Probabilistic Latent Semantic Data Analysis for Grouping and Matching Process using Field Matching Algorithm

A. Ghouse Mohiddin[1], S.Ramakrishna[2], Sheik Mohamed[3]
[1]Research Scholar, Dept. of Computer Science, Dravidian University, Kuppam,
[2]Dept. of Computer Science, Sri Venkateswara University, Tirupathi
[3]Research Scholar, Dept. of Computer Science, Sri Venkateswara University, Tirupathi
A.P., India.

Abstract - We classify data quality problems that are sent by data cleaning and provide an overview of the principal Solution approaches. Data cleansing is particularly needed when integrating heterogeneous information sources and should be sent together with schema-related data transformations.. Data Analysis offers a delineation of data structure, content, rules and relationships by using statistical methodologies to deliver a lot of standard characteristics data types, field lengths and cardinality of columns, granularity, value sets, format patterns, content patterns, implied rules, cross-column, data relationships and cardinality of those relationships. Data de-duplication has been advocated as a promising and effective technique to save the digital space by removing the duplicated data from the data centres or clouds. Data de-duplication is a process of identifying the redundancy in data and then removing it. The resulting unique data/consolidate data into single format using data cleansing and Data standardization. The Soundex generates an alphanumeric code that represents the characters at the start of a string. It creates a code based on how the word sounds and takes variations of spelling into account. Matching will identify related or duplicate records within a dataset or across two datasets. Matching scores records between 0 and 1 on the strength of the match between them, with a score of 1 indicating a perfect match between records. It reads the values in selected input columns and calculates match scores representing the points of similarity between the couples of values. Match Type (Pair Generation), Strategies (Scoring), Match Output (Processing).

Keywords - Data Analysis, Data Cleansing, Data Standardization, Data Grouping, Matching Techniques Algorithms.

I. INTRODUCTION

Data profiling is a specific form of data analysis customer data to detect and characterize important features of data sets. It offers a delineation of data structure, content, rules and relationships by using statistical methodologies to deliver a lot of standard characteristics, information characters, field lengths and cardinality of columns, granularity, value sets, format patterns, content patterns, implied rules, and cross-column and cross-file data relationships and cardinality of those relationships. [4,5] Data cleansing, or Data scrubbing, deals with detecting and removing faults and incompatibilities of data in order to improve the quality of data.. In parliamentary law to provide access to accurate and consistent data, consolidation of different data representations and elimination of duplicate information become necessary.

Data de-duplication is a process of identifying the redundancy in data and then removing it. The resulting unique data/consolidate data into single format using data cleansing and Data standardization. The Soundex generates an alphanumeric code that represents the characters at the start of a string. Matching techniques will identify related or duplicate records within a dataset or across two datasets. Matching scores records between 0 and 1 on the strength of the match between them, with a score of 1 indicating a perfect match between records.[7]

In Section 2 of this paper, we discuss the data analysis and profiling in CRM system. We present Statement of problem-Hypotheses-Customer data validation using Data cleaning and Data Standardization in Section 3. We present Effectiveness of Data Grouping and Matching Technique Algorithms in Section 4. Section 5 we present several considerations in the conclusion.

II. RESEARCH METHOD

A. Data Profiling

Data profile is a specific form of data analysis customer data to detect and characterize important features of data sets. Its content different data rules by using statistical methodologies to deliver a lot of standard characteristics from the customer data, data types, field lengths and issue of data quality[4]. A profile is a set of metadata that describes the content and structure of a dataset. We can run a profile to evaluate the structure of data and verify that data columns are populated with the types of information we expect.

