ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 8(33), DOI: 10.17485/ijst/2015/v8i33/77477, December 2015 ISSN (Online) : 0974-5645

# The Pre Big Data Matching Redundancy Avoidance Algorithm with Mapreduce

**G. Somasekhar[1*] and K. Karthikeyan[2]**

[1]SCSE, VIT University, Vellore – 632006, Tamil Nadu, India; gidd.somasekhar2014@vit.ac.in
[2]SAS, VIT University, Vellore – 632006, Tamil Nadu, India; k.karthikeyan@vit.ac.in

## Abstract

Data matching provides valuable information relevant to complex decisions about programs or policies. For example, information about peer influences on teen behavior, achieved through data matching, can help people decide what kinds of programs would discourage early pregnancy, teenage drinking, and delinquency. If the data used in the data matching process has the big data characteristics and could not be processed on a single machine then it is termed as big data matching, where traditional data matching methods fail. Dedoop1 is the latest tool developed for big data matching. It needs a pair of clusters as input. The state-of-the-art big data matching techniques have a common disadvantage which leads to expensive redundant similarity computations. We focus on the selection process of the pair of clusters given as input to the Dedoop. This approach avoids unnecessary similarity computations. Entire data is subjected to prior selection process before entering into the Dedoop. The technique of canopy clustering is combined with the unique linkage pairs formation technique to solve the redundancy problem. A test sample of article-author information is usedto match the authors related to common subject. Though the basic pre data matching redundancy avoidance approach (BPRA2) solves the problem to some extent, it has some limitations. In addition to considerable preprocessing overhead, it does not solve scalability and incremental issues. The proposed PRAMR approach reduces the preprocessing overhead in BPRA to 'm' times. As the PRAMR uses the big data technique 'MapReduce', the scalability and incremental issues are solved. Hadoop is used for MapReduce jobs. The results are compared with the BPRA and Kolb's approach3. In section IV and V, it is proved that PRAMR is more efficient than the state-of-the-art techniques. PRAMR shows improvementcompared to BPRA and Kolb's approach giving a better solution to the overlapping clusters problem.

**Keywords:** Big Data Matching, Canopy Clustering, Data Point, Overlapping Clusters, Redundancy, Similarity Roller

## 1. Introduction

Modern techniques for data generation, data collection and data storage made huge amount of data available to the users. These huge chunks of data are usually stored as continuous text such as personal demographic data, bibliographic information, phone and mailing lists.This data must be integrated in order to enrich the data quality and draw valuable conclusions. Integration of such data involves two major problems. 1. Structural heterogeneity 2. Syntactical heterogeneity. The former takes place when the source data has no common schema.In this case, schema reconciliation techniques[4,5,6], are used to solve the problem. The latter takes place when syntactically different records refer to same realworld entity. Datamatching is a solution to this problem. If the target data is so huge, then it can be termed as big data matching for which a specific tool called Dedoop can be used. Prior to de-duplication with dedoop, a pair of record collections those are going to be matched should be selected from the pool of records and given as input. For grouping records those related to similar entity,efficient clustering techniques like canopy clustering can be used which leads to the problem of overlapping clusters. Though the similar records are grouped into clusters, some of the clusters are usually overlapped in real time. As the degree of overlapping increases, the degree of redundancy of the data points and their similarity computation increases to a greater extent. Though the Kolb's approach implements the dedoop tool to solve this problem, the mapreduce phase

* Author for correspondence

of de-duplication is more complex involving unnecessary condition checking in every reducer process. The BPRA approach uses clustering and redundant-free pair selection techniques prior to de-duplication(i.e.,datamatching) to reduce the complexity in Kolb's approach during the mapreduce phase of dedoop. When the target data grows to big data range, the BPRA also incurs considerable preprocessing overhead. The PRAMR algorithm reduces the preprocessing overhead to the maximum extent, providing increment and scalability. The entire process is explained in the sections II, III and IV. Section V gives comparison of results. Fraud detection and Anomaly detection in various fields like finance, science and so on is the main advantage of big datamatching. An optimized bigdata platform implementation model[7] through S/W configuration was proposed which is useful in handling big data matching problem.

