International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

An Approach to Detect Record Linkage from Different Relational Databases

Ms. P. L. Gaikwad , Mr. G. A. Patil

Abstract:

The primary reasons Record Linkage is used for exact matching to reduce or eliminate manual review and to make results more easily reproducible. Record or data linkage is an important enabling technology in the health sector [12], as linked data is a cost effective resource that can help to improve research into health policies, and uncover fraud within the health system. Record Linkage has the advantages of allowing better quality control, speed, and better results. Heterogeneous record [10] [11] linkage techniques could be used on the number of different machines for providing the possible matched Records. For achieving this result the Levenshtein Distance: LD, cosine similarity techniques are used. Cosine similarity is given by "dot product". The distance is the number of deletions, insertions, or substitutions required for the transformation. Although it may be possible to use common non key attributes (such as name, address, and date of birth) for this purpose, the result obtained using these attributes may not always be accurate. This is because non key attribute values may not match even when the records represent the same entity instance in reality. The above problem where a real-world entity type is represented by different identifiers in two databases is quite common in the real world and is called the entity heterogeneity. Entity heterogeneity problem is

Solved by using Heterogeneous Record Linkage. Information stored in the original data into a well defined and consistent form. Information may be recorded or captured in various formats, spelled differently, it might be having spaces, some items may be missing or contain errors. Typing errors happen frequently when dates are entered. The Preprocessing and standardization steps attempt to deal with these problems. Transformation of the original input data into a well defined

form, and dividing it into many smaller output fields, gives the record linkage process to be much more accurate.

Index Terms— ***LD, enquiry record, cosine similarity, Record Linkage***

I. INTRODUCTION

Record or data linkage is an important enabling technology in the health sector, as linked data is a cost effective resource that can help to improve research into health policies, reduce costs, and uncover fraud within the health system. Record linkage has applications in customer systems for marketing, customer relationship management, fraud detection, data warehousing, law enforcement and government administration. These applications can be classed as 'administrative', because the record linkage is used to make decisions and take actions regarding an individual entity Significant advances, mostly originating from the data needed to support these decisions are often scattered in heterogeneous distributed databases. In such cases, it may be necessary to link records in multiple databases [11] so that one can consolidate and use the data pertaining to the same real world entity. If the databases use the same set of design standards, this linking can easily be done using the primary key (or other common candidate keys). However, since these heterogeneous databases are usually designed and managed by different organizations (or different units within the same organization), there may be no common candidate key for linking the

records. Although it may be possible to use common non key attributes (such as name, address, and date of birth) for this purpose, the result obtained using these attributes may not always be accurate. This is because non key attribute values may not match even when the records represent the same entity instance in reality. This problem where a real-world entity type is represented by different identifiers in two databases is quite common in the real world and is called the entity heterogeneity problem [9] or the common identifier problem. The key question here is one of record linkage given a record in a local database (often called the enquiry record), how do we find records from a remote database that may match the enquiry record? Traditional record linkage techniques however are designed to link an enquiry record with a set of records in a local master file.

Heterogeneous databases are usually designed and managed by different organizations or different units within the same organization, there may be no common candidate key for linking the records. Although it may be possible to use common non key attributes (such as name, address, and date of birth) for this purpose, the result obtained using these attributes may not always be accurate. This is because non key attribute values may not match even when the records represent the same entity instance in reality. The above problem where a real-world entity type is represented by different identifiers in two databases is quite common in the real world and is called the entity heterogeneity.

APPROACH

Traditional record linkage techniques however are designed to link an enquiry record with a set of records in a local master file. Given the enquiry record and a record from the master file [6]. The Records in data sources are assumed to represent observations of entities taken from a particular population. The records are assumed to contain some attributes (fields or variables) identifying an individual entity. Examples of identifying attributes are name, address, age and gender [1].