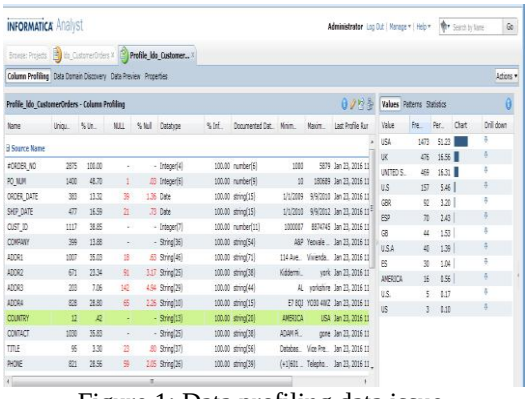


Figure 1: Data profiling data issue.

B. Data Standardization

The Data Standardizer is standardizes characters and strings in data. It can be used to remove noise from a field. It is a passive transformation an input strings and creates standardized versions of those strings. Standardization addresses the data quality issues identified through data profiling. The Parser transformation can parse input data using the following methods:

- Token set.
- Regular expression.
- Reference table.

C. Data cleaning

A data cleaning should find and remove all major faults and inconsistencies both in individual data sources. Data cleaning should not be done in isolation, but together with schema-related data transformations based on comprehensive metadata. The major data quality problems to be puzzled out by data cleaning and information translation[5,6].

III. STATEMENT OF PROBLEM-HYPOTHESES

A. Data Pair Generation

The KeyGen transformation is an active transformation that organizes records into groups based on data values in a column that you select. Use this transformation to sort records before passing them to the Match transformation.

The KeyGen transformation uses a grouping strategy to create group keys for the column you select. The strategies are String, Soundex, and NYSIIS. Records with common values in the selected field have a common group key value. The Match transformation processes records with common group key values together. This enables faster duplicate analysis in the Match transformation.

String: Builds a group key using the first or last number of characters.

Soundex: The Soundex generates an alphanumeric code that represents the characters at the start of a string. It creates a code based on how the word sounds and takes variations of spelling into account.

NYSIIS: The NYSIIS convert a word into its phonetic equivalent.

The number of comparison operations that the Match transformation must perform grows exponentially with the number of records in the data set. This exponential growth can consume significant amounts of computing resources. By creating group keys, the Key Generator transformation enables the Match transformation to compare records in smaller groups, which reduces processing time[6,7].

The Key Generator transformation can also create a unique ID for each record. [6] Each record that enters the Match transformation must contain a unique ID. Use the Key Generator transformation to create IDs for Business data if none exist

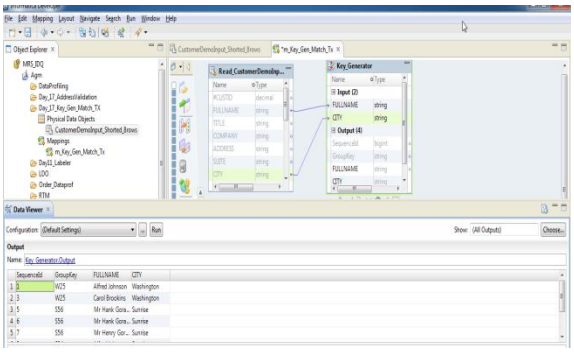


Figure 2: Input data of KeyGen Transformation

SequenceId is generated sequence number for all the matching data and **GroupKey** is generated key columns combination of alpha numeric values, it provide same key values for matching input data. An above example sequenceid 1 and 2 are same groupkey (W25) and sequenceid 5 to 7 have same groupkey (S56) for matching input data.

B. Data Processing

We can analyze the records in a single data set or across two data sets. The Match transformation enables this by creating two copies of each input column. Search for duplicates in a single data set by selecting the cloned copies of a column. Search for duplicates across two data sets by selecting unique columns from each data set. We can match multiple pairs in the Match transformation[11].

The Match transformation contains a set of comparison strategies that compare values in different ways. Select the fields to be compared and the type of strategy to apply to the fields. Every matching strategy we define generates match scores, which means that the transformation can generate multiple scores related to values in a single record. The transformation calculates an average match score that summarizes the degree of similarity between different records and allows we identify the records that are most similar to one another. Use the transformation to set a match threshold for the match scores. [8] The match threshold represents the minimum level of similarity needed to determine that two records are potential duplicates.