Many research works were carried out in the area of datamatching, with many names such as Record Linkage[8,9], De-duplication[10], Entity-Name Matching[11]. Efficient clustering techniques for big data matching were proposed earlier[12,13,14]. A solution was given by Mc Cullum by grouping records in canopies and limiting the expensive similarity computation within each canopy. A two-phase approach was proposed by Chaudhuri, based on nearest neighbors' computation.

Several indexing techniques were summarized for data matching by Peter christen[15]. Fast blocking methods for datamatching were proposed by Baxter, Christen and Churches[16]. Iterative datamatching was mentioned by Bhattacharya and Getoor[17]. A new datamatching method was explained by Hu, Li and Feng[18] which can be applied when the source data is in the form of graphs. Some mapreduce based techniques for datamatching were proposed by Kolb, Thor and Rahm in[19,20]. An incremental clustering algorithm for de-duplication was proposed by Costa, Manco and Ortale[21]. The problem of overlapping clusters was identified by Kolb and a mapreduce based solution was given to it using the dedooptool. The BPRA approach solves the same problem in the pre-dedoop phase with minimal clustering overhead.

## 2. Problem Domain and Definition

Data from multiple sources should be integrated to enrich data quality and to reduce data acquisition cost which facilitates and improves data analysis task.Schema mapping,data fusion and data matching are the different methods evolved in this process where the last method is focused in this paper. The records from initial pool of data may refer to different entities such as customers, patients, employees, students, travelers or tax payers. Grouping of these records pertainingto similar entities is called "data matching". When the target data grows to huge amounts,the same process is called Bigdata matching.Canopy clustering is a datamining technique used for Bigdata matching. Creation of canopy clusters with sample dataset is shown in Figure 1.

In this clustering method, 'n' number of data points or records areselected at random from huge dataset and they are named as canopy centroid points.Two threshold values namely loose threshold($t_l$) and tight threshold($t_t$) are also determined. The canopy cluster formation is having two stages. In the first stage a cheap similarity metric is taken and in the second stage expensive similarity metric is taken. First stage is advantageous because with minimum computation, clusters are formed. Second stage involves maximum computation overhead.

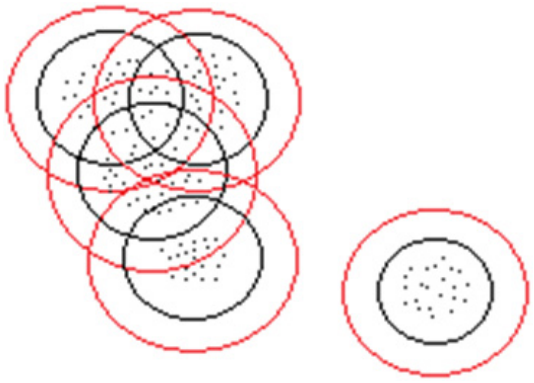

**Figure 1.** Creation of canopy clusters(red circles denote canopies,black circles denote their clusters and dots denote data points or records).

For all canopy centroid points the similarity measure has been calculated with all the remaining points in the whole dataset using cheap similarity metric. Canopies are formed using loose threshold $t_l$. For each canopy, the records having their similarity value with that centroid point less than or equal to $t_t$ are removed from the dataset and a canopy cluster is formed. Similarly 'n' canopy clusters are formed for 'n' canopies. In the latter stage, expensive similarity metric is applied to each cluster and the final set of candidate record pairs is achieved. This is our required data from the huge target dataset.

Advantages of standard Canopy Clustering:

1. It is suitable for Bigdata applications. (i.e., when the target data sets are very huge.)
2. Data sets can be huge in three ways.

- There can be huge number of records in the dataset.
- Each record can have many features.
- There can be many clusters.

his technique is efficient when the problem is big in all these three ways at once.(i.e.,millions ofdatapoints,many thousands of features and many thousands of clusters.)

3. Canopy clustering is most efficient method of clustering.

Though Canopy clustering is more appropriate for Big datamatching, it gives rise to the problem of overlapping clusters. This problem causes redundant pair comparisons when similar datapoints or records share more than one cluster.

