International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

Sampling Selection Strategy for Large Scale Deduplication of synthetic and real datasets using Apache Spark

Gaurav Kumar[1], Kharwad Rupesh[2], Md.Shahid Equabal[3], N Rajesh[4]

[1,2,3,4] *Dept. of Information Science and Engineering, The National Institute of Engineering, Mysuru, Karnataka, India*

Abstract - *Due to the enormous increase in the generation of information by a number of sources, the requirement of several new applications has become mandatory. These applications may be media streaming, digital libraries etc. Data quality is degraded due to the presence of duplicate pairs. This is a very serious issue regarding the quality and authenticity of data.*

Therefore, data deduplication *task is necessary to be performed with the datsets. It detects and removes duplicates and provide efficient solutions to this problem. In very large datasets, it is very difficult to produce the labeled set from the information provided by the user as compared to the small datasets.*

Guilherme Dal Bianco, Renata Galante, Marcos Andre Goncalves, Sergio Canuto, and Carlos A. Heuser proposed a Two-stage sampling selection strategy *(T3S) [17] that selects a reduced set of pairs to tune the deduplication process in large datasets. T3S follows two stages to select the most representative pairs. The first stage contains a strategy to produce balanced subsets of candidate pairs for labeling. The second stage proposes a rule-based active selective sampling incrementally invoked to remove the redundant pairs in the subsets created in the first stage in order to produce an even smaller and more informative training set. This training set can be furthur utilized. Active fuzzy region selection algorithm is proposed to detect the fuzzy region boundaries by using the training set. Thus, T3S reduces the labeling effort substantially while achieving superior matching quality when compared with state-of-the-art deduplication methods in large datasets. But, performing the deduplication in a distributed environment offers a better performance over the centralized system in terms of speed and flexibility. So, in this work, a distributed approach is implemented for the above method using Apache Spark. Also, a comparison is done between* Two-stage sampling selection strategy *and FSDedup. It shows that T3S reduces the training set size by redundancy removal and hence offers better performance than FSDedup. A graph is plotted for the same.*

Index Terms— Scala, Apache spark, Deduplication, Sampling Selection Strategy, T3S

1 INTRODUCTION

Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data. This technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. In the deduplication process, unique chunks of data, or byte patterns, are identified and stored during a process of analysis. As the analysis continues, other chunks are compared to the stored copy and whenever a match occurs, the redundant chunk is replaced with a small reference that points to the stored chunk. Deduplication may occur in-line, as data is flowing, or post-process, after it has been written. With post-process deduplication, new data is first stored on the storage device and then a process at a later time will analyze the data looking for duplication. This is the process where the deduplication hash calculations are created on the target device as the data enters the device in real time. If the device spots a block that it already stored on the system it does not store the new block, just references to the existing block.

There has been a dramatic growth in the generation of information from a wide range of sources such as mobile devices, streaming media, and social networks. Data quality is also degraded due to the presence of duplicate pairs with misspellings, abbreviations, conflicting data, and redundant entities, among other problems. Record deduplication aims at identifying which objects are potentially the same in the data repository. In the context of large datasets, it is a difficult task to produce a replica-free repository. A typical deduplication method is divided into three main phases:

1.1 Blocking:

The Blocking phase aims at reducing the number of comparisons by grouping together pairs that share common features. A simplistic blocking approach, for example, puts together all the records with the same first letter of the name and surname attributes in the same block, thus avoiding a quadratic generation of pairs (i.e., a situation where the records are matched all-against-all).

1.2 Comparison:

The Comparison phase quantifies the degree of similarity between pairs belonging to the same block, by applying some type of similarity function (e.g. Jaccard, Levenshtein, Jaro).

1.3 Classification:

Finally, the Classification phase identifies which pairs are matching or non-matching. This phase can be carried out by selecting the most similar pairs by means of global

thresholds, usually manually defined [1], [2], [3], [4] or learnt by using a classification model based on a training set.

