International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

Handling Duplicate Data in Big Data

[1]Jony Kumar, [2]Mrs.Mamta Yadav
[1]M. Tech Scholar CSE, M.D.U Rohtak,YCET Narnaul, Mahendergarh, India
[2]Assistant Professor CSE, M.D.U Rohtak,YCET Narnaul, Mahendergarh, India

ABSTRACT

The problem of detecting and eliminating duplicated data is one of the major problems in the broad area of data cleaning and data quality in data warehouse. Many times, the same logical real world entity may have multiple representations in the data warehouse. Duplicate elimination is hard because it is caused by several types of errors like typographical errors, and different representations of the same logical value. Also, it is important to detect and clean equivalence errors because an equivalence error may result in several duplicate tuples. Recent research efforts have focused on the issue of duplicate elimination in data warehouses. This entails trying to match inexact duplicate records, which are records that refer to the same real-world entity while not being syntactically equivalent. This paper mainly focuses on efficient detection and elimination of duplicate data. The main objective of this research work is to detect exact and inexact duplicates by using duplicate detection and elimination rules. This approach is used to improve the efficiency of the data. The importance of data accuracy and quality has increased with the explosion of data size. This factor is crucial to ensure the success of any cross enterprise integration applications, business intelligence or data mining solutions.

Keywords : Cross Enterprise Integration, Duplicate Elimination, Semantic Entity, Big Data

I. INTRODUCTION

In the duplicate elimination step, only one copy of exact duplicated records are retained and eliminated other duplicate records. The elimination process is very important to produce a cleaned data. Before the elimination process, the similarity threshold values are calculated for all the records which are available in the data set. The similarity threshold values are important for the elimination process. In the elimination process, select all possible pairs from each cluster and compare records within the cluster using the selected attributes. Most of the elimination processes compare records within the cluster only. Sometimes other clusters may have duplicate records, same value as of other clusters. This approach can substantially reduce the probability of false mismatches, with a relatively small increase in the running time. The following procedures are used to identify or detect duplicates and eliminate duplicates. Detecting duplicate data that represent the same real world object more than once in a certain dataset is the first step to ensure the data accuracy. This operation becomes more complicated when the same object name (person, city) is represented in multiple natural languages due to several factors including spelling, typographical and pronunciation variation, dialects and special vowel and consonant distinction and other linguistic characteristics. Therefore, it is difficult to decide whethe or not two syntactic values (names) are alternative designation of the same semantic entity. Up to authors' knowledge, the previously proposed duplicate record detection (DRD) algorithms and frameworks support only single language duplicate record detection, or at most bilingual. In this paper, two available tools of duplicate record detection are compared. Then, a

generic cross language based duplicate record detection solution architecture is proposed, designed and implemented to support the wide range variations of several languages. The proposed system design uses a dictionary based on phonetic algorithms and support different indexing/blocking techniques to allow fast processing. The framework proposes the use of several proximity matching algorithms, performance evaluation metrics and classifiers to suit the diversity in several languages names matching. The framework is implemented and verified empirically in several case studies. Several Experiments are executed to compare the advantages and disadvantages of the proposed system compared to other tool. Results showed that the proposed system has substantial improvements compared to the well-known tools. Duplicate record detection is the process of identifying different or multiple records that refer to one unique real world entity or object if their similarity exceeds a certain cutoff value. However, the records consist of multiple fields, making the duplicate detection problem much more complicated . A rule-based approach is proposed for the duplicate detection problem. This rule is developed with the extra restriction to obtain good result of the rules. These rules specify the conditions and criteria for two records to be classified as duplicates. A general if then else rule is used in this research work for the duplicate data identification and duplicate data elimination. A rule will generally be of the form:

if <condition >

then <action >

The action part of the rule is activated or fired when the conditions are satisfied. The complex predicates and external function references may be contained in both the condition and action parts of the rule. In existing duplicate detection and elimination method, the rules are defined for the specific subject data set only. These rules are not applicable for another subject data set. Anyone with subject matter expertise can be able to understand the business logic of the data and can develop the appropriate conditions and actions, which will then form the rule set. In this research work, therules are formed automatically based on the specific criteria and formed rules are applicable for any subject dataset. In duplicate data detection rule, threshold values of record pairs and certainty factors are very important.

