International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 11, November-2014

Record Linkage & Deduplication Based on Suffix and Prefix Array Indexing

Warke Yamini,
Dr.D.Y.Patil SOET,Pune
Computer Engineering
Savitribai Phule Pune University
Pune,India

Arti Mohanpurkar
Dr.D.Y.Patil SOET,Lohgaon.Pune
Computer Engineering
Savitribai Phule Pune University
Pune,India

***Abstract*— Record linkage is an momentous process in data soundness which is used in combining, matching and duplicate removal from more than two databases that refer to the same entities. Deduplication is the process of taking off duplicate records in a united database. Now a days,data cleaning and standardization becomes an pompous process. Due to yielding capacity of today's database, discovering matching records in united database is a crucial one. Indexing technique specifically suffix and prefix array is used to efficiently implement record linkage deduplication.**

Keywords— Record linkage, suffix and prefix array, blocking

I. INTRODUCTION

As various government agencies ,business, and research projects assemble exceptionally large amounts of data, skill that permit productive processing, examining and mining of large databases have in recent years admire both academy and industry for holding the attention. Linking or matching records which related to same entity from more than two database become grater task in the phase of assembling data of many data mining project. 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 permit further use of existing data sources for new studies and minimize the cost and determined attempt in data acquisition ,record linkage and deduplication can be used. That is why removing duplicate records in a single database is important one.

In motor servicing station ,refer the example given in table 1. The first name refers to Business name and its residential location, the second is the name of the holder of the business with his home 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 Sumit' which is the holder of motor servicing station. It is possible that different list Associated with the set of businesses may have entries corresponding to anyone of the listed forms of the entity which is the motor servicing station. In such case there may be duplicate Entries found, that 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.illustrates this example.

We can take other 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 Bhirud Sparsh P and in CANARA
bank has kept name as B Sparsh p and in HDFC like Bhirud S P. All these names are referred to same entity that is (Bhirud Sparsh P).In order to find out that whether that all names are referred to same person, record linkage is used. 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 deduplication technique. also one familiar example ,when any

Associated Address	Description
SR.#81/7 Near Yogi Hotel Tathwade, Mulashi,Pune,Maharashtra.	Residential location of business
Patil A sumit 345 Shri ram park Dhanori Toad No.7	Residential location of holder of business.
P A S,Inc C/o sunil pegonkar Dhanori road no. Vishrantvadi chowk .Pune.	Incorporated name of business accountant does books and government forms.

Table 1. Examples of Names and Addresses Referring to the Same Business Entity

faculty in our college send us mail at that time faculty has to send same mail to all students so there are too many duplicate copies of same mail in data server. In this case we can use deduplication .we can keep only reference of that mail on server instead keeping whole copy.Suffix array and prefix is used in pattern searching problem in large database. here we can take example suppose there are two people sunny & joy who are playing the very uninteresting game, sunny has very large string and joy asked sunny that ' does the following substring is substring of yours'? joy had asked too many questions to sunny ,sunny has to give the

answer as early as possible. Sunny is programmer so he think that it would be better to know all the substrings that appear in joys string .before doing all this work sunny is wondering

Identifires	BKVs (Givenname)	Suffixes
R1	Yamini	Yamini,amini ,mini,Ini
R2	Damini	Damini,amini, Mini,Ini
R3	Kamini	Kamini,amini ,mini,Ini
R4	Saudamini	Saudamini,audamini,udamin i,damini,amini,mini, Ini

about how many substring will be there. in joys string. Solution to this is that suppose we assume that sunny has string‘’babc’’ hassubstringb’’,ba’’,bab’’,babc’’,bc’’,a’’,ab’’,abc’’,and c’’. determined by the path starting from the root and going toward nodes 2, 3, 4, 5, 6, 7, 8 and 9 in this order. because building the suffix tree is not always a pleasant job and has a quadratic complexity, an approach using suffix prefix arrays would be much more useful[2][13]. Suffix and prefix array is useful for pattern matching using reduced space on disk suffix array. In our computer lots of data is present..we cant store our whole data in main memory we need secondary storage.like hard disk,cd,dvds.so here we can go by one way that we keep only important data or reference to main data in main memory and remaining one at secondary storage.so available space get minimized .[3][12]

There are some advantages &limitation of suffix and prefix array.when suffix and prefix array is used for pattern matching in disk, save the no of disk access and space.[3] [14]To find longest common substring suffix and prefix array is useful. Limitation is that suffix and prefix array is costly construction process.