1. Match Input and Output Ports - The Match transformation contains predefined input and output ports for data relating to matching operations.

1.1 Input Ports: Match transformation input ports provide the data the transformation requires for matching operations. After we create a Match transformation, we can configure the following input ports.

SequenceId: Provides an ID that uniquely identifies each record in the source data set. Use the Key Generator transformation to create unique IDs if none exist in the data set.

GroupKey: Provides the group key that the Match transformation uses to process records. Identity matching and field matching can use a group key. Ensure that the

group key and sequence ID fields we select come from the same transformation.
To improve matching speeds, configure both the GroupKey input port and the output port that connects to it with the same Precision value[11].

- **Output Ports:** Match transformation output ports provide information about the duplicate analysis that the transformation performs. After we create a Match transformation, we can configure the following output ports [8]

ClusterId: The ID of the cluster to which the record belongs. Used in Clusters match output.
Group Key: The group key of the record.
Cluster Size: The number of records in the cluster to which a record belongs. Records that do not match with other records have a cluster size of 1. Used in Clusters match output.
RowId and RowId1: A unique row ID for the record. The Match transformation creates this ID. This ID may not match the row number in the input data.
DriverId: The row ID of the driver record in a cluster. The driver record is the final record added to the cluster. Used in Clusters match output.
DriverScore: The match score between a record and the driver record in its cluster.
LinkId: The row ID of the record that matched with the current record and linked it to the cluster.
The Link Score defines the contents of the cluster. It must exceed the match threshold. The Driver Score may be higher or lower than the LinkScore, and it may be lower than the match threshold.

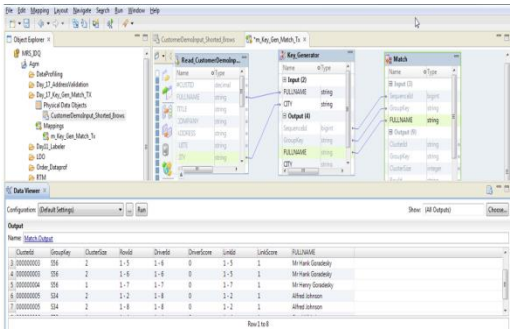


Figure 3: The output ports for the KeyGen transformation

C. Data Match Scoring (Matching Algorithms)

Use field matching to find similar or duplicate data in two or more records. Field matching operations compare the values from two data fields and calculate the similarity between them. We can select one or more pairs of columns from the input data. [10] The Match transformation includes predefined field matching strategies that compare pairs of data values.

1. Bigram Algorithm: It is one of my favorites due to its thorough decomposition of a string. The bigram algorithm matches data based on the occurrence of consecutive characters in both data strings in a matching pair, looking for pairs of consecutive characters that are common to both strings. The larger number of common identical pairs between the strings, the higher the match score. This algorithm is useful in the comparison of long text strings, such as free format address lines. Use the Bigram algorithm to compare long text strings, such as postal addresses entered in a single field.
The Bigram algorithm calculates a match score for two data strings based on the occurrence of consecutive characters in both strings. The algorithm looks for pairs of consecutive characters that are mutual to both strings and divides the actof matching character pairs by the total number of quality pairs[12].

1.1 Bigram Example:

string1	string2
Damien	Darren

The Bigram pairs for the two inputs are as follows

Da,am,mi,ie,en
Da,ar,rr,re,en

Here total 10 pairs yielding 4 matches (i.e. 2 matched pairs "Da,en"). Therefore, the Bigram Distance between these strings is 0.4.
Bigram Distance = Total Matched pairs/Total No.of pairs 4/10=0.4
Example2: Consider the following strings:

- Larder
- lerder

These strings yield the following Bigram groups:

la, ar, rd, de, er
le, er, rd, de, er

Here
Total pairs for both fields=10 pairs,Total Matched pairs=6 pairs, =6/10=0.6

Note that the second occurrence of the string "e r" within the string "lerder" is not matched, as there is no corresponding second occurrence of "e r" in the string "larder".

