Token-Based Data Cleaning Technique for Online Student Registration

May Thet Swe; Dr. Nwe Nwe
University of Computer Studies, Hpa-An
maythet.thet@gmail.com;

Abstract

The data cleaning is the process of identifying and removing the errors in today's data collection database from different sources. The duplicate elimination problem of detecting database records that are approximate duplicates, which describe the same real world entity, is an important data cleaning problem. Data cleaning is very important and most of the organizations are in the need of quality data. The quality of the data needs to be improved in the database before proceeding into next step.

This paper presents data cleaning technique for the student registration system. Students can apply all Majors depend on their choices (1st priority, 2nd priority and so on). Students can apply a specialized major more than once before they have a chance to attend. It is very important for the University Administration to allow a student for only one major. Data Cleaning Process use the system to develop whether data comparison and removing errors. Token-Based Data cleaning Method is one of the simple and easy.

1. Introduction

Instead of detecting the user input of duplication the duplicate data can only be eliminated after storing in the database. A Token based data cleaning algorithm or technique should be used in the system development. A token-Based data cleaning algorithm or technique is useful whenever input data comparing and eliminate duplicate records.

A student can apply a specialized major once before they have a chance to attend. The problem is occurred when the system checked whether the student is accepted for one major or not. A student name Mg Mg and the Student NO. is SHK-0001 applied for three times. In the 1st time, the student typed the student NO. as SHK-0001 , 2nd time is SHK/0001 and 3rd is SHK0001. The system check, there is no data duplication because Student No. are different. Human can understand that the Student No. are the same. A computer cannot understand as it is equal or not equal.

Data cleaning is the process of clearing up databases by detecting and removing errors and inconsistencies from data of different multiple representations of the same real-world entity. It focuses on eliminating variations in data contents and reducing data redundancy aimed at improving the overall data consistency. Data cleaning, also called data cleansing or scrubbing. Data cleaning first detects dirty records by determining whether two or more records represented syntactically different while being semantically equivalent. It cleans the dirty records by retaining only one copy of records that are exact duplicates.

The organization of this paper is as follows: Section 2 presents the related work of the system. Section 3 discusses about the Data Cleaning approaches used in this system. Section 4 presents the proposed system design and Section 5 has the system implementation. Section 6 describes the conclusion of the system.

2. Related Work

There are several approaches for the duplicate detection for the data integration. Bitton et al. [2] sort on designated fields to bring potentially identical records together in a large data file. However, sorting is based on “dirty” fields, which may fail to bring matching records together, and its time complexity is quadratic in the number of records. Hernandez et al. [4], [5] solves the merge/purge problem in a large database by forming keys from some selected fields, sorting the entire data set on the keys, clustering the sorted records and using a scanning window of a fixed size to reduce the number of comparisons. Record comparison is still based on the original dirty records. The equational theory used in the multi-pass version of the work is a time consuming process. The basic field matching algorithm [1] extracts and sorts atomic strings within fields, finds the number of strings that match and computes the match score used to decide if the two fields are the same. The accuracy of the basic field matching algorithm is dependent on the match score threshold for deciding if any two input string match. The work described in [9] introduces the idea of field pre-processing with external data source, prior to sorting and comparison phases as well as “tokenizing of fields”. Pre-processing the dirty records with external data source like birth registry may not always be feasible and final comparison of strings for a match still involves the entire long strings. Work in [3], [7] enhance the data integration and cleaning process with declarative operators that allow for dynamic and interactive cleaning.

While existing techniques have used tokens for bringing likely duplicate records together [4], [5], [6], used pre-determined match score thresholds to decide on a match between two input strings [9], [1], depended on external or interactive input during duplicate detection [3], [6], [7], achieving a high recall (cleaning accuracy) in a reasonable time, which is less dependent on match score thresholds and external intervention, are data cleaning research goals this paper contributes to.

This paper proposes a token-based data cleaning algorithm, which first defines smart tokens from most important fields of records, compares and

identifies duplicate records with those tokens. Token-based technique achieves a better result than the record-based techniques of comparable algorithms. By using short lengthened tokens for record comparisons, a high recall/precision is achieved. The technique also drastically lowers the dependency of the data cleaning on match "threshold" choice.

