Jaro Winkler Fuzzy match algorithm to calculate a similarity index between two strings using open source platform

[1]Mustak Ali, [2]Amlan Saikia, [3]Runjun Baruah, [4]Utpal Sarma
[1, 2]Project Scientist, [3]SSO, [4]PSO & Head i/c
Assam Remote Sensing Application Centre, ASTEC, Bigyan Bhawan
Guwahati, India
[1]mustakali25@gmail.com, [2]amlan.saikia01@gmail.com, [3]brunjun@gmail.com, [4]usarma552@gmail.com

Abstract: *In geospatial domain data structure plays an important role as they are the core component for analysis. In real life situation, data need to be acquired from various sources and hence compatibility, standardization of data structure is a limitation. These are most often variable in nature. Primary key and reliable identifiers allow the user to match records from two or more data sets under various open source/proprietary software using the utility tools. The limitations of a non-primary key between two different data sets create a problem in data integration which results in manual work. An attempt has been made to harness the power of PENTAHO DATA INTEGRATION (PDI) Jaro Winkler fuzzy matching in this regard. The algorithm involved here is based on duplicate-detection and calculates the similarity of two streams of data. The process results in a numeric index between zero and one i.e. zero indicating no similarity and one indicating an identical match. The said algorithm was used to match the name of villages of Assam from two different attribute tables. The main limitation encountered during integration was either one or more characters of the village names were transposed or incorrectly recorded. To overcome the conventional method of correlating the tables, this method is found to be immensely beneficial to save time in correlating and integrating the database.*

Keywords: *GIS, RS, PDI, Jaro Winkler algorithm, Fuzzy matching*

1. INTRODUCTION:

The task of matching entity names has been explored by a number of communities, including statistics, databases, and artificial intelligence. Each community has formulated the problem differently, and different techniques have been proposed [1]. When data sources and sets contain consistent and valid data values, share common unique identifier(s), and have no missing data, the matching process rarely presents any problems. But, when data originating from multiple sources contain duplicate observations, unreliable keys, missing/invalid values, capitalization and punctuation issues, inconsistent matching variables, and imprecise text identifiers, the matching process is often compromised by unreliable and/or unpredictable results [2]. Different techniques, platforms and level of expertise are utilized and applied to accomplish the task of standardizing and integrating the database. There are different proprietary software like SAS (PROC SQL by using COMPGED [3]), MS Excel (VLOOKUP: INDEX, MATCH and IF [4]) and Microsoft SQL Server Integration Services which can easily tackle the problem. In small R&D projects, NGOs, research institutions and academia cannot afford for data specialists along with proprietary software to perform cleaning and standardizing operations on small databases. They rather opt to perform manual task on the databases in GIS software preferably with in-house expertise. In such scenario open source platform holds an edge to overcome the issue related to licensed software. Recently, similar task were carried out implementing an open-source, Java toolkit of name-matching methods [5] that included a variety of different techniques. A comparison of several string distances on the tasks of matching and clustering lists of entity names [6] as a toolkit were conducted too. In addition to existing string-distance methods, a hybrid of cosine similarity and the Jaro-Winkler method [7], were also performed on similar kind of problems. In this paper we introduce a simplified and customized approach to solve the problem related to databases. We chose Pentaho Data Integration (PDI), long known as the Kettle, is an open source ETL that allows designing and implementing data handling and transformation. It is a comprehensive tool with advanced features such as "clustering" of ETL processing. These features are available from the open source version of PDI and are found only in commercial versions of ETLS competitors [8]. Along with this, QGIS and LibreOffice are also used to carry out normal join operation and data handling.

2. STUDY AREA:

Assam, a north eastern state of India, is divided into 33 administrative geographical districts with Dispur as the capital. Assam has 26,784 villages from 33 districts (Table 1). The cadastral maps available at the Director of land records department are converted from paper maps to GIS platform by digitizing them to integrate with the records made available from various other departments. This would help for better land and resource management as well as planning at the grass root level. The process of adding the entire column from different tables need manual human intervention and conventional method of quality checking by editing the rows of database to achieve the final table. We chose the Nalbari district (465 villages) parcels of Assam (Fig 1). The layer needs to integrate the required tables from different sources and see that the process of Jaro-Winkler fuzzy match can help in this to minimize time in efficient way.

