International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Removing Dedepulication Using Pattern Serach Suffix Arrays

Pratiksha Dhande[1], Supriya Kumari[2], Sushmita Tupe[3], Laukik Shah[4]

Department of Computer Science Engineering, Savitribai Phule Pune University, G.H.R.I.E.T, Wagholi, Pune, Maharashtra, India

Abstract: ***With the increase of de-duplication in data sets of voter card or pan card, removing the de-duplication is the major challenge. Record linkage is the process of matching records from several databases that refer to the same entities. When appliedon a single database, this process is known as de-duplication. In this paper the investigation is done to how to remove the de-duplication with the help of suffix arrays.Suffix array is well organized data structure for pattern searching. This paper covers similarity metrics that are commonly used to spot similar field entries, and present a widespread set of duplicate detection algorithms that can identify almost duplicate records in a database. It also covers multiple techniques for improving the effectiveness and scalability of estimated duplicate detection algorithms.Finally, based on the algorithms, the paper presents how to remove the de-duplication from dataset.***

Keywords: String search, pattern matching, suffix array, suffix tree

1. Introduction

Databases play an important role in today's' world. Many different sectors depend on the correctness of databases to carry out operations. Hence, the quality of the data stored in databases can have major implications on the system. An essential step in integrating data from different sources is to identify and eliminate duplicate records that refer to the same entity. This process is known as De-duplication.

String search is a well known problem: given a textA[0 . . . $m-1$] over some alphabet $\Sigma$of size $s=|\Sigma|$ and a pattern Q[0 . . .$k-$ 1], locate the occurrences of Q in A. Several different query modes are possible: whether or not Q occurs (*existence* queries); how many times Q occurs (*count* queries); how many byte locations in A at which Q occurs (*locate* queries); and a set of extracted contexts of A that includes each occurrence of Q(*context* queries) [1].When A and Q are provided on a one-off basis, sequential pattern search methods take $O(m+ k)$ time. When A is fixed, and many patterns are to be processed, it is likely to be more efficient to pre-process A and construct an *index*. The *suffix array* is one such index, allowing *locate* queries to be answered in $O(k+ \log m+ y)$ time when there are y occurrences of Q in A, using $O(m\log m)$ bits of space in addition to A. But suffix arrays only provide efficient querying if A plus the index require less main memory than is available on the host computer, because multiple accesses are required to both. For large texts, two-tier structures are needed, with an in-memory component consulted first in order to identify the data that must be retrieved from anon-disk index.

As many businesses, government agencies and research projects collect increasingly large amount of data, techniques that allow efficient processing, analyzing and mining of such massive databases have in recent years attracted interest from both academic and industry[2]. One task that has been recognized to be of increasing importance in many application domains is the matching of records that relate to the same entities from several databases. Often, information from multiple sources needs to be integrated and combined in order to improve data quality, or to enrich data to facilitate more detailed data analysis. The records to be matched frequently correspond to entities that refer to people, such as clients or customers, patients, employees, taxpayers, students, or travelers.

2. Literature Survey

De-duplication is necessary is for the construction of web portals which combine data from different pages possibly created in a distributed method by millions of people. The key challenge in this task is to find a function that can determine when two records refer to the identical entity in spite of errors and conflicts in the data.

One duty that has been recognized to be of growing importance in many application domains is the matching of records which arecount to the same entities from numerous databases. Many businesses practice de-duplication and record linkage techniques with the objective to de-duplicate their databases to increase data quality or compile mailing lists, or to match their data through organisations, for example for collaborative marketing and e-Commerce projects. Various government organizations are now ever more employing record linkage, for instance within and amongst taxation offices and departments of social security to recognize people who register for assistance many times, or who work and gather unemployment benefits. This isat present not clear which indexing technique is appropriate for what type of data and what kind of record link or de-duplication application. This practice has recently been planned as an efficient domain liberatedapproach to multi-source information combination. The simple idea is to insert the BKVs(Bounded Key Value) and their suffixes into a suffix array based inverted index. A suffix array holds strings or sequences and their suffixes in an alphabetically arranged order. Indexing based on suffix arrays has effectively been used for both English and Japanese databases.One of the finest studied particular cases is edit distance, which permits to delete, insert and replace characters(by a unlike one) in both strings. If the dissimilar operations have dissimilar cost or the cost depend

on the characters involved, weare speaking of general edit distance.

