International Journal of Communication Networks and Security

Volume 2 | Issue 1 Article 17

January 2013

A BIO GENETIC PROCESS TO REPLICAS-FREE REPOSITORY

N. VENKATESWARLU
Search Results Web results Audisankara Institute of Technology,,Nellore,India,
N.VENKATESWARLU@gmail.com

D. SIREESHA SIREESHA
Search Results Web results Audisankara Institute of Technology,Nellore,India, D.SIREESHA@gmail.com

A BIO GENETIC PROCESS TO REPLICAS-FREE REPOSITORY

N. VENKATESWARLU[1], D. SIREESHA[2]

[1]PG student, CSE, ASIT, Nellore, [2]Associate Professor, ASIT, Nellore

Abstract- 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. This is due to the fact that clean and replica-free repositories not only allow the retrieval of higher quality information but also lead to more concise data and to potential savings in computational time and resources to process this data. In this paper, we propose a genetic programming approach to record deduplication that combines several different pieces of evidence extracted from the data content to find a deduplication function that is able to identify whether two entries in a repository are replicas or not. As shown by our experiments, our approach outperforms an existing state-of-the-art method found in the literature. Moreover, the suggested functions are computationally less demanding since they use fewer evidence. In addition, our genetic programming approach is capable of automatically adapting these functions to a given fixed replica identification boundary, freeing the user from the burden of having to choose and tune this parameter.

Index Terms- *Database administration, evolutionary computing and genetic algorithms, database integration.*

1. INTRODUCTION

The increasing volume of information available in digital media has become A challenging problem for data administrators. Usually built on data gathered from different sources, data repositories such as those used by digital libraries and e-commerce brokers may present record with disparate structure. Also, problems regarding low-response time, availability, security, and quality assurance become more difficult to handle as the amount of data gets larger.

Today, it is possible to say that the capacity of an organization to provide useful services to its users is proportional to the quality of the data handled by its systems. In this environment, the decision of keeping repositories with "dirty" data (i.e., with replicas, with no standardized representation, etc.) goes far beyond technical questions such as the overall speed or performance of data management systems. The solutions available for addressing this problem require more than technical efforts, they need management and cultural changes as well.

To better understand the impact of this problem, it is important to list and analyze the major consequences of allowing the existence of "dirty" data in the repositories. These include, for example: 1) performance degradation—as additional useless data demand more processing, more time is required to answer simple user queries; 2) quality loss—the presence of replicas and other inconsistencies leads to distortions in reports and misleading conclusions based on the existing data; 3) increasing operational costs—because of the additional volume of useless data, investments are required on more storage media and extra computational processing power to keep the response time levels acceptable.

To avoid these problems, it is necessary to study the causes of "dirty" data in repositories. A major cause is the presence of duplicates, quasi replicas, or near-duplicates in these repositories, mainly those constructed by the aggregation or integration of distinct data sources. The problem of detecting and removing duplicate entries in a repository is generally known as record deduplication (but it is also referred to in the literature as data cleaning record linkage and record matching).

Existing System:

The impact, it is important to list and analyze the major consequences of allowing the existence of "dirty" data in the repositories.

A function used for record deduplication must accomplish distinct but conflicting objectives: it should efficiently maximize the identification of record replicas while avoiding making mistakes during the process. the existing approaches to replica identification depend on several choices to set their parameters, and they may not be always optimal. Setting these parameters requires the accomplishment.

A major cause is the presence of duplicates, quasi replicas, or near-duplicates in these repositories, mainly those constructed by the aggregation or integration of distinct data sources. The impact, it is important to list and analyze the major consequences of allowing the existence of "dirty" data in the repositories.

The problem of detecting and removing duplicate entries in a repository is generally known as record deduplication.

