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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

An Iterative Approach to Record Deduplication

M. Roshini Karunya, S. Lalitha, B.Tech., M.E.,

II ME (CSE), Gnanamani College of Technology, A.K.Samuthiram, India[1]

Assistant Professor, Gnanamani College of Technology, A.K.Samuthiram, India[2]

ABSTRACT: Record deduplication is the task of identifying, in a data repository, records that refer to the same real world entity or object in spite of misspelling words, typos, different writing styles or even different schema representations or data types [1]. The existing system aims at providing Unsupervised Duplication Detection method which can be used to identify and remove the duplicate records from different data sources. UDD, which for a given query, can effectively identify duplicates from the query result records of multiple web databases. Two cooperating classifiers, a Weighted Component Similarity Summing Classifier (WCSS) and Support Vector Machine (SVM) are used to iteratively identify the duplicate records from the non duplicate record and we also present a Genetic Programming (GP) approach to identify record deduplication. Since record deduplication is a time consuming task even for small repositories, our aim is to foster a method that finds a proper combination of the best pieces of evidence, thus yielding a deduplication function that maximizes performance using a small representative portion of the corresponding data for training purposes. We propose two more algorithms namely Particle Swarm Optimization (PSO), Bat Algorithm (BA) to improve the optimization.

Index Terms – Data mining, duplicate records, genetic algorithm

I INTRODUCTION

1.1 Record deduplication

Deduplication is a key operation in integrating data from multiple sources. The main challenge in this task is designing a function that can resolve when a pair of records refers to the same entity in spite of various data inconsistencies. Record deduplication is the task of identifying, in a data repository, records that refer to the same real world entity or object in spite of misspelling words, typos, different writing styles[1].

1.2 Data mining in record deduplication

Several systems that rely on consistent data 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-duplicate entries in their repositories. Because of that, there have been significant investments from private and government organizations for developing methods for removing replicas from its data repositories.

Deduplication one data set or linking several data sets are increasingly important tasks in the data preparation steps of many data mining projects. The aim of such linkages is to match all records relating to the same entity. Research interest in this area has increased in recent years, with techniques originating from statistics, machine learning, information retrieval, and database research being combined and applied to improve the linkage quality, as well as to increase performance and efficiency when deduplication or linking very large data sets. Data De-duplication is essentially a data compression technique for elimination of coarse-grained redundant data[2].

II EXISTING SYSTEM

In existing system they present a genetic programming (GP) approach to record deduplication. Their GP-based approach is also able to automatically find effective deduplication functions, even when the most suitable similarity function for each record attribute is not known in advance. This is extremely useful for the non-specialized user, who does not have to worry about selecting these functions for the deduplication task. In addition, we show that their approach is also able to adapt the suggested deduplication function to changes on the replica identification boundaries used to classify a pair of records as a match or not. This releases the user from the burden of having to choose and tune these parameter values. . Our approach combines several different pieces of evidence extracted from the data content to produce a deduplication function that is able to identify whether two or more entries in a repository are replicas or not. Since record deduplication is a time consuming task even for small repositories, their aim is to foster a method that finds a proper combination of the best pieces of evidence, thus yielding a deduplication function that maximizes performance using a small representative portion of the corresponding data for training purposes.

III PROPOSED SYSTEM

In the proposed system we presents two approach called particle swarm optimization, bat algorithm to overcome the difficulty and complexity of the genetic programming Approach .The new algorithms finds the best optimization solution for random selection of the input values and removes the duplicate records in the system. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. The particle swarm optimization concept consists of, at each time step, changing the velocity of each particle toward its pbest and lbest locations. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations. The proposed system technique bat algorithm for record deduplication BA is a relatively new population based meta heuristic approach based on hunting behavior of bats. Optimization is nothing but selection of a best element from some set of available alternatives

IV MODULES DESCRIPTION

4.1 Preprocessing

In this module they create the preprocessing stage. They consider the set of documents and collect the all information and keywords from the document. Then extract the keywords from the collection of documents. This keyword extraction is Minimize the analyzing process. The users read each document and gave a positive or negative judgment to the document against a given topic. Because, only users perfectly know their interests and preferences, these training documents accurately reflect user background knowledge. User profiles are acquired by semi automated techniques with limited user involvement. These techniques usually provide users with a list of categories and ask users for interesting questions.

4.2 Generating potential duplicate and non-duplicate vector from web datasets

This module deals with web data sets which can act as an input, that are extracted from the database, after they perform preprocessing method.

