Improved Elastic Search And Efficient Duplicate Data Detection And Removal Using Ensemble Big Data Algorithms

Shaik Subhani , Dr.Nalamothu Naga Malleswara Rao

Abstract: Searching is most widely used in many applications. Elastic search is done for the documents and text within the local or in global databases. Elastic search is very fast comparing with the other normal searching process because elastic search will search the index directly not the text. Searching from huge data becomes more complicated to get the accurate results. Elastic search is developed with java by using the Lucene library. This is open source and highly efficient to search full text and analytics engine. In this paper, the Improved Elastic Search (IES) is introduced which is integrated with efficient duplicate detection by using Ensemble big data algorithms are implemented. This will improve the performance of the Elastic search because of detection and removal of duplicate data. For the improved accuracy the big data with hadoop algorithms are adopted. The dataset used for experimental results is any synthetic document related data.

Keywords: elastic search, big data, hadoop.

1. INTRODUCTION

Many companies are trying to improve the ES (ES) search engines to increase the performance of their systems. ES is mainly related to best index storage which is developed by Lucene and the original algorithm is used for matching the text. The ES can be executed by the Application Programming Interface (API) technique which has high expansible. To install the ES in the system which is very simple and can configure and adjust the environment? The important issue observed to analyze the big data is the space between the object of monitoring and the object of analysis.For example, if the user accounts are taken for objects of observation, it is not always required to every account represents. The record can likewise be utilized by relatives, companions or outcasts. At the time of Big Data analysis, the expectations are commonly made about the nature of the object of monitoring, which is later violated regularly. Verstrepen and Goethals [1] consider the difficulties of the proposal framework applied to shared client accounts. Another issue that prompts the contrast between the object of perception and the object of examination is the exchange off among data and classification. Since classification limits access to information at the individual level, the examination is done dependent on aberrant information. ES was at first created as a framework for full-text search in enormous volumes of unstructured information. At present, ES is an undeniable explanatory framework with different capacities. Information in ES is put away in a reversed record position dependent on Apache Lucene (AL). AL is the most acclaimed internet searcher, initially cantered explicitly on implanting in different projects. Lucene is a library for high speed full-content hunt, written in Java. It gives propelled search capacities, a great file building and capacity framework that can at the same time include, erase reports and perform advancement alongside the inquiry, just as parallel hunt on a lot of lists joining the outcomes. The drawback is similarly low ordering velocity (particularly in correlation with Sphinx), just as the absence of API (which is dealt with by ES).

- *Mr.Shaik Subhani currently pursuing his Ph.D in CSE from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.*
- *Dr. Nalamothu Naga Malleswara Rao is currently working as Professor in Department of CSE, RVR & JC College of Engineering, Guntur, Andhra Pradesh, India.*

ES stores its information in at least one file. Utilizing likenesses from the SQL world, ordering is like a database. It is utilized to store the archives and read them from it. ES utilizes AL library to compose and peruse the information from the file. ES list might be worked off in excess of a solitary AL Index by utilizing "Shards".

2.LITERATURE SURVEY

This section described the various algorithms based on the ES. Nowadays many companies like NetFlix, Linkedin and other companies based on ES for storing, indexing and searching the data because of its advantages such as good expansion models, automatic sharding and other types of replications. More than 15 clusters consist of 800 nodes which are used by Netflix. By using Yahoo Cloud Serving Benchmark (YCSB), they vary the parameters that affect the performance of the databases execution time. Their work does not consider the implementation of ES in real industrial system [2]. They also presented benefits of Amazon ES service. Their work is limited to conceptual working of ES only [3]. Compare with the various searching techniques such as Sphinx with the ES it is a little bit low indexing and searching speed. For the analytical system, it is able to scale and enables sampling of very tedious shapes. This not very easy to utilize, but this presents the many extra features. One of the very important advantages of this search is this is having very little memory and improved indexing is fast which can retrieve the multiple documents at a time. This is the searching that is developed with the full-text search with huge amounts of data that is unstructured. In the present situation, this searching is merged with an analytical system with many capabilities. The data which is stored in this search inverted index format based on Apache Lucene (AL). AL is other improved search engine that aims to develop for the embedding in other programs. The search that divides the data between several machines, that can make it possible for supporting the high-performance operations. Shards are the parts of the data which are divided by this ES. Two types of shards are there such as master and replica. The master operation performs both read and write, while the replica operation performs read-only, and is an exact copy of the

master. In any situation, if the master is failed, the replica becomes the master.