3. Proposed System

Pattern matching is an act of checking a given sequence of tokens for the presence of constituents of some pattern. The pattern,generally have the form of either sequences or tree structures. Pattern matching uses fields of voter card to generate a Bounded Key Value(BKV) which is used for suffix array generation.

A distributed database for the selected region is maintained. The data has d attributes a1; : : : ad, each of which could be textual or numeric. The goal of the system is to find the subset of pairs in the cross-product DxD that can be labelled as duplicates. The data preparation stage includes a parsing, data transformation and a standardization step.

BKV is generated by combining few characters of the selected fields, which can be changed dynamically according to the size of the fields. Once the size is determined it will be same for the rest of the records. This can be a combination of characters and numbers. Size of the fields depends on the importance of various fields. Distinct fields will be given more weightage as compared to other fields.

Similarity algorithm considers a set of function which calculates the similarity match between two records based on the subset generated from the suffix arrays. This algorithm performs operation based on the subset of the suffix array. Either it performs insertion, deletion or substitution.

Suffix array is generated from BKV. A suffix array is a sorted array of all suffixes of a string. Depending on the minimum size of suffix, numbers of suffixes are generated. These suffixes are stored in suffix array. Suffix arrays are closely related to suffix trees. Suffix array can be constructed by performing a depth first traversal on a suffix tree.

A suffix tree for text T is a modified suffix tree in which the parent-child edges represent sequences of symbols from rather than single symbols; and in which internal nodesthat only have a single child are eliminated.

At last the final verification using the face recognition algorithm is done. It will check the eyes to nose width ratio which does not change with time and age. This will help to improve the accuracy of the records. The steps involved in this process will be:-

a) Face Detection:-
   - Locate face in a given image
   - Separate it from the scene
b) Face Normalization:-
   - Adjustment
   - Expression
   - Rotation
   - Lighting
   - Scale
   - Head tilt
   - Eye location
c) Face Identification:-
   - Application of a face recognition algorithm.

4. Architecture

Here the architecture shows who the system is going to work:

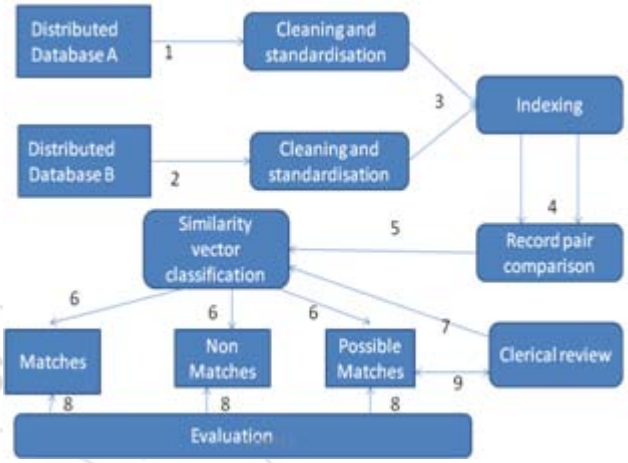

Figure 1: Architecture of the system

1) It is the distributed database which consists of records to be checked. The records must be store in the database.
2) Cleaning and standardization main task is to convert the raw data in well defined data.
3) Indexing step generates the record pairs of candidate records. Candidaterecords thatare generated are compared in detail using comparison function.
4) In record pair comparison all the records are compared that were generated by indexing and for comparison it uses similarity algorithm.
5) Using similarity algorithm like edit distance the strings are compare with each other.
6) If the duplicates are matched or not matched or possible matched evaluation process takes place.
7) Whatever similarity was found was reviewed.
8) Evaluation is required to check what percentage of duplication is there.
9) After the evaluation review the report

5. Algorithm

1: Edit Distance: The edit distance between two strings s1 and s2 is the minimum number of edit operations of single letter needed to transform the string s1 into s2 [2]. There are three types of edit operations:

- insert a letter into the string,
- delete a letter from the string, and
- Replace one letter with a different letter in the string.