Suppose source A has na records and source B has nb records. Each of the nb records in source B is a potential match for each of the na records in source A. So there are na × nb record pairs whose match/non-match status is to be determined. Two disjoint sets M and U can be defined from the cross-product of A with B, the set A × B. A record pair is a member of set M if that pair represents a true match. Otherwise, it is a member of U. The record linkage process attempts to classify each record pair as belonging to either M or U. Many matching problems are more constrained than this statement of the problem. For instance, if each record in data source B refers to a distinct entity, a record in data source A cannot be matched to two records at the same time in data source B. It is more generally referred to as 1-1 linkage.

Architecture of the Record Linkage Problem following record pairs are labeled as:

1. Match, A1.
2. Possible match, A2.
3 .Non-matches, A3.

Searching or blocking is used to reduce the number of comparisons of record pairs by bringing potentially linkable record pairs together. A good attribute variable for blocking should contain a large number of attribute values that are fairly uniformly distributed and such an attribute must have a low probability of reporting error. Errors in the attributes used for blocking can result in failure to bring linkable record pairs together. Status is to be determined [8] Two disjoint sets M and U can be defined from the cross-product of A with B, the set A × B. A record pair is a member of set M if that pair represents a true match. Otherwise, it is a member of U. The record linkage process attempts to classify each record pair as belonging to either M or U.

For identifying similar records between N different sources or grouping of records form N number of different sources the compare and determine score technique is used so it is possible to classify and split the records into separate streams.

3. SYSTEM OVERVIEW

The proposed work consists of the implementation of distributed architecture for record linkage technique fig (b). Six recommended steps will be implemented namely

1. Preprocessing Standardization

2. Classification
3. Split into the Separate Stream
4. Compare Records and Determine Score
5. Split into Separate Match Categorize
6. Analyze Result Of Matches By Clerical Review

1. Preprocessing Standardization:

a. Create common formats & patterns for data values.

b. Preferable data driven rules that can be shared and reused [6].

2. Classification:

a. Choose single/multiple values.

b. Create a concatenated value free or spaces or special characters.

3. Split into the Separate Stream:

a .Create separate data streams to support parallel match processing

4. Compare Records and Determine score:

a. Base on type of value name, select appropriate algorithm [9].

5. Split into Separate Match Categorize:

a. Records are categorized in Match, possible match and non match.

6. Analyze Result of Matches By Clerical Review:

a. Matches need to be reviewed for accuracy; this can be done with tools or in some cases manually.

A general schematic outline of the record linkage process is given in Figure. As most real-world data collections contain noisy, incomplete and incorrectly formatted information, data preprocessing and standardization are important data cleaning steps for successful record linkage, and also before data can be loaded into data warehouses or used for further analysis or data mining . A lack of good quality data can be one of the biggest obstacles to successful record linkage and reduplication, main task of data preprocessing and standardization is the conversion of the raw input data into well defined, consistent forms, as well as the resolution of inconsistencies in the way information is represented and encoded.

1. Preprocessing Standardization:

The basic goal of the Preprocessing and standardization process is to convert the information stored in the original data into a well defined and consistent form. Information may be recorded or captured in various formats, spelled differently, it might be having spaces, some items may be missing or contain errors. For example, if data is captured over the telephone, spelling variations of names are common. Typing errors happen frequently when dates are entered. The Preprocessing and standardization steps attempt to deal with these problems. Transformation of the original input data into a well defined form, and dividing it into many smaller output fields, gives the record linkage process to be much more accurate.