II. Duplicate Record Detection

Big data practitioners consistently report that 80% of the effort involved in dealing with data is cleaning it up in the first place. Duplicate record detection is one of the data cleaning processes. It is the process of identifying records that have multiple representations of the same real-world objects. Sometimes duplicate records are caused by misspelling during data entry, in other cases the duplicated records are resulted from a database integration process. Hence real-world data collections are exposed to be noisy, contaminated, incomplete and incorrectly formatted while being saved in database, data cleaning and standardization is a crucial preprocessing stage. In a data cleaning and standardization step, data is unified, normalized and standardized to be converted into a well-defined form. This step is done because original data may be recorded or captured in various, possibly obsolete formats.

Phonetic Name Matching Algorithms

There are several phonetic name matching algorithms including the popular Russell Soundex (Russell, 1918, 1922) and Metaphone algorithms that are designed for use with English names. The ambiguity of the Metaphone algorithm in some words limited its use. The Henry Code is adapted for the French language while the Daitch-Mokotoff Coding method is adapted for Slavic and German spellings of Jewish names. The Arabic version of the

Soundex algorithm is found in (Aqeel, 2006) and modified in (Koujan, 2008). Its approach is to use Soundex of conflating similar sounding consonants. However a special version of soundex for Arabic person names is proposed in (Yousef, 2013). This enhanced Arabic Combined Soundex Algorithm solved the limitation of the standard soundex algorithm with Arabic names that are composed of more than one word (syllable) like (Abdel Aziz, Abdel Rahman, Aboul Hassan, Essam El Din). The quality of record linking techniques can be measured using the confusion matrix as discussed in (Christen and Goiser, 2007). The confusion matrix compares actual matched (M) and non-matched (U) records (according to the domain expert) to the machine matched (M') and non-matched records (U'). Well known measures include true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP). The measurement of accuracy, precision and recall are usually expressed as a percentage or proportion as follows (Christen and Goiser, 2007):

Accuracy = (TP+ TN) / (TP+FP+TN+FN). (2)

Precision = TP / (TP+FP) (3)

Recall = TP / (TP+FN) (4)

Because the number of negatives TN is very large compared to the number of records in the comparison space, it is widely accepted that quality measures that Cross Language Duplicate Record Detection in Big Data 155 depend on TN (like accuracy) will not be very useful because TN will dominate the formula (Christen and Goiser, 2007).

Remove of duplicate rules from multilevel data Algorithm:-

A multilevel dataset is one which has an implicit taxonomy or concept tree, like the exampleshown in Fig. 1. The items in the dataset exist at the lowest concept level but are part of ahierarchical structure and organization. Thus for example, 'ME' is an item at thelowest level of the taxonomy but it also belongs to the high level concept category of 'Science' and also the more refined category 'Engg'.Each entry in the hierarchy has oneparent (or immediate supertopic) with a path back to the root possible from any where in thehierarchy. The hierarchy information can be encoded with each topic allowing information abouta given topic's ancestry. For example, 'ME' can be encoded as 1_1_2. This first digit in the sequence '1' indicates that it belongs to first category in the first level concept. The second secquence digit '1' indicates that in belongs to first category in the second level concept under belonging category with one level upper. The third digit '2' in sequence ponts to second category in the third level concept under the category from above one level from current level and so on. As per the assumption made the order of the siblings in the this taxonomy is not so important. Thus in this structure the node 'ME'is encoded as 1_1_2 but if made to the first node under the 'Engg' then it would be encoded as 1_1_* and the node 'CSE ' would then be encoded as 1_1_1 . the encoding is done in a simple left to right manner, because of the tree nature of the multi-level dataset a defferent approach to finding frequent itemsets is needed as standard Apriori approach does not take the tree structure in to consideration.

