ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 4, June 2013

A Genetic Programming Approach for Record Deduplication

L.Chitra Devi[1], S.M.Hansa[2] , Dr.G.N.K.Suresh Babu[3]

Research Scholars, Dept. of Computer Applications, GKM College of Engineering and Technology, Chennai, Tamil Nadu, India[1], [2]

Professor and Head, Dept. of Computer Applications, GKM College of Engineering and Technology, Chennai, Tamil Nadu, India[3]

Abstract: In this article we are going to discuss about how genetic programming can be used for record deduplication. Several systems that rely on the integrity of the data in order to offer high quality services, such as digital libraries and e-commerce brokers, may be affected by the existence of duplicates, quasi-replicas, or near-duplicates entries in their repositories. Because of that, there has been a huge effort from private and government organizations in developing effective methods for removing replicas from large data repositories. This is due to the fact that cleaned, replica-free repositories not only allow the retrieval of higher-quality information but also lead to a more concise data representation and to potential savings in computational time and resources to process this data. In this work, we extend the results of a GP-based approach we proposed to record deduplication by performing a comprehensive set of experiments regarding its parameterization setup. Our experiments show that some parameter choices can improve the results to up 30%. Thus, the obtained results can be used as guidelines to suggest the most effective way to set up the parameters of our GP-based approach to record deduplication.

Keywords : Genetic Programming, DBMS, Duplication, Optimisation

I. INTRODUCTION

Normally, organizations become conscious of practical precise disparities or inconsistencies while integrating data from diverse sources to implement a data warehouse. Such problems belong to the category called data heterogeneity. With the increase in size of the database the problem intensifies taking into account the huge amount of computational resource required for examination and removal of duplicate records. Duplicates can occur out of numerous scenarios, for instance when a large database is updated by an external source and registry numbers are not accessible or are in error. File systems often contain superfluous copies of information: identical files or sub-file regions, perhaps stored on a single host, on a shared storage cluster, or backed-up to secondary storage. Deduplicating storage systems take advantage of this redundancy to decrease the essential space needed to contain the file systems. Deduplication can work at either the sub-file or whole-file level. Data deduplication policies can be classified according to the basic data units they handle. In this context, mainly two main data deduplication strategies can be defined: File-level deduplication, in which only a single copy of each file is stored. Two or more files are known as identical if they have the same hash value. This is a very popular service imbibed in multiple products. Block-level deduplication, which breaks files into blocks and stores only a single copy of each block. The system could either use fixed-sized blocks or variable-sized chunks. The architecture of the deduplication solution is modeled by two basic approaches. In the target-based approach deduplication is managed by the target data storage perhipherals or service, while the client is ignorant of any deduplication that might occur. Source based deduplication is performed on the data at the client side only before it is transferred. Particularly, there is communication between client software and the backup server to check for the presence of files or blocks. Two well-known source de-duplication methods, source local chunk-level deduplication and source global chunk level de-duplication have been proposed in the past to address the above mentioned problem by erasing the redundant data chunks before transferring them to the remote backup destination. The studies expose that, due to the out-of-memory fingerprint accesses to massive backed-up data, chunk-level de-duplication has an inherent latency and suffers throughput problem that affects the backup performance. In source global chunk-level de-duplication, this overhead of massive disk accesses will regulates the deduplication process and which will result in increase of backup window. While in source local chunk-level de-duplication, the overhead is reduced by searching the duplicate chunks at the same client. This alleviated overhead, however, limits the compression ratio, which results in increases of backup window due to the increased data transmission cost. Therefore, there is immediate need to achieve a balance between de-duplication efficiency and deduplication overhead for the maintenance of a shorter backup window than existing solutions. There are several other methods that have been proposed for the deduplication purpose which are having efficiency and accuracy. The methods are deduplication using genetic algorithm, semantic methods, cloud services. Above mentioned problems have been solved by the deduplication methods which have been modeled using GA. This research has been done to find the optimization techniques that are having some performance superiority over these existing methods.

