Asian Journal of Computer Science and Technology
ISSN 2249-0701 Vol. 2 No. 1, 2013, pp.39-44

Heterogeneous Record Linkage Using CAA

K. Kumaresan

PG Scholar, Department Of Information Technology, Anna University of Technology, Coimbatore - 641 047, Tamil Nadu, India

E-mail : Kumaresh2020@gmail.com

(Received on 10 December 2012 and accepted on 15 May 2013)

Abstract - **Record linkage is a scheme to retrieve the related data's from more than one table which are not in the same structure and not reside in the same places. Matching techniques facing following problems, (1) no common attribute to match Records between the data tables. (2) Record linkage in online is not an efficient and which provide traffic and may some connectivity failures will occur. (3) Previous techniques will not concentrate on unduplicated error Record (spelling mistakes). Using CAA (Concurrent Attribute Acquisition) and UGK (User Generated Key) approach not all the attributes of the entire remote attribute Records are taken into local site [LS]. Rather only the related attribute Records are taken into LS. So the communication traffic is reduced. Then Local Entity [LE] will be compared with each other Downloaded Remote table Records. Traditional Blocking (Group the record which have relationship from the Data set) to identify the required Records. Misspelled original Record also identified. After this process related Record identified with their identifier and table information. Insert this information on the new table [NT]. Publish NT as a global access Databases.**

Keywords : **Record linkage, Data Linkage, Data Matching, Record Blocking, Datamining.**

I. Introduction

Record linkage (RL) refers to the process of finding records that refer to the same records from different data sources. Record linkage is an useful technique when we have to join data sets that do not share a common identifier[1] (e.g., database key, URI, National identification number, Social Security Number), Due to differences in record shape, storage location this is refer to as heterogeneity[1][12][7] databases. Record linkage is a useful tool when performing data mining tasks, where the data originated from different sources or different organizations. Most commonly, performing RL on data sets involves joining records of persons based on name, DOB, address, pin code, since no National identification number or similar is recorded in the data.

National Identity/Insurance Card number is used by many countries for follow their citizens, permanent address, and temporary address for the purposes of work, taxation, government benefits, health care, and other governmentally-related functions. Record linkage is important to social history research since most data sets, such as census records and rural community registers were recorded long before the invention of National identification numbers. When old sources are digitized, linking of data sets is a requirement for longitudinal study. This process is difficult by lack of standard spelling of names, family names that change according to place of lodging, changing of administrative boundaries, and problems of checking the data against other sources.

Many organizations in the health sector are collecting, storing, processing and analyzing more and larger data collections with millions of records. Most of this data is about patients and contains identifying (such as names, addresses, and dates of birth), as well as confidentiality information (such as details of medical procedures and tests). Analyzing this data over and over again requires information from multiple data source to be linked and in order to enable more detailed analysis, and study to link those data otherwise its quite impossible to link. Nowadays, healthcare record linkage not only faces computational and operational challenges due to the increasing size of data collections and their complexity, but also faces privacy and confidentiality challenges when we integrate the record from other data sources. Nowadays, data linkage techniques are applied in and between government organizations to find information about taxation, census, immigration, social welfare, in crime and fraud detection, and also in terrorism intelligence.

Computer-assisted data linkage [5] goes back as far as the 1950s, and the mathematical foundation of probabilistic data linkage (as developed by Fellegi and Sunter in 1969) is still the basis of many current linkage systems. Often the linkage process is challenged by the lack of a common unique entity

identifier, and thus becomes non-trivial. In such cases, person identifiers (like names and dates of birth), demographic information (like addresses) and other specific information (like medical details) have to be used to achieve good linkage results. These attributes, however, can contain typographical errors, they can be coded differently, parts can be out-of-date or swapped, or even be missing.