Proposed System:

They propose a number of algorithms for matching citations from different sources based on edit distance, word matching, phrase matching, and subfield extraction. a matching algorithm that, given a record in a file (or repository), looks for another record in a reference file that matches the first record according to a given similarity function. The matched reference records are selected based on a user-defined minimum similarity threshold. individuals The 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. we presented a GP-based approach to record deduplication. Our approach is able to automatically suggest deduplication functions based on evidence presentin the data repositories. The suggested functions properly combine the best evidence available in order to identify whether two or more distinct record entries are replicas (i.e., represent the same real-world entity)

Module Description

1. Matching Algorithm

A method that exploits general similarity functions adapted to a specific domain, we can mention There the authors propose a matching algorithm that, given a record in a file (or repository), looks for another record in a reference file that matches the first record according to a given similarity function. The matched reference records are selected based on a user-defined minimum similarity. A number of algorithms for matching citations from different sources based on edit distance, word matching, phrase matching, and subfield extraction

2. Record De duplication

The problem of detecting and removing duplicate entries in a repository is generally known as record de-duplication (but it is also referred to in the literature as data cleaning record linkage and record matching. we present a genetic programming (GP) approach to record de-duplication. Our approach combines several different pieces of evidence extracted from the data content to produce a deduplication function

3. Genetic Programming

Genetic Programming is one of the best known evolutionary programming techniques. It can be seen as an adaptive heuristic whose basic ideas come from the properties of the genetic operations and natural selection system. we use GP as a generic framework to address the record deduplication problem independently of any other technique. As we showed in our experiments, our GP-based approach achieves better results than a state-ofthe- art method

4. Evolutionary process

It is a direct evolution of programs or algorithms used for the purpose of inductive learning (supervised learning), initially applied to optimization problems. GP, as well as other evolutionary techniques, is also known for its capability of working with multi objective problems, that are normally modeled as environment restrictions during the evolutionary process. The main aspect that distinguishes GP from other evolutionary techniques (e.g., genetic algorithms, evolutionary systems, genetic classifier systems) is that it represents the concepts and the interpretation of a problem as a computer program.

System design

UML Diagram

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Data Flow Diagram

A data floe diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. The process in the context level diagram is exploded into other process at the first level DFD. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

Use case diagrams

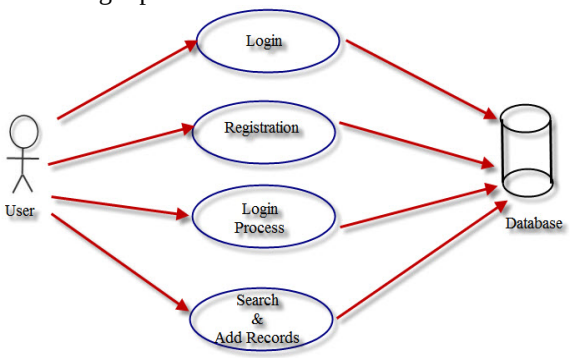
A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

Actors:

- User
- Data base

Use cases:

- Login
- Registration
- Login process



Sequence Diagram

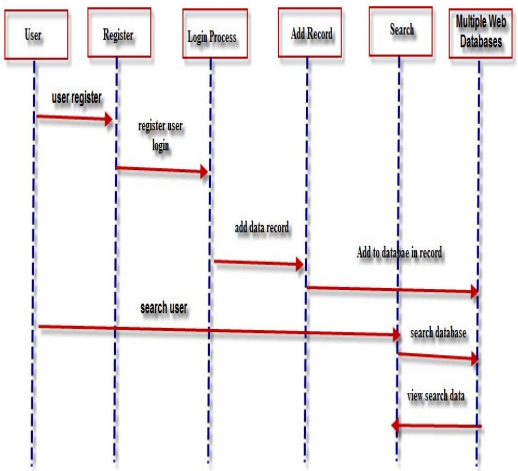
A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of message exchanged between the objects needed to carry out the functionality of the scenario. The diagrams are read left to right and descending. The example below shows an object of class 1 start the behaviour by sending a message to an object of class 2. Messages pass between the different objects until the object of class 1 receives the final message.

