Detecting Duplicates and near Duplicates Records in Large Datasets

Shailesh Singh
Department of Computer Science & Engineering, Jamia Hamdard (Hamdard University), New Delhi, India.
shaileshjuly@gmail.com

Syed Imtiyaz Hassan
Department of Computer Science & Engineering, Jamia Hamdard (Hamdard University), New Delhi, India.
s.imtiyaz@gmail.com

***Abstract*— The rapid growth in data volumes and the need to integrate data from various heterogeneous resources bring to the fore the test of making the efficient detection of the duplicate copy of records in databases. Since the data sources are incoherent and autonomous, they may adopt their own conventions and often, integrating data from different sources may lead to erroneous redundancy of data. To ensure high quality data, the database must validate and filter the incoming data from the external sources. In this regard, data normalization has become a necessity to ensure the high quality of the data stored in these databases. The process of identifying the record pairs that represent the same entity is commonly known as duplicate record detection making it one of the most important tasks in the process of data cleansing. The proposed work suggests an approach to improve the accuracy of the duplicate record detection process which when used in combination with two other concepts of text similarity and edit distance leads to a well filtered data. The background of implementation trials for these concepts was chosen as Scholarship Portal data developed for various organizations where finding and identifying of such records to the most possible extents as well as enabling the genuine students not to be debarred from getting scholarships as it has various kind of reservation/quota mechanism was a dire need.**

Keywords- Big Data; Trigrams; Similarity; Lavensthein Edit Distance; Database data mining; Scholarships

I. INTRODUCTION

Databases play important role in contemporary digitalized world where emphasis is on encouraging paper less alternatives. A number of organizations require quality data for critical decision making like various entitlements, concessions or may it be a distribution system. Often the quality in datasets leads to problems which arise with the rapidly increasing volumes of data stored in real-world databases that are assured by the vital data cleaning process. Data quality problems are encountered in the single data collections, like the files and databases. For example, owing to misspellings during data entry, mistakenly omitted information or other erroneous data or due to the combination of multiple data sources in data warehouses results in significant rise in the need for data normalization. Data cleaning deals with the detection and removal of errors and duplicities as well as inconsistencies from the data to improve the quality of data.

Data cleaning plays a significant role in the process of data mining. It is necessary to enrich the quality of data in a data warehouse prior to the data mining process. Numerous data cleaning techniques are being employed for diverse purposes. The fundamental element of data cleaning is usually termed as duplicate record identification that is the process of identifying the record pairs signifying the same entity (duplicate records). The process of duplicate detection is preceded by a data preparation stage which includes a parsing, a data transformation, and data standardization during which data entries are stored in a uniform manner in the database, resolving the structural heterogeneity problem. Data preparation is also described using the term ETL (Extraction, Transformation, Loading)[12]

Multiple versions of the same record are often accumulated when databases are constructed from multiple autonomous sources. The task of detecting these different versions is known as record deduplication. Generally, the similarity of duplicate records is higher than the random pairs of records. The problem of identifying different records that describe unique entities is denoted by record linkage or duplicate detection.

II. MOVIVATION

In India various kind of scholarships schemes [4] are running by the various ministries with a common goal of empowering the students of the weaker section and motivating the meritorious students to attain education. Ministries like Ministry of Minority Affairs have the schemes targeting the minority communities i.e. Muslims, Christians, Sikhs, Jains, Parsi etc. Similarly Department of Person with disability, Ministry of Social Justice, Ministry of Tribal Affairs, Ministry of Labour, Dept. of Higher Education and Department of School Education and Literacy etc have targeted specific kind of subset of students and awards them with the scholarships

empowering them and use heterogeneous and autonomous data collection sets which are often not in sync with each other.

Since there are different ministries /department participating for the common cause and a student may be eligible for more than one schemes hence there was a need of harmonization to bring all of them under one roof with ease to students to compare the scholarships if he is eligible under more than one criteria at the same time, enabling the ministries to avoid awarding more than scholarships to one student hence extending their reach and supports to more students. Although we have UIDAI for assigning a unique aadhaar number to each citizens in India, And also it is being asked from the students on the portal, it has its own benefits to the students like DBT to their aadhaar seeded account etc., it cannot be mandatorily asked from each students and cannot be used a factor to deprive students from getting scholarships.