3. Data Cleaning

Data quality refers an 'error-free' approach in the data warehouse. The quality of data needs to be increased by using the data cleaning techniques. Existing data cleaning techniques used to identify record duplicates, missing values, record and field similarities and duplicate elimination [3]. The main objective of data cleaning is to reduce the time and complexity of data processing and increase the quality of data in the corresponding database.

There are several existing data cleaning techniques that are being used for different purposes. 'Similarity functions' are used to find the similarity between records and fields [19]. 'Duplicate elimination functions' are used to determine whether two or more records represent the same real world object [4].

3.1. Dirt in the Source Data

There are two levels of dirt exists, namely, field or attribute level dirt and record level dirt. The field level dirt is the dirt that occurs when each field in a record is considered in isolation. Other field level dirt include (1) typographic errors, (ii) different addressing conventions (in address field), etc. Record level dirt is the combination of all the fields' dirt in a given row. It is the kind of duplication in other records. An obvious implication of record level dirt is "that duplicates are easily determined". This system uses Token-based Cleaning algorithm to detect the record level dirt, "duplications".

3.2. Token based Cleaning Algorithm

Token based data cleaning algorithm accepts "dirty" sources tables and returns "cleaned" data tables. A user selects two or three most important fields of records, compares and identifies duplicate records with those tokens, compares and ranks them based on their power to uniquely identify records. The elements in the selected fields area tokenized, those uniquely identifying fields of the table are used as different domain sort keys to produce sorted token-tables. Token-records in close neighborhood are compared for a match and a new Id is generated for records. The steps of the Token-based cleaning algorithm are as follows:

- **Step 1**: Selection and Ranking of fields: The user expected to select and rank the fields that could be confined to perfectly discriminate one record from another.
- **Step 2**: Extraction and Formation of Token: This step requires a scanner to tokenize and decompose the elements in the ranked fields.

The scanner will discard some tokens which are considered unimportant, such as Title token like "Mr.", "Ms." and "Dr." are excluded in "name" elements, the word like " the", "of", "for", "in" are considered unimportant in publication title.

4. Proposed System

This system presents the data cleaning approach used in eliminating duplicate records from user registration system. A student may apply more than one program and only one program has to be accepted. In such case, duplicate detection must be performed. Token based Cleaning approach is used for duplicate detection. It selects the main attributes as the token and sentence similarity algorithm is used to detect the duplicates. In this system, student's Name, student's Roll No, NRC (ID) Number and Address are used as tokens in this system. Figure 1 presents the process flow of the system.

Token-Based Data Cleaning is simple and easy to understand to solve the problem. Token means a unique set of string to use data comparison and checking data duplication. A unique set of string is a string getting from the original set of string by eliminating the delimiters. The delimiters for the Student No. are '-', '/', ',', and '#', When the system sees the delimiters, It eliminates those and produce pure Student No. The pure Student No.. is called a 'Token', unique set of string.

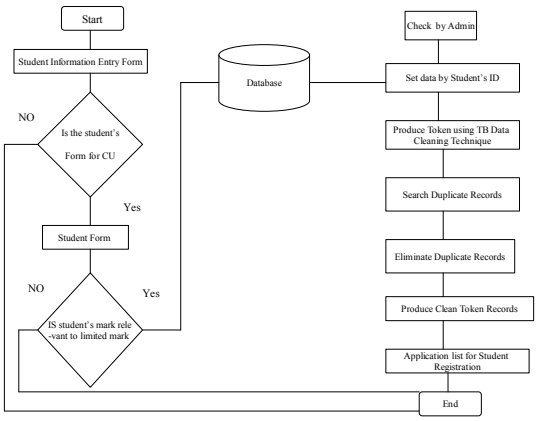

Figure 1: Process Flow of the System

6. System Implementation

This system is developed as Web based University site, where students can apply programs that University provides. A student may apply more than one program as his / her desire. To remove the duplicate application, this system uses Token-based cleaning approach.

Two cleaning tasks to be carried out on the student registration table are: (i) duplicate detection, and (ii) duplicate elimination. Duplicate detection requires a combination of (pieces of) information from two or more fields to find if two or more records are the same. Duplicate elimination task ensures that only one copy of records found to be duplicates is retained.

This system is developed using Microsoft Visual Studio .Net 2008. ASP .Net C# is used to implement the system. Microsoft Access 2003 is used to store the students' registration data.