To calculate the Bigram match score, the transformation divides the number of matching pairs by the total number of pairs in both strings An above example, the text are 60% similar and the match score is 0.60.

2. Edit Distance Algorithm: The Edit Distance algorithm is an implementation of the Levenshtein distance algorithm where matches are computed based on the minimum number of operations required to transform one string into the other. These operations can include an insertion, deletion, or substitution of a single character. This algorithm is easily suited for matching fields containing a short text string such as a name or short address field. The edit-distance between strings x and y is the minimal number of Insertions, Deletions and Substitutions. Use the Edit Distance algorithm to compare words or short text strings, such as names, F_Name, L_Name, and M_Name etc.
The Edit Distance algorithm calculates the minimum “cost” of transforming one string to another by inserting, deleting,

or replacing characters. Score is number of unchanged characters/longest string. Strings of arbitrary length, derives a match score for two values by calculating the minimum cost of transforming one string into another by the insertion, deletion and replacement of characters. It determining the minimum number of edit operations necessary to change one string into another. A single edit operation may be shifting a single symbol of the chain into another, canceling, or entering a symbol. The length of the edit sequence provides a measure of the distance between the two strings

EditDistance Algorithms:

EDITDISTANCE(s1,s2)
1. int m[i,j]=0
2. for i <- 1 to [s1]
3. do m[i,0]=i
4. for j<- 1 to [s2]
5. do m[0,j]=j
6. for i <- 1 to [s1]
7. do for j<-1 to [s2]
8. do m[i,j]=min{m[i-1,j-1]+ if (s1[i]=s2[i]) then 0 else 1 fi,
9. m[i-1,j]+1,
10.m[i,j-1]+1}
11 return m [[s1],[s2]].

Example 1:

String1 String2
LEVENSTON LEVENSHTEIN

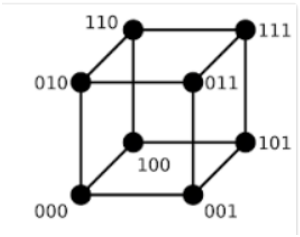
Here

String1 length size=9
String2 length size=11
Unchanged character=8
Unchanged character/max length characters
Result=8/11=0.7

3. Hamming Distance Algorithm: Use the Hamming Distance algorithm when the position of the data characters is a critical factor, for example in numeric or code fields such as telephone numbers, ZIP Codes, or product codes.

This algorithm calculates to match score for two data strings by computing the number of positions in which characters differ between the data strings. For strings of different length, each additional character in the longest string is counted as a difference between the strings.

The Hamming distance algorithm, for instance, is particularly useful when the positions of the characters in the string are important. Examples of such strings are telephone numbers, dates and postal codes. The Hamming Distance algorithm measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other[12]. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. A major application is in coding theory, more specifically to block codes, in which the equal-length strings are vectors over a finite field.



3.1 Hamming Distance Example

Consider the following strings:

- Morlow
- Marlowes

The highlighted characters indicate the positions that the Hamming algorithm identifies as different.

To calculate the Hamming match score, the transformation divides the number of matching characters by the length of the longest string. The strings are 62.5% similar and the match score is 0.625.

IV. RESULTS AND ANALYSIS

A. Data Association

It processes output data from a Match transformation. It makes links between duplicate records that are attributed to different match clusters, so that these discs can be related together in data consolidation and master information management operations. It generates an Association ID value for each row in a group of associated records and writes the ID values to an output port. It use a Consolidation transformation to produce a master record based on books with common association ID values.

The following table contains three records that could identify the same individual.

Table 1: Contains three records of Association.