In recent years, computer science researchers have started to explore the use of various techniques taken from machine learning, data mining, database research, information retrieval, and artificial intelligence to improve the linkage process[6]. Techniques investigated include learning the optimal parameters for approximate string comparison techniques[6] (like edit-distance costs); representing records as document vectors (an approach taken from information retrieval); applying active learning (a technique where the learning system selects difficult pairs of records for manual classification, thereby reducing human intervention); using supervised learning[5] approaches (where manually prepared training data, i.e. pairs of classified records, are needed to train a classifier); and clustering[6] (unsupervised learning techniques that explore the structure of the data without the need of manual training examples). Many of these new approaches, however, do require training data, which is often not available in real world situations, or only obtainable via manual preparation (a costly process similar to manual clerical review). Additionally, many of the recent publications in this area present experimental linkage studies that are based on only small data sets with a couple of thousand records.

Geocoding [6] is a technique related to data linkage or linking of addresses (that can contain typographical errors, or it may be incomplete, or out-of-date, or other errors) to a reference database and validated addresses and their geographic locations (latitude and longitude). Geocoding is important, as it is the initial step before record can be loaded into geographical information systems, and before it can be spatially analyzed and viewed. Spatial record analysis is critical, for example When researching of rapidly spreading infectious diseases, or when investigating bio-terrorism. Accurate linkage of addresses is important, as any subsequent data processing, visualization and analysis depends upon the quality of the linked records.

In the classical probabilistic approach[6][9] pairs of records from two data sets are compared using various similarity functions (like exact or approximate string, numerical, date, or age comparisons) and then classified into matches (if the compared attributes mainly agree), non-matches (if the compared attributes mainly disagree), or as possible matches (if the linkage system cannot make a clear decision).The class of possible matches are those record pairs for which manual clerical review is needed to decide their final linkage status. Data linkage of two data sets A and B considers record pairs in the product space |A| X | B| and determines which pairs are matches. Thus, the total number of record pairs equals the product of the two data sets, i.e. |A| X | B|, where |. | denotes the number of records in a data set. Comparing all pairs is computationally only feasible for small data sets containing up to several thousand records each, as, for example, linking two data sets with 100000 records each would result in 1010 (ten billion) record pair comparisons. Techniques known as blocking [6] are applied to reduce the number of record pair comparisons. They cluster records into blocks and only compare records within the same block, thereby reducing the complexity of the overall linkage process. By analyzing record linkage, two essential characteristics identified by authors of [1]:

The databases exhibiting entity heterogeneity [1][7] are distributed, and it is not possible to create and maintain a central data storage area or warehouse where pre computed Linkage results can be stored. [1] Found out a centralized solution may be impractical for several reasons

(I) First, if the databases span several organizations, the ownership and cost allocation issues associated with the warehouse could be quite difficult to address.

(II) Second, even if the warehouse could be developed, it would be difficult to keep it up-to-date. As updates occur at the operational databases, the linkage results would become stale if they are not updated immediately.

This staleness may be unacceptable in many situations. For instance, in a criminal investigation, one maybe interested in the profile of crimes committed in the last 24 hours within a certain radius of the crime scene. In order to keep the warehouse current, the sites must agree to transmit incremental changes to the data warehouse on a real-time basis. Even if such an agreement is reached, it would be difficult to monitor and enforce it. For example, a site would often have no incentive to report the insertion of a new record immediately.

Therefore, these changes are likely to be reported to the warehouse at a later time, thereby increasing the staleness of the linkage tables and limiting their usefulness. In addition, the overall data management tasks could be prohibitively time-consuming, especially in situations where there are many databases, each with many records, undergoing real-time changes. This is because the warehouse must maintain a linkage Table for each pair of sites, and must update them every time one of the associated databases changes.