Objects:

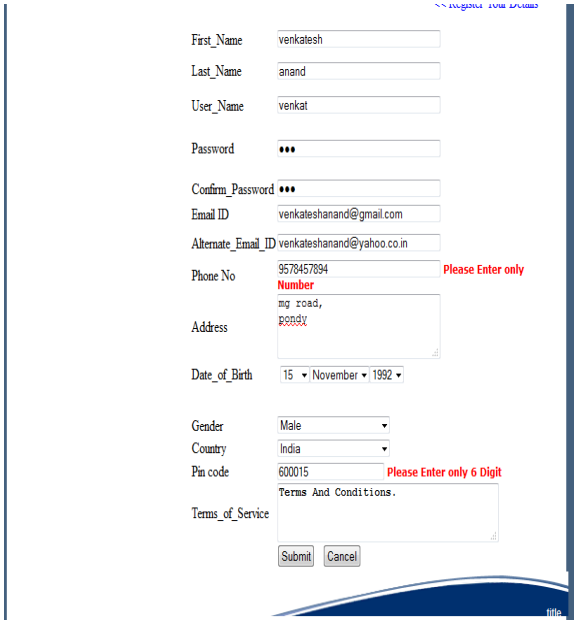
- user
- data base

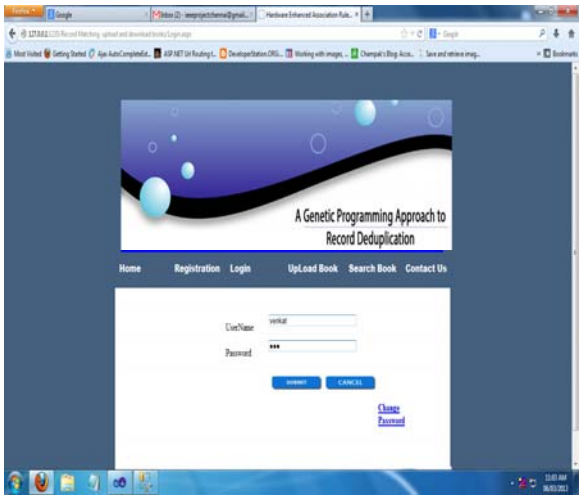
Sequences:

- Login
- Registration
- Login process
- Research and add records
- Multiple web data basses



7. RESULTS





CONCLUSION:

Identifying and handling replicas is important to guarantee the quality of the information made available by data intensive systems such as digital libraries and e-commerce brokers. These systems rely on consistent data to offer high-quality services, and may be affected by the existence of duplicates, quasi replicas, or near-duplicate entries in their repositories. Thus, for this reason, there have been significant investments from private and government organizations for developing methods for removing replicas from large data repositories. the two real data sets used in the comparison experiments, our approach outperformed the baseline best result in at suggested functions used less evidence which means that they are computationally less demanding. Additionally, unlikely most deduplication approaches described in the literature that use exactly the same similarity function for all available attributes, ours is

capable of combining distinct similarity functions that best fit each attribute considered. Thus, our approach is able to automatically choose the best function for each specific case, which is certainly one of the reasons of our better results.

REFERENCES

[1] M. Wheatley, “Operation Clean Data,” CIO Asia Magazine, http://www.cio-asia.com, Aug. 2004.

[2] N. Koudas, S. Sarawagi, and D. Srivastava, “Record Linkage:Similarity Measures and Algorithms,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 802-803, 2006.

[3] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, “Robust and Efficient Fuzzy Match for Online Data Cleaning,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 313-324, 2003.

[4] I. Bhattacharya and L. Getoor, “Iterative Record Linkage for Cleaning and Integration,” Proc. Ninth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery, pp. 11-18, 2004. DE CARVALHO ET AL.: A GENETIC PROGRAMMING APPROACH TO RECORD DEDUPLICATION

[5] Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.

[6] practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia.

[7] Data Communications and Networking, by Behrouz A Forouzan.

[8] Computer Networking: A Top-Down Approach, by James F. Kurose.

[9] Operating System Concepts, by Abraham Silberschatz. http://www.sourcefordgde.com, http://www.networkcomputing.com/