II. RELATED WORK

Record deduplication is a research topic that has attracted a lot of attention in the database field and related areas. As already mentioned, the existence of replicas in data repositories can lead to inconsistencies that may severely affect several types of service, causing damage for companies and governmental institutions. In an attempt to solve these inconsistencies, some works have proposed the creation of similarity functions capable of combining information contained in the data repositories in order to identify when a pair of records constitutes or not a replica. Elmagarmid et al. [2007] classify these approaches into two categories, exemplified next: Ad-Hoc. These are methods that usually depend on some knowledge of a particular domain or on specific distance metrics (for example, for strings characters) Training-based. These are methods that depend on some type of training, supervised or semi supervised, for the identification of replicas, such as probabilistic and machine learning approaches. In the following, we briefly describe some relevant work that illustrates these two categories of methods.

A.Ad-hoc Methods

Chaudhuri et al. [2003] have proposed a linkage algorithm that receives a record from an archive or data repository as input and searches for another record in a reference archive that "matches" with the former, according to some predefined similarity function. The matched records are selected according to some user defined minimum similarity threshold, allowing that more than one candidate record be returned as a possible answer. In this case, the user is responsible for choosing the duplicated record most similar to the original one.

An information retrieval style method is used in WHIRL [Cohen 2000], a database management system that supports similarity joins between relations that include textual attributes. In this system, textual attributes are represented according to the vector space model [Salton 1989] and term weights are calculated using the well-known TF-IDF scheme in order to determine when tuples from two relations should be joined based on the similarity of their attribute values. Carvalho and da Silva [2003] also use the vector space model for computing similarity among fields from different data sources and evaluate four distinct strategies to assign weights and combine the similarity scores of each field. As a result of their experiments, they found that using evidence extracted from individual attributes improves the result of the replica identification task.

B. Training-based Methods

Since these are more related to the work presented in this article, we cover training-based methods in more details in this section. Newcombe et al. [1959] were the first ones to address the record deduplication problem as a Bayesian inference problem and proposed the first method to automatically handle replicas. However, their method was considered empirical [Elmagarmid et al. 2007] since it lacks a deeper statistical ground. After Newcombe et al.'s work, Fellegi and Sunter [1969] have proposed an elaborated probabilistic method to deal with the problem of evidence combination. Their proposed method requires the definition of two thresholds for replica identification. If the similarity value between two records is above the positive threshold, these records are considered as replicas; if it is below the negative threshold, the records are considered non-replicas; and if the similarity value is between these two thresholds, the records are classified as "possible

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 4, June 2013

replicas", requiring a manual classification by a specialist. The Fellegi and Sunter's method does not require explicit training examples but, on the other hand, it does require that the data administrator explicitly defines the two aforementioned thresholds, a non trivial task since this definition is repository-dependent. This method has dominated the field for more than two decades until new deduplication methods were developed by the statistical and machine learning communities. Febrl2 [Christen 2008] is one of the best known software packages that implement this method. On the other hand, pure machine learning based methods need some training examples, which should present characteristics similar to those of the duplicated data, so that they can generalize their solutions (deduplication functions) for the entire repository. The main problem with this type of method is the cost of creating such training data, which in some cases may be infeasible. Bilenko and colleagues [Bilenko et al. 2003; Bilenko and Mooney 2003] exploit a machine learning technique to improve both the similarity functions that are used to compare record fields and the way pieces of evidence are combined to identify replicas. In their work, extracted evidence is encoded as feature vectors, which are then used to train an SVM (Support Vector Machines [Joachims 2002]) classifier to better combine them for the replica identification task that is performed by a system called MARLIN (Multiply Adaptive Record Linkage with INduction). This system also uses several blocking strategies to improve its effectiveness when clustering similar records. Active Atlas [Tejada et al. 2001] is an object identification system that aims at learning mapping rules for identifying similar objects (records) from distinct data sources. The process involves two steps. First, a candidate mapping generator proposes a set of possible mappings between the two sets of objects by comparing their attribute values and computing similarity scores for the proposed mappings. Then, a mapping rule learner determines which of the proposed mappings are correct by learning the appropriate mapping rules for that specific application domain. This learning step is exectued by using a decison tree. What mainly differs the method behind this system from other machine learning based methods is that it tries to reduce the training effort by relying on user-provided information for selecting the most relevant training examples. In [de Carvalho et al. 2006], the authors follow a genetic programming approach to improve the results of the Fellegi and Sunter's method. They apply this machine learning technique to balance the weight vectors produced by the Fellegi and Sunter's method in order to generate a better evidence combination than the simple linear summation used by that probabilistic model. Following their previous work, de Carvalho et al. [2008] have proposed a new GP-based method that finds the best evidence combination for generating a deduplication function within a generic framework that is independent of any other technique. Since record deduplication is a very costly task, even for small repositories, the proposed method tries to find the best evidence combination that also maximizes performance, using for this only a subset of the repository as training data.