Supreme Court guidelines also defers it to make as mandatorily filled fields. In Absence of the unique key factors there were the fair amount of chances of getting of duplicates / near duplicates data as with the demographic data like students name, date of birth, father name, mother name, guardian name, Institution name, course in which studying, multiple registrations can be done by the applicants or registrations can be done in both category fresh as well as renewal. And with the text the task of finding duplicates and near duplicates become so much important as every scholarships schemes have the limited number of sheets which should be filled with these kind of records as well as one can be received two times a scholarships. An intelligence mechanism needs to be built which can detect the duplicates and near duplicates records. Various mechanism were analyzed for carrying out this task like soundex, metaphone, double metaphone, lavensthien edit distance, similarity based on trigraphs studied, Sondex earlier used in the US Census data analytics from 1890 to 1920 works well with English only names .This paper focus on the task of identifying the duplicates and near duplicates records detection also dictionary based typographical error corrections mechanism with help of lavensthein edit distance. Using the proposed strategy we have identified nearly 9 lacs records that was duplicates hence enabling the schemes to accommodate more genuine students.

III. PROBLEM STATEMENT

A. Lack Of Unique Entity Identifiers And Data Quality

In the absence of unique identification in the record set the problem of record matching [1] or duplicate record detection becomes worse. In the view of having no unique identification we have to define a rule for record duplicity let us say R is a record set that consist of n attributes R [r1,r2,…..,rn] , S is another record S [s1,s2,….sn]

Then R is similar to S

If

r1 ~ s1

r2 ~ s2

……

rn ~ sn

For example we have following two dataset

1. [<Rahul singh> , <Guruvinder Singh>,<Shital Shingh>,<Male>,<Uttar Pradesh>]
2. [<Rahul> ,<Guruwinder Singh>, <Sheetal Singh> ,<Male> , <UP>]

Above records will be duplicate or similar only if following pairs are duplicate i.e. guruvinder and guruwinder shital and sheetal ,UP and Uttar Pradesh hold equivalency.

Here we have to exploit dimensional hierarchies to measure co-occurrence among tuples for detecting equivalence errors and for reducing false positives. This is in conjunction with the textual similarity functions employed traditionally for detecting duplicates.

Using the weighted predictions we need to find an algorithm suitable enough to detect the useful similarity between two data sets. The weight of predictions is indicative of the importance of the importance of information used to arrive at the prediction.

In a generic term the problem of duplicate or near duplicate detection is to trace out whether a single real world's entity pretends to be the two or multiples. The text data set, intentionally or unintentionally can be easily prepared as the multiple different entity data by adding typo graphical errors or prevailed name writing practices. Also if dataset grew long then no of comparisons need to be performed becomes very high O(n2). Hence somehow we also need to carefully localize the data and find out the algorithm to reduce no of operations.

Table No.1: Sample Student Data Format

Student Name	Date of Birth	Gender	Father Name	Mother Name	District	State	Institute Name	Course Name

Above table shows the sample data format from the scholarship portal, Column in the table are considered as the subset of the students data used for the activity of duplicate records detection.

While writing the name there is possibility of typing errors like Salim , Salem. Similarly collected data also encounters issue of name writing practices like R.M. Singh and Radha Mohan Singh might be the same person.

We also need to deploy the process of data cleaning in the process of knowledge discovery from the database. Name may or may not have the initials eg. Mr. Mrs. Shri, Smt etc .Name may contain words like s/o , d/o , c/o etc.

As we know that text based comparison cost is more and if no of comparisons are more, then its executions times will increase drastically. We will have to keep no of comparison less as much as possible and use of the data localization/grouping as much as possible.

For some field in the item set we have fixed range of data for example states is India are well known and fixed. This information can serves as the dictionary for correction of these fields' data using some similarity measure or any other mechanism.

IV. RELATED WORK

To the problem of identifying near duplicates we need a score [14] metrics and associated threshold for decision making We have reviewed the available literature to attain better understanding of the existing work and its relevance to our problem domain: the algorithms like Soundex [8], Metaphone, Demetaphne[11][5], Hash Based text similarity[10], Text similarity based on the genetic algorithm[9], Smith-Waterman Algorithm[2] etc.

