Domain Dependent and Independent Data Cleansing Techniques

Dr. Payal Pahwa, Rashmi Chhabra

Bhagwan Parshuram Institute of Technology, I.P.University, Delhi, India
Research Scholar, CSE Department, NIMS University, Jaipur (Rajasthan)

Abstract
Data warehousing is emerging as the cornerstone of an organization’s information infrastructure. Today every business organization needs accurate and large amount of information to make proper decisions. For taking the business decisions the data should be of good quality. To improve the data quality data cleansing is needed. Data cleansing is fundamental to warehouse data reliability, and to data warehousing success. There are various methods for data cleansing. We classify the methods in two categories domain dependent and domain independent. This paper presents a survey and review of data cleansing methods, classification of existing methods and comparison between them.

Keywords
data cleansing, data warehouse, data cleansing techniques, domain dependent, domain independent.

I. Introduction
A data warehouse is a subject-oriented, integrated, time variant, non-volatile collection of data in support of management’s decision-making process [1]. It is an overall strategy, or process, for building decision support systems and a knowledge-based applications architecture. The primary objective of Data Warehousing is to bring together information from disparate sources and put the information into a format that is conducive for making business decisions. When records from different sources are integrated then they may not agree in schema and also there is redundancy problem. Therefore Detection and elimination of duplicate records from data warehouse is a challenging task. There are different approaches to remove the duplicates. In this paper we discuss the existing methods of duplicate detection and then comparison between them. The layout of paper is as follows: Section II discusses Data cleansing, Data Cleansing methods are discussed in Section III, Section IV includes comparison of different cleansing methods and finally Section V concludes our work.

II. Data cleansing
Data cleaning or scrubbing is the process of removing the errors form the data. It is an inherent activity related to database processing, updating and maintenance. Data fed from various operational systems prevailing in the different departments/sub-departments of the organization, has discrepancies in schemas, formats, semantics etc. due to numerous factors [2]. While integrating data from these heterogeneous sources, multiple instances referring to the same real-world entity are generated which need to be pre-processed before loading in the data warehouse. One of the most difficult tasks is to distinguish between multiple occurrences of same real-world data sets scattered over different sources. The conflict arises when these heterogeneous sources have to be accumulated into a large data warehouse. The data cleaning algorithms works on these types of problems.

III. Data cleansing algorithm
There are several approaches for duplicate elimination or record linkage which is a part of data cleansing. Every duplicate detection method requires an algorithm for determining whether two or more records are duplicate representations of the same entity. There are two categories of data cleansing algorithms one is domain-dependent and the other is domain-independent.

A. Domain dependent algorithms
In these algorithms domain knowledge is required before applying cleansing algorithm. The equivalence of records are defined by a set of equational axioms, which is termed equational theory [2]. Determining record equivalence requires much more information about database. This is an complex inferential process by which duplicates are determined. A high level declarative rule language is used for expressing the equivalent logic. Different instances of the same entity in different databases can be identified by making inferences using other shared attributes [3]. For example to check whether two records are equivalent the rule applied is [6]:

Given two records r1 and r2
If the last name of r1 equals the last name of r2,
And the first name differ slightly,
And the address of r1 equals the address of r2 then
r1 is equivalent to r2.

A. A sorted neighborhood method
A sorted neighborhood method is used to detect approximate duplicates. This method consists of three steps: First, a key is computed by extracting the relevant fields or portion of fields from the database. The choice of the key depends on domain knowledge. Efficiency of algorithm heavily depends on the key chosen. The second step is to sort the database using the key calculated in the first step. Finally a fixed size window is moved to compare the records. Every new record entering in the window is compared with previous records in the window. Suppose the size of the window is w records, then every new record entering the window is compared with the previous w−1 records to find “matching” records. The complexity of this algorithm is calculated at different phases of the algorithm, the key creation phase requires O(n) operations, the sorting phase requires O(n log n), merging requires O(wn) where n is no. of records in database. So total complexity of sorted neighborhood method is O(n log n) if w<log n otherwise O(w n) [10]. The drawback of the method is that it depends heavily on the proximity of duplicate records after sorting. If duplicate records are far apart after sorting, it is unlikely that they will appear in the same window during the scanning process, and hence, will be missed.

B. Domain independent algorithm
Domain-independent approach does not required domain knowledge for data cleansing. These techniques assume that records have the same high level schema and are made up of alphanumeric characters. These methods consider records as strings and compute record similarity using string comparison algorithms. Basically three field matching algorithms exist for identification of duplicates [8]: Basic Field matching algorithm, Recursive field matching and Smith-Waterman algorithm. In basic
field matching algorithm, the degree to which two fields match is the no. of their matching atomic strings divided by their average no. of atomic strings. In recursive field matching algorithm , two strings match with degree 1.0 if they are the same atomic string or one abbreviates the other else their degree of match is 0.0 .The Smith Waterman algorithm computes an n x m matrix with given two strings of length n and m whose [i, j] component is the edit-distance between the prefix 1...i of one string and the prefix 1...j of the second string. When the prefixes (or the entire strings) match exactly, then the optimal alignment follows the main diagonal. For approximate matches, the optimal alignment is within a small distance of the diagonal. The cleaning algorithms uses any of the three matching algorithm. There are different domain independent data cleaning techniques:

1. **Token based data cleansing algorithm**

In token based approach instead of taking all the fields as a key we use smart tokens. Tokens are formed using the most important fields of records . This heavily depends on domain knowledge . This is an important step because performance of the whole algorithm depends on it. Then the tokens are sorted and compared to identifies duplicate records . Token-based technique achieves a better result than the record-based techniques of comparable algorithms. The sorting phase requires O(n log n) operations where ‘n’ is maximum number of atomic strings in either field.