C.GP based Record DeDuplication Method

In this section we provide an overview of the GP-based record deduplication method used in this work and first introduced in [de Carvalho et al. 2008]. We start by presenting a brief introduction to the genetic programming technique followed by an explanation of how the record deduplication problem was modeled using this technique. We conclude with a more detailed description of the main steps of the method.

III. GENETIC PROGRAMMING OVERVIEW

Genetic Programming (GP) is one of the best known and mostly used techniques of evolutionary computing and can be seen as an adaptive heuristics whose basic ideas are inspired in the natural selection process. It is a direct evolution of programs or algorithms used for inductive (supervised) learning, initially applied to optimization problems. One of the main characteristics of evolutionary techniques is their capability to deal with problems with multiple objectives, normally modeled as environmental constraints during the evolutionary process. These techniques are also known for their capability of searching for solutions in large, possibly unbounded, search spaces in which the optimal solution may be unknown, thus producing solutions close to the optimum. What mainly differentiates GP from other evolutionary techniques (such as genetic algorithms and evolutionary systems) is the representation of concepts and the solution as a computer program, being the data manipulated only by these programs. Computer programs have the flexibility needed to deal with a multitude of problems. Moreover, the program structures being evolved do not have size limitations and can dynamically vary during the process according to the requirements of the problem. The most used GP representation for solutions of a specific problem are trees and graphs. Besides choosing a representation for the solution of the problem, it is also necessary to define a set of terminals and functions to perform the task of solving the problem. Terminals are inputs, constants, or nodes with no input, which are the leaves of the trees, while the set of functions comprises operators, declarations, and basic or user-defined functions used to manipulate the values of the terminals [Koza 1992]. The leaf nodes are located at the end of the branch while functions are located in the internal nodes. The search space is the space built from all possible programs that can be constructed with the functions and terminals defined for the problem domain. A GP algorithm evolves a population of tree-represented individuals, i.e., a set of possible solutions for the problem. In each generation of the evolutionary process, individuals are evaluated according to a specific quality metric that measures how well they solve the problem, calculated by a fitness function. The fitness value is used as a criterion to select the best individuals, which will transmit their characteristics to the future generations through operations like crossover and mutation. In this work, we use the ranking method [Koza 1992] as a selection method. This method chooses the k best individuals based on the fitness values to be included in the next generation and to participate in the genetic operations. While the number of new individuals is smaller than the desired population size n (a parameter to be set), two individuals are picked among those selected as described above and have their "genetic material" exchanged in the crossover operation with a certain probability (another parameter to be set) to generate a new individual that combines the genetic material of their "parents". More specifically, the crossover operation is implemented by randomly choosing one node of each of the two trees (the two individuals) and exchanging the subtrees below them. The role of the crossover operation is to combine good solutions towards the most promising solutions in the search space. Finally, also with a certain probability (yet another parameter), the mutation operation is applied, to introduce new individuals (i.e., solutions) in the population, increasing its diversity. This is useful, for example, to avoid that the whole evolutionary process gets trapped in local minima during the search process for a good solution. In this work, the mutation of an individual (i.e., a tree) is performed by first randomly selecting one of its nodes. We then replace the node (and its corresponding subtree) by a new randomly generated subtree, without exceeding a maximum tree depth d. The whole process is repeated until a target fitness value f or a maximum number of generations g is reached. In the end of the process, the best individual, i.e., the one with the best fitness value, which usually belongs to the last generation in the evolutionary process, is chosen as the final solution for the problem in hand.

IV. GP APPLIED TO THE RECORD DEDUPLICATION PROBLEM

In this work, as in [de Carvalho et al. 2008], we use a representation based on trees for the individuals of the GP process which, in our case, represent possible record deduplication functions. More specifically, to perform the record deduplication, we use evidence combination functions in which each evidence E is a pair <attribute, similarity function> that represents the application of a specific similarity function on the values of a given attribute of the data repository. For example, to deduplicate a table from a relational database with attributes name, surname, age and address, using the Jaro-Winkler (JW) [Winkler 1999] similarity function, we would have the following pieces of evidence: E1<name, JW>, E2<surname, JW>, E3<age, JW> and E4<address, JW>.