ID	Name	Address	City	State	ZIP	SSN
1	David Jones	100 Admiral Ave.	New York	NY	10547	987-65-4321
2	Dennis Jones	1000 Alberta Ave.	New Jersey	NY		987-65-4321
3	D. Jones	Admiral Ave.	New York	NY	10547-1521	

It does not identify all three records as duplicates of each other, for the following reasons:

If we define a duplicate search on Name and Address data, records 1 and 3 are identified as duplicates but record 2 is excluded. If we define a duplicate search on Name and SSN data, records 1 and 2 are identified as duplicates but record 3 is omitted[11,12].

A different match clusters, so that records that share a cluster ID are given a common [6] AssociationID value. In this example, all three records are given the same AssociationID, as shown in the following table:

Table 2: Consolidated the three records from Association.

ID	Name	Address	City	State	Zip	SSN	Name and Address Cluster ID	Name and SSN Cluster ID	Association ID
1	David Jones	100 Admiral Ave.	New York	NY	10547	987-65-4320	1	1	1
2	Dennis Jones	1000 Alberta Ave.	New Jersey	NY		987-65-4320	2	1	1
3	D. Jones	Alberta Ave.	New York	NY	10547-1521		1	2	1

B. Data Consolidation

It is an active transformation that analyzes groups of related records and creates a consolidated record for each group. Use the Consolidation transformation to consolidate record groups generated by transformations such as the Key Generator, Match, and Association transformations. The Consolidation transformation generates consolidated records by applying strategies to groups of related records. The transformation contains an output port that indicates which record is the consolidated record. We can choose to limit the transformation output to include only consolidated records[11].

Row-Based Strategies: A row-based strategy analyzes rows in the record group and selects one row. The Consolidation transformation uses the port values from that row to create a consolidated record. The default strategy is "most data." Choose one of the following row-based strategies:

Most data: Selects the row with the highest character count. If the highest character count is shared by two or more rows, the strategy returns the last qualifying value.

Most filled: Selects the row with the highest number of non-blank columns. If the highest number of non-blank columns is shared by two or more rows, the strategy returns the last qualifying value.

Modal exact: Selects the row with the highest count of the most frequent non-blank values. For example, consider a row that has three ports that contain the most frequent values in the record group. The count of the most frequent values for that row is "3." If the highest count of the most frequent non-blank values is shared by two or more rows, the strategy returns the last qualifying value. Fetch Golden Record based on IsSurvivor ="Y"

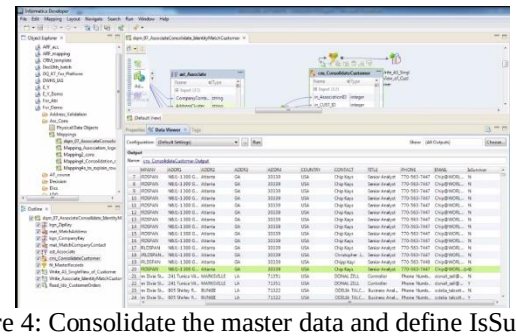


Figure 4: Consolidate the master data and define IsSurvivor column.

IsSurvivor= "N" means Transaction data **(Duplicate Records)**

IsSurvivor= "Y" means Master record data **(Unique Records).**

V. CONCLUSION

Data de-duplication is a process of identifying the redundancy in data and then removing customer data. A set of processes that measure and improve the quality of important data on an ongoing basis, ensures that data dependent business processes and applications deliver expected results. Data Standardization is the problems with the data have been identified, to cleanse the data through standardization process, enrichment and validate the good data. The Address validation is the data quality issues identified through data profiling to transform and parse data from single fields to multiple fields. Data deduplication is a process of identifying the redundancy in data and then removing it. The resulting unique data/Consolidate data into single format using data cleansing and Data standardization. Using scorecards to measure data quality progress and shared URL link to the stakeholder.

VI. REFERENCES

[1]. Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record 26(1), 1997.

[2]. Batini, C.; Lenzerini, M.; Navathe, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration.In Computing Surveys 18(4):323-364, 1986.

[3]. Bouzeghoub, M.; Fabret, F.; Galhardas, H.; Pereira, J; Simon, E.; Matulovic, M.: Data Warehouse Refreshment. In [16]:47-67.