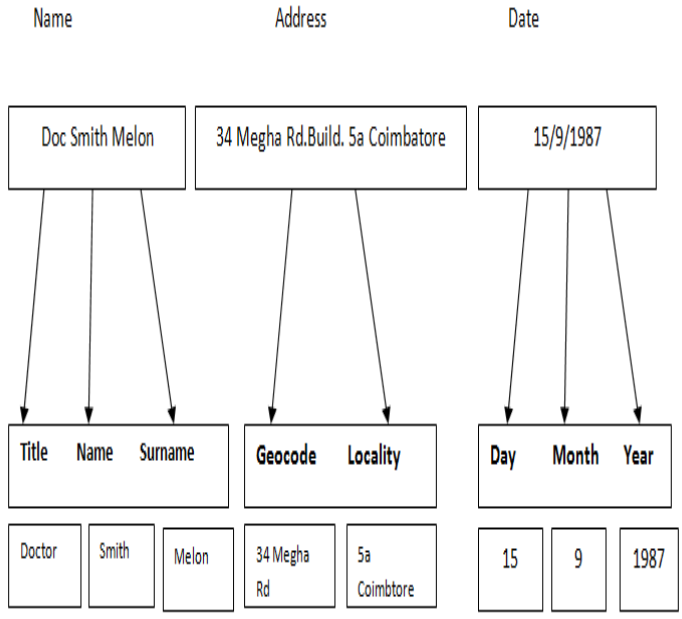


Fig (a).Example of Standardization personal information

As an example, the record in Figure (a) with four input components is cleaned and split into 14 output fields. Comparing these output fields with the respective fields of

other records results in a much better linkage quality than just comparing for example the whole name or the whole address as a string with the name or address from other records. Personal data used for record linkage can be broadly categorized into five classes: names, addresses, dates and times, categorical attributes and scalar quantities such as height or weight. The main criteria for such data is that they are relatively invariant over time, they should not change, or at least not change often. For these reasons attributes such as diagnoses or medical findings, are generally not used for record linkage purposes. Similarly, scalar attributes are also rarely used because they are subject to change, although it depends on the specific application.

Levenshtein Distance: LD

Levenshtein distance (LD) is a measure of the similarity between two strings [5], which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t.

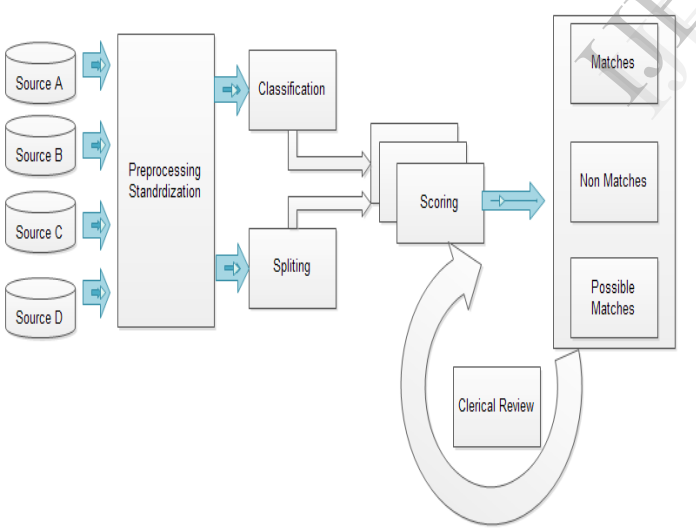


Fig (b). The distributed architecture for Heterogeneous Record Linkage

If s is "test" and t is "test", then LD(s,t) = 0, because no transformations are needed. The strings are already identical.

If s is "test" and t is "tent", then LD(s,t) = 1, because one substitution (change "s" to "n") is sufficient to transform s into t.

The greater the Levenshtein distance, the more different the strings are.

The Algorithm

Steps: 1. Set n to be the length of s.
Set m to be the length of t.
If n = 0, return m and exit.
If m = 0, return n and exit.
Construct a matrix containing 0..m rows and 0..n columns.

2. Initialize the first row to 0..n.
Initialize the first column to 0..m.

3. Examine each character of s (i from 1 to n).

4. Examine each character of t (j from 1 to m).

5. If s[i] equals t[j], the cost is 0.
If s[i] doesn't equal t[j], the cost is 1.

6. Set cell d[i,j] of the matrix equal to the minimum of:
a. The cell immediately above plus 1: d[i-1,j] + 1.
b. The cell immediately to the left plus 1: d[i,j-1] + 1.
c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.

