JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Bilingual Matching Algorithm for Data De-Duplication in E-Governance Large Scale Heterogeneous Datasets

[1]K.Balachander, [2]D.Paulraj

[1]Associate Professor, Dept of CSE, Velammal Institute of Technology, Chennai, India

[2]Professor, Dept of CSE, R.M.K College of Engineering and Technology, Chennai, India

Received: 14 Feb 2020 Revised and Accepted: 25 March 2020

Abstract

The rapid growth and advancements in the field of Information Technology, Cloud Computing and Internet of Things (IoT), paved the way for unprecedented increase in the data that they have produced. The huge data generated from different sources have distinct characteristics in terms of its size, forms and the speed at which the data is generated. This situation has led to the difficulty in processing these datasets by using the traditional tools and techniques. In this paper, a bilingual matching algorithm is proposed to identify and eliminate duplicate data from the large scale datasets collected across various public sector administrative departments such as Public Distribution System (PDS), Old Age Pension (OAP), Health, Farmers Welfare etc. The proposed algorithm is implemented to match entries of E-governance dataset available across various public sector departments, particularly focusing regional languages of India (especially Tamil to Tamil matching and Tamil to English matching). Tamil language to Tamil language matching is carried out to avoid the redundant records, which is unique and novel contribution in processing the large scale heterogeneous datasets. Experimental results are provided to ascertain the effectiveness of the proposed algorithm along with implementation of font conversion process and a customized PostgreSQL plugin. The experimental validation of the proposed matching algorithm proves the capability of the algorithm to identify the duplicate records for the unique set of data considered for the experimental study.

Key words: Bilingual Algorithm, Data De-Duplication, Big Data, Data Standardization

1 Introduction

Over the past decade, there is an exponential growth in the generation of data across various domains. Based on the report released by International Data Corporation (IDC) in 2011 [1], the amount of data created and utilized around the world is approximately 1.8 ZB and this will get doubled in every two years. The term “Big Data” has been coined to reflect its characteristics with respect to different dimensionalities such as volume, velocity, value, variety and veracity. Because of its inherent characteristics and generated at different sources, big data poses large amount of unstructured data. The composition of this unstructured data increases the complexity of data processing.

Thus, the Big Data era has emerged due to the rapid growth across all the domains such as health care, bioinformatics, vehicular networks, biological science etc. This situation has resulted in acceleration of big data research through various initiatives taken by industries and government agencies across the globe. Big data processing is most predominant in major domains such as healthcare, public sector administration, retail industry, manufacturing industry and personal data location. Over the years, the application of big data has been extended to fields such as atmospheric science, medicine, genomics and other interdisciplinary and complex research domains.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

The most important issue in big data processing across all the domains is that how to efficiently process, effectively organize and manage such huge datasets for performing real-time analytics. Data acquisition, storage, management and analysis are the key issues to be handled during the real-time processing. Processing huge datasets at real time poses various challenges using the traditional data analytics tools. The traditional data analysis and management systems is suitable only for the structured data and also these systems utilizes expensive hardware. Hence, parallel processing infrastructure is highly essential for processing huge datasets at real-time in an efficient way. Any processing methodology should definitely address all the challenges without compromising in any aspect [1]. The key challenges associated during big data analytics are:

a. Data representation – which focus on making the available data more meaning full for subsequent analysis and interpretation. If the data is not represented correctly, it may lead to reduction in the value of the original data and may also be a bottleneck during the data analysis.

b. Data compression and redundancy reduction – concentrates on reducing the cost involved during the management of data without compromising on the precision of the data, such that the value of the data is not affected.

c. Data life-cycle management – due to the availability of huge data, a strategy should be framed for deciding which of the data should be stored for future use and which data can be discarded.

d. Analytical methodology – Big data analytics relies on fundamental infrastructure for efficient processing. The use of non-relational database proven to be advantage for processing unstructured data, while the traditional RDBMS lacks expandability and scalability. Any analytic approach should focus on providing accurate results, with less time and cost.

e. Data confidentiality – Most of the data analytics solutions are provided by the experts, who are not the owners of the data or service providers. Hence, it is highly essential to take measures to protect most sensitive data for ensuring its safety.

f. Energy management – power consumption management and control at the system level should be implemented for consuming less energy during the data processing, transmission and storage.

g. Scalability – The solution methodology should be capable of processing the current datasets and also if the size of the dataset is increased in the future. In both cases, the solution should provide accurate analytics.

h. Co-operation – Experts from different disciplines should come forward and work together in order to provide a comprehensive solution to the data analytics objective, because this field requires solutions from interdisciplinary domains.

Data representation, analysis and redundancy reduction are the three main challenges considered in this paper.

The digitization and automation of various government departments/agencies have led to the data collection of every citizen of the country/state by individual departments. This provides an efficient way for the citizens to approach the concerned departments for applying to any schemes that the government has introduced. The primary objective for collecting such data is to maintain each citizen details online for effective reach out of the government plans to the intended beneficiaries. The data collected thus provides an easier and faster way for the government departments to satisfy the requirements of the citizens. The collected data primarily constitutes the demographic details of every citizen.

The paper is organized as follows, Section 2, summarizes the related work. Architecture for processing E-Governance data and Problem formulation are described in Section 3. The proposed Bilingual matching algorithm and the dataset considered is presented in Section 4. Section 5 provides the experimental results. Finally, conclusion is presented in Section 6.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

2 Related Work

Text Similarity

String based Text similarity approaches are broadly classified into character based and term-based [2]. In character based similarity measures considers similarity between two strings based on the distance (Damarau-Levenshtein algorithm) [3], length of contiguous characters (Longest common substring algorithm) number and order of common characters (Jaro) [4], using prefix length (Jaro-Winkler) [5].

Also based on dynamic programming technique which is utilized for biological sequence comparison for matching the best alignment (Needle-Wunsch) [6], dynamic programming for dissimilar sequences for character based matching (Smith-Waterman) [7]. Sub- sequences matching using N-gram similarity approach [8] is implemented in which the distance is measured by ratio of number of similar N-gram to maximal number of N-grams.

In term-based similarity measure Manhattan distance is used (block distance) [9] , cosine similarity measures [10] the similarity between two vectors and angle the strings Euclidean distance measures the sum of squared distance between the two vectors. Jaccard similarity [11] identifies the shared terms with the number of unique terms available in both the strings and simple vector based approach matches the coefficients along the dimensions of both the vectors, by simply counting the number of non-zero similar terms. Corpus-based similarity approach [12-14] is similar to dictionary based approach which compares the similarity index from the information gained from large dictionary.

Knowledge-based similarity approach is also semantic similarity measures based on information desired from semantic networks [15]. Knowledge based similarity approaches can be further divided into two approaches. Semantic similarity and measures of semantic relatedness [16]. Several Hybrid similarity methods [17] has also been implemented which provides better performance.

Data Duplication:

Ahmed.H.Yousef [18] proposed a framework which utilizes phonetic algorithms which support different indexing/blocking techniques for duplicate record detection. The framework uses various proximity matching algorithms suitable for matching names in several languages. Also the work evaluates the performance with standard metrics.

A comprehensive analysis of duplicate record detection is studied by Elmagarmid et al 2007[19], in which several duplicate detection algorithms with similarity metrics are discussed. Duplicate record detection mechanism can be evaluated with following metrics.

I. The first metric is based on measure of complexity by the ratio of number of generated record pairs with the reduction ratio.

II. The second metrics relates to the quality of the results which is measured by determining the positive predictive value (precision) and True positive rate (recall).

Duplicate record detection strategies can be classified to three types [18].

1. Deterministic which can be applied only if the datasets possess high quality accurate unique entity identifiers.
2. Probabilistic approach is considered as consistent, cost-effective and more reliable strategy for duplicate record detection.
3. Expectation maximization algorithm and approximate string comparison are the two methods which is categorized under modern approaches for duplicate record detection.

A redundant removal strategy for stock commodities is proposed [20] to delete the redundant data and efficiently processed the data during the map phase. A pairwise record matching algorithm [21] is proposed to detect only approximate duplicate data in a database.

Several machine learning algorithms [22-25] are also available for duplicate record detection, which requires training data, which is not available in many situations and should be prepared manually, which leads to be a time-consuming process.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Various tools are available for detecting duplicate records in English language. Some of the tools big-match, TAILOR [26] and FEBRL [27]. These tools utilize different techniques for identifying similar entities from data sources in the same language.