2 RELATED WORKS

Record deduplication studies have offered a wide range of solutions exploiting supervised, semi-supervised, and unsupervised strategies. Supervised and unsupervised strategies rely on expert users to configure the deduplication process. The former assumes the presence of a large training set consisting of the most important patterns present in the dataset (e.g., [8], [9]). The latter relies on threshold values that are manually tuned to configure the deduplication process (e.g., [1], [2], [10], [4]).Committee-based strategies for deduplication, called ALIAS and Active Atlas respectively, are outlined in [7] and [11]. The committee identifies the most informative pairs to be labeled by the user as the unlabeled pairs that most classifiers disagree regarding their prediction. Active Atlas employs a committee composed by decision trees, while ALIAS uses randomized decision trees, a Naive Bayes and/or a SVM classifier. An alternative active learning method for deduplication was proposed in [5], where the objective is to maximize the recall under a precision constraint. The approach creates an N-dimensional feature space composed of a set of similarity functions that are manually defined, and actively selects the pairs by carrying out a binary search over the space. However, the N-dimensional binary search may lead to a large number of pairs been queried, increasing the manual effort [6]. In [6], a strategy, referred as ALD, is proposed to map any active learning approach based on accuracy to an appropriate deduplication metric under precision constraints. This kind of approach projects a quality estimation of each classifier by means of points in a two-dimensional space. ALD conducts a binary search in this space to select the optimal classifier that respects the precision constraint. The space dimensions correspond to the classifiers' effectiveness, estimated by means of an ―oracle‖. The pairs used for training are selected by the IWAL active learning method [12].

3 PROPOSED MODEL AND SYSTEM DESIGN

3.1 Terminologies

Sig-Dedup has been used to efficiently handle large deduplication tasks. It maps the dataset strings into a set of signatures to ensure that similar substrings result in similar signatures. The signatures are computed by means of the inverted index method. To overcome the drawback of quadratic candidate generation [15] prefix filtering [16] is used. The prefix filtering is formally defined below: Definition 1: Assume that all the tokens in each record are ordered by a global ordering ϑ. Let p-prefix of a record be the first p tokens of the record. If Jaccard(x,y) ≥ t, then the (p)-prefix of x and (p)-prefix of y must share at least one token. where, Jaccard(x,y) is defined as: J(x,y) = ____ Prefix length of each record u is calculated as |u| − t · ⌈ ⌉ +1 , where t= Jaccard similarity threshold.

3.2. Framework

The framework for large scale deduplication using the two stage sampling selection strategy is illustrated in Figure 4.1. First, the candidate pairs are produced after identifying the blocking threshold. Next, T3S strategy is employed. In its first stage, T3S produces small balanced subsamples of candidate pairs. In the second stage, the redundant information that is selected in the subsamples is removed by means of a rule-based active sampling. These two steps work together to detect the boundaries of the fuzzy. Finally, the classification approach is introduced which is configured by using the pairs manually labeled in the two stages.

All these steps are implemented in the distributed environment using Apache Spark.

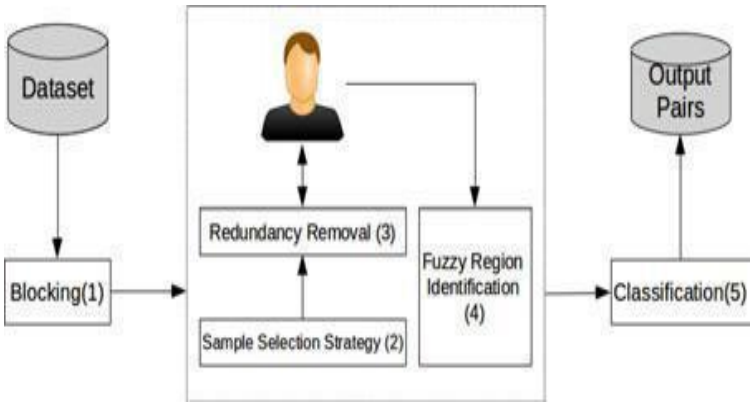

Figure-3.1 Framework for Large Scale Deduplication using T3S

3.2.1. Determining Blocking Threshold

In large datasets, it is not feasible to run the Sig-Dedup filters with different thresholds due to the high computational costs. So, a stopping criterion is introduced. The method employed is defined as:

Definition 2: Consider a subset S, created from a randomly sampled dataset D and a range of thresholds with fixed step thj = 0.2, 0.3,. . ., and 0.9. The subset S is matched using each threshold value thj. The initial threshold will be the first thj that results in a number of candidate pairs smaller than the number of records in S.