7. After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

II. Cosine Similarity and Term Weight

The DOT Product is defined as “DOT Product=Term Counts/ Documents *Query Term Counts”

Cosine similarity is given by "dot product / (document + query magnitudes) = cosine".

Term vector theory makes no provision for document normalization.

a. local nature; i.e. at the level of documents
b. global nature; i.e. at the level of database collections
c. scaling nature; i.e., through length scales

1) **The Product DOT:**

If we multiply the coordinates of A and B and add the products together we get the "mythical" DOT Product, also known as the inner product and scalar product. So the A•B DOT Product is given by

Equation 1: A•B = x1*x2 + y1*y2

If points A and B are defined in three dimensions then their coordinates are (x1, y1, z1) and (x2, y2, z2) and these points can be referred to as A(x1, y1, z1) and B(x2, y2, z2). The A•B DOT Product is now given by

Equation 2: A•B = x1*x2 + y1*y2 + z1*z2

To define a straight line we need at least two points. So, if we draw a straight line from C to either A or B, we can define the distance, d, between the points. This is the so-called Euclidean Distance, which can be computed in four easy steps. For any two points defining a straight line:

1. take the difference between the coordinates of the points
2. square all differences
3. add all squared differences
4. square root the final result

Since we have defined x0 = 0 and y0 = 0, then to find out how far A is from C the Euclidean Distance as

Equation 3: $d_{AC} = ((x_1 - x_0)^2 + (y_1 - y_0)^2)^{1/2} = (x_1^2 + y_1^2)^{1/2}$

Similarly, to find out how far B is from C

Equation 3: $d_{BC} = ((x_2 - x_0)^2 + (y_2 - y_0)^2)^{1/2} = (x_2^2 + y_2^2)^{1/2}$.

$$Sim(A, B) = \text{cosine } \theta = \frac{A \bullet B}{|A||B|} = \frac{x1*x2 + y1*y2}{(x1^2 + y1^2)^{1/2} \ (x2^2 + y2^2)^{1/2}}$$

Figure . The cosine angle between A and B.

As the angle between the vectors shortens the cosine angle approaches 1, meaning that the two vectors are getting closer, meaning that the similarity of whatever is represented by the vectors increases.

This is a convenient way of ranking documents; i.e., by measuring how close their vectors are to a query vector. For instance, let say that point A(x1, y1) represents a query and points B(x2, y2), D(x3, y3), E(x4, y4), F(x5, y5), etc represent documents. To do this we need to construct a term space. The term space is defined by a list (index) of terms. These terms are extracted from the collection of documents to be queried. The coordinates of the points representing documents and queries are defined according to the weighting scheme used.

If weights are defined as mere term counts ($w = tf$) then point coordinates are given by term frequencies; however, we don't have to define term weights in this manner. As a matter of fact, and as previously mentioned, most commercial search engines do not define term weights in this way, not even in terms of keyword density values.

$$Sim(Q, D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}}$$

where the sigma symbol means "the sum of", Q is a query, D is a document relevant to Q and *w* are weights (see reference 4). How these weights are defined determines the significance and usefulness of the cosine similarity measure. By defining t_{max} as maximum term frequency in a document, *N* as number of documents in a collection and *n* as number of

documents containing a query term, we can redefine term weights as

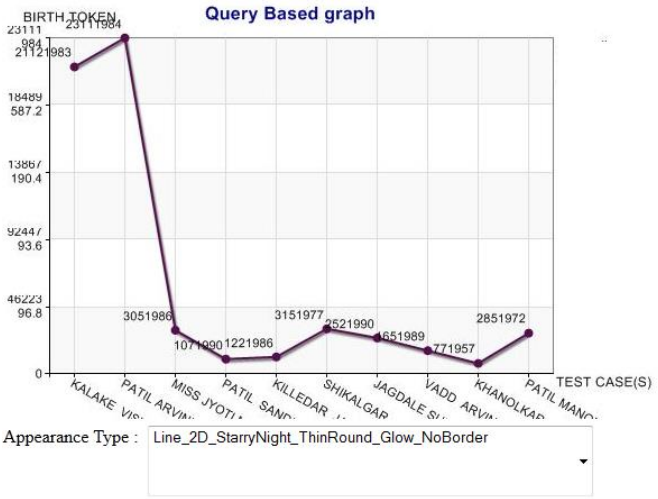
w = tf /tfmax

w = IDF = log(N/n)