The participating sites [1] allow controlled sharing of portions of their databases using standard database queries, but they do not allow the processing of scripts, stored procedures, or other application programs from another organization. The issue here is clearly not one of current technological abilities, but that of management and control. If the management of an organization wants to open its databases to outside scripts from other organizations, there are, of course, a variety of ways to implement it. However, the decision to allow only a limited set of database queries (and nothing more) is not based on technological limitations [1]; rather it is often a management decision arising out of security concerns. More investment in technology or a more sophisticated scripting technique [1], therefore, is not likely to change this situation. A direct consequence of this fact is that the local site cannot simply send the lone enquiry record to the remote site and ask the remote site to perform the record linkage and send the results back.

An important issue associated with record linkage in distributed environments [1] is that of schema integration. For record linkage techniques to work well, one should be able To identify the common nonkey attributes between two databases. If the databases are designed and maintained independently as in most heterogeneous Environments [1] it would be necessary to develop an integrated schema before the common attributes can be identified.

EXAMPLES

Consider the situation of a state in India consisting of about 40 districts. Each district has criminal data processing systems and their respective data models. The district share a important portion of the stored criminal records among themselves as, it has long been decided that it is not practical to create a central data warehouse that consolidates all the information.

Currently, a police inspector investigating a crime at the spot makes a phone call to a control room operator, who searches through the different databases. The process is quite incompetent. The search keys are satisfied by many records in several databases providing all the information back to the police officer over the phone is difficult, error-prone, and time-consuming. Finally, if all control room operators are busy working on other investigations, inspector may have to wait for a long time.

In order to address this problem, a proposal [1] is currently under consideration whereby the field personnel would be provided with handheld devices. The basic idea in this proposal [1] is that a crime investigator should be able to quickly download relevant information on these devices, instead of having to wait for a control room operator to do the necessary research.

Unfortunately, there are several challenges that has been found out by authors [1] in implementing this proposal. First, since no centralized data warehouse exists, an investigating officer may have to send queries to several databases separately to download the relevant information. Second, the handheld devices do not have enough storage capacity to download all the remote Databases [1] in a batch process and store them locally. Third, the connection speed on these machines is not very high, making it impossible to download millions of records on a real-time basis. Therefore, the practicality of the entire proposal depends on finding a way to download only the relevant criminal records to the handheld devices to be complex.

II. Related Works

In the work [1] Author proposed a new technique, called "Concurrent Attribute Acquisition", where the Remote records are partitioned repeatedly, until we obtain the desired path of all the related records. This recursive partitioning can be done in one of the following two ways: 1) by transferring the attributes of the remote records and comparing them locally 2) by sending a local attribute value, comparing it with the values of the remote records, and then transferring the identifiers of those remote records that match on the attribute value. In the concurrent partitioning scheme, we make a database query that selects the relevant remote records directly, in one single step. Hence, there is no need for identifier transfer. Once the relevant records are identified, all their attribute values are transferred. In this paper authors do not concentrated on heterogeneity problems.

In the work[5] author tells An alternative technique is to use artificially created data, which provides advantages that content and error rates can be controlled, and the deduplication or linkage status is known.

In the work [2] Author proposes a technique, which is called "Traditional Blocking". In this the number of possible comparisons increases with the file size, this can make it unwieldy the files are large, such as in record linkage. Comparisons were therefore restricted to comparisons of "blocks" or "Packets" of cases where one or more variables matched exactly. This process is referred to as "blocking" and it minimize the comparisons that must be undertaken at a given time.

In the Work [3] Author said "The simpler approaches, like traditional blocking is the overall fastest techniques. Among the other fast techniques are the robust suffix array and adaptive sorted neighborhood approaches."They also providing indexing techniques for better record linkage.

In the work [4] Author describes a new machine learning approach that creates expert-like rules for field matching. In this approach, the relationship between two field values is described by a set of heterogeneous transformations. Previous machine learning methods used simple models to evaluate the distance between two fields. However, this approach enables more sophisticated relationships to be modeled, which better capture the complex domain specific, common-sense phenomena that humans use to judge similarity. We compare our approach to methods that rely on simpler homogeneous models in several domains. By modeling more complex relationships we produce more accurate results.