After studying various literature, to the best of our knowledge, there is no tools or techniques exists for detecting duplicate records in Tamil language. Hence, our proposed bilingual algorithm concentrates on duplicate record detection in terms of Tamil-Tamil language mapping and Tamil-English language mapping.

3 Architecture for Processing E-Governance Data

The Government has digitized all the records available in various departments with the vision to benefit the intended citizen's. To control the existing online data, it was decided to use the National Population Register (NPR) based demographic data in various department schemes like Public Distribution System (PDS), Old Age Pension, etc., in a data platform to authenticate resident information across NPR and Unique identity number. The aim of this initiative is to link NPR and Unique identity number and seed Unique identity number into the individual department's database.

The volume of data is collected from every citizen is huge in size, and mostly the data collected is of heterogeneous in nature because of data representation in different regional languages across states, in country like India. The data collected is entered into the database in Tamil and English languages. Some departments may use Tamil language and some other departments may use English language for the representation of the citizen's data. The collected data exhibits much redundancy across different public sector administrative departments such as Public Distribution System (PDS), Old Age Pension (OAP) Scheme, Health Scheme, Birth and Death Registration, Employment Exchange, Driving License, Student's Scholarship Scheme, etc. Same citizen data is available in one or more departments in different or same languages.

The Architecture for processing E-Governance Data for matching the records is shown in figure 1. National Population Register (NPR) dataset is used for matching the similar dataset available from various public sector departments

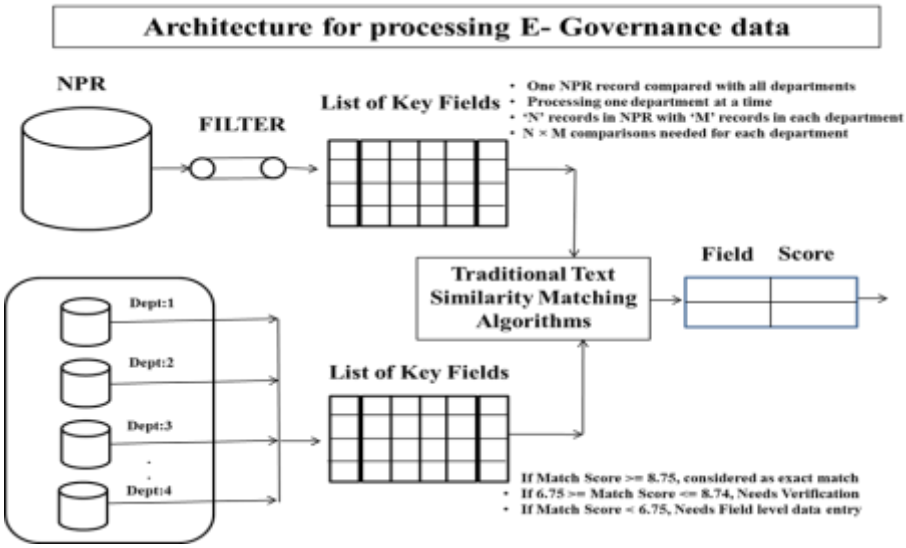


Figure 1: Architecture for Processing E-Governance Data

For matching the records, a list of key fields is extracted by a filter. Similarly, key fields are also extracted from the datasets available in individual departments One NPR record is compared with the records stored in all the departments. Processing of data can be one department at a time. The list of key fields extracted is matched using traditional text similarity matching algorithms. If Match Score >= 8.75, considered as exact match, If the match

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

score lies between 6.75 >= Match Score <= 8.74, it needs Verification and If Match Score < 6.75, Needs Field level data entry.

If there are 'N' records in NPR and 'M' records in each department, N*M comparisons are needed for each department. Moreover, collected data from each department is in different language. This poses a bigger challenge for finding a solution to the problem.

For example, considering the records between Public Distribution System department (PDS) and Health department, it was not possible to find matching records because of data representation as shown in Table 1. In this example, the dataset is in Tamil language in which significant fields pertaining to the resident looks same, whereas name itself varies slightly from each department. Standard facilities are available with analysis tools which support matching in English language. The challenge is to incorporate the matching of records from Tamil to English and as well as Tamil to Tamil languages, which is not available in the literature so far and thus the proposed algorithm proves to be novel and unique for solving the problem under consideration.

Table 1: Snapshot of Data Representation between PDS and Health Datasets

Sl. #	Health Dataset	PDS Dataset
1.	"அம்மாசி"	"அம்மாசி S"
2.	"மீனகம்மாள்"	"மொளகம்மாள் "
3.	"லட்சுமி "	"லெச்சுமி "
4.	"இராமர்"	"ராமர்"

In order to mitigate such issues, a Bilingual similarity matching algorithm is proposed, which is unique for the considered dataset, because of its application to match Tamil language to Tamil language and Tamil language to English language.

3.1 Problem Formulation

The government's initiative toward E-governance has led to the digitization of fundamental details about each department. For providing effective and efficient service to every citizen, each department collects data from individual citizen for maintaining the record online and to automate the processing of request initiated by the citizen. Since each department collected data individually from every citizen, the records exhibits redundancy in the collected data across all the departments. Also, the collected data is in different languages such as Tamil, English, etc. In order to streamline and leverage the available data, to benefit the intended beneficiaries, National Population Register (NPR) based demographic data across various departments have been initiated by the government.

The main objective of NPR based demographic data is to integrate the available data across various departments into one unique source of data. This integration causes the reduction in anomalies and eliminates the redundant data, which ensures the presence of precise authenticated resident data across all the departments.

The scope of the problem is to formulate a strategy to link all the available dataset from individual departments with a single source of unique identity used in various countries (Such as Social Security Number (SSN), UID, etc.). Since the departmental databases require the verification and authentication of each beneficiary, it

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

is challenging task as the primary data available for beneficiary identification is either contradictory, unverifiable or incomplete.

The following issues should be addressed for arriving at a solution to the considered problem:

1. Each department use different font types in entering the data.
2. Need to convert the available data into a uniform standard format.
3. Collected data should be cleaned for making it suitable for processing.
4. Apply de-duplication techniques on the dataset.

Processing and authenticating the citizen's information across different departments is difficult because of the presence of redundant data and different languages. The legacy data obtained from different departments where manually entered using different platforms, environments and languages. The real challenge is to first streamline and standardize the data collected.

The contribution of the work is three fold:

1. Font conversion strategy for converting the Non-Unicode representation of the database records into Unicode font.
2. A Bilingual matching algorithm is proposed and implemented to identify and remove the duplicate data from the dataset, which is unique and concentrates on Tamil language to English language matching and Tamil to Tamil matching.
3. A customized PostgreSQL plugin is written in order to implement the proposed matching algorithm for performing further analysis.

3.2 Methodology

A complete study of database schema of various public sector departments is performed. In examining the schema, it was found that there were lacunas in perceiving its structure at the data entry level. Although the schema has been well defined, it is evident that there was a poor standard followed during the data entry stage. For example, in filling the demographic details across named fields like door number, address 1, address 2, locality, street, district, city, state, etc., it is found that the details were merged in one field or wrongly entered into another field. This has led to the presence of NULL values in the mandatory fields. There are no specified constraints to avoid data entry errors and to identify beneficiary data uniquely.

The data collected by various departments were in a variety of formats. The first task is to convert the available dataset into standard uniform format. Several open source tools and techniques were used for converting the dataset into a standard uniform format. Then all the dataset represented in the proprietary fonts are converted to standard Unicode font representation. Unicode is an international standard for word processing where multiple languages are used. It is a two-byte encoding scheme. It represents each character as a two-byte number. Each two-byte number represents a unique character used. A number can be represented as one character and vice versa. Unicode encodes only vowels and consonant characters and a set of modifiers. The modifiers represent situations where the vowel and consonant pair appear in Tamil language. Unicode file stores text information at character level. There are many different regional encodings for different languages and operating systems. Converting the propitiatory codes into a standard code with a centralized point is known as Unicode.

After analyzing the datasets, it is found that various data elements required cleansing [28]. In order to improve the quality of data, data cleaning is needed. It deals with detecting and removing errors and discrepancies from data. In single data collection, the errors may occur due to misspellings during data entry, invalid data or missing data. When multiple datasets are to be integrated, the need for data cleansing is increases to a significant

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

level. Since, the data sources contain redundant data in different data representations. In order to get accurate and consistent data, linking of different data representations and removal of duplicate data becomes essential.

The cleansing process is carried out on all the considered datasets pertaining to Public Distribution System department, NPR, Old Age Pension department, Employment exchange department. In these datasets, each column is analyzed for valid non-null values and appearance of any special characters other than characters, in case of name field.

