Scientific Journal of Impact Factor (SJIF): 4.14

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 5, May -2016

Web Framework for Data Deduplication at Scale

Chirag Nagpal[1], Neha Malik[2], Ajit Singh[3], Shushma Shirke[4], Shinsmon Varghese[5]

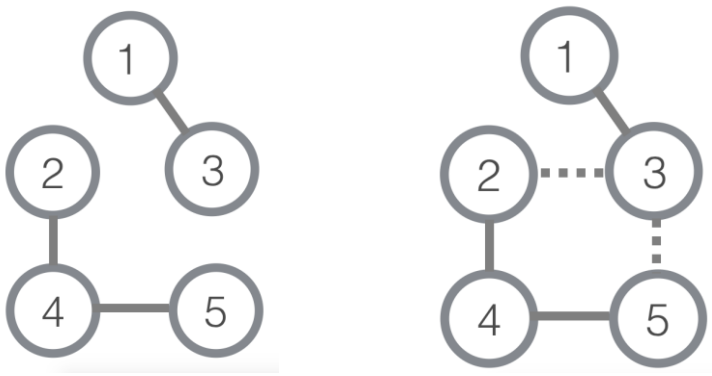
[1,2,3,4,5]*Dept. Of Computer Engineering, Army Institute of Technology,*

Abstract —*In todays time of multiple heterogenous sources of data, data dedupliction is a difficult challenege. We present a Web Based framework that utilizes the popular MVC paradigm to provide the end user with a functionality allowing Data Deduplication. In the backend the framework allows the use of distributed computing using a Map Reduce strategy to scaleu p our de deuplication schme.*

Keywords*-component; formatting; style; styling; insert (key words) (minimum 5 keyword require) [10pt, Times new roman,Italic, line spacing 1.0]*

I. INTRODUCTION

We live in the day and age of big data with multiple heterogeneous sources of data from the Internet of Things to the Semantic Web. With multiple such data records, there is inherent possibility of data redundancy in these strict schema driven databases. Thus a need is felt to utilize recent trends in machine learning techniques in order to resolve such inherent redundancies arising from various sources including human error.



(a) (b)

Fig. 1 (a) The graph nodes represent the entities before Data Deduplication. (b) Represents the records after Deduplication the dashed edges represents the records that are found to be similar using the deduplication framework.

In this paper we present a web based framework, along with a Map Reduce backend in order to perform this large scale fuzzy matching based Entity Resolution. The framework allows users without any prior domain knowledge of machine learning or distributed computing to perform data deduplication on these datasets. The web framework is flexible and allows decoupling of the underlying Map Reduce backend with a Model-View-Controller based frontend.

II. MVC FRAMEWORK

The MVC[1] is an extremely popular design pattern encountered in software development. MVC frameworks are the preferred for various large scale web platforms. MVC frameworks allows the decoupling of components into the application into three basic parts – The Model, The View and the Controller.

2.1. Model

It refers to the data of the application. The model comprises of the rules and logic that govern how data is stored and manipulated. Popular MVC frameworks support various different Model paradigms like RDBMS, where data is stored in strict schema driven relational databases, such a strategy is employed by SQL or Structured Query Language. Newer MVC frameworks also support the use of paradigms like NoSQL databases, which are extremely flexible in terms of the schema of the data to be stored.

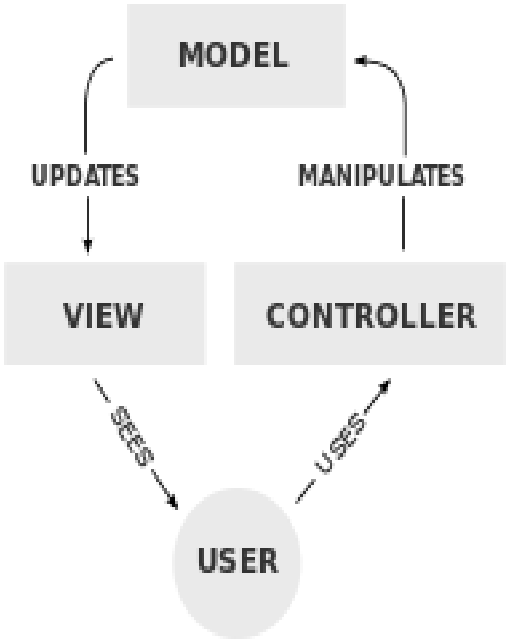


Fig. 2 A typical collaboration of MVC components

2.2. View

View refers to the component that is responsible for the user interface. It displays the output to the user and is also responsible for taking the input from the user. Most popular MVC frameworks involves web based views which support user interaction through web pages displayed in Web browsers. Each webpage consists of various forms, with different input elements including, textboxes, drop down menus etc. The data to be input from the user is delivered in the form of simple HTTP requests like GET, POST etc. One such engine allowing templating is the *jinja* engine which allows inline python code to be put with HTML code.

2.3. Controller

The controller refers to the component that is responsible for the underlying logic that interacts with the users request and produces the required response. The controller acts as an intermediary between the Model and View and abstracts the control logic. Popular scripting languages like Python and PHP are utilized to serve as the controlling language.

2.4. Flask

For our application we utilize the popular python based micro framework called 'Flask'[2]. Flask allows considerable abstraction from the underlying HTTP logic and allows the creation of large scale production level Web Platforms rapidly. Flask supports interactive views like jinja templating making it easy to design and user interfaces.
Plotting Server: This module allows the end user to seamlessly utilize certain external plotting tools like Matplotlib [3] and Gephi [4], inorder to plot the resulting output of the Entity Resolution process, as the output clusters can be visualized in terms of a Graph, with edges representing the matched entitities.