III. Proposed Model

In this section, the authors [1] introduces a competent solution to the online, distributed environment record linkage problem. The main advantage of the sequential approach is that, unlike the usual full-information case, not all the attributes of all the remote records are taken to the local site; instead, attributes are taken one at a time. After retrieving an attribute, the matching possibility is revised based on the realization of that attribute, and a decision is made whether or not to retrieve more attributes from the remote site.

A. Concurrent Attribute Acquisition

The main drawback of the sequential schemes find out by authors [1] (SAA and SIA) is that the related information to the remote records must be transferred back and forward between the Local and Remote sources; by this way the resulting transparency could be huge, particularly when the number of remote records is large. When we consider the latency-related [1] delays as well, this backward and forward nature communication may make them particularly inappropriate in many situations. To completely eliminate the overhead that occurs in a recursive partitioning scheme embedded in SAA or SIA authors [1] introduce new approach, Concurrent Attribute Acquisition (CAA). In this CAA, we make a database query which is posed to the remote database to retrieve only the relevant records.

Let us consider the vector $V = (V_1, V2, V3 \ldots V_k)$ certain realizations of this vector leads to a matching probability greater than α; call these the favorable realizations. Our main objective is to retrieve only the remote records that are nearer to the favorable realizations of V. We intend to use the tree effectively to identify these realizations. Any path in the tree that have a "STOP" node with a matching probability greater than α, which provides a favorable realization is called an acceptance path[1]. Such a path can be expressed as a conjunctive condition.

Since various paths have different favorable realizations, the overall query condition should be a disjunction of all the acceptance paths starting at the root. For the situation, if there are m acceptance paths out of the root, and if e_j denotes the condition of path j, j=1, 2, 3…m then the overall query condition can be written as: $e_1 \vee e_{2 \ldots} e_m$. This query condition, however, is quite complex and can be compressed further.

In order to explain how this can be done, we denote E(z) as the complete query condition rooted at node z. Because of the completeness property of the matching tree every relevant record (a record with matching probability above α) must satisfy E (z), and every irrelevant record (a record with matching probability below α) must satisfy ¬E(z), the negation of E(z). Let l(z) and r(z) be the left and right children of node x, respectively. Denoting the attribute at node z as Y(z), E(z) can be expressed as a recursion:

$E(z) \equiv ((Y(z) = a(Y(z))) \Lambda E(l(z))) V ((Y(z) \neq a(Y(z))) \Lambda E(r(z)))$.

Assume, without loss of generality, that a match on Y(z) is a favorable realization. Now consider the revised query Condition.

$E'(z) \equiv ((Y(z) = a(Y(z))) \Lambda E(l(z))) V E(r(z))$.
Clearly, $E(z) => E'(z)$,

so a relevant record would not be excluded from consideration if E(z) is replaced by E'(z); the question is whether irrelevant records would be erroneously included as a result of this replacement. Suppose that b ε R does not satisfy E(z), i.e., b is irrelevant, but b satisfies E'(z). Therefore, b must satisfy

$E'(z) \Lambda \neg E(z) \equiv (Y(z) = a(Y(z))) \Lambda E(r(z)) \Lambda \neg E(l(z)) => (Y(z) = a(Y(z) = a(Y(z))) \Lambda E(r(z))$.

Since we assumed that a match on Y(z) is a favorable realization, the matching probability of the above condition must be greater than the matching probability associated with the condition $(Y(z) \neq a(Y(z))) \Lambda E(r(z))$. However, the latter condition corresponds to an acceptance path in the tree and has a matching probability greater than α, so b has a matching probability greater than α. This is a contradiction to the assumption that b is irrelevant. Therefore, by rewriting the expression of E(z) as E'(z) and using it recursively starting at the root, we can ensure that each node is included in the query exactly once, thereby reducing the size of the query significantly. The size of the compressed query is, therefore, the size of tree and is equal to $\sum_{z \varepsilon Z} s(z)$.