[4]. Erhard Rahm and H. Hai Do. Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 23(4):3--13, December 2000.

[5]. Lee, M.L.; Lu, H.; Ling, T.W.; Ko, Y.T.: Cleansing Data for Mining and Warehousing. Proc. 10th Intl. Conf.Database and Expert Systems Applications (DEXA), 1999.

[6]. Quass, D.: A Framework for Research in Data Cleaning. Unpublished Manuscript. Brigham Young Univ., 1999.

[7]. Hernandez, M.A.; Stolfo, S.J.: Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge discovery 2(1):9-37, 1998.

[8]. Batini, C.; Lenzerini, M.; Navathe, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration.In Computing Surveys 18(4):323-364, 1986.

[9]. Haas, L.M.; Miller, R.J.; Niswonger, B.; Tork Roth, M.; Schwarz, P.M.; Wimmers, E.L.: Transforming Heterogeneous Data with Database Middleware: Beyond Integration. In [26]:31-36, 1999.

[10]. Kashyap, V.; Sheth, A.P.: Semantic and Schematic Similarities between Database Objects: A Context-Based Approach. VLDB Journal 5(4):276-304, 1996.

[11]. Y. Tan, H. Jiang, D. Feng, L. Tian, Z. Yan, and G. Zhou, \Sam: A semantic-aware multi-tiered source de-duplication framework for cloud backup," in Parallel Processing (ICPP), 2010 39th International Conference on, pp. 614 {623, Sept. 2010.

[12]. F.Rashid, A.Miri, and I.Woungang, A secure data deduplication framework for cloud environments," in Privacy,

Security and Trust (PST), 2012 Tenth Annual International Conference on, pp. 81 {87, July 201

[13]. K.S.N.Prasad, S.Ramakrishna "Text Analytics to Data Warehousing" (IJCSE) International Journal on Computer Science and Engineering" Vol.02,No.06,2010,PP:2201-2207.

[14]. A.Ghouse Mohiddin, S.Ramakrishna "Tactics for Dynamic Data Cleansing and Data Profiling Using Dimensions for Data Quality Assessment" (IJCSE) International Journal on Computer Science and Engineering" Volume-6, Issue-4 E-ISSN: 2347-2693

Authors Profile

Mr. A.Ghouse Mohiddin Master of Computer Application from M.K.University of Madurai, Tamil Naidu, in year 2003 and Master of Philosophy in Computer Science from Periyar University,Salam,Tamil Naidu,India in year 2008. He is currently pursuing Ph.D. and currently working as Senior Technical Consultant in Capgemini Technology Services India Limited, Bangalore. His main research work focuses on Data warehousing, Data De-duplication and Data Standardization, fuzzy Logic Algorithms, Data Base Management System, Cloud Security and Privacy, Big Data Analytics, and Data Mining.

Mr. S.Ramakrishna Master of Science from S.V.University of Tirupathi, A.P. India in year 1983. Doctor of Philosophy from S.V.University of Tirupathi, A.P. India in year 1988. He is currently working as Professor in Department of Computer Science, S.V.University of Tirupati, A.P. India. He has published more than 100 research papers in reputed international journals. He has more than 30 years of teaching experience and more than 10 years of Research Experience.

Mr. Sheik Mohamed Master of Computer Application from Bharathidasan University for,Tiruchirappalli, Tamil Naidu. He is currently pursuing Ph.D. S.V.University, Tirupathi, A.P. and currently working as Asst. Professor, MCA Dept., SITAMS, Chittoor, AP. He has published more than 18 research papers in reputed national and international journals.. He has more than 18 years of teaching experience and more than 9 years of Research Experience. His main research work focuses on Artificial Intelligence, Neural Networking, Data warehousing, fuzzy Logic Algorithms, Data Base Management System, Cloud Security and Privacy, Big Data Analytics, and Data Mining.