Data standardization is performed on the cleansed data, to find relations with other available datasets. Data de-duplication is performed in the next stage, which is one of the main contribution of this paper. After scrutinizing the NPR dataset pertaining to one district with 344265 records, it is found that it contains 389 redundant entries. These redundant entries were removed from the original table, because NPR dataset is used as the base record for authenticating the other department's dataset. The following sub section elaborates the example scenario which describes the necessity of employing a data de-duplication mechanism for the considered datasets.

3.3 Scenario – Considering PDS and NPR Datasets

The data de-duplication technique is explained by considering PDS datasets and NPR datasets. Both the datasets contain crucial records like personal and the demographic details of the citizen. The distinction is that the PDS datasets contain the details in Tamil language, whereas the NPR dataset contains records both in Tamil and English language.

A direct comparison is carried out on sample of both the PDS and NPR datasets, for matching the entries. Table 2 describes the total number of records compared in NPR and PDS datasets with number of matching records.

Table 2: Total Number of Records Compared in NPR and PDS Datasets

Total Records in NPR dataset: 344265 **Total Records in PDS dataset: 654025**		
S.No	**Criteria for Comparison (NPR & PDS)**	**Number of Matching Records**
1	Name and Application Local (Tamil)	198147
2	Name and Family Head Member Name	127657
3	Name, Family Head Member Name and Postal code	16590
4	Name, Family Head Name and Street	6643
5	Name, Family Head Name, Postal code and Street	2988

It is observed from the table that the count of matching records across the two datasets keeps on decreasing as more conditions are added. The following are the identified reasons, after careful analysis of the datasets:

1. Postal code is usually represented in 6 digits. But the postal code entered is in non-standard format, for example: 25614 (i.e. other than 6 digits).

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

2. Non-uniformity found in expressing names, both in Tamil and English languages. Spelling of names differ in both datasets.

Also, the datasets from other departments such as Old Age Pension department, Employment Exchange, Health, etc., are considered for validating the proposed algorithm. The sample dataset considered for the experimental study contains four schemas and thirty database tables. The database tables considered for rectifying the issues is presented in Table 3. Also, some of the linking tables, which specifies the demographic details such as place, street, and district are also considered during the development of matching strategy.

Table 3: Dataset Records Considered for Matching Problem

S. No	Name of the Database Table	Number of Row Count	Description
1	Card1	184022	Contains the Head Member of the family
2	Card members1	665642	Contains all the members of the family

Due to several issues stated regarding the characteristics of the collected data, the linkage of records across the database tables require a special strategy to be implemented for matching the records across all the departments. The following section presents the Bilingual matching algorithm and its working with certain examples.

4 Bilingual Matching Algorithm

The objective of the proposed algorithm is to match the records available across different departments and to remove the duplicate entries in the dataset. The algorithm must also support matching of Tamil language to Tamil language and Tamil language to English language records as well. Initially a character map is created for different languages, which will be used as the basic character set during the matching process. Table 4 presents examples of output of the algorithm after the comparison in various languages. The detailed algorithm is as follows:

Step1: For each letter in the word except first letter, get the corresponding digit from the character map.

Step2: If the letter is not found in character map, then assign the digit for that letter as 0.

Step3: Skip the duplicate consecutive codes.

//For Example in Tamil கே.கே will be considered as கே.

Step4: Replace first digit with first alphabet character.

Step5: Remove all 0s from the code.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Step6: Return code padded to the required length

//For Example: if the required length of code is 5. and is ൬BCD, then returned code will be ൬BCD0.

Table 4: Sample Input and Output of the algorithm after the comparison in various languages

S.No	Input	Language of the Input	Output
1	சந்தோஷ்	Tamil	சLKES000
2	सन्तौष	Hindi	सLKES000
3	കാർത്തിക്	Malayalam	കAPKBF00

4.1. Comparison Logic

If the comparison is to done between the names सन्तौष (Hindi) and சந்தோஷ் (Tamil) and need positive result, a comparison logic should be implemented for making the comparison language independent.

Step1: Compare the two strings, if they are same, then return 0

Step2: Calculate the codes for both strings, if the match is found, then return 1. i.e., both strings are from the same language and similar.

Step3: If strings have different first letter and rest of the part is same, check whether the first letters are with same digit. If so, both words from different languages, but similar.

Return code for this case will be 2.

Step4: If none of the above conditions match, return -1, indicating both strings are completely different.

Matching Strategy

As the considered datasets are associated with various departments, in order to establish resident linkage among the data, it is essential to devise a strategy to ascertain a particular link. Table 5 provides the details of the mechanism adopted for finalizing the matching strategy.

Table 5: Matching Strategy

S.No	Field Name	Criteria
1	Postal Code	Exact Match
2	Name and Father Name	Partial Match (Bilingual Match – 100%)
3	Village Name	Partial Match (Bilingual Match – 80% and above)
4	Door Number	Partial Match – 60% and above

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

The main aim is to sort out the resident information such as name, father name and with address across departments. So, the field name mentioned in the table 5 are considered in the order they are mentioned along with criteria to find the match. This matching strategy is applied on the datasets along with bilingual matching algorithm.

The next sub sections present the implementation of PostgreSQL plugin for the experimental study and the mathematical formulation of the weightage during the mapping process. Even though open source database offers wide range of functionalities, the default rich set of feature may not suitable for all the use cases. Hence, the provision to extend its functionality is utilized for the specific use cases considered for the proposed study.

4.2 Implementation of PostgreSQL Plugin

The PostgreSQL plugin is used to connect and execute SQL statements on a PostgreSQL database. After the execution of the statements, it reads back the results. Depending on the configuration, the returned values are then converted into collected list of values. The inbuilt functions available in PostgreSQL is not sufficient for processing the queries for the considered problem. This situation resulted in the implementation of a customized package in the existing PostgreSQL. The extension facilitates in collecting all the objects into a single package and to simplify the database management.

The following two files are needed to configure extension skeleton in PostgreSQL:

- A control file in the format "extension_name.control", which tells PostgreSQL some basics about newly created extension
- An extension's SQL script file in the format extension--version.sql

Once the two files are created, then these files should be added into the project directory.

The skeletons of the two files and the partial code written in Python for processing are as follows:

1) File Name : tamilfuzzy.control

```
#tamilfuzzy extension
comment = 'tamilfuzzy datatype'
default_version = '0.0.1'
relocatable = true
```

2) tamilfuzzy-0.0.1.sql

```
#above sql code.
```

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Partial Python code

```
BEGIN
  WHILE LENGTH(input) < 4 LOOP
    CHAR = UPPER(substr(INPUT, pos, 1));
    pos = pos + 1;
    CASE CHAR
    WHEN '' THEN
      -- End of input string
      IF input = '' THEN
        RETURN '';
      ELSE
        RETURN rpad(input, 4, '0');
      END IF;
    WHEN 'ஆ', 'ஐ', 'க' THEN
      symbol = '1';
    WHEN 'இ', 'உ', 'ச' THEN
      symbol = '2';
    WHEN 'அ', 'ஈ','ஊ','ஏ','க','ங' THEN
      symbol = '0';
    WHEN 'ள' THEN
      symbol = '5';
    WHEN  'ஒன' THEN
      symbol = '3';
    ELSE
      -- Not a consonant; no output, but next similar consonant will be re-recorded
       symbol = '';
```

4.3 Mathematical formulation of the algorithm

The weightage calculation is used in the matching process. Higher percentage of weightage is given when the postal code and names are compared within the two dataset. If the postal code is different the conclusion is that two records are not similar. Similarly, the weightage is assigned for other attributes depending upon their importance, such that the matching strategy does not provide wrong results during the matching. The example of weightage calculation is as follows:

Example:

```
select uid, keycardnumber,membername, res_name_local, placename, res_addr_vtc_name,
pincode, res_addr_pincode, replace(res_addr_building,'¬','/'), replace(doorno,'¬','/')
from npr_final  , card_t
where res_addr_pincode = pincode and
tamilfuzzy(trim(res_name_local) , trim(membername) ) > 0.7
and res_addr_vtc_name = placename
and tamilfuzzy(replace(substring(res_addr_building, 0,4), '¬','/') , replace(doorno,'¬','/')) > 0.5
and tamilfuzzy(streetnameinlocal, res_addr_street_local) > 0.5
and substring(res_addr_building, 0,4) % replace(doorno,'¬','/')
```

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

and tamilfuzzy(res_addr_building, doorno) > 0.5

The weightage calculation takes the fields like uid, keycard no, name, place, pin code, door no from different datasets pertaining to departments (NPR and PDS) for comparison. Here, uid matches for all the records but with differing keycard no and name. To resolve this situation additional parameters like place, pin code and door no is considered with weightage.