Felix Naumann [10] in his document gives overview of the problem of Text similarity: As per Neumann[10] similarity between the two data sets can be identified by observing the similarity score of following parameters

a. Edit measures
b. Token Based
c. Hybrid
d. Phonetic
e. Domain Dependent

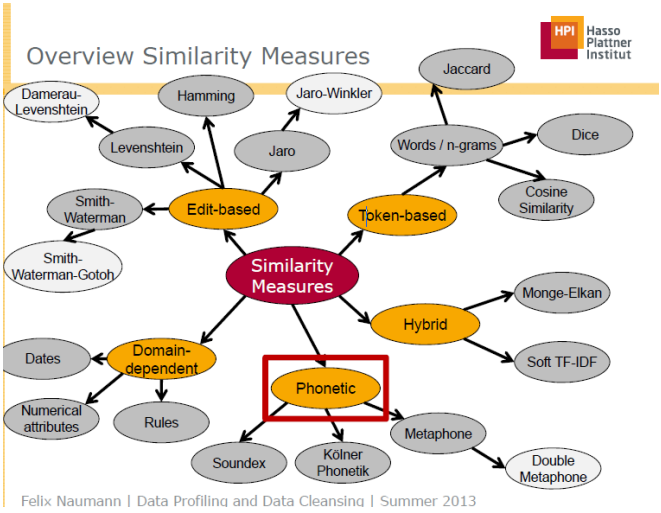

Figure No 1: Overview of Similarity measures [10]

Based on the devised threshold value from the training dataset we make the decision for the real data as whether it is considered as the duplicate records or not.

A. LAVENSTHEIN EDIT DISTANCE

Mathematically Lavensthein Edit Distance [3] between two strings a, b (of length |a| and |b| respectively is given by leva,b(|a|,|b|) where

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Where 1ai ≠ bj is the indicator function equal to zero when ai = bj and equal to one otherwise.

In general term lavensthein edit distance tells the minimum no. of insert/update/delete operations need to perform on the first string to transform it into second strings. Let us take an example of salim and Salem . if we replace i with e then first word can be transformed into second word . So here lavensthein distance between the two string is one.

The Levenshtein distance algorithm has been used in:

- Spell checking
- Speech recognition
- DNA analysis
- Plagiarism detection

1) Steps

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.
6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: d[i-1,j] + 1. b. The cell immediately to the left plus 1: d[i,j-1] + 1. c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

Example: Calculating Lavensthien Edit Distance b/w the words GUMBO and GAMBOL

		G	U	M	B	O
	0	1	2	3	4	5
G	1	0	1	2	3	4
A	2	1	1	2	3	4
M	3	2	2	1	2	3
B	4	3	3	2	1	2
O	5	4	4	3	2	1
L	6	5	5	4	3	2

a)

Figure No 2 : Transposition Matrix for Lavensthein Distance b/w word “GUMBO” and “GAMBOL”.

Normalized Edit Distance is calculated by dividing the lavensthein distance by the max length of the two string under comparison. For above example Levensthien distance = 2 and Normalised distance will be = 2/6 = 0.33

B. TRIGRAMS OR TRIGRAPHS

A trigram [4] is a group of three consecutive characters taken from a string. We can measure the similarity of two strings by counting the number of trigrams they share. This simple idea turns out to be very effective for measuring the similarity of words in many natural languages. A string is considered to have two spaces prefixed and one space suffixed when determining the set of trigrams contained in the string

For example we have a word: shailesh, the trigram of it will be as follows:

{" s"," sh",ail,esh,hai,ile,les,"sh ",sha}

And Trigram of the test sailesh will be :

{" s"," sa",ail,esh,ile,les,sai,"sh "}

C. TRIGAPH SIMILARITY

Returns a number that indicates how similar the two arguments are [4]. The range of the result is zero (indicating that the two strings are completely dissimilar) to one (indicating that the two strings are identical).

For Example similarity score between the name : Shailesh & Shalesh Singh are : 0.642857

D. SOUNDEX

Soundex [7] is a phonetic algorithm for indexing names by sound, as pronounced in English. The goal is for homophones to be encoded to the same representation so that they can be matched despite minor differences in spelling. The algorithm mainly encodes consonants; a vowel will not be encoded unless it is the first letter

Soundex was developed by Robert C. Russell and Margaret King Odell and patented in 1918] and 1922. A variation called American Soundex was used in the 1930s for a retrospective analysis of the US censuses from 1890 through 1920[8].

But soundex does not work well in case of non-English names. Since we have the data set of all non-English names hence use of soundex is not preferred

V. PROPOSED SOLUTION

We have divided problem stamen in the two part:

A Dictionary based error correction of misspelled word

B Finding similarity and normalized edit distance between the data sets

A. DICTIONARY BASED ERROR CORRECTION OF MISSPELLED WORD

Since we have captured the data via online and offline mode in form of excel sheets there are possibilities of data inconsistency in case of data being collected like typographical errors while typing in excels. For example while typing for district name it may be mistyped due to human error. For correcting these errors we proposed to use dictionary based correction. In this process we lavensthein edit distance will be used to correct the word based on the edit distance from the dictionary word.