After finding the global initial threshold value for the blocking process, the entire dataset is matched to create the set of candidate pairs.

3.2.2. First stage of T3S: Sample Selection Strategy

The first stage of T3S adopts the concept of levels to allow each sample to have a similar diversity to that of the full set of pairs. The ranking, created by the blocking step, is fragmented into 10 levels (0.0-0.1,0.1-0.2, 0.2-0.3,. . ., and 0.9-1.0), by using the similarity value of each candidate pair.

The similarity value of each candidate pair is found using Jaccard similarity. This fragmentation produces levels composed of different matching patterns to prevent non-matching pairs dominating the sample.

3.2.3. Second Stage of T3S: Redundancy Removal

Several pairs selected inside each level are composed of redundant which does not help to increase the training set diversity. Selective Sampling using Association Rules is used to remove redundancy in the information randomly selected as shown in Figure 3.2.

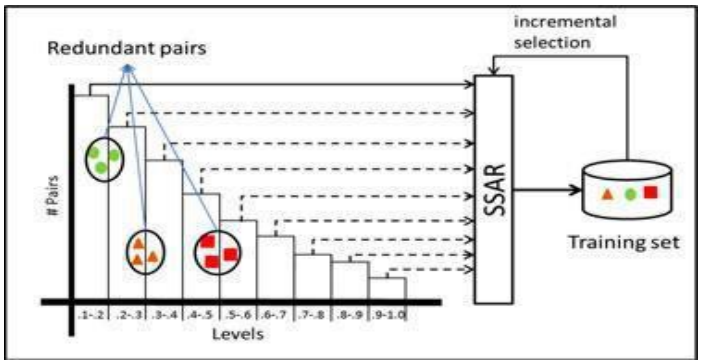

Figure-3.2 Illustration of SSAR

3.2.3.1. SSAR Method

The second stage of T3S aims at incrementally removing the non-informative or redundant pairs inside each sample level by using the SSAR (Selective Sampling using Association Rules) active learning method [14]. In the beginning, when the training set D is empty, SSAR selects the pair that shares most feature values with all other unlabeled pairs to initially compose the training set. SSAR selects an unlabeled pair ui for labeling by using inferences about the number of association rules produced within a projected training set specific for ui. The projected training set is produced by removing from the current training set D instances and features that do not share features values with ui. When compared with the current training set, the unlabeled pair with less classification rules over the projected training set represents the most informative pair. If this pair is not already present in the training set, it is labeled by the user and inserted into the training set. After this, a new round is performed and the training set must be re-projected for each remaining unlabeled pair to determine which one is most dissimilar when compared to the current training set. If the selected pair is already present in the training set, the algorithm converges.

3.2.3.2. Computational Complexity

The computational complexity of SSAR is O(S * |U| * 2m), where —S|| is the number of pairs selected to be labeled, —|U||| represents the total number of candidate pairs and —m|| is the number of features. —|U||| pairs must be re-projected each time that a labeled pair is attached to the current training set, producing a computationally unfeasible time to process large datasets.

3.2.4. Fuzzy Region Detection

Definition 3: Let Minimum True Pair-(MTP) represent the matching pair with the lowest similarity value among the set of candidate pairs.

Definition 4: Let Maximum False Pair-(MFP) represent the non-matching pair with the highest similarity value among the set of non-matching pairs.

The fuzzy region is detected by using manually labeled pairs.

The user is requested to manually label pairs that are selected incrementally by the SSAR from each level. First, SSAR is invoked to identify the informative pairs incrementally inside each level to produce a reduced training set. The pairs labeled within each level are used to identify the MFP and MTP pairs.

MTP and MFP pairs define the fuzzy region boundaries the similarity value of the MTP and MFP pairs identifies α and β values. The pairs belonging to the fuzzy region are sent to the Classification Step.

3.2.5. Classification

The Classification step aims at categorizing the candidate pairs belonging to the fuzzy region as matching or non-matching.

The classifier, T3S-NGram maps each record to a global sorted token set and then applies both the Sig-Dedup filtering and a defined similarity function (such as Jaccard) to the sets. The NGram Threshold is required to identify the matching pairs inside the fuzzy region using the NGram tokenization.First, the similarity of each labeled pair is recomputed by means of a similarity function along with the NGram tokenization. After this, the labeled pairs are sorted incrementally by the similarity value and a sliding window with fixed-size N is applied to the sorted pairs. The sliding window is relocated in one position until it detects the last windows with only non-matching pairs.