Table 6 provides the details of matching names in Tamil across PDS and NPR datasets with 95% matching. If there is a space in between, if there is a difference in one letter, then the match between the PDS and NPR datasets will be 95%.

Table 6: Name with more than 95% matching

S.No	Name in PDS dataset	Name in NPR dataset
1	ராமசாமி	ராம சாமி
2	இராமசாமி	ராமசாமி
3	முணியசாமி	முணுசாமி
4	முணியசாமி	முனியசாமி

Table 7 provides the details of matching names in Tamil across PDS and NPR datasets with 85% to 95% matching. If the difference between two datasets is an initial and a space or initials at the end or the difference is 2 or more characters, then the matching lies between 85% to 95% and so on.

Table 7: Name with more than 85% to 95% matching

S.No	Name in PDS dataset	Name in NPR dataset
1	ராமசாமி பூ	பூ இராமசாமி
2	முணியசாமி	எ முணு சாமி
3	மெரி மரியதாஸ்	*ம மெரி மரியதாஸ்*
4	சுப்பிரமணி	சுப்பிரமணியன்

Table 8 provides the details of matching Tamil names in PDS dataset with English names in NPR datasets with 85% to 95% matching. If the difference between two datasets is an initial and a space or initials at the end or the difference is 2 or more characters, then the matching lies between 85% to 95% and so on.

Table 8: Name with more than 95% matching

S.No	Name in PDS dataset	Name in NPR dataset
1	ராமசாமி	Rama samy
2	இராமசாமி	Ramasamy

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

3	முணியசாமி	Munusamy
4	முணியசாமி	Muniasamy

5 Experimental Results and Discussions

In the proposed approach, records in each department is processed for matching, using the bilingual matching algorithm. In order to streamline the considered dataset, which is composed of fields with Non-Unicode font representation, initially a font conversion strategy is implemented. The font conversion strategy converts the Non-Unicode font representation into Unicode representation. This Unicode representation of the records ease the subsequent processing stages. After the font conversion process, the entire dataset is cleaned before it is utilized for further processing. The next subsections provide the details of the font conversion process and validation of bilingual matching algorithm.

5.1 Font Conversion Process

Data pertaining to the PDS dataset is encoded with proprietary fonts (Ramya, Bamini). The conversion mechanism is applied to convert from Ramya, Bamini to Unicode font.

The accuracy of the data cleansing depends majorly on the data format consistency. The data can be compared and cleansed only if all the data which are taken as inputs are of same format, say if it is in English locale (en_US), all the data should be in English (en_US). In this case, there are several tables where the fields are combination of Unicode and Non-Unicode encodings. Comparison of such fields will provide undesired results and the final result will not be accurate. The font conversion module takes care of converting Non-Unicode fonts to Unicode fonts.

Non-Unicode to Unicode Conversion:

The data in State Database contains fields of Non-Unicode encodings using Tamil Ramya font in few fields and Unicode encoded with English fonts in few fields. The Non-Unicode Ramya Fonts in Tamil local need to be converted to Tamil Unicode font. The conversion process involves

- Locating those fields which are encoded with Non-Unicode standards
- Extraction of all data from those fields
- Identification of the Non-Unicode format used in the field
- Creating a conversion table for the given Non-Unicode format to Unicode
- Creating temporary database for conversion process
- Applying Conversion algorithm on the extracted data
- Merging converted data back to original database
- Committing the conversion process

The above process is followed to convert the Non-Unicode encoded fields into Unicode encodings and the resultant records is used for further processing stages. Table 8 describes one sample of database pertaining to a district (or province) with Non-Unicode encoded fields, representing record count, font status and number of fields with Tamil ‘Ramya’ font.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Table 8: Font Conversion Status of One District (or Province)

S.No	Name	Record Count	Font Status	Number of Fields with 'Ramya' Font
1	Boguscard	0		0
2	Boguscardmembers	0		0
3	Card	170805	Ramya	7 – applicantnamelocal , fmhname , addressline1 , addressline2, addressline3, richiname, tsosename
4	Cardmembers	614077	Ramya	3 – fmhname , relationm , spousename
5	Currentauthentication	46		0
6	District	31	Ramya	1 - district_name
7	Familycardmaster	161596	Ramya	2 – membername , address
8	Familycardmaster_temp	162231	Ramya	2 – membername , address
9	Familymembermaster	555160	Ramya	1 - membername
10	Familymembermaster_t	555160	Ramya	1 - membername
11	Historyauthentication	42514		0
12	Renewelcard10072102	144473	Ramya	1 - familyhea
13	renewalcarddetail14092012	156062	Ramya	1 - familyhea
14	renewalcarddetail21072012	143153	Ramya	1 - familyhea
15	renewalcarddetail24082012	156202	Ramya	1 - familyhea
16	renewalcarddetails13122012	149814	Ramya	1 - familyhea
17	renewalcarddetails13122012_1	147064	Ramya	1 - familyhea
18	renewalcarddetails13122012_2	2750	Ramya	1 - familyhea
19	renewalcarddetails13_temp	158983		0
20	renewalcarddetails20122012	152532	Ramya	1 - familyhea
21	renewalcarddetails20122012_1	149693	Ramya	1 - familyhea
22	renewalcarddetails20122012_2	2761	Ramya	1 - familyhea
23	renewalcarddetails21122012	156690	Ramya	1 - familyhea
24	renewalcarddetails21122012_1	151719	Ramya	1 - familyhea
25	renewalcarddetails21122012_2	4847	Ramya	1 - familyhea
26	renewalcarddetails23122012	156685	Ramya	1 - familyhea
27	renewalcarddetails23122012_1	152695	Ramya	1 - familyhea
28	shoplocality	34534		0
29	villagemaster1	17388	Bamini	1 - village_name

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

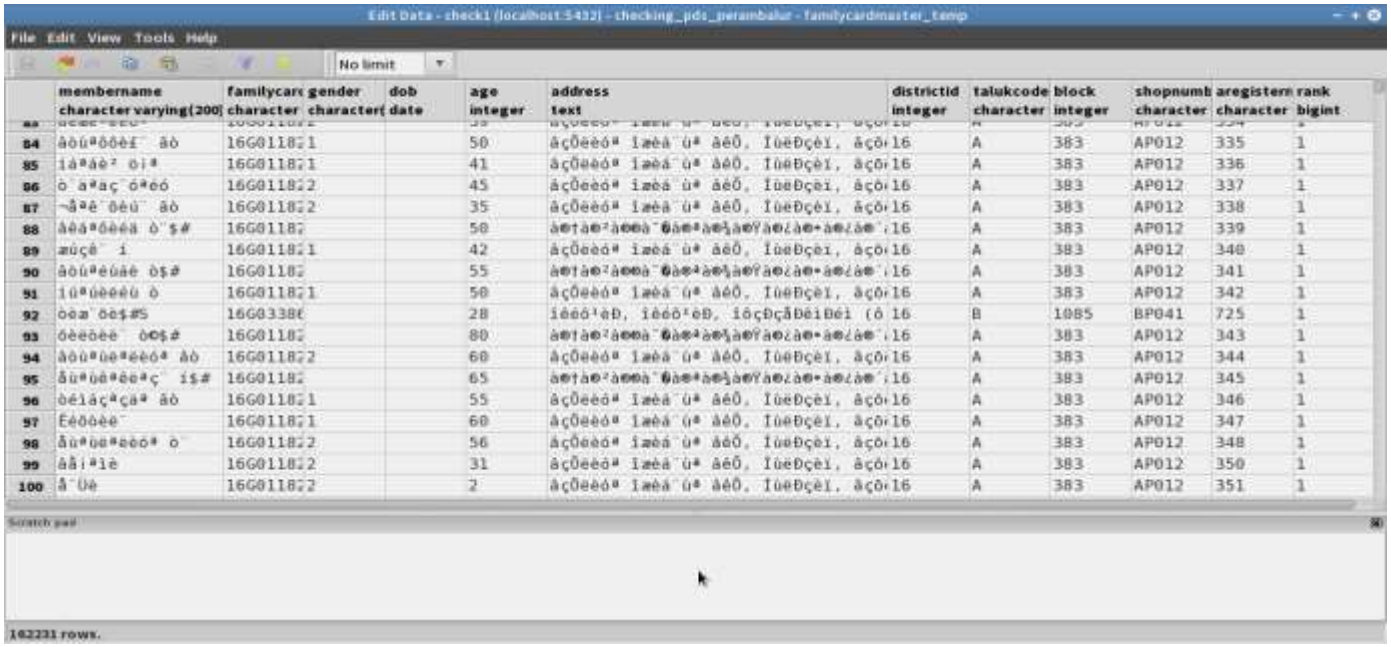
Table 9 presents the details of the processor configuration, number of records in Non-Unicode fonts, number of fields, and time taken for the font conversion process.