District	Village(No)	District	Village(No)	District	Village(No)
Baksa	692	Dima Hasao	695	Lakhimpur	1184
Barpeta	838	Goalpara	838	Majuli	248
Biswanath	832	Golaghat	1127	Morigaon	640
Bongaigaon	566	Hailakandi	322	Nagaon	1018
Cachar	1059	Hojai	405	Nalbari	465
Charaideu	345	Jorhat	619	Sibsagar	531
Chirang	509	Kamrup	1091	Sonitpur	1052
Darrang	561	Kamrup Metropolitan	321	South Salmara Mankachar	315
Dhemaji	1328	Karbi Anglong	1269	Tinsukia	1180
Dhubri	938	Kar		Jdalguri	802
Dibrugarh	1356	Kokrajhar	912	West Karbi Anglong	1653
Total	**9024**	**Total**	**8672**	**Total**	**9088**
	Grand total No. of Villages: 26784				

TABLE 1: NO. OF VILLAGES IN EACH DISTRICT OF ASSAM
Source: http://dilrmp.nic.in

3. DATA USED:

The paper map of Assam from the Directorate of Land record department is digitized to acquire the cadastral village parcel layer. The space based information system for Decentralized planning (SIS-DP) layer driven from the Thana map is joined with LAC (Legislative Constituency) map. The population attribute of Nalbari (D_Nalbari_PCA) is being extracted from the spreadsheet of population census of India, 2011. A_nalbari_cad, B_nalbari_sisdp and C_Nalbari_lac are the GIS layer of Nalbari district of Assam. Each layer is comprised of shapefile (.shp), shape projection (.prj), database file (.dbf), shape index file (.shx).The database file (.dbf) of these layers are converted into spreadsheet in libreoffice which contain the village name. The layers are from different sources and contain same number of parcel (473) against each village name except the Nalbari Population Census spreadsheet which contains (482) parcels. The two layers (B_nalbari_sisdp and C_Nalbari_lac) and the spreadsheet (D_Nalbari_PCA) are managed and edited in a way to pick the desired rows based on the village name column finally to aggregate into the cadastral layer (A_nalbari_cad).