- Cora data set , which is a collection of research being a text string and its Manually segmented into multiple fields such as Author, Title, Year ,etc.
- The preprocessing technique was done by Exact Matching method.

- Cora is a noisy data set in which there are some fields that are not Correctly segmented. It is used to identify potential vector and non-duplicate record from the data sets.

4.3 Component Weight Assignment

This module deals with the records that are retrieved from the web database, first it will assign weights to the fields and identify similar and dissimilar records iteratively.

```
Implementation of UDD algorithm
Input : Potential duplicate vector set P
        Non-duplicate vector set N
Output: Duplicate Vector set D
$C_1$: a classification algorithm with adjustable parameters
W that identifies duplicate vector pairs from P
$C_2$: a supervised classifier
Algorithm:
1. D=Ø
2. Set the parameters W of $C_1$ according to     N
3. Use $C_1$ to get a set of duplicate vector pairs $d_1$ from P
4. Use $C_1$ to get  a set duplicate vector pairs f from N
5. P=P – $d_1$
6. While $|d_1| \neq 0$
7. N'=N-f
8. D=D+$d_1$+f
9. Train $C_2$ using D and N'
10. Classify p using $C_2$ and get a set of newly identified duplicate vector pairs $d_2$
11. P=P - $d_2$
12. D=D+$d_2$
13. Adjust the parameters W of $C_1$ according to N' and D
14. Use $C_1$ to get a new set of duplicate vector pairs $d_1$ from P
15. Use $C_1$ to get a new set of duplicate vector pairs f from N
16. N=N'
 Return D
```

4.4 Weighted Component Similarity Summing Classifier (WCSS)

C_1—Weighted Component Similarity Summing (WCSS) Classifier

At the beginning, it is used to identify some duplicate vectors when there are no positive examples available. Then, after iteration begins, it is used again to cooperate with C_2 to identify new duplicate vectors. Because no duplicate vectors are available initially, classifiers that need class information to train, such as decision tree and Naive Bayes, cannot be used. An intuitive method to identify duplicate vectors is to assume that two records are duplicates if most of their fields that are similar presented in the web database. To evaluate the similarity between two records, it combines the values of each component in the similarity vector for the two records. Suppose in different fields may have different importance when this decide whether two records are duplicates. The importance is usually data-dependent, which, in turn, depends on the query in the Web database scenario and w_i Є [0,1] is the weight for the *i*th similarity component, which represents the importance of the *i*th field and it evaluate the duplicate entries.

4.5 Support Vector Machine (SVM) Classifier

C_2 - Support Vector Machine (SVM) Classifier

After detecting a few duplicate vectors whose similarity scores are bigger than the threshold using the WCSS classifier. The positive examples are identified duplicate vectors in D, and negative examples, namely the remaining non duplicate vectors in N'. Hence, it can train another classifier C2 and use this trained classifier to identify new duplicate vectors from the remaining potential duplicate vectors in P and the non duplicate vectors in N'.

4.6 Genetic algorithm for identifying the record deduplication

GP evolves a population of length-free data structures, also called individuals, each one representing a single solution to a given problem. During the evolutionary process, the individuals are handled and modified by genetic operations such as reproduction, crossover, and mutation, in an iterative way that is expected to spawn better individuals (solutions to the proposed problem) in the subsequent generations.

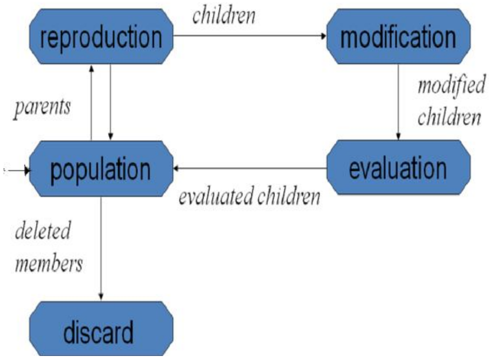

Fig. The GA Cycle of Reproduction

In this work, the GP evolutionary process is guided by a generational evolutionary algorithm. This means that there are well defined and distinct generation cycles. We adopted this approach since it captures the basic idea behind several evolutionary algorithms. The steps of this algorithm are the following:

1. Initialize the population (with random or user provided individuals). [1]
2. Evaluate all individuals in the present population, assigning a numeric rating or fitness value to each one.
3. If the termination criterion is fulfilled, then execute the last step. Otherwise continue.
4. Reproduce the best n individuals into the next generation population. [1]
5. Select m individuals that will compose the next generation with the best parents.
6. Apply the genetic operations to all individuals selected. Their offspring will compose the next
Population. Replace the existing generation by the generated population and go back to Step 2.
7. Present the best individual(s) in the population as the output of the evolutionary process. [1]

4.7 Bat Algorithm

BA is a relatively new population based metaheuristic approach based on hunting behavior of bats. [8] In this algorithm possible solution of the problem is represented by bat positions. Quality of the solution is indicated by the best position of a bat to its prey. [5] BA has been tested for continuous constrained optimization problems. The new algorithm finds the best optimization solution for random selection of the input values and removes the duplicate records in the system.