Experiments performed in [de Carvalho et al. 2008] show that the GP-based record deduplication method is able to adapt the suggested deduplication functions according to changes in the replica identification thresholds necessary to classify pairs of records. Therefore, the user does not need to worry about setting up the "best" values for this threshold according to the data repository, given that the suggested deduplication functions can adapt automatically, maintaining the effectiveness of the solutions, despite changes in this threshold.

V. CONCLUSIONS AND FUTURE ENHANCEMENTS

The duplicate information is stored from different sources, which require more spaces to store replicas. The bitter experience of the several events proved that data loss is common to a modern enterprise. Different types of approaches are involved to solve those problems. SQL based data deduplication, Chunk based clustering architecture and Genetic approach models all are very tedious in their respective field of implementation because of their huge dependence on file organization or file system. Moreover all of these have indecent time complexity which is a matter of great concern. But in the case of this new proposed model as huge as be the relational database with huge number of tables (which is the most anticipated consequence) this proposed model will perform more optimistically than a tiny database. The domain identifier, key identifier, row id, column id are the key role players in this model. Apart from that the binary search and the table index

position are the main optimisation factor for this model. Hence the length of a data of a particular field of a particular row and column are not the matter of concern
regarding space wastage, rather it proves our model more practically, in that case. Since the entire algorithm is basically based on the row column approach that is why this can only be implemented in RDBMS initially, though our goal will be to renovate this concept in a broader sense so that it can cover up the all the branches of DBMS. Identifying and handling replicas is important to guarantee the quality of the information made available by modern data storage services. Services that rely on the integrity of the data in order to offer high quality services, such as digital libraries and e-commerce brokers, may be affected by the existence of duplicates, quasi-replicas, or near-duplicates entries in their repositories. Thus, for this reason, there has been a huge effort in developing effective and efficient methods for removing replicas from large data repositories. This article is relevant since deduplication is a time and resource consuming task. Because of that, any gain (even very small ones) may represent a relevant saving in time and computational resources.

REFERENCES

[1] Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). Modern Information Retrieval. ACM Press/Addison-Wesley.

[2] Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998). Genetic Programming - An Introduction: on the Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers.

[3] Bell, R. and Dravis, F. (2006). Is you data dirty? and does that matter? Accenture Whiter Paper - http://www.accenture.com.

[4] A.K.Elmagarmid, P.G. Ipeirotis, and V.S. Verykios, (2007) "*Duplicate record detection: A survey*", IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 1, pp 1-16.

[5] H. Biggar, (2007) "*Experiencing data de-duplication: Improving efficiency and reducing capacity requirements*" White Paper, the Enterprise Strategy Group.

[6] Bilenko, M. and Mooney, R. J. Adaptive duplicate detection using learnable string similarity measures. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, pp. 39–48, 2003.

[7] Carvalho, J. C. P. and da Silva, A. S. Finding similar identities among objects from multiple web sources. In Proceedings of the ACM International Workshop on Web Information and Data Management. New Orleans, USA, pp. 90–93, 2003.

[8] Chaudhuri, S., Ganjam, K., Ganti, V., and Motwani, R. Robust and efficient fuzzy match for online data cleaning. In Proceedings of the ACM SIGMOD International Conference on Management of Data. San Diego, USA, pp. 313–324, 2003.

[9] Geer, D. Reducing the storage burden via data deduplication. IEEE Computer 41 (12): 15–17, 2008.

[10] Gu, L. and Baxter, R. Decision models for record linkage. Selected Papers from Australasian Data Mining Conference vol. 3755, pp. 146–160, 2006.

[11] Joachims, T. Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. Kluwer Academic Publishers, Norwell, USA, 2002.

[12] Vrable, M., S. Savage and G.M. Voelker, 2009. Cumulus: Filesystem backup to the cloud. ACM Trans. Storage. DOI: 10.1145/1629080.1629084

[13] Winkler, W.E., 2001 Record linkage software and methods for merging administrative lists. The Pennsylvania State University.

[14] Zhu, B., K. Li and H. Patterson, 2008. Avoiding the disk bottleneck in the data domain deduplication file system. Proceedings of the 6th USENIX Conference on File and Storage Technologies, (FAST '08), USENIX Association Berkeley, USA.