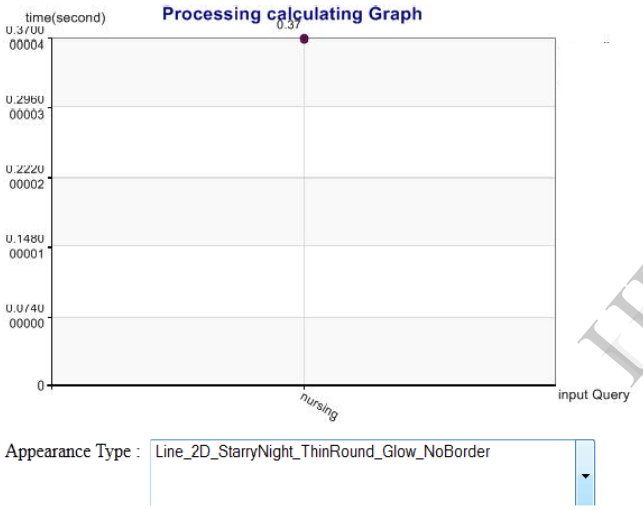
w = tf*IDF = tf*log(N/n)

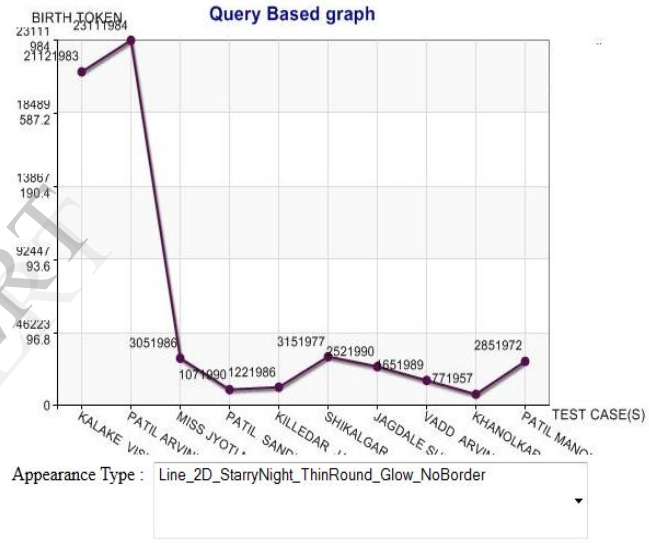
w = tf*IDF = tf*log((N - n)/n)

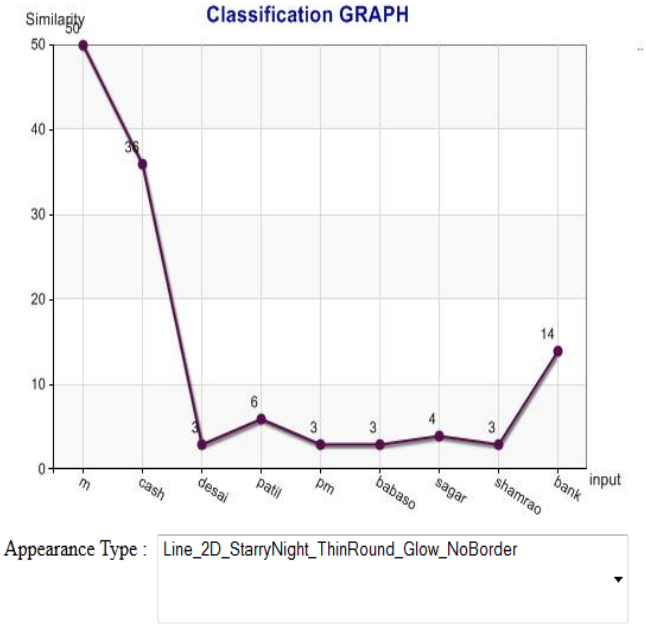
or even in terms of variants of *tf* and *IDF*, each one with their own customized definition and theoretical interpretation.