Let us take example: we have dictionary of following words for District Relation

Table No.2: Sample Data for Dictionary

Kanpur	Chandigarh	Allahabad
Bangalore	Delhi	Etawah
Tiruvanantpuram		

And user types the word: chandigarh

Then based on the edit distance from the dictionary words we can correct this misspelled word to Chandigarh

Lavensthein (‘chandigarh’ , ‘chandigarh’) =1

Similarly Bengalore , Bangaloree, bangaloore can be corrected to Bangalore

Using the dictionary based correction errors may be corrected from the text data and cleaned data will be used for further steps.

B. FINDING SIMILARITY AND NORMALIZED EDIT DISTANCE BETWEEN THE DATA SETS

First Step is important to correct misspelled word of district and state relation, because for the next steps where dictionary based word correction is not possible, to reduce the no of comparison we will be localize our dataset based on the district relation. Hence dividing the complete dataset to n number of subsets. And we need to perform second step within this subset, hence optimizing no of comparisons

For the next step we have used following attributes of data to identify the duplicates and near duplicated entity:

- Applicant Name
- Date of Birth
- Gender
- Father Name
- Mother Name
- District
- State
- Institute Name
- Course Name

For the above chosen attributes major issue is with the textual attribute data like Applicant Name, Father Name and Mother Name Since these are text data and we cannot create any dictionary for it as we did for the correction of district name and state name. Here finding the pattern via various pattern matching methodology may not be used and may not provide us the appropriate results. Neither regular expression nor text equality can be used.

Knowledge Discovery in Database process:

Extraction of hidden knowledge[13] from unstructured data is explained by author in his paper .Here our target is to gain the knowledge from the raw data stored in the relational database format in terms of delicacy amongst them. For this first of all Data cleaning needs to be done. For Example We may found in some name character like Mr., Mrs. , Shri, Smt. Dr. etc. . These words should be removed as part of data cleaning process.

Similarly if name consist of s/o, d/o, u/g keywords meaning it consist the name as well as father/guardian name for the single name attributes, it must be cleaned based on the pattern matching. Here if we found these key words then we keep the value exist before key words. For example if we have name like Roshan Singh s/o Mr. Havinder Singh then after applying data cleansing process[Regular Expression Pattern Matching] we will retain Roshan Singh only.

After these steps we applied the trigraph similarity algorithm along with the lavensthein distance b/w the different entity set. For the reducing the no of comparisons for the getting trigraph similarity we will be deviding the complete data set into small cluster here we will group all data belong to one district to one cluster. And entity need to be compared for the similarity within rest entities of the clusters.

Here no of comparisons are very costly operation if we had n item set then we required n*n comparisons, when n becomes larger the comparisons execution becomes costlier.

We can minimizes the number of comparisons with a little modification.

Suppose we have n item sets then we will rank them from 1 to n. And 1st item set need to be compared with the rest n-1 item sets, 2nd item set need to be compared to rest n-2 item sets. Similarly mth item set need to be compared with the n-m item sets only. Hence no if comparison can be reduced to n*n to n+(n-1)+(n-2)+…..+(n-m) +……..1 = n*(n+1)/2 .

Let us suppose n=25000 then earlier we required 6.25 x 108 comparisons while using modified version we required 3.12 x 108 that is nearly half of earlier required comparisons.

Summarizing the above methodology following steps were adopted in the process:

1. Pre-process the name [rule based data cleaning]
2. Data set Partitioning with help of geographical localization of data i.e. over district relation of dataset
3. Compare each attribute of dataset to another data set in the same bucket
4. Compute trigraph similarity as well as the normalized edit distance between the attributes of the dataset
5. Apply the threshold over the computed parameters, i.e. decision making based on threshold [grouping of near duplicates]
6. After we finished comparing all data set against each other, we were left with the groups of duplicates records based on the set threshold limit.

Table No.3 Similarity and normalized edit distance(Nled) score

Sr. no	Table			
	String one	String two	Sim Score	Nled score
1	IMRAN HUSSAIN	IMRAN HUSAIN	0.8	0.153
2	MOHAIDEEN ABDUL KADAR JAILANI M	A M MOHAIDEEN ABDUL KADAR JAILANI	0.97	0.181
3	ROONAQ QAYOOM	ROONAK QAYOOM	0.75	0.076
4	ZAID MANZOOR	ZAHID MANZOOR	0.69	0.076
5	CHANDANA SURESH RAMESHBHAI	CHANDANA SURESHBHAI RAMESHBHAI	0.96	0.133
6	GAZALAH SAKEENA	GAZALA SAKEENA JOHN	0.64	0.315
7	AFSANA RAJESAB INAMDAR	APASANA RAJESAB INAMADAR	0.6	0.296

VI. Results and Discussion

In this section we will present our experimental result and also a formal discussion over the findings. The result demonstrates following things.