Table 9 Details of the Font Conversion Process

Details of the Font Conversion Process	
Testing Machine Configuration	Intel I7 processor with 4 GB RAM
Database Tested for Conversion	All Databases
Number of Rows originally in database with 'Ramya' font	4088714
Number of Rows originally in database with 'Bamini' font	17388
Number of fields with Ramya font	22 tables with 32 fields
Number of fields with Bamini font	1 table with 1 field
Number of Rows converted to Unicode	**4106102**
Time taken for conversion for 1Lakh record	**35 mins (approx)**

Figure 2 represents the screen shot of the original database before conversion and Figure 3 represents the screen shot of the converted database from 'Ramya' font to Unicode font.

Figure 2: Sample Screen-shot of the Original Database before conversion



ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

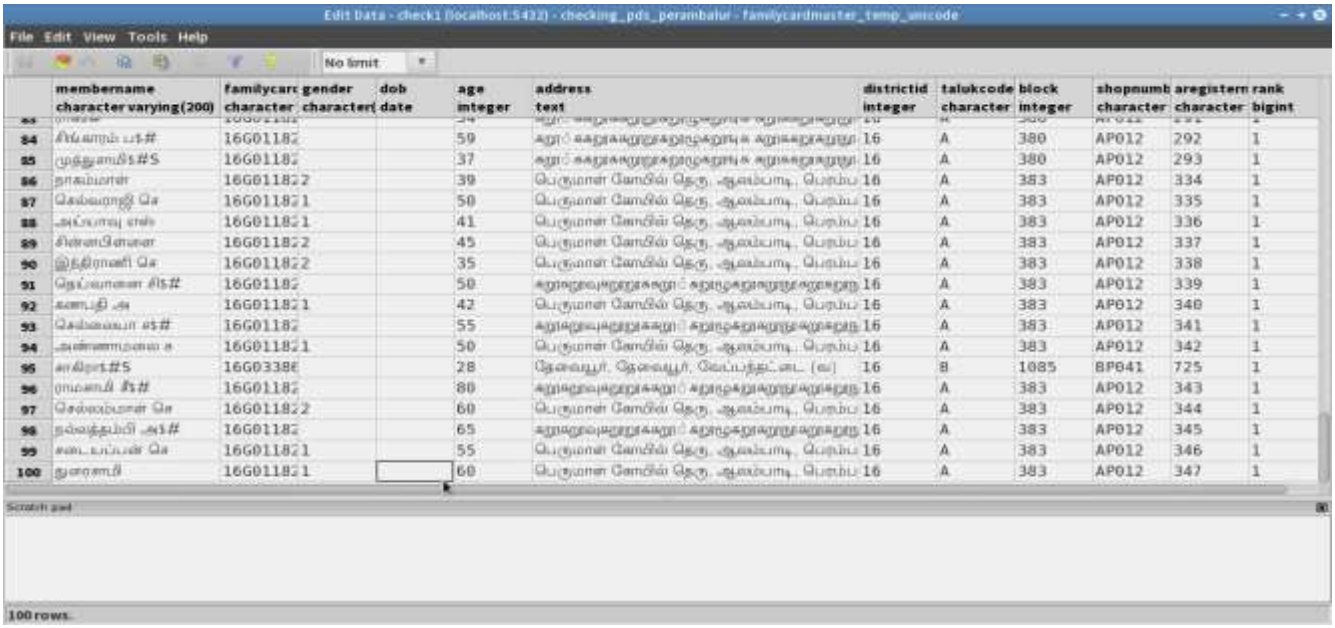


Figure 3: Screen-shot of Database after Converting 'Ramya' font to Unicode Font

5.2 Validation of Bilingual Matching Algorithm

The validation of the proposed Bilingual matching algorithm is done by considering a unique dataset pertaining to the Tamil Nadu State government records [29,30]. The government has introduced various schemes for the welfare of the citizens. Each scheme is within the control of various department. Each department has initiated the data collection process of every citizen separately and thus there exist duplicate records across the department database as mentioned in the Section 3. The details of the dataset dimensionality considered for the experimental study, i.e. number of records in each department and number of columns is reported in Table 10.

Table 10: Details of the Dataset Dimensionality

Department	Number of Records (Rows)	Number of Fields (Columns)
PDS	654025	22
NPR	345043	32
Health	125150	15
Employment Department	89312	25
Old Age Pension (OAP)	60384	54
Festival Benefit Department	143768	33
Farmer Welfare	258900	38
AD&TW and BC&MBC Scholarship	105200	35
Birth and Death	231000	29

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

The analysis of these dataset revealed that all the dataset has three common data types such as Integer, String and Date. The presence of other data types such as Boolean, Decimal and BLOB is identified in some part of the dataset.

Based on the data summary analysis of the Old Age Pension dataset, Festival benefit dataset, Health dataset, Employment dataset and Farmer Welfare dataset has keycard number (PDS unique ID) for references. Hence, these departments are identified as PDS referenced departments. This indicates that mapping of PDS and NPR will ease the number of iteration during the mapping process, by matching the PDS records with the name available in the PDS referenced departments. Figure 4 shows the relationship of the attributes of the dataset across various departments.

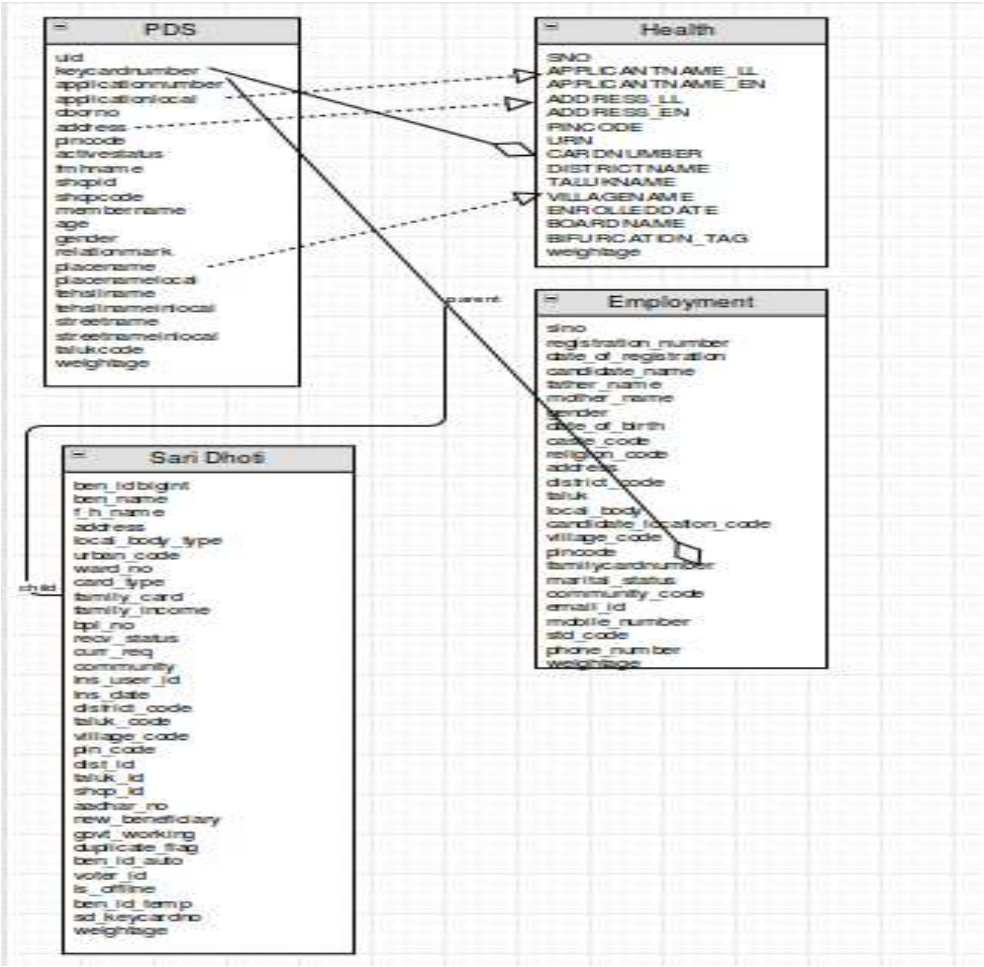


Figure 4: Relationship of Dataset across Various Departments

The real challenge in the implementation of the algorithm is to resolve the ambiguities present in the dataset. The first ambiguity is that, some dataset has captured names only in local language (For example, Tamil, applicationlocal in PDS), some other dataset does not capture the name in local language (For example, candidate_name in Employment dataset). These ambiguities are resolved by the proposed algorithm during the matching process. The second ambiguity is the data entry error in which Father Name/Husband Name/Mother Name is specified in single attributes. This ambiguity is resolved through the application of data standardization procedure. Table 11 provides the details of the captured dataset across all the departments.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Table 11: Details of the Captured Dataset Across all the Departments