2. **Priority queue algorithm**

The priority queue algorithm is used for approximate duplicate detection. It uses a priority queue of sets of records belonging to the last few clusters detected [5]. The algorithm sequentially scans the database and find out whether each record belongs to a cluster represented in the queue or not. If the record is an existing member of cluster then next record in database is scanned otherwise the record is kept in the priority queue. This record is then again checked against existing clusters. If it is not the member of any cluster of them then it is assumed as a new cluster in priority queue. With this approximate similar records are kept in same cluster. Hence we identify the duplicate records easily. The complexity is calculated for fixed ‘t’, ‘k’, ‘l’ and ‘m’ where ‘t’ is number of records in database each of length ‘l’ and ‘k’ is maximum number of sets in the queue with maximum size of each set ‘m’, all union-find and priority queue operations take small constant amounts of time. For one pass over the database, the number of invocations of the Smith-Waterman algorithm is bounded above by ‘tkm’. The overall time complexity of one pass is therefore O(tl log tl +tklm) =O(tl log tl) as there is small fixed number of passes, the total time complexity is the same [4].

3. **Duplicate record elimination using external merge sort**

This method uses modified two way merge sort which removes duplicate records along with sorting. If two input runs contain duplicate records, then the output run produced by merging them should retain only one copy of each record. Whenever two input tuples are compared and found to be identical, only one of them is written to the output run and the other is discarded by advancing the appropriate pointer to the next tuple [7]. The complexity of the duplicate elimination process using modified merge-sort is determined by two factors: the number of phases and the size of the output runs produced at each phase. The number of phases/operations required to sort a file along with duplicate elimination is O (log2n) where ‘n’ is no. of records in file regardless of the number of duplicate tuples.

IV. **Comparison of cleansing algorithms**

The comparison of all the above discussed techniques are given in the Table 1. We compare them on the basis of what are the key steps followed by the technique, whether the technique find approximate or exact duplicate of records, there complexities, the field matching approach and there limitations.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Key Steps</th>
<th>Duplicate Detection</th>
<th>Complexity</th>
<th>Domain knowledge</th>
<th>Field matching approach</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorted Neighborhood method</td>
<td>Key creation + sorting + Merging using sliding window</td>
<td>Approximate duplicate detection</td>
<td>O(nw) where ‘n’ is no. of records and ‘w’ is window size.</td>
<td>Domain dependent</td>
<td>Edit distance, phonetic similarity typewriter distance</td>
<td>It depends heavily the proximity of duplicate records after sorting</td>
</tr>
<tr>
<td>Token Based Algorithm</td>
<td>Selection of fields + formation of tokens + sorting + duplicate detection and elimination</td>
<td>Exact duplicate detection</td>
<td>O(n log n) Where ‘n’ is maximum number of atomic strings in either field.</td>
<td>Domain independent</td>
<td>Basic field matching algorithm</td>
<td>Performance heavily depends on Token selection</td>
</tr>
<tr>
<td>Priority Queue Algorithm</td>
<td>Sorting + merging + duplicate elimination using priority queue</td>
<td>Approximate duplicate detection</td>
<td>O(tl log tl) Where ‘t’ is no. of records with Length ‘l’</td>
<td>Domain independent</td>
<td>Smith water man algorithm</td>
<td>Difficulty of choosing the prime representative to represent the whole data set.</td>
</tr>
<tr>
<td>Duplicate record elimination using external merge sort.</td>
<td>Sort the records using modified external merge sort that also remove the duplicates</td>
<td>Exact duplicate detection</td>
<td>O(log2 n)</td>
<td>Domain independent</td>
<td></td>
<td>This method is suitable for files which have uniform duplication factor.</td>
</tr>
</tbody>
</table>
V. Conclusion

There are various duplicate elimination approaches for data cleansing. These approaches either detect the exact duplicate or approximate duplicate. Detection of approximate duplicate is most challenging task. There are two categories of cleansing algorithm one is domain dependent that requires domain knowledge for duplicate detection and the second is domain independent. Sorted neighborhood method is the only domain dependent algorithm all other are domain independent algorithms. In this paper we discuss all the approaches and present a comparison between them and concluded that the complexity of duplicate elimination using external merge sort is minimum. Further each algorithm has its own benefits as well as limitations.

References