## 3. Problem Statement

In the datamatching process, each cluster is taken at a time and subjected to expensive pairwise similarity computation. As each cluster has overlapping regions, these regions may be repeated while computing similarities.

Given 'n' number of clusters those are initially created from a bigdata pool by using canopy clustering.Then 'k' number of sub-clusters can be generated from 'n' clusters, where a subset of 'm'sub-clusters from 'k' sub-clusters may be repeated 'v' times or less than 'v' times. 'v' is the maximum degree of redundancy. This redundancy should be avoided in order to reduce the total datamatching time. The Kolb's approachuses the dedoop tool to solve this problem. But in the mapreduce phase of dedoop, unnecessary condition checking and large number of redundant similarity computations are observed. The BPRA approach avoids the condition checking overhead and redundant computations in the pre dedoop phase. BPRA does not need post-processing of final result. But it has large pre-processing overhead due to big data in real time. This should be reduced using any big data technique to improve the performance and scalability of big data matching.

## 4. Problem Solving and Innovative Content

In the ultimate cluster set, each cluster is divided into sub-clusters where the component sub-clusters of a cluster when collected form the total cluster. Now the target data pool is the sets of sub-clusters of all the clusters. For each cluster, a pair of sub-clusters is selected for big data matching with dedoop. In each such pair, a same sub-cluster may be taken twice to form the pair, or different sub-clusters of the same cluster may be taken. Finally a set of pairs is obtained. From this set of pairs, the repeated(redundant) pairs are deleted. Finally a resultant set of unique linkage pairs is achieved which is the input to the big datamatching with a specific tool "dedoop". Parallel dedoop executions can be carried out to get the big datamatching result as early as possible. The BPRA approach is improvedand simplified by using the mapreduce technique. The process is incremental in nature and it is scalable. As the data is added continuously the clustering scenario also changes frequently leading to frequent updates to the clusters. The Hadoop Distributed File System(HDFS) does not support updation or modification of the data. Soanupdation process is equivalent to a delete operation followed by a write operation or vice versa. In the step 18 of PRAMR algorithm, write operationis performed first by storing the new written data in the variable $SC_{TOTAL}$. After each new write, all the data stored in D are deleted and the data in $SC_{TOTAL}$ are placed in D. This process is repeated for each test data set. The big data matching process is isolated from thesefrequentdeletes and writes. To increase the performance, incrementality, and scalability of the BPRA approach, the frequent deletes and writes are performed using one mapreduce job where as another mapreduce job is dedicated to generate candidate pairs based on expensive similarity computations by the dedooptool. The former mapreduce job is given higher priority than the latter. Based on priority scheduling, 'P' number of higher priority mapreduce jobs can be run followed by 'Q' number of lower priority mapreduce jobs periodically (Here P>>Q). i.e., big data matching is performed periodically where as deletes and updates are performed so frequently. The following are the algorithms for solving the problem. This improved BPRA approach is named as PRAMR algorithm. (Pre big data matching Redundancy Avoidance algorithm with MapReduce.)

The algorithm PRAMR satisfies the incrementality requirement. We intend to cope with the clustering problem in an incremental setting. 'D' is the clustered initial data pool. Each centroid point in C represents a cluster. Test data set 'nd' must be integrated within a

previously clustered data pool 'D'. Each data point in 'nd' must either be associated with atleast one cluster in C or lead to creation of a new cluster. If a data point in 'nd' is associated with multiple clusters, then it's cluster membership information CMI is updated to reveal all the clusters related to that data point.For example if a datapoint $d_i$ is associated with multiple clusters namely, $C_1$,$C_2$ and $C_3$then it's CMI is updated to $C_1\$C_2\$C_3\$$. The Jaccard similarity distance is taken to find whether a datapoint $d_i$ relates to a cluster $C_k$.