In this case, there is no identifier overhead, and the included record overhead is still $np|_{\theta} sR$. Therefore, the normalized overhead in this case is given by

Normalized Total Overhead = $+ p|_{\theta}$

Using this CAA approach the communication overhead is reduced .In following section we apply some blocking techniques along with CAA.

B. Blocking and UGK

With the probabilistic linkage approach, the number of possible comparisons increases with the file size. This can make it unwieldy when the files are large, such as in this project. Comparisons were therefore restricted to comparisons of "blocks" or packet "pocket"[3] of cases where one or more variables matched exactly. This process is referred to as "blocking"[2][3][4][6] and simply stratifies the linkage process to minimize the number of comparisons that must be undertaken at a given time. Multiple passes through the data were used for each separable blocking variable.

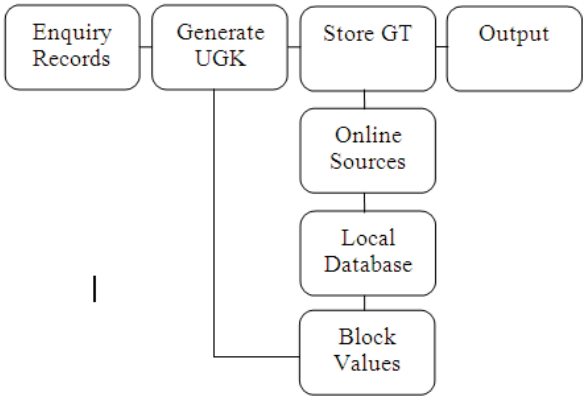

Fig.1 System Architecture

The record linkage approach described here assumes that we are linking records in two database tables, X and Y, such that there are corresponding attributes in each table, That is, the i th column in each table contains elements of the same type. In many applications, there are additional complexities; for instance, one table might have two attributes, such as "first name" and "last name", and the other table might have attributes such as "full name". These complexities can generally be handled in a pre-processing phase [3] (e.g., concatenating "first name" and "last name").

Our record linkage process has several phases [3]. First, we parse each cell in each record into a set of tokens [3], where each token is an individual word, number, or symbol. Optionally, we also label the tokens with a semantic category (e.g., parsing a full name into first name, optional middle initial, and last name), and also optionally apply a set of normalization operators to standardize the tokens. Second, we

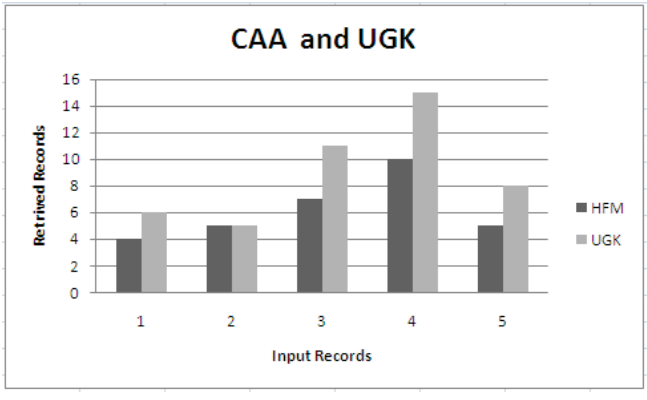

Fig 2.Comparision Chart

use a blocking algorithm [3][4][11] to identify pairs of records that have the potential to match. This eliminates the need to evaluate the entire cross product. In our implementation, the key to our approach is the use of transformations to relate two values. Transformations that we use for string values include: Equal, Synonym, Misspelling, Abbreviation, Prefix, Acronym, Concatenation, Suffix, Soundex and Missing.