Duplicate Data Detection (DDD)

In any datasets or database data duplication becomes more complicated to process the data or records. With the duplicate data present in any document or datasets the processing time will take more compare with the normal data. The quality of data is calculated based on the availability of the duplicates and noise in the data; this is called as data quality problems [4]. DDD plays a major role in record linkage, near-duplicate detection and filtering queue [5]. DD is utilized to find some of the real-world examples that present in various formats or representations in the database [6, 7]. In this paper, it is very important to remove the duplicates in the datasets is more difficult to overcome that. This paper mainly focuses on the removing the duplicates on the huge datasets with the integration of bigdata.

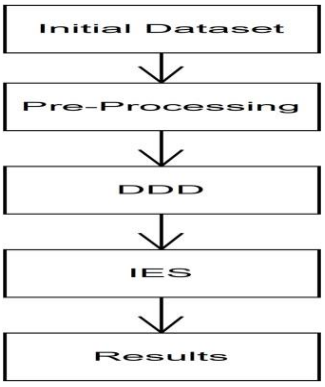

Figure: 1 *System Architecture*

Dataset Description:

In this paper, for the elastic search there is a need of dataset. The dataset used in this paper is synthetic company employee's details which contains 10 attributes and one lakh records and this in the form of Json. Among these records the duplicate records are processed in the pre-processing stage and the quality of data is ready for the IES.

{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"KARLA","LastName":"QUINLIVAN","Designation":"
20170","Gender":"Female","Age":48,"MaritalStatus":"Married","I
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"MAURICE","LastName":"HELT","Designation":"Pre
01085","Gender":"Female","Age":52,"MaritalStatus":"Unmarried"
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"MAURO","LastName":"MACINNES","Designation":
11201","Gender":"Male","Age":63,"MaritalStatus":"Unmarried","I
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"TOD","LastName":"GOODSPEED","Designation":"
30741","Gender":"Male","Age":45,"MaritalStatus":"Married","Inte
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"HUNG","LastName":"ALDERETE","Designation":"I
11553","Gender":"Male","Age":47,"MaritalStatus":"Unmarried","I
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"LEONIA","LastName":"ZOLDAK","Designation":"Pr
55406","Gender":"Female","Age":51,"MaritalStatus":"Unmarried"
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"MAURITA","LastName":"LISKE","Designation":"Pr
37876","Gender":"Female","Age":65,"MaritalStatus":"Married","I
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"MARGOT","LastName":"TROUTT","Designation":"
18966","Gender":"Female","Age":57,"MaritalStatus":"Married","I
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"MARNA","LastName":"DEISS","Designation":"Vice
29708","Gender":"Female","Age":62,"MaritalStatus":"Unmarried"
{"index":{"_index":"companydatabase","_type":"employees"}}
{"FirstName":"DWIGHT","LastName":"KONDO","Designation":"Vi
37643","Gender":"Male","Age":46,"MaritalStatus":"Unmarried","I

Figure: 2 *Dataset instances*

Enhanced De-Duplication Algorithm Steps:

Duplication of records can be deleted by the using various traditional DDD algorithms. The enhanced De-Dup algorithm used to find the duplicate records which are same within the datasets. To calculate the similar string values in every row jaro distance (JD) is used. This is used to compare the first name and last name. The two strings are σ1 and σ2. The following are the steps for duplicates detection.

The string lengths are calculated with | σ1 | and | σ2 |.

This will find the similar characters c with two strings; which is done with $|i - j| \leq \frac{1}{2}\min\{|\sigma_1|, |\sigma_2|\}$.