Department	Name captured in English	Name captured in Tamil	Name captured in Tamil - Partially	Number of Records in Tamil
PDS	No	Yes	No	654025
NPR	Yes	Yes	Yes	187254
Health	Yes	Yes	Yes	76749
Employment	Yes	No	No	0
Old Age Pension	Yes	No	No	0
Festival Benefit Department	Yes	No	No	0
Farmer Welfare	No	Yes	Yes	87952
AD &TW and BC & MBC Scholarship	Yes	Yes	No	105200
Birth and Death	Yes	Yes	No	231000

The objective of processing data de-duplication in E-Governance data is not for reducing the space. It has higher objective to eliminate the redundant beneficiary and make government scheme to reach every right citizen. After careful descriptive analysis on given datasets, the effectiveness of the proposed algorithm in identifying the duplicate records is proved through the following three different use cases:

Use Case 1: Finding and Removing Exact Duplicates

- The easiest and low latency execution is finding exact duplication.
- The Challenge in this case is that, given PDS dataset there are repeated entries based on the combination of "name" and "relation name" in other datasets, For Example;

1. "Ramasamy";"Perumal" - repeated 1220 times
2. "Periyasamy";"Ramasamy" - repeated 1215
3. ……
4. "Adhimoolam";"Chinnusamy" - repeated 508 times

Figure 5 shows the result of finding exact duplicates by matching only names, names with relation

and finally considering names, relation and pin code. The matching process is carried out for the three departments namely, Old Age Pension, Festival Benefit and Farmer Welfare. As the number of important attributes is included for the identification of duplicate records, the algorithm works correctly in identifying the duplicate entries. When only one attribute "name" is considered for the matching, number of matching records reported is high, which is obvious that same name could be there for different citizen. When two attributes "name" and "relation" is considered for the matching, the number of matching records reduces. Finally, when three attributes

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

"name", "relation" and "pincode" is considered for the matching, the number of matching records reduces further, which indicates the better performance of the algorithm.

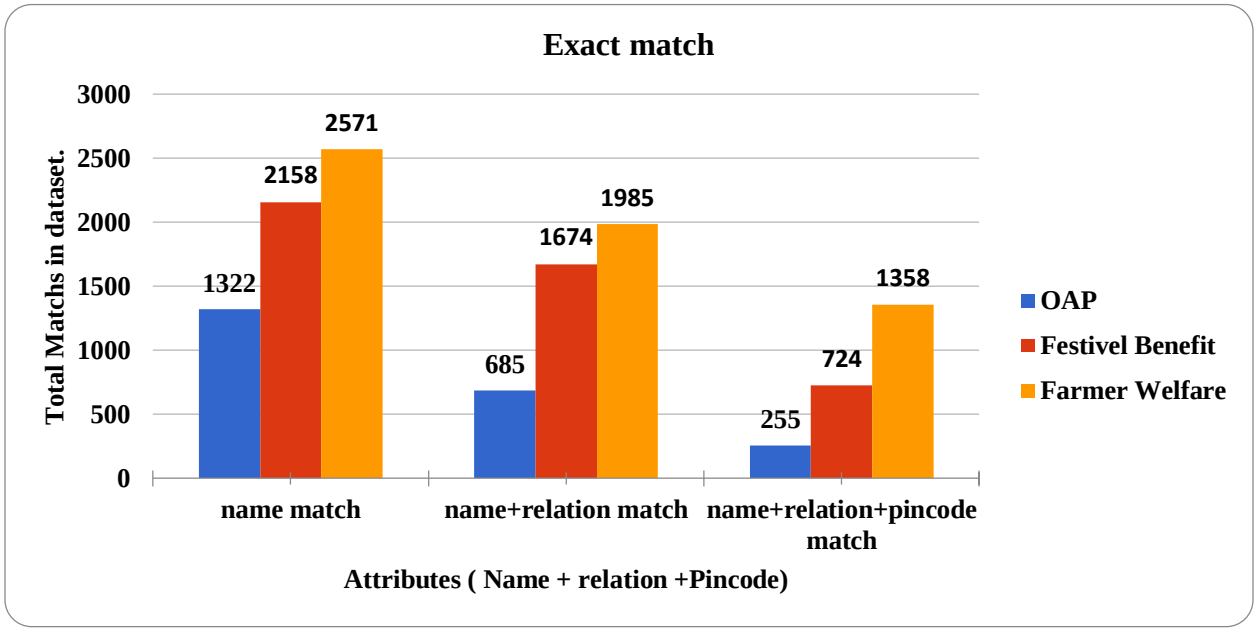


Figure 5: Result of Exact Match

Use Case 2: Finding and Removing Partial Duplicates

In this use case, the effectiveness of the proposed algorithm is validated by considering partial match in the fields across three different department datasets. Both single attribute matching and multiple attribute matching is taken into consideration. Figure 6 presents the result of single attribute comparison and Figure 7 presents the result of multiple attribute comparison. Both the results show a comparative percentage of increase in the number of matched records, compared to that of exact matching case, because of partial character mapping is considered in this experimental study.

