© 2015 IJSRSET | Volume 1 | Issue 5 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Detecting Duplicate Records - A Case Study

T. Parimalam, R. Deepa, R. Nirmala Devi, P. Yamuna Devi
Department of Computer Science, Nandha Arts and Science College, Erode, Tamil Nadu, India

ABSTRACT

Databases play an important role in today's IT based economy. Many industries and systems depend on the accuracy of databases to carry out operations. Therefore, the quality of the information stored in the databases, can have significant cost implications to a system that relies on information to function and conduct business. Often, in the real world, entities have two or more representations in databases. Duplicate detection is the process of identifying multiple representations of same real world entities. The purpose of this paper is to provide a thorough study on different methods used for detecting duplicate records. And also this paper discussed about the different duplication detection tools in detail.

Keywords: Database, Duplicate Detection, Records

I. INTRODUCTION

Data quality has become a key issue in computer-based management systems. Inadequate data causes serious operational difficulties as well as direct financial losses. Operational databases store information generated by business transactions, and this information is used by management to support business decisions. Data accuracy assurance is vital, as data is the cornerstone of a company's business operations. In addition to serious implications on decision making, the quality of the data may affect customer satisfaction, resulting in unnecessary and possibly high costs to repair damage caused by low-quality data. In an ideal situation, each data item should have a global or unique identifier, allowing these records to be identified, linked, and related across tables. Unfortunately, this is not the case in real-life, complex databases. Many organizations have multiple data collection systems (e.g. Oracle, legacy systems), and these may differ not only in values or identifiers, but also in format, structure, and schema of databases. Additionally, data quality is affected by human error, such as data entry errors, and lack of constraints.

When data is entered manually or gathered from different sources, whether from different systems or different locations, duplicate records may result. Describe duplicate records as "all cases of multiple representations of same real-world objects, i.e., duplicates in a data source". Heterogeneous data often lacks a global identifier, or a primary key, which would uniquely identify real-world objects.

II. METHODS AND MATERIAL

A. Data Preparation

Duplicate record detection is the process of identifying different or multiple records that refer to one unique real-world entity or object. Typically, the process of duplicate detection is preceded by a data preparation stage, during which data entries are stored in a uniform manner in the database, resolving (at least partially) the structural heterogeneity problem.

The data preparation stage includes the following steps.

i. Parsing

It locates, identifies and isolates individual data elements in the source files. Parsing makes it easier to correct, standardize, and match data because it allows the comparison of individual components, rather than of long complex strings of data.

ii. Data transformation

It refers to simple conversions that can be applied to the data in order for them to conform to the data types of their corresponding domains. This type of conversion focuses on manipulating one field at a time, without taking into account the values in related fields. The most common form of a simple transformation is the conversion of a data element from one data type to another.

iii. Data standardization

It refers to the process of standardizing the information represented in certain fields to a specific content format. This is used for information that can be stored in many different ways in various data sources and must be converted to a uniform representation before the duplicate detection process starts. Without standardization, many duplicate entries could erroneously be designated as non-duplicates, based on the fact that common identifying information cannot be compared. One of the most common standardization applications involves address information. There is no one standardized way to capture addresses so the same address can be represented in many different ways. Address standardization locates (using various parsing techniques) components such as house numbers, street names, post office boxes, apartment numbers and rural routes, which are then recorded in the database using a standardized format.

Even after parsing, data standardization, and identification of similar fields, it is not trivial to match duplicate records. Misspellings and different conventions for recording the same information still result in different, multiple representations of a unique object in the database.

B. **Detecting Duplicate Records**

i. Matching Records with Individual Fields

One of the most common sources of mismatches in database entries is the typographical variations of string data. Therefore, duplicate detection typically relies on string comparison techniques to deal with typographical variations.