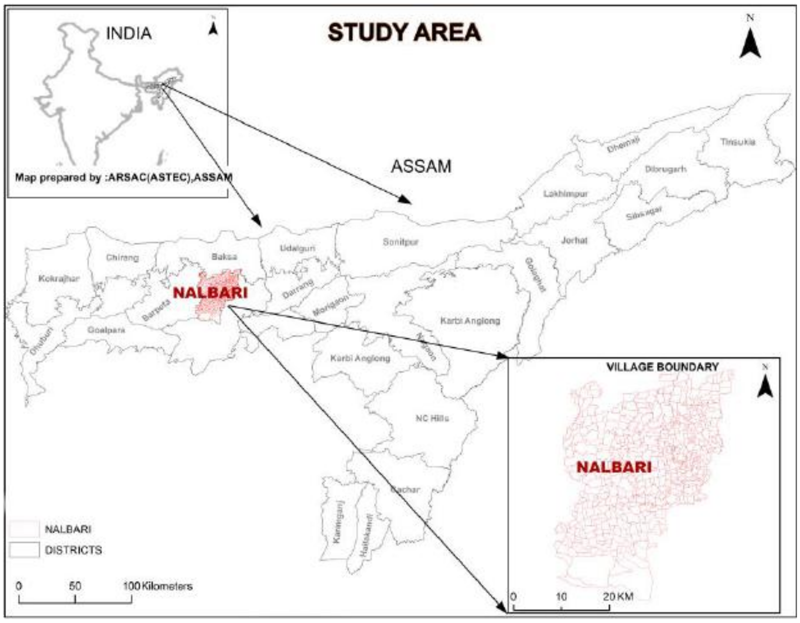

Fig. 1 MAP HIGHLIGHTING STUDY AREA IN RED COLOR

4. METHODOLOGY AND ANALYSIS:

The database attached to GIS layer used by different department in multi-disciplinary domain projects need accurate attribute for generating comprehensive and correct analysis. The capabilities of GIS software to add, edit, remove, relate, join and merge attributes has therefore become first go to choice for the earth science and allied Professionals. This tool allows executing their task in very easy manner. Complex nature of different database more often compels the data to switch into the different proprietary database software engine like ORACLE and SAS using SQL. A high level of experts are required to perform different matching algorithm to find the best possible matching cases for the rows to make it a primary key to perform join for the databases. The methodology below becomes interesting as GIS experts from non-SQL background can get the desired result in few steps (Fig.2) using the Jaro-Winkler fuzzy match algorithm in Pentaho data integration software, along with QGIS and LibreOffice. These earlier were done manually which cost time, money to procure the proprietary licence and hire experts.

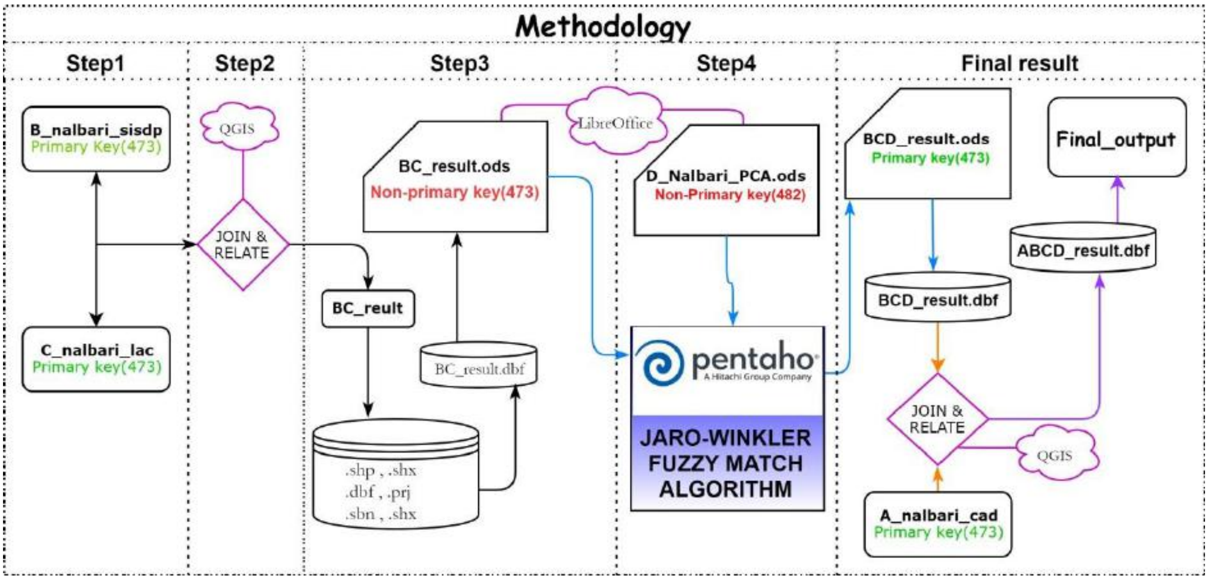

Fig. 2 METHODOLOGY DETAILING THE PROCESS OF STRING METRIC ALGORITHM IMPLEMENTATION

4.1 DATA PROCESSING:

Pentaho is a powerful Business Intelligence open source suite that offers many features, including reporting, OLAP pivot tables and dash-boarding. Data integration can be seen as the process that combines data from a variety of sources in order to provide a coherent view. PENTAHO DATA INTEGRATION provides a graphical interface "Spoon" (based on SWT), from which one can create two types of treatment: transformations and tasks (jobs). Transformations work with streams of data, transforming the rows according to the declared steps. Jobs contain a sequence of transformations and other auxiliary tasks [9]. In the event of our data ETL process, the desired data is identified and extracted from different sources (Fig 3a). Transformations are then applied to the extracted data (BC_result & D_Nalbari_PCA) utilizing the Jaro-Winkler Fuzzy Match algorithm (Fig 3b). Finally, resulting data (BCD_result) is obtained in the form of files or loaded into target database (Fig 3c). The entire process is summed up in the following sections.