1. recovered € Ø
2. for all r R exactbasis
3. candidatebasis rules ä Ø
4. determine if any of the items x in the antecedent X of rule r: X => Y are the ancestor of any generator g in the list of generators G and if so store g in list A
5. determine all of the possible subsets of list A and store as S
6. for all s RS check to ensure every x RX for rule r has a descendant in s and if not add x to s so that s R x
7. if s has no ancestors in Y & s has no descendants in Y & for all items iRs there are no ancestor-descendant relations with item i' Rs & for all items i

RY there are no ancestor- descendant relations with item i'RY

8. insert {r : s => Y} in candidatebasis rules

9. end loop

10. if for all x R X test to see that they have a descendant item i R A and if not add x to A

11. if A has no ancestors in Y & A has no descendants in Y & for all i RA there are no ancestor-descendant relations with item i RA & for all items iRY there are no ancestor-descendant relations with item i' R Y

12. insert {r : A => Y} in candidatebasis

13. for all c : B => D "R candidate basis rules

14. if B , D ? itemset i R closed itemset list C & B ? g R Gi

15. insert {r : B => D, g.supp} in recovered

16. end loop

17. end loop

18. return exactbasis , recoverer

The DUPOUT= Option

III. REFERENCES

1. Radu-Ioan,Ciobanu,Valentin Cristea, Ciprian Dobre and Florin Pop, Big Data Platforms for the Internet of Things,2014,Springer
2. Flavio Bonomi, Rodolfo Milito, Preethi Natarajan and Jiang Zhu,Fog Computing: A Platform for Internet of Things and Analytics, springer (2014)
3. Shintaro Yamamoto, Shinsuke Matsumoto,Sachio Saiki, and Masahide Nakamura Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo 657-8501, Japan,Using Materialized View as a Service of Scallop4SC for Smart City Application Services (2014)
4. Mukherjee, A.; Datta, J.; Jorapur, R.; Singhvi, R.; Haloi, S.; Akram, W. "Shared disk big data analytics with Apache Hadoop" (18-22 Dec. 2012)
5. Kudakwashe Zvarevashe1, Dr. A Vinaya Babu, Towards MapReduce Performance Optimization: A Look into the Optimization Techniques in Apache Hadoopfor BigData Analytics (2014)
6. Gartner: Hype cycle for big data, 2012. Technical report (2012)
7. IBM, Zikopoulos, P., Eaton, C.:Understanding BigData: Analytics for Enterprise Class Hadoop and Streaming Data. 1st edn. McGraw-Hill Osborne Media,New York (2011)
8. Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., Tufano, P.: Analytics: The realworld use of big data. IBM Institute for Business Value—executive report, IBM Institute for Business Value (2012)
9. Evans, D.: The internet of things—howthe next evolution of the internet is changing everything. Technical report (2011)
10. Cattell, R.: Scalable sql and nosql data stores. Technical report (2012)
11. Apache: Hadoop (2014) (Online 20 Oct 2015)
12. Jo Foley, M.: Microsoft drops dryad; puts its big-data bets on hadoop. Technical report (2011)
13. Locatelli, O.: Extending nosql to handle relations in a scalable way models and evaluation framework (2012012)
14. Robinson, I., Webber, J., Eifrem, E.: Graph Databases. O'Reilly Media, Incorporated (2013)
15. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels,W.: Dynamo: amazon's highlyavailable key-value store. SIGOPS Oper. Syst. Rev. 41, 205–220 (2007) Big Data Management Systems for the Exploitation 89
16. Riak: Riak (Online Oct 2015)
17. Apache: Couchdb (Online; Oct 2015)
18. MongoDB: Mongodb (Online; Oct 2015)
19. Hypertable: Hypertable (Online; Oct 2015)
20. Rabl, T., Gómez-Villamor, S., Sadoghi, M., Muntés-Mulero, V., Jacobsen, H.A., Mankovskii, S.: Solving big data challenges for enterprise application performance

management. Proc. VLDB Endow. 5, 1724–1735 (2012)

21. Neo Technology, I.: Neo4j, the world's leading graph database. (Online;Oct 2015)
22. Amato, A., DiMartino, B., Venticinque, S.: Semantically augmented exploitation of pervasive environments by intelligent agents. In: ISPA, pp. 807–814.(2012)
23. Jing Zhang, "A Distributed Cache for Hadoop File Distribution system in Real time Cloud Services ", 2012 ACM/IEEE 13th International Conference on Grid Computing.
24. Pig.apachi.org (online Oct 2015).