PRAMR(C,D,d,μ)
**Input:** A pool of target data D;
Initial set of cluster centroids C with 'p' number of centroids;
Similarity threshold μ;
New set of datapoints or records nd; (i.e., test set.)
**Output:**Non-empty sets ULP and $SC_{TOTAL}$ which can be taken as input for parallel Dedoop applications.
1: Let $C=\{C_1,C_2,\ldots..,C_p\}$,$D=\{d_1,d_2,\ldots..,d_n\}$nd=$\{nd_1,nd_2,\ldots..,nd_m\}$;
2:**for**i=1....m **do**
3: $CMI(nd_i)= \emptyset$;
4: $C^1= \emptyset$;
5:**for**k=1....p**do**
6:**if** $SIM(nd_i,C_k)>= \mu$ **then**
7: $CMI(nd_i)$ $CMI(nd_i)$ U "$C_k$$";
8:**end if**
9:**end for**
10:**if**$CMI(nd_i)= \emptyset$ **then**
11: $C_{p+1}= nd_i$ ;
12:$C^1C^1$ U $\{C_{p+1}\}$;
13: p=p+1;
14:**end if**
15: D = D U $\{nd_i\}$ ;
16: C = C U $C^1$;
17:**end for**
18: $SC_{TOTAL}$ = MR_CLUSTER_INFO_UPDATE(D,$C^1$); // write operation.
19: Erase all the data in D. // delete operation.
20: D= $SC_{TOTAL}$.
21: $SCI_{ALL}$ = SCI($SC_{TOTAL}$);
22: ULP =FORM-ULPS($SCI_{ALL}$);

(Note: $C_k$ is the $k^{th}$centroid in the set of centroids C.
$CMI(nd_i)$ extracts the cluster membership information of datapointnd$_i$.
SCI denotes sub-cluster information.
$SCI_{ALL}$ extracts the sub-cluster information of all sub-clusters in the set $SC_{TOTAL}$.
$SCI_k$extracts the sub-cluster information of the member sub-cluster $SC_k$in the set $SC_{TOTAL}$.
$SIM(nd_i,C_k)$ computes the Jaccard similarity distance between data points $nd_i$ and $C_k$.
ULP denotes set of Unique Linkage Pairs.
$SC_{TOTAL}$ denotes set of all sub-clusters which are obtained by combining all reducer process outputs i.e. all $SC_{SUB}$ s.
MAP_CLUSTER_INFO_UPDATE( ) and REDUCE_CLUSTER_INFO_UPDATE( ) denote the mapper and reducer process of the mapreduce job MR_CLUSTER_INFO_UPDATE( ). )

MAP_CLUSTER_INFO_UPDATE(DPART,C1)
Input:A partition of target data set D= DPART = {dp1,dp2,....,dpq} where q is the total number of data points in DPART,b=0,s=0;
Output:parts of SCbs where SCb is a sub-cluster.
M1: for each dpi in DPART do
M2: for each Cj in C1do
M3: ifSIM(dpi ,Cj)>=μthen
M4: CMI(dpi) CMI(dpi) U "Cj$" ;
M5: end if
M6: end for
M7: for b=0…s do
M8: ifCMI(dpi) = SCI(SCb) then
M9: SCbSCbU {dpi};
M11: end if
M12: end for
M13: b=b+1;
M14: SCb= Ø;
M15: SCb = SCbU{dpi};
M16: SCI(SCb) CMI(dpi);
M17: s=b;
M18: end for

REDUCE_CLUSTER_INFO_UPDATE(SCUNION)
Input:Union of all parts of all SCbs= SCUNION
Output:A subset of all sub-clusters SCSUB
R1: Group all parts of SCb,which are having the same SCI value.

Jaccard similarity distance=SIM($d_i$,Centroid($C_k$))=($d_i \cap C$ entroid($C_k$))/($d_i$ U Centroid($C_k$)) (1)

If this distance is greater than or equal to the selected threshold 'μ' then the data point $d_i$ is associated with the cluster $C_k$.

The higher priority mapreduce job MR_CLUSTER_INFO_UPDATE( ) generates component sub-clusters for each cluster based on SCI or CMI value. The region with same SCI value is treated as a sub-cluster. Possible Unique Linkage Pairs of all the sub-clusters are formed using FORM_ULPS( ). For de-duplication within each sub-cluster,samesub-cluster is repeated in the UL(Unique Linkage Pair).The remaining unique linkage pairs are formed by checking the condition of common cluster in their corresponding SCI values. When both sub-clusters in the pair belong to common cluster then only that pair is a valid pair.