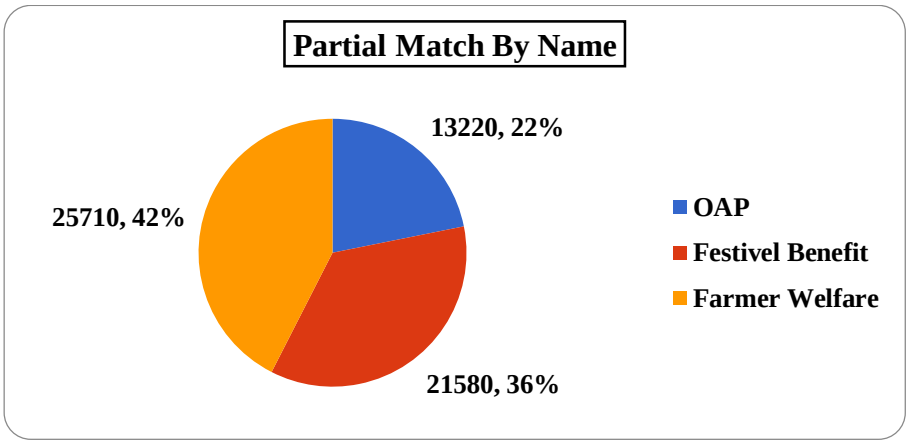


Figure 6: Result of Single Attribute Comparison

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

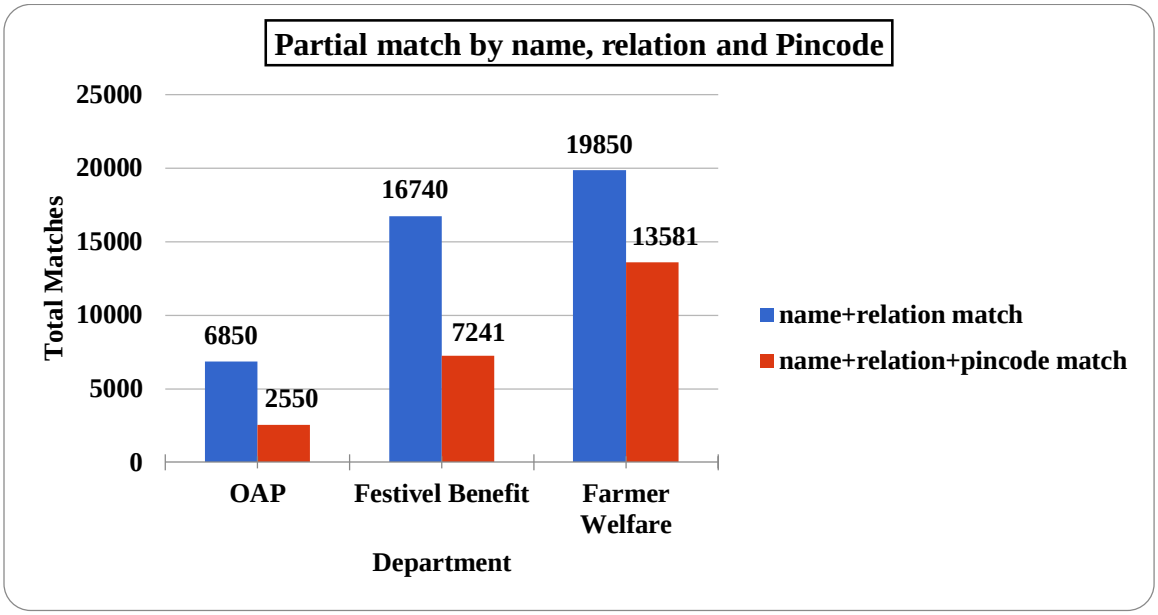


Figure 7: Result of Multiple Attribute Comparison

Use Case 3: Rule Based Matching

The proposed algorithm utilizes rule based matching strategy to find the matching records more precisely. This use case is presented to prove the effectiveness of the proposed algorithm in two scenarios. The rule based approach uses nested constraints for finding the match between the datasets and thus proves to be an accurate mapping solution to the considered problem.

Scenario 1:

```
IF Name + relation name MATCHES THEN
        IF date_of_birth MATCHES THEN
                IF member_keyCard MATCHES THEN
                        DO compare ALL members in PDS
                        RETURN match status //Greater than or equal to 95%
                RETURN match status //Greater than or equal to 70% and lesser than 95%
        RETURN match status // Greater than or equal to 45% and lesser than 70%
```

In this scenario 1, three levels of mapping strategy are carried out in which, if all the conditions are satisfied for a corresponding record, the matching percentage reported will be 95%. If only two conditions are satisfied, the matching percentage reported will be 70% and if only one condition is satisfied the matching percentage will be 45%.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Scenario 2:

```
IF Name + relation name MATCHES THEN
        IF date_of_birth MATCHES THEN
                IF member_keyCard MATCHES THEN
                        DO compare ALL members in PDS
                        RETURN match status //Greater than or equal to 95%
                ELSE IF caste_code MATCHES THEN
                        IF community_code MATCHES THEN
                                RETURN match status //Greater than or equal to 95%
                IF voter_id OR epic_no MATCHES THEN
                        RETURN match status //Greater than or equal to 99%
                 RETURN match status
                //Greater than or equal to 70% and lesser than 99%
                RETURN match status
                // Greater than or equal to 45% and lesser than 70%
```

In this scenario 2, more number of additional constraints are specified in order to further increase the accuracy of the matching process. Table 12 presents the outcome of implementing the rule based matching within the proposed algorithm, for both the scenarios. Applying scenario 1 rules for OAP, Festival Benefit and Farmer Welfare datasets, results in finding 587,1259 and 1960 duplicate records respectively, whereas for the same three datasets, when applying scenario 2 rules the number of duplicate records is decreased to 280, 701 and 1074 respectively.

Table 12: Result of Rule Based Matching for the Two Scenarios Considered

Department	Scenario 1	Scenario 2	Total Number of Records Compared
OAP	587	280	60384
Festival Benefit	1259	701	143768
Farmer Welfare	1960	1074	258900

The following subsections provides the performance analysis of the proposed algorithm quantitatively.

5.2.1 Performance Analysis

To further ascertain the effectiveness of the proposed algorithm, two different quantitative validations are provided. The first validation with respect to the time required to complete the processing of different queries in the traditional RDBMS setup is observed. The processing time is observed for each query with respect to Health department dataset. Table 13 provides the details of the processing time with indexing and without indexing.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

Figure 8 shows the comparison of the processing time taken with indexing and without indexing, when querying for single name, name along with parent name and finally name, parent name and village name respectively.

Table 13: Processing Time in RDBMS Setup with Indexing and without Indexing

Information	Processing Time in RDBMS (in seconds)	
Query	No Index	With Index
Comparing one name from Health with all names in PDS	9.58	9.26
Comparing all name from Health with all names in PDS - group based on demographic	610.59	472.18
Comparing one {name, parent name} from Health with all names and parent names in PDS	14.72	14.72
Comparing all {name, parent name} from Health with all names and parent names in PDS - group based on demographic	1187.15	847.92
Comparing one {name, parent name and village name} from Health with all {name, parent name and village name} in PDS	20.52	20.40
Comparing all {name, parent name and village name} from Health with all {name, parent name and village name} in PDS - group based on demographic	1526.14	1178.65

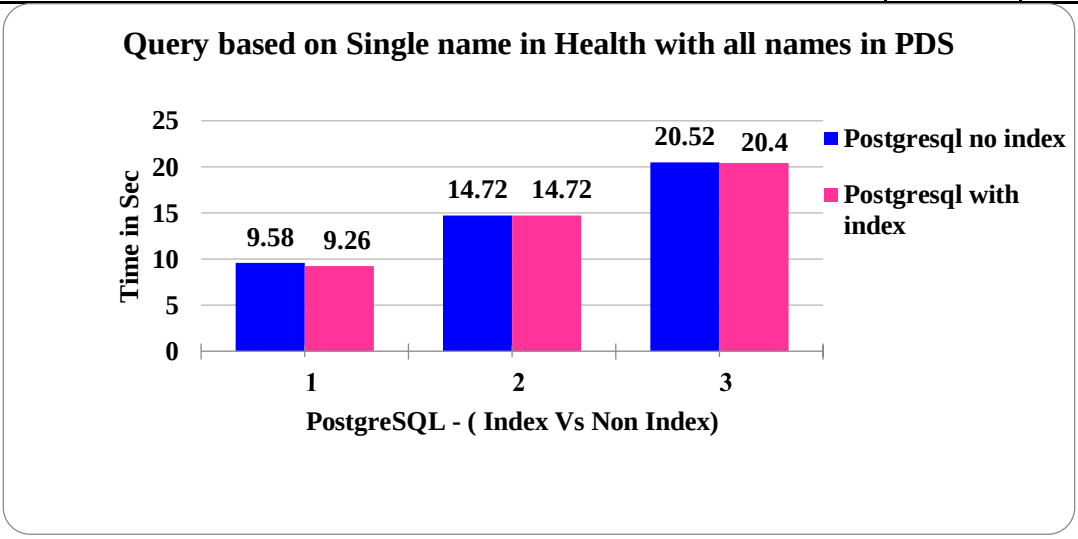


Figure 8: Comparison of Processing Time with Indexing and without Indexing

The second validation of the proposed algorithm is provided with respect to the four different performance measures, namely Precision, Recall, F1 Measure and Sensitivity. These parameters are used to

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

highlight the capability of the algorithm in identifying the duplicates records correctly. A sample of 1200 records that matches in NPR, PDS, Health, OAP, Employment, Farmer Welfare and Scholarship is taken as test dataset for this study. A total of 562 duplicate records is added manually along with the 1200 records in each department (the 562 duplicate records inserted in each department are not similar). After inserting 562 duplicate records in each department dataset, now the total records available for the validation study is 1762. The main objective of this validation is to prove how much percentage of duplicate records is identified correctly by the proposed algorithm.