2. EXISTING SYSTEM

In existing suffix array based indexing only suffixes down to minimum length lm are inserted into suffix array. for example ,for BKV pitambar and lm=5,the values ‘pitambar’,’itambar’,’tambar’,’ambar will be generated ‘and identifiers of all records that have this BKV will be inserted into corresponding four inverted index list. To limit the maximum size of blocks a second parameter, bm,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 ‘amini’, ‘mini’ and ‘ini’will be removed since it contains four record identifiers[4]. As can be seen in fig.1, one problem with suffix array based indexing is that errors and variations at the end of BKVs will result in records being inserted into unusual blocks, and true matches get lost.

3. PROPOSED SYSTEM

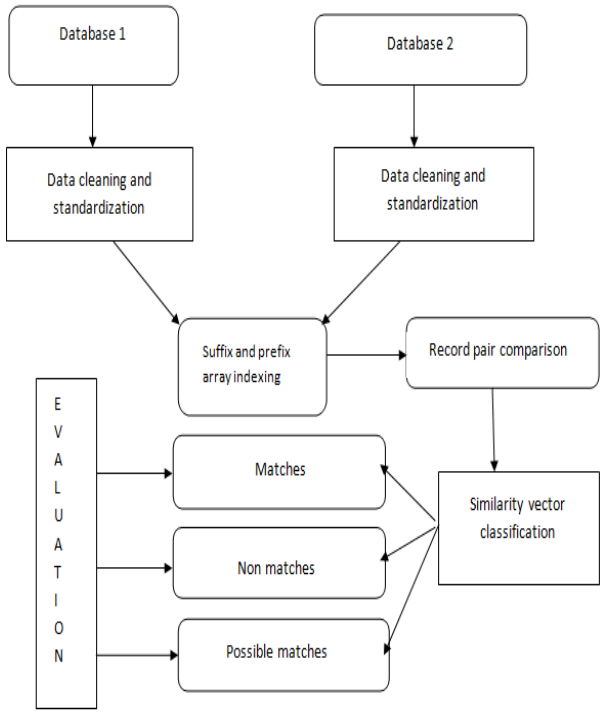
The aim of this technique is to generate the true suffixes as well as prefixes of BKVs with all substrings down to the minimum lengths of lm .For example, for the BKV ‘pitambar’ and lm=5,this approach would generate the substrings: ‘pitambar’, ’pitamba’, ’pitam’, ‘itambar’, ‘ptamb’ ,‘tambar’ ‘ambar’ it can better overcome the errors and variations at different positions in the BKVs creating more blocks and inserting record identifier into larger number of blocks compare to original suffix array technique.

Following fig2 shows general architecture of record linkage process using suffix and prefix array blocking indexing. As most of real world data are soiled and contain rowdy, deficient and erroneous formatted information, a decisive first step in any record linkage and deduplication.

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.the table on right hand side show the resulting sorted suffix array.The block with suffix value ‘amini’ and ‘mini’ will be removed because it contains more than bm record identifiers

project is data cleaning and standardization [6]. It has been recognized that lack of good quality data can be one of the biggest obstacles to successful record linkage[7].

Fig2.general architecture of record linkage process using suffix and prefix array.



Suffix	Identifiers
Yamini	R1
Amini	R1,R2,R3,R4
Mini	R1,R2,R3,R4
Damini	R2,R4
Kamini	R3
Suadamini	R4
Udamini	R4
Audamini	R4
Ini	R1,R2,R3,R4

The main task of data cleaning and standardization is the conversion of the raw input data into well defined consistent form[8][9]. The second step is the suffix and prefix indexing and detailed explanation is given in algorithm. This indexing step generates pairs of candidate records which are compared in detail in the comparison step using variety of comparison function. several fields are normally compared for each record pair, resulting in vector which contain numerical similarity values calculated for that pair. using this similarity values next step is to classify compared candidate record pairs into equal,un-equal,and likely equal depending on decision model used[10][11].

The Suffix Array blocking method is appropriate for a considerable range of applications, but has one limitation. If two BKVs are identical apart from an error positioned less than lms characters away from the end of the BKV string, standard Suffix Array blocking will fail to group these records into the same block. To overcome this problem we propose suffix and prefix array blocking with grouping operation carried on similar suffix and prefix in ordered suffix ,prefix index list.Praposed idea is shown in detail in given algorithm1

Algoritm 1 sufix and prefix array blocking

Input:

1. Qa and Qb, the sets of records to find matches between.
2. The suffix and prefix comparison function similarity threshold rj.
3. The minimum suffix length lms and the maximu block size lmbs.