The edit distance thus describes how similar or dissimilar two strings are by the number of steps it takes to turn from oneinto the other, where a step is defined as a single letter change.

Example for edit distance is as follows:
What is edit distance between "kitten" and "sitting" ? A minimal edit script that transforms the former into the latter is:

- **k**itten → **s**itten (replace of "s" for "k")
- sitt**e**n → sitt**i**n (replace of "i" for "e")
- sittin → sittin**g** (insertion of "g" at the end).

Therefore the edit distance between "kitten " and "sitting" is 3.

2. Jaro-Winkler Distance: Jaro-Winkler distance string comparison algorithm is mainly used for comparisons of last names and first names.The higher the Jaro–Winkler distance for two strings is, the more similar the strings are. The Jaro–Winkler distance metric is designed and best suited for short strings such as person names. The score is normalized such that 0 equates to no similarity and 1 is an exact match.

6. Future Direction and Conclusion

One can use Huffman code to classify the duplication in database more efficiently. This needs extra efforts and more expertise. Also, one can use artificial intelligence to find out the duplications and it becomes more powerful with time and experience. This will provide more accuracy as compared to any other alternative. But this increases the complexity of the system. Another modification that can be done will be to use fully automated image processing as it will not require any manual inspection. It will select the possible duplicate records and give the appropriate result. In short, the future is bright indeed. Much will change in the years ahead, but one thing is certain: No more unethical means will be used to vote or to use privileged application.

Proposed system will remove all the de-duplication from the data sets so in sensitive things such as voting there duplication of voter cards can be avoided. We have carried out the detailed investigation on our proposed system and also the algorithm we are using that is edit distance algorithm.

7. Acknowledgment

We are working on this project under the guidance of Mrs. Pratiksha Dhande. , Assistant Professor at G. H. Raisoni Institute of Engineering & Technology, Wagholi, Pune.

References

[1] Simon Gog, Alistair Moffat, J. Shane 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
[2] Peter Christen,"A Survey of Indexing Techniques for Scalable Record Linkage and De-duplication", IEEE transactions on knowledge and data engineering, vol. z, no. y, zzzz 2011
[3] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios,Duplicate Record Detection: A Survey", IEEE transactions on knowledge and data engineering, vol. 19, no. 1, january 2007
[4] SunitaSarawagi,AnuradhaBhamidipaty,"InteractiveDed uplication using Active Learning", IIT Bombay
[5] Gonzalo Navarro." A Guided Tour To Approximate String Matching",Dept.of Computer Science,University of Chile,BlancoEncalada 2120 –Santigo-Chile.
[6] M.A. Jaro, "Advances in Record-Linkage Methodology as Appliedto Matching the 1985 Census of Tampa, Florida," J. Am. StatisticalAssoc., vol. 84, no. 406, pp. 414-420, June 1989.
[7] W. W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison ofstring distance metrics for name-matching tasks," in *Workshop onInformation Integration on the Web, held at IJCAI'03*, Acapulco, 2003.
[8] A. Moffat, S. J. Puglisi, and R. Sinha, "Reducing space requirementsfor disk resident suffix arrays," in *Proc. 14th Int. Conf.DASFAA*, Brisbane, QLD, Australia, 2009, pp. 730–744.
[9] V. Mäkinen and G. Navarro, "Compressed compact suffix arrays,"in *Proc. Symp. CPM*, Istanbul, Turkey, 2004, pp. 420–433.
[10] A. Moffat, S. J. Puglisi, and R. Sinha, "Reducing space requirements for disk resident suffix arrays," in *Proc. 14th Int. Conf. DASFAA*, Brisbane, QLD, Australia, 2009, pp. 730–744.