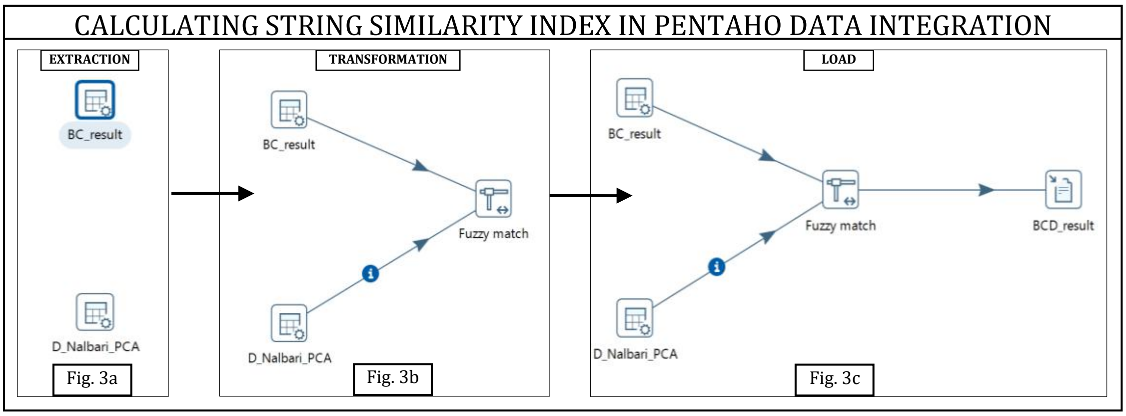

Fig. 3 CALCULATING STRING SIMILARITY INDEX IN PENTAHO DATA INTEGRATION

5. MATHEMATICAL MODEL:

The fuzzy match algorithm applied above is based on **Jaro–Winkler distance,** which is a string metric for measuring the edit distance between two sequences. It is a variant proposed in 1990 by William E. Winkler of the **Jaro distance** metric [10]. Informally, the Jaro distance between two words is the minimum number of single-character transpositions required to change one word into the other.

The Jaro distance is a measure of similarity between two strings. The higher the Jaro distance for two strings is, the more similar the strings are. The Jaro distance $\boldsymbol{d_j}$ of two given strings $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ is

$$d_j = \begin{cases} 0 & \text{If, m = 0} \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{Otherwise} \end{cases}$$

Where:

- ***m*** is the number of matching characters; • ***t*** is half the number of transpositions

Each character of $\boldsymbol{s_1}$ is compared with all its matching characters in $\boldsymbol{s_2}$. The number of matching (but different sequence order) characters divided by 2 defines the number of *transpositions*.

Example: Given the strings $\boldsymbol{s_1}$ **kumrikata** and $\boldsymbol{s_2}$ **kumarikata**, we find:

•$\boldsymbol{m} = 9$ •$|\boldsymbol{s_1}| = 9$ •$|\boldsymbol{s_2}| = 10$ •$\boldsymbol{t} = 4.5$

We find a Jaro score of:

$$d_j = \frac{1}{3}\left(\frac{9}{9} + \frac{9}{10} + \frac{9-4.5}{9}\right) = 0.89889$$

6. RESULT AND DISCUSSION:

Jaro-Winkler distance metric algorithm was applied to village name (data columns) from BC_result & D_Nalbari_PCA table with data rows amounting to 472 and 483 respectively. The algorithm resulted to an index measure between 0 and 1. Values obtained in our case were subdivided into four continuous ranges on the basis of data anomaly shared among them to better understand the results. A total count for each range is depicted with the help of a pie chart (Fig 5). Perfect match resulted to 37% and the score associated with the same equates to 1(Fig 4(a)). These data columns are inevitably selected as the unique identifier.