1. Start
2. Let I be the inverted index structure used.
3. Let Ci be the resulting set of candidates to be used when matching with a record qai
4. // Interpretation of Index structure:
5. For record qbi Є Qb:
6. Construct BKV bbi of given name
7. Generate suffixes and prefixes from bbi,
8. Insert Sbi , Pbi and reference to qbi into I
9. //Dismiss Large Block
10. For every unique suffix Sf and prefix Pf in I
11. If the number of record reference paired with Sf ,Pf > Lmbs
12. Remove all suffix-reference pairs where the suffix, prefix is Sf, Pf respectively.
13. //suffix and prefix grouping
14. For each, unique suffix Sf and prefix Pf in I
15. Compare All suffix Sf and prefix Pf with previous suffix Sg and prefix Pg
16. Using chosen comparison function (e.g.jaro)
17. If Jaro(Sf,Sg) and jaro (Pf,Pg) > rj
18. Group together the suffix and prefix reference pairs
19. Corresponding to Sf, Sg and Pf,Pg respectively.
20. //querying to gather candidate sets for matching:
21. For record qai Є Qa:
22. Construct BKV bai of given name
23. Generate suffixes and prefixes from bai
24. Query I for list of record references which match Qbi
25. Add these references to the set Ci (No duplication)

26 .STOP

4. CONCLUSION AND FUTURE SCOPE

Suffix and prefix 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. Our improvement derives these qualities, but significantly improves the accuracy at the cost of very small amount of extra processing.

In future work we can use link list instead of using suffix and prefix array. As in our proposed system array is used so there is limit for taking array size.Using link list we can solve this problem

REFERENCES

[1] "Winkler, William E. "Overview of record linkage and current research directions."US Bureau of the Census. 2006.," Tech. Rep. RR2006/02, 2006.

[2] Vladu, Adrian, and Cosmin Negruşeri. "Suffix arrays–a programming contest approach." (2005).

[3] Gog, Simon, Alistair Moffat, J. Culpepper, Andrew Turpin, and Anthony Wirth. "Large-scale pattern search using reduced-space on-disk suffix arrays." IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 8, AUGUST 2014

[4] Christen, Peter. "A survey of indexing techniques for scalable record linkage and deduplication." Knowledge and Data Engineering, IEEE Transactions on 24.9 (2012): 1537-1555.

[5] P. Christen, "A comparison of personal name matching: Techniques and practical issues," in Workshop on Mining Complex Data, held at IEEE ICDM'06, Hong Kong, 2006.

[6] Christen, P., Churches, T., & Hegland, M. (2004). Febrl–a parallel open source data linkage system. In Advances in knowledge discovery and data mining (pp. 638-647). Springer Berlin Heidelberg

[7] Clark, D. E. (2004). Practical introduction to record linkage for injury research. Injury Prevention, 10(3), 186-191.

[8] Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4), 3-13.

[9] Churches, Tim, et al. "Preparation of name and address data for record linkage using hidden Markov models." BMC Medical Informatics and Decision Making 2.1 (2002): 9.

[10] Christen, Peter, and Karl Goiser. "Quality and complexity measures for data linkage and deduplication." Quality Measures in Data Mining. Springer Berlin Heidelberg, 2007. 127-151. [11] L. Gu and R. Baxter, "Decision models for record linkage," in Selected Papers from AusDM, Springer LNCS 3755, 2006

[11] Su, Weifeng, Jiying Wang, and Frederick H. Lochovsky. "Record matching over query results from multiple web databases." Knowledge and Data Engineering, IEEE Transactions on 22.4 (2010): 578-589.

[12] Dey, Debabrata, Vijay S. Mookerjee, and Dengpan Liu. "Efficient techniques for online record linkage." Knowledge and Data Engineering, IEEE Transactions on 23.3 (2011): 373-387..

[13] Bernecker, Thomas, et al. "Scalable probabilistic similarity ranking in uncertain databases." Knowledge and Data Engineering, IEEE Transactions on 22.9 (2010): 1234-1246

[14] Bilenko, Mikhail, Beena Kamath, and Raymond J. Mooney. "Adaptive blocking: Learning to scale up record linkage." Data Mining, 2006. ICDM'06. Sixth International Conference on. IEEE, 2006.