Finally, the similarity value of the first matching pair encountered after the last windows with only non-matching pairs, defines the NGram threshold value. The candidate pairs that survive the filtering phase and meet the Ngram threshold value are considered as matching ones.

4 CONCLUSIONS

In this project, we have proposed a distributed algorithm for large scale deduplication using sampling selection strategy which produces the same result as the centralized system but speeds up the process by a considerable amount. As followed from our experiments, our distributed approach

performs the same processes in a lesser time with a greater flexibility and scalability. We have also compared the T3S approach with the FSDedup. T3S reduces the user effort by reducing the training set size and results in a lesser number of matching pairs.

ACKNOWLEDGEMENT

The authors can acknowledge any person/authorities in this section. This is not mandatory.

REFERENCES

[1] R. J. Bayardo, Y. Ma, and R. Srikant, —Scaling up all pairs similarity search,|| in Proc. 16th Int. Conf. World Wide Web, pp. 131–140, 2007.

[2] S. Chaudhuri, V. Ganti, and R. Kaushik, —A primitive operator for similarity joins in data cleaning,|| in Proc. 22nd Int.Conf. Data Eng., p. 5, Apr. 2006.

[3] J. Wang, G. Li, and J. Fe, —Fast-join: An efficient method for fuzzy token matching based string similarity join,|| in Proc. IEEE 27th Int. Conf. Data Eng., 2011, pp. 458–469.

[4]C. Xiao, W. Wang, X. Lin, J. X. Yu, and G. Wang,—Efficient similarity joins for near-duplicate detection,|| ACM Trans. Database Syst., vol. 36, no. 3, pp. 15:1–15:41, 2011.

[5]A. Arasu, M. Gotz, and R. Kaushik, —On active learning of record matching packages,|| in

[6]K. Bellare, S. Iyengar, A. G. Parameswaran, and V. Rastogi,—Active sampling for entity matching,|| in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 1131–1139.

[7]S. Sarawagi and A. Bhamidipaty, —Interactive deduplication using active learning,|| in Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2002, pp. 269–278.

[8] M. G. de Carvalho, A. H. Laender, M. A. Goncalves, and A. S. da Silva, —A genetic programming approach to record deduplication,|| IEEE Trans. Knowl. Data Eng., vol. 24, no. 3, pp. 399–412, Mar. 2012.

[9]J. Wang, G. Li, J. X. Yu, and J. Feng, —Entity matching: How similar is similar,|| Proc.

[10]R. Vernica, M. J. Carey, and C. Li, —Efficient parallel set-similarity joins using mapreduce,|| in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2010, pp. 495–506.

[11]S. Tejada, C. A. Knoblock, and S. Minton, —Learning Domain-independent string transformation weights for high accuracy object identification,|| in Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2002, pp. 350–359.

[12]A. Beygelzimer, S. Dasgupta, and J. Langford, —Importance weighted active learning,|| in Proc. 26th Annu. Int. Conf. Mach. Learn., pp. 49–56, 2009.

[13] G. Dal Bianco, R. Galante, C. A. Heuser, and M. A. Gonalves, —Tuning large scale deduplication with reduced effort,|| in Proc. 25th Int. Conf. Scientific Statist. Database Manage. 2013, pp. 1–12.

[14] R. M. Silva, M. A. Gonc¸alves, and A. Veloso, —A two-stage active learning method for learning to rank,|| J. Assoc. Inform. Sci. Technol., vol. 65, no. 1, pp. 109–128, 2014.

[15]M. Bilenko and R. J. Mooney, —On evaluation and Training-set construction for duplicate detection,|| in Proc. Workshop KDD, 2003, pp. 7–12.

[16]A. Arasu, C. R_e, and D. Suciu, —Large-scale deduplication with constraints using dedupalog,|| in Proc. IEEE Int. Conf. Data Eng., 2009, pp. 952–963.

[17] Guilherme Dal Bianco, Renata Galante, Marcos Andr_e Gonc¸alves, Sergio Canuto, and Carlos A. Heuser, —A Practical and Effective Sampling Selection Strategy for Large Scale Deduplication,|| IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 9, September 2015.