The algorithm REDUNDANCY-CHECK checks whether any pair is repeated. If a pair is repeated then it skips that pair. If a pair is a new one then it is added to the set ULP (Unique Linkage Pairs set). Let us once examine a sample overlapping clusters scenario for understanding.

FORM-ULPS($SCI_{TOTAL}$)
**Input:** empty set ULP, $SCI_{TOTAL}$={$SCI_1$, $SCI_2$,…, $SCI_N$}, N= Sizeof($SCI_{TOTAL}$) ;
**Output:**Non-empty set ULP.
FU1: **for** l=1 … N **do**
FU2: UL1=$SCI_l$^$SCI_l$;
FU3: b=REDUNDANCY-CHECK($UL_1$);
FU4:**if** b=0 **then**
FU5: ULPULP U {$UL_1$};
FU6:**end if**
FU7:**for** p=l+1…N **do**
FU8:**if** $SCI_l$=$SCI_p$ **then**
FU9: go to FU8;
FU10:**end if**
FU11:**if** $SCI_l \cap SCI_p \neq \emptyset$ **then**
FU12: UL2=$SCI_l$^$SCI_p$;
FU13: b=REDUNDANCY-CHECK(UL2);
FU14:**end if**
FU15:**if** b=0 **then**
FU16: ULPULP U{UL2};
FU17:**end if**
FU18:**end for**
FU19: **end for**

REDUNDANCY-CHECK(UL)
R1: SZ=sizeof(ULP);
R2:**for** q=1…SZ **do**
R3:**if** UL=ULPq**then**
R4: return 1;
R5:**end if**
R6:**end for**
R7: return 0;

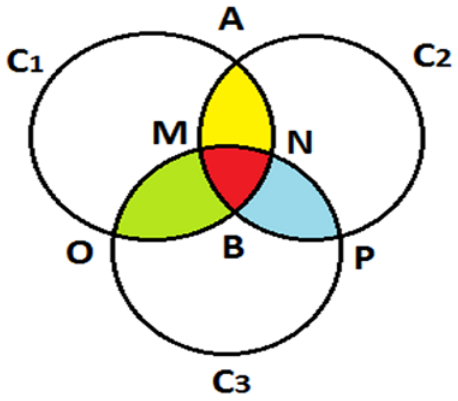


**Figure 2.** Sample overlapping among three canopy clusters C1,C2,C3.

The scenario of sample overlapping among three canopy clusters is shown in Figure 2. For simplicity sub-clusters and ULPs are named in Table 1. Let $C_1^1$,$C_2^1$,$C_3^1$ denote non-overlapping sub-cluster regions of $C_1$,$C_2$ and $C_3$respectively. The above scenario is a best example to understand the problem.There may be several overlapping clusters in realtime. Finally, the sets ULP and $SC_{TOTAL}$ are the results of the proposed algorithm.These can be taken as input for the parallel Dedoop applications. Each pair in the set ULP is considered as input to the big datamatching tool Dedoop where each sub-cluster from that pair is extracted from the set $SC_{TOTAL}$. The implementation of Dedoop tool is explained by Kolb, Thor and Rahm. Even though BPRA approach does not need post-processing of the final big datamatching results and it is advantageous compared to Kolb's approach avoiding unnecessary condition checking and similarity computation, It has considerable preprocessing overhead when the data reaches big data range. This overhead caused by BPRA is reduced to a maximum extent by PRAMR algorithm improving the performance, incrementality and scalability of the entire process.