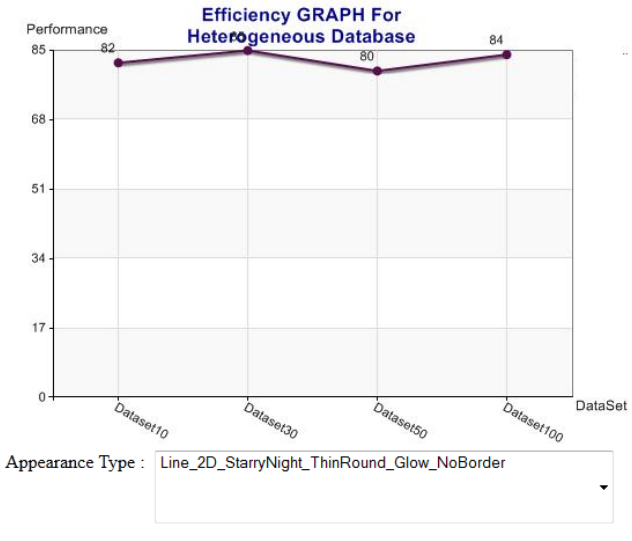


RESULTS









5. CONCLUSIONS

Record linkage is an important Technique in heterogeneous database system. Records representing the some type are identified using different identifiers in different databases. In the absence of a common identifier, it is often difficult to find records in a remote database that are similar to a given enquiry record.

REFERENCES

[1] R. Ash, Information Theory. John Wiley and Sons, 1965..

[2] T.R.Belin and D.B. Rubin, "A Method for Calibrating False-Match Rates in Record Linkage," J. Am. Statistical Assoc., vol. 90, no. 430, pp. 694-707, 1995.

[3] P. Bernstein, "Applying Model Management to Classical Meta Data Problems," Proc. Conf. Innovative Database Research (CIDR),pp. 209-220, Jan. 2003.

[4] D. Dey, "Record Matching in Data Warehouses: A Decision Model for Data Consolidation," Operations Research, vol. 51, no. 2, pp. 240- 254, 2003.

[5] D. Dey, S. Sarkar, and P. De, "A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 3, pp. 567-582,
May/June2002.

[6] A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios, "Duplicate Record Detection: A Survey," IEEE Trans. Knowledge and Data Eng.,vol. 19, no. 1, pp. 1-16, Jan. 2007.

[7] M.J. Goldacre, J.D. Abisgold, D.G.R. Yeates, and V. Seagroatt, "Risk of Multiple Sclerosis after Head Injury: Record Linkage Study," J. Neurology, Neurosurgery, and Psychiatry, vol. 77, no. 3,pp. 351-353, 2006.

[8] L. Gu and R. Baxter, "Adaptive Filtering for Efficient Record Linkage," Proc. Fourth SIAM Int'l Conf. Data Mining (SDM '04),pp. 22-24, Apr. 2004.

[9]D. Dey, S. Sarkar, and P. De, "A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases," Management
Science, vol. 44, no. 10, pp. 1379-1395, 1998.

[10] D. Dey, S. Sarkar, and P. De, "A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases," IEEE Trans. Knowledge and Data Eng.,vol. 14, no. 3, pp. 567-582, May/June2002

[11] J.A.Baldwin,"Linked Record Health Data Systems," The Statistician, vol. 21, no. 4, pp. 325-338, 1972.

[12] Record Linkage: Current Practice and Future Directions Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford

[13] **Febrl** – Freely extensible biomedical record linkage Release 0.2.2 Peter Christen , Tim Churches November 13, 2003