- Character-based similarity metrics
- Token-based similarity metrics
- Phonetic similarity metrics
- Numeric Similarity Metrics

While multiple methods exist for detecting similarities of string-based data, the methods for capturing similarities in numeric data are rather primitive. Typically, the numbers are treated as strings (and compared using the metrics described above) or simple range queries, which locate numbers with similar values.

ii. Matching Records with Multiple Fields

In most real-life situations, however, the records consist of multiple fields, making the duplicate detection problem much more complicated. In this section, we review methods that are used for matching records with multiple fields. The presented methods can be broadly divided into two categories:

- Probabilistic approaches and supervised machine learning techniques.
- Approaches that rely on domain knowledge or on generic distance metrics to match records. This category includes approaches that use declarative languages for matching, and approaches that devise distance metrics appropriate for the duplicate detection task.

1). Probabilistic Matching Models: Newcombe et al.[1] were the first to recognize duplicate detection as a Bayesian inference problem. Then, Fellegi and Sunter formalized the intuition of Newcombe et al. and introduced the notation that we use, which is also commonly used in duplicate detection literature. The comparison vector $\underline{x}$is the input to a decision rule that assigns $\underline{x}$ to Uor to M. The main assumption is that $\underline{x}$is a random vector whose density function is different for each of the two classes. Then, if the density function for each class is known, the duplicate detection problem becomes a Bayesian inference problem.

2). **Supervised and Semi-Supervised Learning**: The probabilistic model uses a Bayesian approach to classify record pairs into two classes, M and U. This model was widely used for duplicate detection tasks, usually as an application of the Fellegi-Sunter model. While the Fellegi-Sunter approach dominated the field for more than two decades, the development of new classification techniques in the machine learning and statistics communities prompted the development of new deduplication techniques. The supervised learning systems rely on the existence of training data in the form of record pairs, pre-labeled as matching or not.

One set of supervised learning techniques treat each record pair $\langle\alpha,\beta\rangle$ independently, similarly to the probabilistic techniques of probabilistic matching models. Cochinwala et al.[2] used the well-known CART algorithm, which generates classification and regression trees, a linear discriminant algorithm, which generates linear combination of the parameters for separating the data according to their classes, and a "vector quantization" approach, which is a generalization of nearest neighbor algorithms. The experiments which were conducted indicate that CART has the smallest error percentage. Bilenko et al.[3] use SVM light to learn how to merge the matching results for the individual fields of the records. Bilenko et al. showed that the SVM approach usually outperforms simpler approaches, such as treating the whole record as one large field. A typical post-processing step for these techniques (including the probabilistic techniques of probabilistic matching models is to construct a graph for all the records in the database, linking together the matching records. Then, using the transitivity assumption, all the records that belong to the same connected component are considered identical.

The transitivity assumption can sometimes result in inconsistent decisions. For example, $\langle\alpha,\beta\rangle$and $\langle\alpha,\gamma\rangle$can be considered matches, but $\langle\beta,\gamma\rangle$ not. Partitioning such "inconsistent" graphs with the goal of minimizing inconsistencies is an NP-complete problem. Bansal et al.[4] propose a polynomial approximation algorithm that can partition such a graph, identifying automatically the clusters and the number of clusters in the dataset. Cohen [5] proposed a supervised approach in which the system learns from training data how to cluster together records that refer to the same real-world entry. The main contribution of this approach is the adaptive distance function which is learned from a given set of training examples. McCallum and Wellner learn the clustering method using training data; their technique is equivalent to a graph partitioning technique that tries to find the min-cut and the appropriate number of clusters for the given data set, similarly to the work of Bansal [4] et al..

3).Active-Learning-Based Techniques: One of the problems with the supervised learning techniques is the requirement for a large number of training examples. While it is easy to create a large number of training pairs that are either clearly non-duplicates or clearly duplicates, it is very difficult to generate ambiguous cases that would help create a highly accurate classifier. Based on this observation, some duplicate detection systems used active learning techniques to automatically locate such ambiguous pairs. Unlike an "ordinary" learner that is trained using a static training set, an "active" learner actively picks subsets of instances from unlabeled data, which, when labeled, will provide the highest information gain to the learner.