1	BC_result	D_Nalbari_PCA	measure value
2	Madhapur	Madhapur	1
3	Jaha	Jaha	1
4	Arara	Arara	1
5	Naptipara	Naptipara	1
6	Barnibari	Barnibari	1

Fig. 4(a)

369	BC_result	D_Nalbari_PCA	measure value
370	Pitanipara	Pitnipara	0.898888889
371	Khakhrisal	Kharsitha	0.898240741
372	Madhaya Kajia	Madhya Kazia	0.898018648
373	Batahgila	Niz-Batahgila	0.897435897
374	Bar Makhibaha	Bar Makhibaha(Barmakueibna)	0.896296296

Fig. 4(c)

177	BC_result	D_Nalbari_PCA	measure value
178	Sidalkuchilachima	Sidalkuchi Lachima	0.988888889
179	Khudra Chenikuchi	Khudrachenikuchi	0.988235294
180	Dehar Kalakuchi	Deharkalakuchi	0.986666667
181	Larma Batakuchi	Larmabatakuchi	0.986666667
182	Khudrakulhati	Khudra Kulhati	0.985714286

Fig. 4(b)

464	BC_result	D_Nalbari_PCA	measure value
465	Hublakha	Bhelakhaiti	0.789502165
466	No 1 Bardhanara	No.1.Barbala	0.786666667
467	No 2 Bardhanara	No.2.Barbala	0.786666667
468	Damdamar Pathar	Damal	0.782222222
469	Rongaphali	Ghorathal	0.778306878

Fig. 4(d)

Fig. 4: STRING SETS WITH INDEX SCORE RANGES VARYING BETWEEN (0.7 AND 1)
Fig. 4(a): STRINGS WITH PERFECT MATCH SCORE 1, Fig 4(b): STRINGS WITH NEAR PERFECT MATCH SCORE (0.99 - 0.9),
Fig. 4(c): STRINGS WITH SCORE (0.89 - 0.8) & Fig. 4(d): STRINGS WITH LEAST MATCH INDEX SCORE (0.79 - 0.7)

For index score varying between a measure of (0.99 – 0.90), data in the range is considered near perfect match amounting to 40% of the lot. An in-depth analysis of the data in this range highlights blank spaces between groups of strings as the primary anomaly resulting to the index score. However, this set of data (Fig 4(b)) can also be considered for unique identifier selection without any manual intervention. Remaining categories varying between (0.89 – 0.80) and (0.79 – 0.70) respectively need special attention. On critical examination for data in the range (0.89 – 0.80), letter transposition on a scale of +/- 2 in one of the two data columns (Fig 4(c)) has been found. Although, on application of manual rectification, the data is deemed fit for selection as unique identifier. The final set of data (Fig 4(d)) in the range (0.79 – 0.70) tend to be the least matching case. As the algorithm tries to map data from one column to other, best match possible is drawn by the algorithm. This matching results in a low index score eventually leading to manual rectification. The overall process is a lot easier than manually interpreting each data row, which not only is time efficient but also takes up fewer steps.

