniques, when tested using the same data sets. Based on statistical classification, Dunn [6] and Marshall [14], and Fellegi and Sunter [8] proposed a theory which is referred as Record linkage.

As various large organizations, businesses have collectively large amount of data. In order to process and analyze that data, matching of records that relate to the same entities from several databases is necessary.

There are several different indexing approaches are available, including traditional blocking, q-gram base indexing, canopy clustering, string- map based indexing, suffix array indexing. The time complexity of traditional blocking is O(dn log n) where n is the number of records in each of the two data sets that are being matched and d is the number of key fields chosen [7].

Basic idea behind suffix array indexing is to insert the BKVs and their suffixes into a suffix array based inverted index. In this indexing technique, only suffixes down to a Minimum length, lm, are inserted into the suffix array.

For example, for a BKV ‘bannana’ and lm = 5, the values ‘bannana’, ‘annana’, ‘nnana’ will be generated, and the identifiers of all records that have this BKV will be inserted into the corresponding four inverted Index lists.

3. METHODOLOGY

3.1 Existing System

In existing improved suffix array blocking [2][17]only suffixes down to minimum length lm are inserted into suffix array. for example ,for BKV ‘abhijit’ and lm=4,the values ‘abhijit’,’bhijit’, ‘hijit’ and ’ijit’will be generated, and the identifiers of all records that have this BKV will be inserted into corresponding four inverted index lists.

To limit the maximum size of blocks and candidate record pairs generated a second parameter, bm, which permit the maximum number of record identifiers in block to be set .Blocks which contain more than bm record identifiers will be removed from suffix array. For example in fig 1.,block with bm=2 having suffix value ‘avi’will be removed. Fig .1.shows suffix array based indexing with given name used as BKVs, a minimum suffix length lm=3 and a maximum block size bm=2.The table 2 shows the resulting sorted suffix array. The block with suffix value ‘avi’will be removed because it contains more than bm record identifiers.

Table.2

Suffix	Identifiers
Vaibhavi	R1
Aibhavi	R1
Ibhavi	R1
Bhavi	R1
Havi	R1
Avi	R1,R2,R3
Vaishnavi	R2
Aishnavi	R2
Ishnavi	R2
Shnavi	R2
Hnavi	R2
Navi	R2
Avi	R2,R1,R3
Pallavi	R3
Allavi	R3
Llavi	R3
Lavi	R3
Avi	R3,R2,R1

Identifires	BKVs(Given-name)	Suffixes
R1	Vaibhavi	Vaibhavi, aibhavi, ibhavi, bhavi, havi, avi
R2	Vaishnavi	Vaishnavi, aishnavi, ishnavi, shnavi, hnavi, navi,avi
R3	Pallavi	Pallavi, allavi, llavi , lavi,avi

Fig .1. suffix array based indexing with given name used as BKVs, a minimum suffix length lm=3 and a maximum block size bm=2

3.2 Proposed System

As in existing system suffixes are generated only down to minimum length. One problem with existing improved suffix array blocking is that errors and variations at the end of BKVs will result in records being inserted into different blocks, missing true matches.To overcome this drawback, a modification of the suffix generation process is to not only generate the true suffixes of BKVs, but all sub-strings down to the minimum lengths of lm in a sliding window fashion. For example, for the BKV ‘Abhijit’ and lm =4 , this approach would generate the sub-strings: ‘abhijit’,’bhijit’, ‘hijit’ , ’ijit,’abhiji’,’bhiji’,’hiji’ .

In existing system BKV is generated by concatenating name and surname .but there is problem with this approach as there may be possible that two person is having same name so at the time de-duplication it results in Missing true matches. In proposed system BKV is generated by concatenating name surname and record id.which impressively overcome problem of existing system. Also comparison function used in proposed system that is Jaro-Winkler and edit distance gives more similarity than existing Jaro. Following fig.2 shows proposed system architecture.

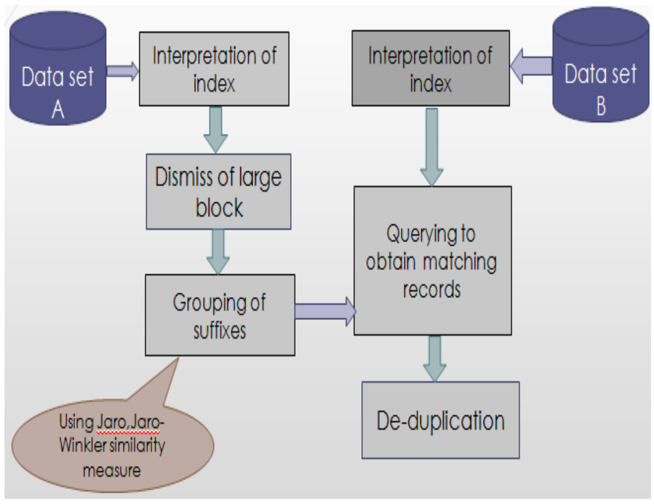

Fig. 2. System architecture

In proposed system, as shown in fig .2, in first step one dataset is taken as input. Dataset including any Japanese and bibliographic data.

In second step, Interpretation of index firstly blocking key value (BKV) is generated by concatenating key fields .then suffixes are generated in sliding widow fashion 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.

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).