III. MAP REDUCE

Map Reduce is a popular programming strategy for parallel processing over a distributed infrastructure[5]. The MAP step is used to find the appropriate data records from the database and provide the records parallel to the multiple nodes in the databased. The Reduce operation is popular function that is used to perform a particular operation on multiple node.

In our application the Map Operation can be considered a bucketing scheme, wherein the data to be operated upon is distributed in. Thus the Map operation depends upon a single strong feature which it uses to decompose the data into the buckets. Inside each bucket, the reduce operation carries out pairwise entity matching, using some previously learnt match function.

An important consideration is too ensure the number of buckets with respect to the size of each bucket. Higher number of buckets, of smaller size would reduce the computation to an $O(n^2)$ time complexity while Lower number of large

buckets too will have a similar impact. Inorder to tune the parameters considerable amount of heuristic is required. The bucket size also is significantly dependent on the domain in which the application is designed or intended to be implemented on. [6] describes some of the strategies that are used to find the correct number of buckets.

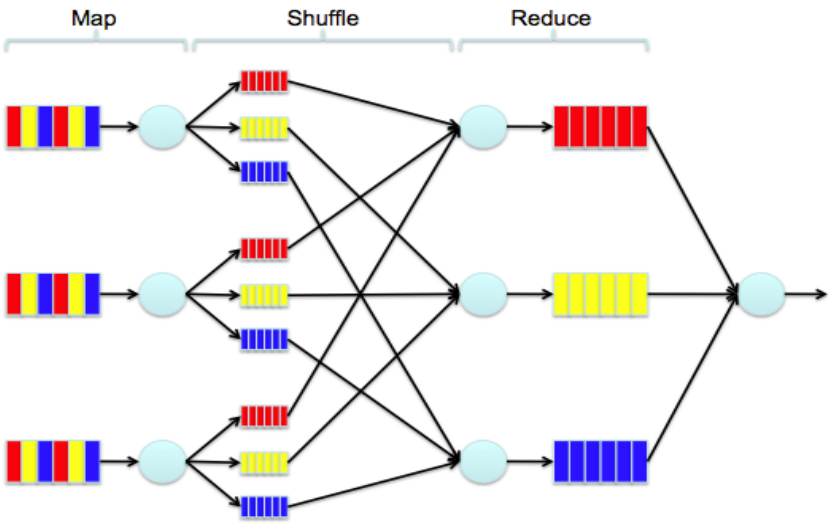


Fig. 3 Typical Map Reduce operation.
Map divides the input based on its type while Reduce operates on parallely on the input.

The match function, is a computationally expensive operation and hence is not suitable to be applied to the entire dataset, and hence is applied only to individual buckets, reducing the number of operations exponentially.

$$n_1^2 + n_2^2 + \ldots + n_d^2 << N^2$$

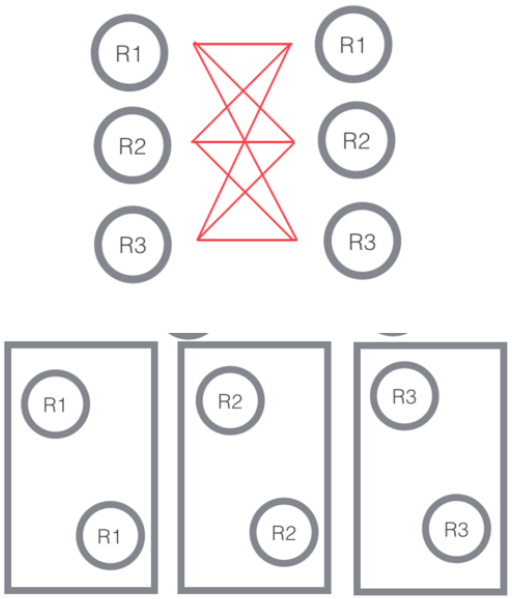


Fig. 3 Instead of comparing each add 'R' to other add we divide the ads to blocks and perform ER within each block. Reducing the amount of computations, exponentially

IV. CHALLENGES

While the system is distributed in nature, a MapReduce system depends strongly on how well the system is designed to be distributed, as components are loosely coupled. Significant bottlenecks can arise from communication errors or network bandwidth. Thus, the better designed distributed systems tend to outperform highly distributed, poorly designed systems
A significant challenge arises from the learning of a confidence threshold to perform Data Deduplication. Low confidence lead to a breakdown with very large clusters while high thresholds, result in extremely granular resolved clusters. The creeping in of false positives, results in the breakdown at even very high thresholds. Further research needs to be carried out in order to ensure the prevention of such a breakdown.

REFERENCES

[1] Leff, Avraham, and James T. Rayfield. "Web-application development using the model/view/controller design pattern." Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International. IEEE, 2001.
[2] Grinberg, Miguel. Flask Web Development: Developing Web Applications with Python. " O'Reilly Media, Inc.", 2014.
[3] Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science and engineering 9.3 (2007): 90-95.
[4] Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy. "Gephi: an open source software for exploring and manipulating networks." ICWSM 8 (2009): 361-362.
[5] Papadakis, Georgios. Blocking Techniques for efficient Entity Resolution over large, highly heterogeneous Information Spaces. Diss. Technische Informationsbibliothek und Universitätsbibliothek Hannover (TIB), 2013.