We have examined the proposed solutions output for the smaller data set /training data set to set our threshold parameter values to 0.6 for similarity score and normalized edit distance to 0.3

Then we have executed proposed algorithms to the nearly 1.8 crore datasets and we are able to identify the 9.3 lacs records that are near duplicates.

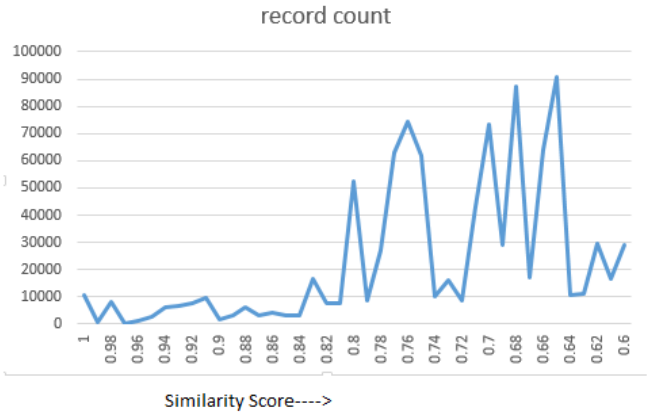

Figure No 2: Similarity score vs record count

VII. Conclusions

This work has presented an algorithm for overcoming the issue of inconsistent data and detecting the near duplicates with an advance approach. With the combined scientific concepts of text similarity and the edit distance the accuracy of the results has been improved to further extent. And proposed work can further extended to integrate it with the online system for intelligent near duplicity detection system.

References

[1] Peter Christen "Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution" Springer Heidelberg, ISBN: 978-3-642-31164-2

[2] Alvero E. Monge "An adaptive and efficient algorithm for detecting approximately duplicate database record" California State University Lond Beach CECS Department CA 90840 8302

[3] Li Yujian And Liu Bo "A Normalized Levenshtein Distance Metric" IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 29, No. 6, June 2007

[4] Oleg Bartunov, Teodor Sigaev "Effective Similarity Search In PostgreSQL " Lomonosov Moscow State University , PGCon-2012, Ottawa

[5] Deepa, K. and Rangarajan, R. "A Comprehensive Review of Significant Researches on Duplicate Record Detection in Databases", Advances in Computational Sciences and Technology, Vol. 2, No. 2 pp. 117-134, 2009.

[6] National Scholarship Portal -URL: www.scholarships.gov.in

[7] Frankie Patman and Leonard Shaefer"The Hidden Risks of Soundex-Based Name Searching" IBM Global Name Recognition G507-1515-00
[8] "Information on the Soundex Indexing System" The United States Cencus Bearau URL: https://www.census.gov/history/www/genealogy/decennial_census_records/soundex_1.html
[9] "Duplicate Record Detection Using Soft Computing Approaches" Thesis By Deepa K Faculty Of Information And Communication Engineering Anna University Chennai 600 025 November 2013
[10] Felix Naumann "Similarity measures " 11.6.2013 Hasso Pattner Institute,,IT Systems Engineering University of Potsdam
[11] Ashis Kumar Mandal, Md. Delowar Hossain, Md.Nadim "Developing an Efficient Search Suggestion Generator, Ignoring Spelling Error for High Speed Data Retrieval Using Double Metaphone Algorithm " Proceedings of 13th International Conference on Computer and Information Technology (ICCIT 2010) 23-25 December, 2010, Dhaka, Bangladesh
[12] Erhard Rahm & Hong Hai Do "Data Cleaning: Problems and Current Approaches" University of Leipzig, Germany http://dbs.uni-leipzig.de
[13] Syed Imtiyaz Hassan, "Designing a flexible system for automatic detection of categorical student sentiment polarity using machine learning", International Journal of u- and e- Service, Science and Technology, vol. 10, issue.3, Mar 2017, pp. 25-32, ISSN: 2005-4246.
[14] Syed Imtiyaz Hassan, "Extracting the sentiment score of customer review from unstructured big data using Map Reduce algorithm", International Journal of Database Theory and Application, vol. 9, issue 12, Dec 2016, pp.289-298, doi: 10.14257/ijdta.2016.9.12.26, ISSN: 2005-4270.