This Fig.1 shows the process of our technique. In this approach User Generated Key (UGK) is generated from enquiry record. For example enquiry record's first name is Amuthan, Last name Laxman, Data of birth 04-09-1990, born city Coimbatore, Postal Code 641 023.System will generate Key using this value AMLA04091990641023. If key value already exists in the Global Table (GT) the system produce the output. GT contains original source information with their identity value. If the key value is not available in GT, System finds the related records from the online sources. In our system the records are retrieved using CAA which is explained deeply in the section 3.1 provided by author [1] and the values are stored into the local temporary database. System retrieves only the related record's attributes and its values, even though it contains 50 attributes it only retrieves necessary attributes which are useful for blocking. If data base contains 100000 related records with 50 attribute values, retrieving those records to local site is totally time consuming and requires large space for storage.

With the use of temporary table records, the system finds the similar values and generates the UGK using our blocking techniques [3][4]. In some instances record value may be empty or totally unmatched. In such cases the system matches the entity with another attribute of the record. If one record's first name is spelled wrongly but its DOB, born city values are correct. For finding correct first name,we match this two attribute values with another table's attribute values. This Key value is stored in GT with their original data source information like data base name, table name, identity field and its value. This process can be repeated for other databases. Finally, the output can be generated through this GT information.

IV. Conclution

An efficient record linkage technique is developed using above approach reduces the time and communication overhead. It also reduces complexity by retrieving filtered attributes instead of retrieving all attribute values from remote to local site. The accuracy of the record can be efficiently increased using our blocking techniques and different comparison techniques.

With the use of clustering technique while retrieval of relevant information from the remote databases, time consuming can be reduced, which is to be done as future work. In addition, indexing, while retrieving records from online databases further increases the efficiency in terms of time .

References

[1] Debabrata Dey, Member, IEEE, Vijay S. Mookerjee, and Dengpan Liu (2011), 'Efficient Techniques For Online Record Linkage', I*EEE Transactions On Knowledge And Data Engineering*, Vol. 23, No. 3, March 2011

[2] Steven N. Minton and Claude Nanjo,Craig A. Knoblock, Martin Michalowski, and Matthew Michelson 'A Heterogeneous Field Matching Method For Record Linkage' in part by the Air Force Office of Scientific Research under grant number FA9550-04-1-0105

[3] Peter Christen, The Australian National University, "A Survey Of Indexing Techniques For Scalable Record Linkage And Deduplication", *IEEE Transactions On Knowledge And Data Engineering,* Vol. Z, No. Y, Zzzz 2011 2

[4] William E. Winkler, U.S. Bureau of the Census "Matching And Record Linkage".

[5] Peter Christen,"Probabilistic Data Generation For Deduplication And Data Linkage" Australian Research Council (ARC) Linkage Grant LP0453463 and partially funded by the NSW Department of Health

[6] Peter Christen, Tim Churches "Secure Health Data Linkage And Geocoding: Current Approaches And Research Directions" Australian Research Council (ARC) Linkage Grant LP0453463.

[7] Matthew Michelson, Craig A. Knoblock "Mining Heterogeneous Transformations For Record Linkage" Air Force Office of Scientific Research under grant number FA9550-04-1-0105

[8] Liang Jin, Chen Li, Sharad Mehrotra , University of California, Irvine, CA 92697, USA "Efficient Record Linkage In Large Data Sets"

[9] Peter Christen August 2007 TR-CS-07-03 "Towards Parameter-Free Blocking For Scalable Record Linkage"

[10] Soufiane Boufous, Caroline Finch, Andrew Hayen, Ann Williamson "Data Linkage Of Hospital And Police Crash Datasets In Nsw" NSW Injury Risk Management Research Centre University of New South Wales, Sydney NSW 2052, Australia.

[11] Peter Christen, Tim Churches "Febrl – Freely extensible biomedical record linkage" Australian National University.

[12] Lifang Gu,Rohan Baxter,Deanne Vickers, Chris Rainsford " Record Linkage: Current Practice and Future Directions "CSIRO Mathematical and Information Sciences GPO Box 664, Canberra, ACT 2601, Australia,CMIS Technical Report No. 03/83.