Sarawagi [6] designed ALIAS, a learning based duplicate detection system, that uses the idea of a "reject region" (see Reject region) to significantly reduce the size of the training set. The main idea behind ALIAS is that most duplicate and non-duplicate pairs are clearly distinct. For such pairs, the system can automatically categorize them in $U$and Mwithout the need of manual labeling. ALIAS requires humans to label pairs only for cases where the uncertainty is high. This is similar to the "reject region" in the Fellegi [7]model, which marked ambiguous cases as cases for clerical review. Tejada et al. used a similar strategy and employed decision trees to teach rules for matching records with multiple fields. Their method suggested that by creating multiple classifiers, trained using slightly different data or parameters, it is possible to detect ambiguous cases and then ask the user for feedback. The key innovation in this work is the creation of several redundant functions and the concurrent exploitation of their conflicting actions in order to discover new kinds of inconsistencies among duplicates in the data set.

4). Distance-Based Techniques: Even active learning techniques require some training data or some human effort to create the matching models. In the absence of such training data or ability to get human input, supervised and active learning techniques are not appropriate. One way of avoiding the need for training data is to define a distance metric for records, which does not need tuning through training data. Using the distance metric and an appropriate matching threshold, it is possible to match similar records, without the need for training.

One approach is to treat a record as a long field, and use one of the distance metrics described in field matching to determine which records are similar. Monge [8] proposed a string matching algorithm for detecting highly similar database records. The basic idea was to apply a general purpose field matching algorithm, especially one that is able to account for gaps in the strings, to play the role of the duplicate detection algorithm.

Distance-based approaches that conflate each record in one big field may ignore important information that can be used for duplicate detection. A simple approach is to measure the distance between individual fields, using the appropriate distance metric for each field, and then compute the weighted distance between the records. In this case, the problem is the computation of the weights, and the overall setting becomes very similar to the probabilistic setting that we discussed in probabilistic matching models.

5). Rule-based Approaches: A special case of distance-based approaches is the use of rules to define whether two records are the same or not. Rule-based approaches can be considered as distance-based techniques, where the distance of two records is either 0 or 1. Wang and Madnick[9] proposed a rule-based approach for the duplicate detection problem. For cases in which there is no global key, Wang and Madnick suggest the use of rules developed by experts to derive a set of attributes that collectively serve as a "key" for each record. For example, an expert might define rules such as

IF age< 22 THEN status = undergraduate ELSE status = graduate

IF distanceFromHome > 10 THEN transportation = car ELSE transportation = bicycle

By using such rules, Wang and Madnick hoped to generate unique keys that can cluster multiple records that represent the same real-world entity.

6). Unsupervised Learning: As we mentioned earlier, the comparison space consists of comparison vectors which contain information about the differences between fields in a pair of records. Unless some information exists about which comparison vectors correspond to which category (match, non-match, or possible-match), the labeling of the comparison vectors in the training data set should be done manually. One way to avoid manual labeling of the comparison vectors is to use clustering algorithms, and group together similar comparison vectors. The idea behind most unsupervised learning approaches for duplicate detection is that similar comparison vectors correspond to the same class.

The idea of unsupervised learning for duplicate detection has its roots in the probabilistic model proposed by Fellegi and Sunter (see probabilistic matching models). As we discussed in probabilistic matching models, when there are no training data to compute the probability estimates, it is possible to use variations of the Expectation Maximization algorithm to identify appropriate clusters in the data.

Ravikumar and Cohen[11] follow a similar approach and propose a hierarchical, graphical model for learning to match record pairs. The foundation of this approach is to model each field of the comparison vector as a latent binary variable which shows whether the two fields match or not. The latent variable then defines two probability distributions for the values of the corresponding "observed" comparison variable. Ravikumar and Cohen show that it is easier to learn the parameters of a hierarchical model than to attempt to directly model the distributions of the real-valued comparison vectors.