The comparison of strings are done with following transportations are as follows.

$$\text{Jaro}\,(\sigma_1, \sigma_2) = \frac{1}{3}\left(\frac{c}{|\sigma_1|} + \frac{c}{|\sigma_2|} + \frac{c - t/2}{c}\right)$$

The result of the pre-processing is done with using the java and jdk 1.2 to show the results. In this stage the duplicates are removed and in the synthetic employees dataset and the results are shown in table 1,

TABLE 1 *PERFORMANCE OF THE DDD*

Total No of Instances	Original Instances	Duplicate Instances
1,00,000	79,987	20,013

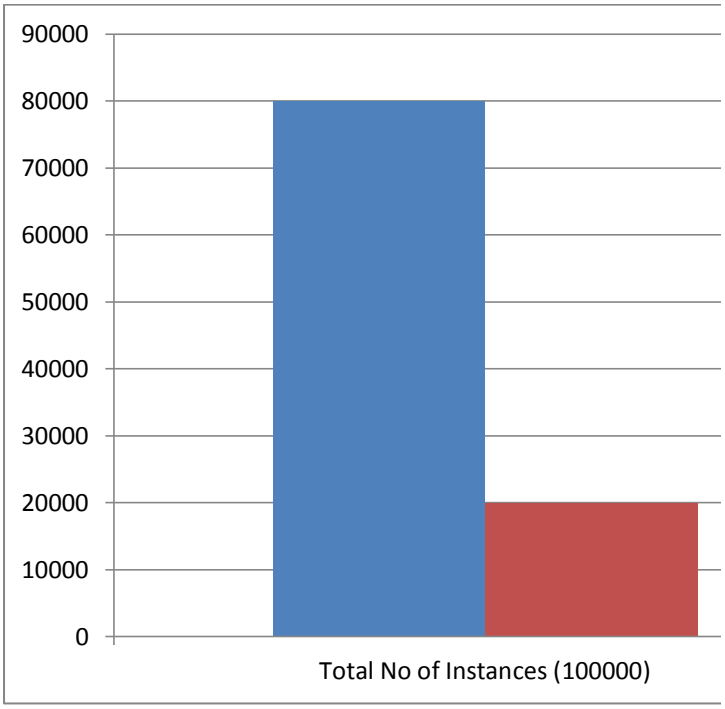

***Figure: 3** Performance of DDD*

Performance Evolution

Various performance measures are given below for the calculation of False Positive Rate (FPR), False Negative Rate (FNR), Sensitivity, Specificity, and Accuracy.

FPR

Based on the given data the information is divided into normal and abnormal..

$$FPR = \frac{FP}{FP + TN}$$

FNR

The percentage of cases where a data was classified to abnormal data, but in fact it did.

$$FNR = \frac{FN}{FN + TN}$$

Accuracy: This will calculate the overall accuracy of the data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The Data Processing Time calculator computes the amount of time needed to process an amount of data (S) at a specified rate (R).

(S) This is the size of the file or data object being processed.

(R) This is the processing rate

$$A = \frac{S}{R}$$

Improved Elastic Search (IES) using Map Reduce

The improved elastic search using map reduces gives the better results compare with the existing algorithms. The big data is mainly focuses on processing the huge data. To process the original instances 79,987 map reduce algorithm is adopted with the IES. Espicially map reduce divide the process parallel to with all the workers. In map reduce workers plays the major role. They divide the processing of dataset into shards. Every worker will take one shard and process the records according to the elastic search. The following steps to process the elastic search.

1.) After pre-processing.
2.) Enter the search based on requirements.
3.) User can enter total no of persons with similar hobbies and any queries related to the elastic search.
4.) Now the map reduce implemented and no of workers assigned according to the dataset records.
5.) Implement IES.
6.) Show results.

Table 2 shows the total no of workers shard the processing of dataset with various requirement of the user elastic search.