**Table 1.** Sample canopy clusters,their sub-clusters and unique linkage pairs for Figure 2.

| Canopy cluster | subclusters | Corresponding Unique Linkage subcluster pairs |
|---|---|---|
| $C_1$ | $C_1^1$,AMN, MBN,- MOB | $C_1^1$-$C_1^1$, AMN-AMN, MBN-MBN, MOB-MOB, $C_1^1$-AMN, $C_1^1$-MBN, $C_1^1$-MOB, AMN-MBN, AMN-MOB, MBN-MOB |
| $C_2$ | $C_2^1$,AMN, MBN,NBP | $C_2^1$- $C_2^1$, NBP-NBP, $C_2^1$-AMN, $C_2^1$-MBN, $C_2^1$-NBP, AMN-NBP, MBN-NBP |
| $C_3$ | $C_3^1$,MOB, MBN,NBP | $C_3^1$- $C_3^1$, $C_3^1$-MOB, $C_3^1$-MBN, $C_3^1$-NBP, MOB-NBP |

## 5. Results and Comparison

**Table 2.** Execution times for different data pools

| Different sizes of data pools | Kolb's approach | BPRA | PRAMR |
|---|---|---|---|
| 1k | 15 | 14 | 10 |
| 2k | 31 | 28 | 23 |
| 3k | 46 | 42 | 36 |
| 4k | 60 | 53 | 44 |
| 5k | 78 | 68 | 54 |

The algorithm is checked for different datasets such as bank credentials data set and medical diagnosis datasets to group records pertaining to same bank and same disease respectively. The algorithm execution times in minutes corresponding to different datapools are tabulated in Table 2. Assume all clusters are uniformly distributed. The Hadoop 2.2.0 and ubuntu 13.10 operating system environment is used for mapreduce jobs.

The algorithm is compared with Kolb's approach and BPRA approach as shown in Figure-3 for different degrees of redundancy. A test sample of article-author information is used to match the authors related to common subject. 100k of data points are taken in the data pool and it is partitioned into 100 clusters.We successively increase the initial cluster size by 1000 data points up to s = 10,000. This results in a pair overlap of 25% for s = 2000 and 81% for s = 10,000. In the Kolb's approach checking is made for each pair in a cluster whether it has to be evaluated or redundant.

The BPRA approach eliminates this checking overhead. But when the data reaches big data range (>Tb), the BPRA also

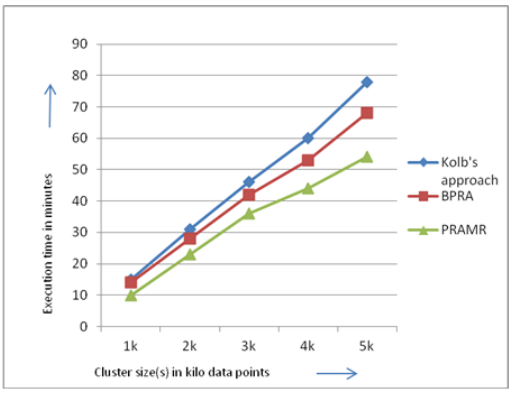


**Figure 3.** Execution times comparison for different degrees of redundancy.

Incurs a considerable preprocessing overhead. This preprocessing overhead can be reduced to a maximum extent using PRAMR algorithm. If the number of unnecessary checking are 'u' for a candidate pair, the total number of unnecessary checking for 'n' candidate pairs is 'nu'. If the time complexity of Kolb's approach is O(K), then the time complexity is 'nu' times reduced for BPRA approach excluding the preprocessing overhead. When data volume is huge, the time complexity for BPRA is O(K)/nu + O(P) where O(P) is the time taken for preprocessing. As the preprocessing is subjected to mapreduce in PRAMR algorithm, the total data is partitioned into 'm' number of partitions by using 'm' mappers in the mapreduce job MR_CLUSTER_INFO_UPDATE( ). So the time complexity of PRAMR becomes O(K)/nu + O(P)/m . As the time complexity is less when compared to other two approaches while giving the same result, this algorithm is more suitable for a big data scenario improving the scalability and performance of the entire process. The incremental nature of the algorithm is preserved by repeating the process for each test data set generated in equal time intervals. The experimental

results shown in Figure-3 prove that PRAMR is far better when compared to the other two approaches.