The sensitivity is the ratio of the total number of duplicate records identified with total number of available duplicate records within the dataset. The mathematical formulation for Precision, Recall and F1 Measure is given in the following equations 1 to 3. Figure 9 shows the results of performance measure values for six different departments. The values obtained are consistently more than 90% for all the performance measures which indicates that the algorithm potentially performs better in identifying the duplicate records correctly.

$$Precision = \frac{|\{Relavant\ items\} \cap \{Retrieved\ items\}|}{|\{Retrieved\ items\}|} \quad \text{---------- (1)}$$

$$Recall = \frac{|\{Relavant\ items\} \cap \{Retrieved\ items\}|}{|\{Relavant\ items\}|} \quad \text{---------- (2)}$$

$$F = \frac{2(P * R)}{(P + R)} \quad \text{- ---------- (3)}$$

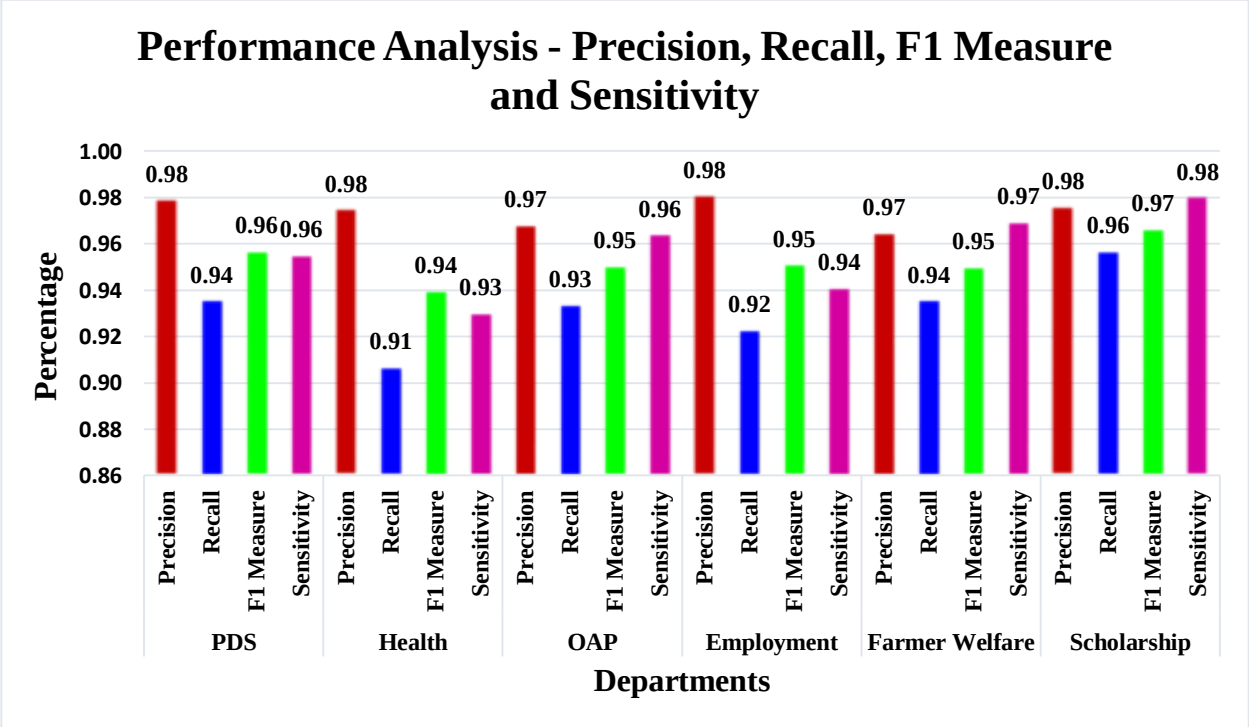


Figure 9: Performance Analysis – Precision, Recall, F1 Measure and Sensitivity

6 Conclusion

A Character based bilingual matching algorithm was proposed in this paper to detect the duplicate records in the large scale e-governance datasets across various public sector departments. The Dataset considered is unique in which certain ambiguities were present such as non-Unicode font representation, presence of null values in various fields. Initially, a font conversion methodology is implemented to convert all the dataset into standard Unicode format. Further, data standardization procedure is exploited to remove the ambiguities related to the

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

presence of null values. The bilingual matching algorithm implemented is able to detect the presence of duplicate records present across various department datasets. Experimental study provided with several use cases exhibits the effectiveness of the proposed algorithm in detecting the duplicate records. The quantitative validation of the proposed algorithm with respect to standard performance metrics such as precision, recall, F1 measure and sensitivity show the efficiency and capability of the algorithm in detecting the duplicate records, considering large scale dataset.

7 References

1. Min Chen, Shiwen Mao, Yunhao Liu, Big Data: A Survey, The Journal of Mobile Network and Application, 19 (2014), pp171-209, Springer, DOI 10.1007/s11036-013-0489-0
2. Wael H. Gomaa and Aly A. Fahmy, A Survey of Text Similarity Approaches, International Journal of Computer Applications, (0975 – 8887), 68(13) (2013), pp: 10-15.
3. Hall, P. A. V. & Dowling, G. R, Approximate string matching, Comput. Surveys, , 12 (1980), pp: 381-402
4. Jaro, M. A, Advances in record linkage methodology as applied to the 1985 census of Tampa Florida, Journal of the American Statistical Society, 84 (1989), 406, pp 414-420.
5. Jaro, M. A, Probabilistic linkage of large public health data file, Statistics in Medicine, 14 (1995) (5-7), pp: 491-498.
6. Needleman, B. S. & Wunsch, D. C., A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology, ,48(3) (1970): pp:443–53.
7. Smith, F. T. & Waterman, S. M. Identification of Common Molecular Subsequence, Journal of Molecular Biology, (1981) 147: 195–197.
8. Alberto, B. , Paolo, R., Eneko A. & Gorka L.,Plagiarism Detection across Distant Language Pairs, In Proceedings of the 23rd International Conference on Computational Linguistics, (2010). pp 37–45.
9. Eugene F. K.,Taxicab Geometry , Dover. (1987), ISBN 0-486-25202-7.
10. Jaccard, P., Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles, 37(1901), pp: 547-579.
11. Landauer, T.K. & Dumais, S.T., A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge", Psychological Review, (1997). pp:104.
12. Matveeva, I., Levow, G., Farahat, A. & Royer, C., Generalized latent semantic analysis for term representation. In Proceedings of RANLP, (2005)
13. Gabrilovich E. & Markovitch, S,Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, Proceedings of the 20th International Joint Conference on Artificial Intelligence, . (2007). pages 6–12.
14. Juan-manuel, Torres-Moreno, Gerardo Sierra and Peter Peinl, A German Corpus for Text Similarity Detection Tasks, International Journal of Computational Linguistics and applications, 2(2) (2014) pp:9-24
15. Sowmya.V, Kranthi Kiran. K and Tilak Putta, Semantic textual similarity using machine learning algorithms, International Journal of Current Engineering and Scientific Research, 4(8) (2017), pp: 10-15.
16. Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D. & Miller, K., WorldNet: An online lexical database. Int. J. Lexicograph, 3 (4) (1990), pp. 235–244.
17. Arifah Che Alhadi, Aziz Deraman, Masita, Masila, Abdul Jaleel, Wan Nural Jawahir Wan Yussof and Akashah Amin Mohamed, An ensemble similarity model for short text retrieval, Springer lecture notes on Computer Science 10404, (2017), pp:20-29.
18. Ahmed Hassan Yousef, Cross language duplicate record detection in Big Data, Big Data in Complex Systems (2015)., pp:147-171.
19. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering, 19(1) (2007), 1–16.
20. Alvaro E,Monge, Matching Algorithms within a Duplicate Detection System, IEEE Computer Society,(2000).
21. Seungwoo Jeon, Bonghee Hong, Joonho Kwon, Yoon-sik Kwak and seok-il Song, Redundant data removal Technique for efficient Big Data Search Processing, International Journal of Software Engineering and its applications, 7(4) (2013), pp:427- 435.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

22. Christen, P.: Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2008a).
23. Christen, P.: Automatic Training Example Selection for Scalable Unsupervised Record Linkage. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), 5012 (2008b), pp. 511–518. Springer, Heidelberg.
24. Shanmughavadivu.P and Baskar.N, An improving Genetic Programming Approach Based Deduplication using KFINDMR, International Journal of Computer Trends and Technology, 3(5) (2012), pp: 694 -701.
25. Divine Muhivuwonmunda, Data De-Duplication through Active Learning, MCS thesis, University of Ottawa (2010).
26. Elfeky, M.G., Verykios, V.S., Elmagarmid, A.K.: TAILOR: a record linkage toolbox. In: Proceedings of the 18th International Conference on Data Engineering, (2002), pp. 17–28.
27. Christen, P.: Febrl: a freely available record linkage system with a graphical user interface. In: Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management, vol. 80. Australian Computer Society, Inc., Wollongong (2008c)
28. Erhard Rahm, Hong Hai Do, Data Cleaning: Problems and Current Approaches, IEEE Computer Society (2000).
29. https://tn.data.gov.in/
30. https://data.gov.in/resources-from-webservices
31. Muthukumar, B., Dhanagopal, R. & Ramesh, R. KYP modeling architecture for cardiovascular diseases and treatments in healthcare institutions. *J Ambient Intell Human Comput* (2019). https://doi.org/10.1007/s12652-019-01653-z
32. Dhanagopal, R., Muthukumar, B. A Model for Low Power, High Speed and Energy Efficient Early Landslide Detection System Using IoT. *Wireless Pers Commun* (2019). https://doi.org/10.1007/s11277-019-06933-7