Smart Removal of Redundant Data using Progressive Techniques

K. D. Mane1 M. H. More2 V. B. Bansode3 P. S. Gawade4
1,2,3,4B.E. Student
1,2,3,4Department of Computer Engineering
1,2,3,4S.B. Patil C.O.E. Indapur, 413106, Savitribai Phule, Pune University, Maharashtra, India

Abstract—Data are among the most important assets of a company. But due to data changes and sloppy data entry, errors such as duplicate entries might occur, making data cleaning and in particular duplicate detection indispensable. However, the poor size of today's data sets render duplicate detection processes expensive. Online retailers, for e.g., offer huge catalogs comprising a constantly growing set of items from many different suppliers. As independent persons change the product portfolio, duplicates arise. Although there is an obvious need for de-duplication, progressive duplicate detection identifies most duplicate pairs early in the detection process. Instead of reducing the overall time needed to finish the entire process, progressive approaches try to reduce the average time after which a duplicate is found early termination, in particular, then yields more complete results on a progressive algorithm than on any traditional approach.

Key words: Data Cleaning, Data Duplication, Progressiveness

I. INTRODUCTION

Today, duplicate detection methods need to process ever larger datasets in ever shorter time: maintaining the quality of a dataset becomes increasingly difficult. We present two novel, progressive duplicate detection algorithms that significantly increase the efficiency of finding duplicates if the execution time is limited: They maximize the gain of the overall process with in the time available by reporting most results much earlier than traditional approaches. But due to data changes and sloppy data entry, are among the most important assets of a company such as duplicate entries might occur, making data cleaning and in particular duplicate detection in dispensable. However, the pure size of today's datasets render duplicate detection processes expensive. Online retailers, for example, offer huge catalogs comprising a constantly growing set of items from many different suppliers. As independent persons change the product portfolio, duplicates arise. Although there is an obvious need for de-duplication, online shops without downtime cannot afford traditional de-duplication.

II. LITERATURE SURVEY

A. “PayAsYou-Go Entity Resolution” - Steven Euijong Whang, David Marmaros, and Hector Garcia Molina[1]

In this article, evaluated the overhead of constructing hints as well as the runtime benefits for using hints. It provides the solution is a pay-as-you-go approach for ER. Work in reasonable amount of time or when there is a time limit.


In this paper they detect approximately duplicate records in a database they provide the solution for that field matching technique. Improvement in the current state-of-the-art.

C. “Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem” - Mauricio A. Hernandez, Salvatore J. Stolfo [3]

This paper presents a Basic Data Cleansing Solutions, develop simulation based and generative models to analyze. Work with more real world data with same sort and less error.


This Duplicate detection in real world. They uses the the Duplicate Count Strategy. In this paper, they specifically describe the search for duplicates in very large data sets within reasonable time.


[5]

They intend to to identify similar or matching digital library entities such as citations or authors, and consolidate them into a single canonical entity. Sorted neighborhood method (SNM), and demonstrate how we can achieve improved accuracy and performance by adaptively changing its fixed sliding window size. Adaptive framework, the same idea can be applied to other well-known record linkage algorithms.

III. SYSTEM MODEL

The system contains the different modules as

1) Data Collection: To collect and/or retrieve data about activities, results, context and other
2) Pre-processing method: Data Pre-processing or Data cleaning. Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.
3) Data Separation: After completing the pre-processing, the data separation to be performed.
4) Duplicate detection.

Fig. 1: System Architecture
IV. SUMMARY & CONCLUSION

Progressive sorted neighborhood technique and progressive obstruction. Each algorithm increases the potency of duplicate detection for things with restricted execution time; they dynamically amend the ranking of comparison candidates supported intermediate results to execute promising comparisons initially and fewer promising comparisons later. To see the performance gain of our algorithms, we have a tendency to planned a unique quality live for progressiveness that integrates seamlessly with existing measures. Victimization this live, experiments showed that our approaches vanquish the normal SNM by up to 100.

REFERENCES