## 6. Conclusion

This paper has presented an efficient big data algorithm to avoid redundant similarity computations caused by overlapping clusters. The experimental results showed that the proposed approach is better when compared to Kolb's approach and BPRA approach. It reduces the preprocessing overhead to the maximum extent while maintaining the consistent results. It does not require post-processing of final results. In the future, our focus would be on other big data problems.

## 7. References


1. Kolb L, Thor A, Rahm E. Dedoop:Efficient deduplication with Hadoop. Proceedings of the VLDB endowment; Istanbul, Turkey. 2012; 5(12):1878–81.
2. Somasekhar G, Karthikeyan K. Dealing with the problem of overlapping clusters prior to data matching using clustering techniques. Proceedings of the 5th international conference on Advances in Computer Engineering; Kochi, India. 2014 Dec. p. 588–97.
3. Kolb L, Thor A, Rahm E. Don't match twice: redundancy-free similarity computation with mapReduce. DanaC'13 Proceedings of the 2nd Workshop on Data Analytics in the Cloud; Newyork: USA. 2013. p.1–5.
4. Agichtein E, Ganti V. Mining reference tables for automatic text segmentation.Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining; Newyork: USA. 2004. p. 20–9.
5. Monge AE, Elkan CP. Automatic segmentation of text into structured records. Proceedings of the ACM SIGMOD conference on management of data; Newyork: USA. 2001; 30(2):175–86.
6. Cesario E, Folino F, Locane A, Manco G and Ortale R. Boosting text segmentation via progressive classification. J KnowlInfSyst 2008;15(3): 285–320.
7. Kyoo-sung N, Doo-sik L. Bigdata Platform Design and Implementation Model, Indian Journal of Science and Technology. 2015 Aug; 8(18):1–8.
8. Fellegi IP, Sunter AB. A theory for record linkage. J Am Stat Assoc.1969; 64(328):1183–210.
9. Winkler WE. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. Proceedings of the section on survey research methods, American Statistical Association; Alexandria, Egypt. 1990. p. 354–59.
10. Sarawagi S, Bhamidipaty A. Interactive deduplication using active learning. Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining; New York, USA. 2002. p. 269–78.
11. Cohen WW, Richman J. Learning to match and cluster large high-dimensional data sets for data integration. Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining; New York, USA. 2002. p. 475–80.
12. McCallum A, Nigam K, Ungar LH. Efficient Clustering of High Dimensional Datasets with application to Reference Matching.Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining; New York, USA. 2000. p. 169–78.
13. Chaudhuri S, Ganti V, Motwani R. Robust identification of fuzzy duplicates. Proceedings of the international conference on data engineering; Washington DC, USA. 2005. p. 865–76.
14. Anderberg MR. Cluster Analysis for Application. Academic Press; 1973.
15. Christen P. A survey of indexing techniques for scalable record linkage and deduplication.IEEE Transactions on Knowledge and Data Engineering. IEEE; 2012 Sep; 24(9):1537–55.
16. Baxter R, Christen P, Churches T. A comparison of fast blocking methods for record linkage. Proceedings of ACM-SIGKDD'03 workshop on Data Cleaning, Record Linkage and Object Consolidation; Washington DC, USA. 2003. p. 1–6.
17. Bhattacharya I, Getoor L. Iterative record linkage for cleaning and integration. Proceedings of the SIGMOD workshop on research issues on data mining and knowledge discovery; New York,USA. 2004. p. 11–8.
18. Hu H, Li G, Feng J. Fast Similar Subgraph Search with Maximum Common Connected Subgraph Constraints.IEEE International Congress on Big Data. Santa Clara, CA: IEEE; 2013. p. 181–88.
19. Kolb L, Thor A, Rahm E. Multipass Sorted Neighborhood Blocking with MapReduce. Journal:Computer Science - Research and Development. 2012 Feb; 27(1):45–63.
20. KolbL, Thor A, Rahm E. Parallel sorted neighborhood blocking with mapreduce. Proceedings of BTW'11; Kaiserslautern, Germany. 2011. p. 45–64.
21. Costa G, Manco G, Ortale R. An incremental clustering scheme for data de-duplication. Journal:Datamining and Knowledge Discovery. 2010 Jan; 20(1):152–87.