C. Duplicate Detection Tools

i. FEBRL SYSTEM (Freely Extensible Biomedical Record Linkage)

It is an open-source data cleaning toolkit, and it has two main components: The first component deals with data standardization and the second performs the actual duplicate detection. The data standardization relies mainly on hidden-Markov models (HMMs); therefore, Febrl typically requires training to correctly parse the database entries. For duplicate detection, Febrl implements a variety of string similarity metrics, such as

Jaro, edit distance, and q-gram distance. Febrl supports phonetic encoding (Soundex, NYSIIS, and Double Metaphone) to detect similar names.

ii. Tailor

It is a flexible record matching toolbox, which allows the users to apply different duplicate detection methods on the data sets. The flexibility of using multiple models is useful when the users do not know which duplicate detection model will perform most effectively on their particular data. TAILOR follows a layered design, separating comparison functions from the duplicate detection logic. Furthermore, the execution strategies, which improve the efficiency, are implemented in a separate layer, making the system more extensible than systems that rely on monolithic designs. TAILOR reports statistics, such as estimated accuracy and completeness, which can help the users understand better the quality of the given duplicate detection execution over a new data set.

iii. Whirl

It is a duplicate record detection system available for free for academic and research use. WHIRL uses the tf.idf token-based similarity metric to identify similar strings within two lists. The Flamingo Project is a similar tools that provides a simple string matching tool that takes as input two string lists and returns the strings pairs that are within a prespecified edit distance threshold. WizSame by WizSoft is also a product that allows the discovery of duplicate records in a database.

iv. BIGMATCH

It is the duplicate detection program used by the U.S. Census Bureau. It relies on blocking strategies to identify potential matches between the records of two relations, and scales well for very large data sets. The only requirement is that one of the two relations should fit in memory, and it is possible to fit in memory even relations with 100 million records. The main goal of BigMatch is not to perform sophisticated duplicate detection, but rather to generate a set of candidate pairs that should be then processed by more sophisticated duplicate detection algorithms.

III. CONCLUSION

As database systems are becoming more and more commonplace, data cleaning is going to be the cornerstone for correcting errors in systems which are accumulating vast amounts of errors on a daily basis. Despite the breadth and depth of the presented techniques, we believe that there is still room for substantial improvements in the current state-of-the-art. Data preparation, detecting duplicate records and duplicate detection tools were discussed in this paper. Finally, large amounts of structured information are now derived from unstructured text and from the web. This information is typically imprecise and noisy; duplicate record detection techniques are crucial for improving the quality of the extracted data. The increasing popularity of information extraction techniques is going to make this issue more prevalent in the future, highlighting the need to develop robust and scalable solutions. This only adds to the sentiment that more research is needed in the area of duplicate record detection and in the area of data cleaning and information quality in general.

IV. REFERENCES

[1] Newcombe, Howard B. James M. Kennedy and S.J. Axford and A.P. James (1959). "Automatic Linkage of Vital Records". Science 130 (3381): 954-959.

[2] Cochinwala, Munir; Verghese Kurien and Gail Lalk and Dennis Shasha (2001). "Efficient data reconciliation". Information Sciences 137 (1-4): 1-15.

[3] Bilenko, Mikhail; Raymond J. Mooney and William Weston Cohen and Pradeep Ravikumar and Stephen E. Fienberg (2003). "Adaptive Name Matching in Information Integration". IEEE Intelligent Systems 18 (5): 16-23.

[4] Bansal, Nikhil; Avrim Blum and Shuchi Chawla (2004). "Correlation Clustering". Machine Learning 56 (1-3): 89–113.

[5] Cohen, William Weston; Jacob Richman (2002). "Learning to Match and Cluster Large High-Dimensional Data Sets For Data Integration". Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002).

[6] Sarawagi, Sunita; Anuradha Bhamidipaty (2002). "Interactive Deduplication using Active Learning". Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). pp. 269-278.

[7] Fellegi, Ivan Peter; Alan B. Sunter (1969). "A theory for record linkage". Journal of the American Statistical Association 64 (328): 1183-1210.

[8] Monge, Alvaro E.; Charles P. Elkan (1996). "The Field Matching Problem: Algorithms and Applications". Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 267-270.

[9] Wang, Y. Richard; Stuart E. Madnick (1989). "The Inter-Database Instance Identification Problem in Integrating Autonomous Systems". Proceedings of the Fifth IEEE International Conference on Data Engineering (ICDE 1989). pp. 46-55.