3.3 Pseodocode

```
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 ri1 Є 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 suffix 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 ri1 Є 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 duplications.
```

4. RESULT AND COMPARISON

As in existing system Jaro similarity measure is used to compare suffixes.Praposed system uses Jaro-Winkler which gives more similarity than existing Jaro. Table 3 shows result of Jaro and Jaro-Winkler similarity for suffixes. From this it is clear that Jaro-Winkler is more efficient and gives more similarity than Jaro for suffixes..

From fig 3 shows time required for improved suffix array blocking and proposed suffix array blocking.

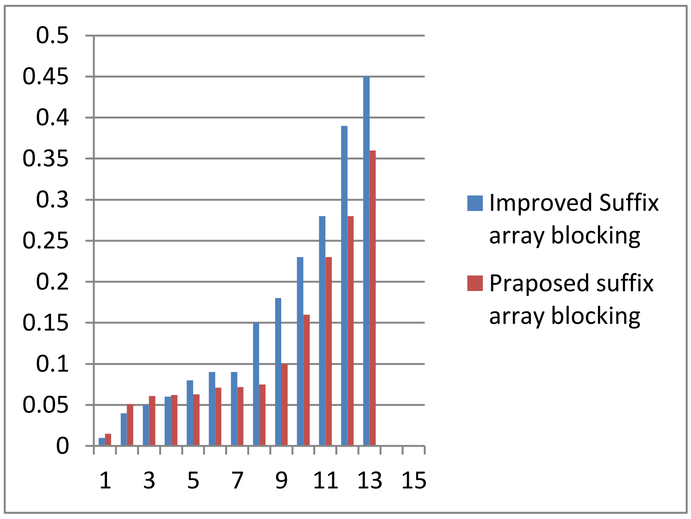

Fig. 3Time comparison graph

Table.3

Suffixes	Jaro similarity	Jaro-winkler Similarity
2353-23534	0.84	0.87
2353-235345	0.79	0.83
2353-235345k	0.76	0.81
2353-235345ka	0.74	0.79
2353-235345kar	0.72	0.77

2353-235345kara	0.70	0.76
23534-235345	0.86	0.89
23534-235345k	0.82	0.86
23534-235345ka	0.80	0.84

5. CONCLUSION
Suffix array blocking 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. Experimental result shows that proposed approach is more scalable than traditional approach for data sets containing millions of records. 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.

6. ACKNOWLEDGMENTS
I would like to thank to all my friends, colleges and researchers.

7. REFERENCES
[1] Winkler, William E. "Overview of record linkage and current research directions." In *Bureau of the Census*. 2006.

[2] Christen, Peter. "A survey of indexing techniques for scalable record linkage and deduplication." *IEEE transactions* on *Knowledge and DataEngineering, vol.* 24.9, 2012,1537-55.

[3] Baxter, Rohan, Peter Christen, and Tim Churches. "A comparison of fast blocking methods for record linkage." In *ACM SIGKDD*, vol. 3, 2003. 25-27.

[4] Christen, Peter. ’’*Towards parameter-free blocking for scalable record linkage’’*. Department of Computer Science, Faculty of Engineering and Information Technology, Australian National University, 2007.

[5] Aizawa, A., & Oyama, K. ‘’A fast linkage detection scheme for multisource information integration.’’ IEEE In *Web Information Retrieval and Integration,. Proceedings. International Workshop on Challenges ,vol.05* , April 2005, 30-39.

[6] Dunn, Halbert L. "Record Linkage*." *American Journal of Public Health and the Nations Health*, vol. 12, 1946,1412-1416.

[7] Elfeky, Mohamed G., Vassilios S. Verykios, and Ahmed K. Elmagarmid. "TAILOR: A record linkage toolbox."IEEE,*18th International Conference on, Data Engineering, Proceedings*, 2002, 17-28.

[8] Fellegi, I. P., & Sunter, A. ‘’A theory for record linkage.’’ *Journal of the American Statistical Association*, vol.*64* (328), 1969. 1183-1210.

[9] Gill, Leicester, Michael Goldacre, Hugh Simmons, Glenys Bettley, and Myfanwy Griffith. "Computerised linking of medical records: methodological guidelines." *Journal of Epidemiology and Community Health*,vol.4, 1993, 316-319.

[10] Gu, Lifang, Rohan Baxter, Deanne Vickers, and Chris Rainsford. "Record linkage: Current practice and future directions." *CSIRO Mathematical and Information Sciences Technical Report* .vol.3, 2003,13-38.

[11] M. A. Hernandez and S. J. Stolfo.’’ Real-world data is dirty: Data cleansing and the merge/purge problem.’’Data Mining and Knowledge Discovery,vol 2(1), 1998,9–37.

[12] Newcombe, Howard B., and James M. Kennedy. "Record linkage: making maximum use of the discriminating power of identifying information."*Communications of the ACM* , vol. 11, 1962, 563-566.

[13] Jin, Liang, Chen Li, and Sharad Mehrotra. "Efficient record linkage in large data sets." *IEEE Eighth International Conference on Database Systems for Advanced Applications.* 2003, 137-146.

[14] Marshall, J. T. "Canada's national vital statistics index." *Population Studies*,vol. 2, 1947, 204-211.

[15] McCallum, A., Nigam, K., & Ungar, L. H. ‘’Efficient clustering of highdimensional data sets with application to reference matching.’’ In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, August 2000 , 169-178.

[16] William, E. W. ‘’*Overview of record linkage and current research directions*.’’ Technical Report, 2006.

[17] De Vries, Timothy, Hui Ke, Sanjay Chawla, and Peter Christen. "Robust record linkage blocking using suffix arrays." *ACM conference on Information and knowledge management*, 2009, 305-314