4.8 Particle swarm optimization to identify record deduplication

In this module we implement the particle swarm optimization for finding the record deduplication. PSO algorithm is another example of population based algorithms[8]. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms. The system is initialized with a population of random solutions and searches for optima by updating generations. PSO is initialized with a group of random particles and then searches for optima by updating enerations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution it has achieved so far. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. Based on the local and global best value, we can easily find the re duplication records. Compared with genetic algorithms, the information sharing mechanism in PSO is significantly different

.

4.9 Comparison of algorithms

In this module we compare the performance of genetic algorithm, bat algorithm, particle swarm optimization. Every algorithm is carefully monitored and finally particle optimization shows the best performance in finding the most duplicate records.

V CONCLUSION AND FUTURE WORK

In this project we have seen concepts about Record Deduplication , Genetic Algorithm, Bat Algorithm, and PSO. Implementing Record Deduplication we have been able to find duplicate records. Duplicate detection is an important step in data integration and this method is based on offline learning techniques, which requires training data. In the Web database scenario, where records to match are greatly query-dependent, a pertained approach is not applicable as the set of records in each query's results is a biased subset of the full data set. To overcome this problem, it presents an unsupervised, online approach UDD, for detecting duplicates over the query results of multiple Web databases. Two classifiers, WCSS and SVM are used cooperatively in the convergence step of record matching to identify the duplicate pairs from all potential duplicate pairs iteratively.

The genetic programming approach combines several different pieces of evidence extracted from the data content to produce a deduplication function that is able to identify whether two or more entries in a repository are replicas or not. Their aim is to foster a method that finds a proper combination of the best pieces of evidence, thus yielding a deduplication function that maximizes performance using a small representative portion of the corresponding data for training purposes. In our proposed system we increased the best optimization solution to deduplication of the records.

The final design criterion in choosing a deduplication vendor is where the deduplication occurs. Once again you have two choices: at the source or at the destination. In future work we plan to introduce new algorithm to find the duplication records at the third party vendors also, and reduce the time to find the duplication records, without affecting the speed of the process. We also introduce various data samples and find the duplication records. Since deduplication is time consuming and the optimization is less. Other enhancements must be employed to increase the optimization. Particle Swarm Optimization is one such enhancement. Many other enhancements can also employed to improve finding duplicate records. The future work is improving the result of formerly used algorithms

VI REFERENCES

[1] Moise´s G. de Carvalho, Alberto H.F. Laender, Marcos Andre´ Gonc¸alves, and Altigran S. da Silva, A Genetic Programming Approach to Record Deduplication, IEEE transactions on knowledge and data engineering, vol. 24, no. 3, march 2012.

[2] Atish Kathpal, Matthew John and Gaurav Makkar , Distributed Duplicate Detection in Post-Process Data De-duplication.

[3] A. E. Monge and C. P. Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In Proceedings of the SIGMOD 1997 Workshop on Research Issues on Data Mining and Knowledge Discovery, pages 23–29, Tuscon, AZ, May 1997.

[4] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery.

[5] Altringham, J. D.: Bats: Biology and Behaviour, Oxford University Press, (1996).

[6] Bilenko, M. and Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. Proceedings of the 9th ACM SIGKDD conference, Washington DC, August 2003.

[7] Baxter, R., Christen, P. and Churches, T.: A Comparison of Fast Blocking Methods for Record Linkage. ACM SIGKDD '03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, August 27, 2003, Washington, DC, pp. 25-27.

[8] Koffka Khan, Ashok Sahai, A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context, I.J. Intelligent Systems and Applications, 2012, 7, 23-29, Published Online June 2012 in MECS.

[9] Bertolazzi, P., De Santis, L. and Scannapieco, M.: Automated record matching in cooperative information systems. Proceedings of the international workshop on data quality in cooperative information systems, Siena, Italy, January 2003.

[10] M.A. Hernández and S.J. Stolfo, –Real- World Data Is Dirty: Data Cleansing and the Merge/Purge Problem, Data Mining and Knowledge Discovery, vol. 2, no. 1, pp. 9-37, Jan. 1998.