***TABLE 2** PERFORMANCE OF IES*

S. No	Total No of Workers	Record assigned to every workers	Result Show based on query	Time taken	Accuracy
1	Worker-1	9876	4532	0.010	98.8
2	Worker-2	8976	3213	0.0054	98.8
3	Worker-3	9807	3421	0.0045	98.8
4	Worker-4	7890	4121	0.0034	98.8
5	Worker-5	8765	3211	0.0045	98.8
6	Worker-6	9098	2134	0.0056	98.8
7	Worker-7	8976	2321	0.0065	98.8
8	Worker-8	6785	2431	0.006	98.8
9	Worker-9	5674	1232	0.003	98.8
10	Worker-10	4140	876	0.004	98.8
Overall Results	10 Workers	79987	27492	0.054	98.8

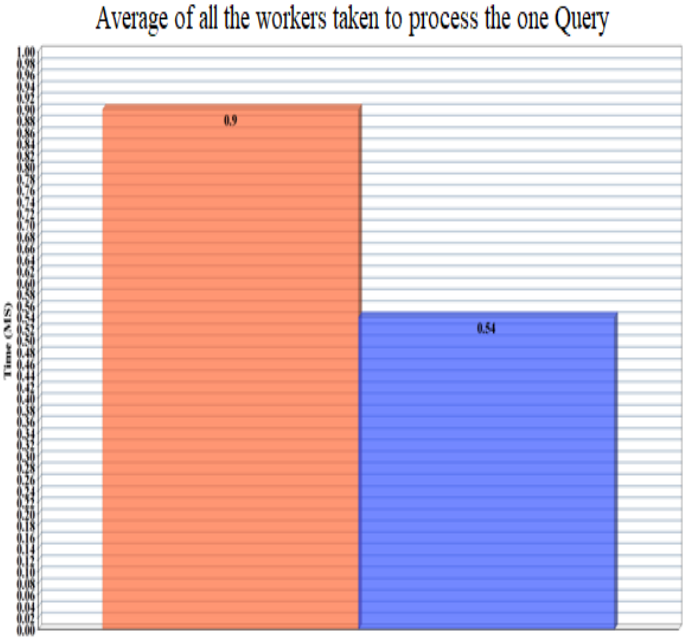

***Figure 4**: Query processing starting time*

3.CONCLUSION

In this paper, the IES is integrated with the map reduce algorithm to process the huge datasets. The two parameters

are overall processing time and accuracy is calculated. Map reduce plays the major role to improve the performance of the results. Map reduce is used to shard the dataset and process the data according to the elastic search. The data de-duplication is done fully on all the dataset to increase the quality of the data. The other name for quality is accuracy. It is very time taking process for every query if the map reduce algorithm is not integrated.

4.REFERENCES

[1] Verstrepen K, Goethals B. “Top-n recommendation for shared accounts”. In: Proceedings of the 9th ACM Conference on Recommender Systems, NY, 2015, ACM, pp. 59–66.

[2] Abubakar, Y., Adeyi, T. S., & Auta, I. G. (2014), "Performance Evaluation of NoSQL Systems Using YCSB in a Resource Austere Environment". IJAIS, 7 - No.8, 23-27.

[3] Gupta, P., & Nair, S. (2016). "Survey Paper on Elastic Search”. IJSR, 5(1), 333-336.

[4] G. Beskales, A., M. Soliman, F., I. Ilyas, S.i Ben-David, and Y. Kim, "ProbClean: A Probabilistic Duplicate Detection," ICDE, 2010 IEEE 26th International Conference.

[5] J. Kim and H. Lee, "Efficient Exact Similarity Searches using Multiple," in IEEE 28th ICDE, 2012.

[6] M. Ektefa, F. Sidi, H. Ibrahim, and M. A. Jabar, "A Threshold-based Similarity Measure for Duplicate Detection," ICOS, 2011 IEEE Conference on, Langkawi, pp. 37 - 41.

[7] M. Herschel, F. Naumann, S. Szott, and M. Taubert, "Scalable Iterative Graph Duplicate Detection," in IEEE TTKDE, 2012.