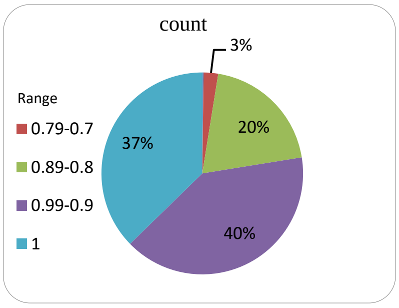

Fig. 5 PIE REPRESENTATION OF SCORES

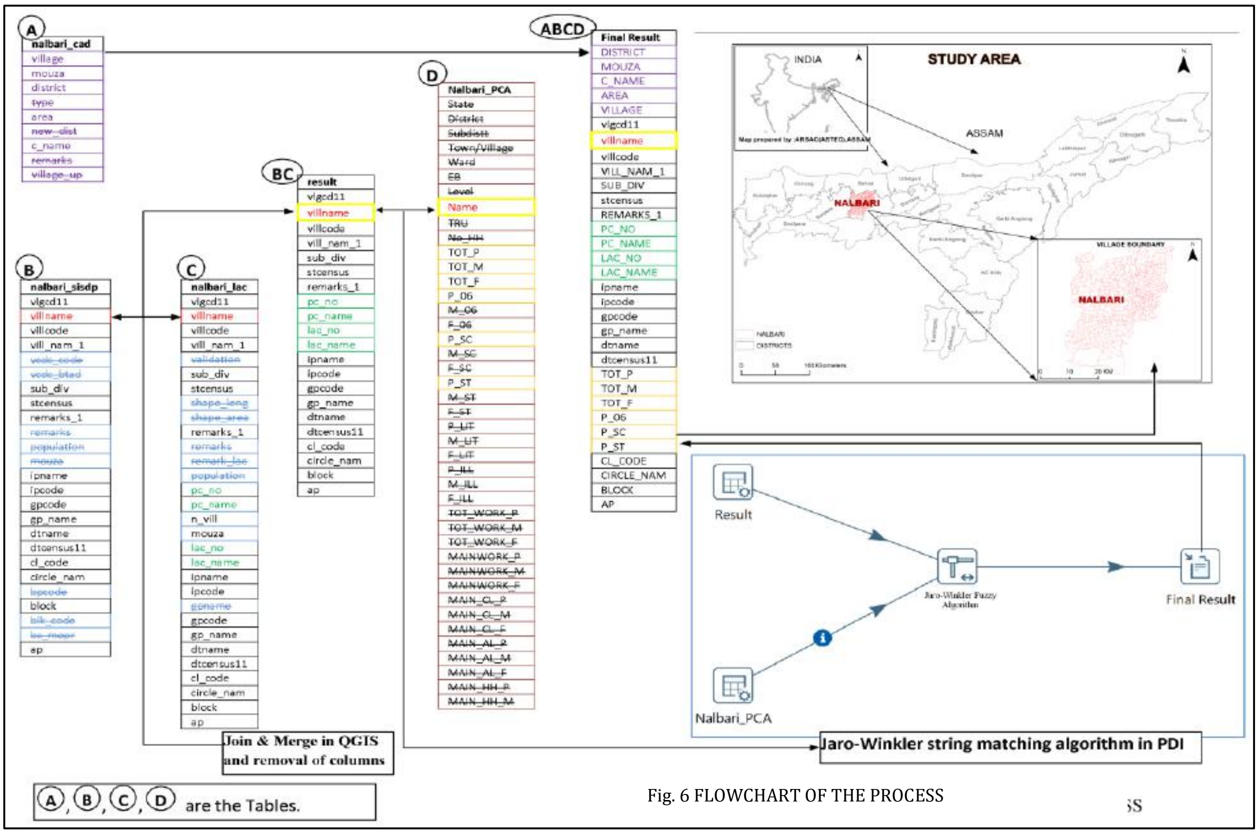

Fig. 6 FLOWCHART OF THE PROCESS

The outcome of the methodology for the study area gave us nearly 334 unique keys against 473 and 482 data rows, thus as a result we can conclude that 70% of the data rows were easily joined in QGIS while leaving out the remaining 30% for manual intervention.

7. CRITICAL ANALYSIS:

The PDI tool allows for a long list of String metric algorithms to choose from (**Levenshtein and Damerau-Levenshtein, Needleman-Wunsch, Pair letters similarity, Metaphone, Double Metaphone, Soundex, and RefinedSoundEx)**. These algorithms have one thing in common: They are aimed at matching strings. The way they do that, however, varies, which makes some algorithms more suited for some specific projects based on understanding the nature of the databases. Jaro-Winkler is an excellent choice for finding matching duplicates possibly containing misspellings. Caution is needed, however; **Jos van Dongen** and **Davidson** have a higher Jaro-Winkler similarity score than **Jos van Dongen** and **Dongen, J van**, but no human being would pick the former over the latter as a possible duplicate candidate [11].

In the event of our ETL process, highlighted data rows fig 7 reflects the abnormality in data matching. Index score in both the cases is around 0.8, which certainly is assumed to be a good score. However, on close examination corresponding data rows are found to be non-matching and thus contradicting with reference to the algorithm performance.

Jaro-Winkler algorithm has been adjudged as the best, with the slowest taking 2 to 3 times as long as the fastest [12]. Of course these times are dependent on the lengths of the strings and the implementations, and there are ways to optimize these algorithms that may not have been used.

11	BC_result	D_Nalbari_PCA	measure value
12	Badrukuchi	Bhuyarkuchi	0.882121212
13	Arara	Arara	1
14	Amoyapur	Amaya-Pur	0.869312169
15	Serabari	Cherabari	0.884259259
16	Sariya	Sariahtali	0.866666667
17	Pub Kalakuchi	Pub-Kalakuchi	0.964102564
18	Naptipara	Naptipara	1
19	Diruwa	Dirua	0.966666667
20	Barnibari	Barnibari	1
21	No 3 Bartala	No.3.Barbala	0.866666667
22	No 4 Bartala	No.4.Barbala	0.866666667
23	No 1 Kaplabori	No.1.Kaplabari	0.885714286
24	Damdamar Pathar	Damal	0.782222222
25	Roumari Damdama	Rowmari Domdoma	0.848888889
26	Bonpura	Bonpura	1
27	Kharkaldi	Kharkaldi	1
28	Kochuar Pathar	Khudra Katra	0.804285714
29	Bamunditari	Bamundittari	0.983333333
30	No 1 Naruwa	No.1.Narua	0.873939394
31	No 1 Kekankuchi	No.2.Kekankuchi	0.893333333

Fig 7: HIGHLIGHTED ROWS REFLECT ON THE SHORTCOMINGS OF THIS ALGORITHM AS DISCUSSED ABOVE.

8. CONCLUSION:

This paper demonstrates an easy five step approach, where user can perform to calculate a similarity index of the two tables, conducting data transformation using the Fuzzy match algorithm in the open source environment (PDI) to standardize, integrate and join/combine data sets together.PDI, along with QGIS and LibreOffice can be very helpful in organizations/research institutes. Jaro-Winkler fuzzy match algorithm in simple steps using GUI (spoon) was used over two databases from a single district of Assam. Missing primary key and unavailability of proprietary software have resulted in working out the proposed method which is user friendly, allows for faster data processing and is available free of cost. The results are satisfactory and have immensely benefitted the project task to carry out join, relate and merge in such varied database.

REFERENCES:

1. W. Cohen, William & Ravikumar, Pradeep & E. Fienberg, Stephen. (2003). A Comparison of String Metrics for Matching Names and Records. Proc of the KDD Workshop on Data Cleaning and Object Consolidation.
2. Kirk Paul Lafler, Stephen Sloan (2017) Western Users of SAS Software- A Quick Look at Fuzzy Matching Programming Techniques Using SAS® Software. 24th SAS Conference Proceedings, Long Beach, California, sep. 20-22, 2017, PP-129.
3. Jede diah J. Teres. (2011) NorthEast SAS user group- Using SQL Joins to Perform Fuzzy Matches on Multiple Identifiers.25th SAS Conference Proceedings, Portland, Maine,sep.11-14,2011,PS-07
4. Aldo Benini(2008) Merging two datasets on approximate values Matching on groups as well as on the nearest value of a numeric variable, in MS Excel and in STATA.
5. Cohen, W. W., and Ravikumar, P. 2003. Secondstring: An opensource java toolkit of approximate string-matching techniques.
6. Cohen, W. W.; Ravikumar, P.; and Fienberg, S. E. 2003. A comparison of string distance metrics for name-matching tasks. In Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03).
7. Winkler, W. E. 1999. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04.
8. Abdellah Amine, Rachid Ait Daoud, Belaid Bouikhalene (2016) Efficiency Comparison and Evaluation between Two ETL Extraction Tools- Indonesian Journal of Electrical Engineering and Computer Science 3(1):174-181
9. Alexandra Maria Ioana Florea, Vlad Diaconita, Ramona Bologa(2016). Data integration approaches using ETL. Database Systems Journal vol. VI, no. 3/2015:19-27.
10. A. Jaro, Matthew. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of The American Statistical Association - J AMER STATIST ASSN. 84(406): 414-420. 10.1080/01621459.1989.10478785.
11. Casters M , Bouman R and Dongen J.V (2010) Pentaho kettle Solutions (1st edit). Wiley Publishing, Indianapolis, USA
12. Christen Peter (2006) A Comparison of Personal Name Matching: Techniques and Practical Issues. Technical Note: TR-CS-06-02, The Australian National University, Canberra, Australia

Web References:

- http://aldo-benini.org/Level2/HumanitData/Benini_NearMergesByGroup_120326
- http://secondstring.sourceforge.net
- http://www.census.gov/srd/www/byname.html