6.1. Processes of the Token based Cleaner

Token-Based Data Cleaning Technique can clean the unclean data from database and produce as Token. Token uses data matching and comparison. There are four steps included in this approach.

- Selection and Ranking of fields
- Extraction and Formation of Tokens
- Sorting of Tokens and
- Duplicate Detection

6.1.1. Selection and Ranking of Fields

The system stores every student application forms in the database. The admin user searches for the application forms of the student that apply the computer university. Total mark validation process is also performed for applied program. For the token based cleaning process, the system select Student's ID, Student's Name, NRC and Student's Address are used to detect the duplication.

6.1.2. Extraction and Formation of Tokens

This system entails divisible tokens and indivisible tokens. Token records in the close neighborhood are compared from a match from the database. There are 3 types of tokens.

Numeric Tokens:
Consists only digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
Example; Date of Birth , Student Number
The delimiter for that Numeric Tokens will be "/", "-", ",", "#" and so on.

Alphabetic Tokens:
Consists of only alphabets (a A-z Z)
Example ; Name of person and Company names

Alphanumeric Tokens:
Consists both numeric and alphabetic tokens
Example ; Student Number , NRC Number
First, decomposes a given alphanumeric element (Student No.
SHK-0001) decomposes into SHK0001)
Then decomposes each of the alphanumeric part to its numeric and alphabetic part (0001 and SHK)
Sort the set of tokens in certain (0001SHK)

6.1.3. Storing Tokens

The table of Tokens from step 2 is sorted separately according to the system users.

6.1.4. Detection and Elimination Duplication

It is the main cleaning tasks of the system. It finds / detects duplicate records in the list which got after the step 3. Then, it eliminates those before showing the user to his/her request.

6.2 Experimental Result

This system is implemented on the desktop computer with Processor Intel(R) Pentium(R) Dual CPU 2.20GHz, Memory 2 GB of RAM. We have used 100 test data sets with 20 duplicates. It is run with different threshold level. Table 1 represents the sample Registration data of the students containing duplicate records. (only a few fields are displayed in this table).

Table 1: Student Registration Table

No.	StudentID	Name	BDate	Program	NRC
1	ICS-00004	Soe Htoo Aung	30-Dec-1985	Under grad-Dip	12/ MDG(N) 000098
2	ACS-00004	Soe Htoo Aung	30-Dec-1985	Under grad-Degree	12/ MDG(N) 000098
3	ICS-00001	Mie Mie	25-Nov-1983	Graduate Degree	01/ BBB(N) 689068
4	ICS-00016	Nay Linn Tun	27-Sep-1980	Post-Graduage-Dip	04/ AAA(N) 008791

After the duplication process, Table 2 shows the duplicate records detected by the system.

Table 2: Duplicate Detection

No.	Duplicate Set	First Set

1	{1, 2}	1
2	{3}	3
3	{4}	4

This system is tested with 97 student records with 20 duplications. Performance of the system is measured against four parameters: (a) recall, (b) false-positive errors (FPE), (c) reverse false-positive error (RFP) and (4) threshold. Recall is the ratio indicating the number of duplicates correctly identified by a given algorithm. For example, if "x" number of duplicates were identified out of "y" number of duplicates, then the recall is x/y , which when expressed in percentage is 100 * x / y. False positive error is a ratio of wrongly identified duplicates. Formally, False-positive errors,

FPE = 100 * number of wrongly identified duplicates / total number of identified duplicates.

Reverse false-positive error (RFP) indicates the number of duplicates that a given algorithm could not identify. Formally,

RFP = 100 * number of duplicates that escaped identification / total number of duplicates.

A given algorithm must not be fluctuated with varied thresholds. This system is tested with different threshold 0.25, 0.44 and 0.88. The experimental results show that this Token based cleaning algorithm maintains high recall, low FPE, low RFP and maintain a steady behavior as threshold varies. This system is tested against Basic Algorithm and Lee's algorithm. Table 3 describes the performance analysis of those three algorithms.

Table 3: Performance Analysis of Three algorithms

Thresh.	Algo	RC	FPE	RFP
0.25	Token	98.89	1.2	1.27
0.25	Basic Algo	85.56	12	5
0.25	Lee's Algo	82.23	16	12
0.88	Token	98.89	1.2	1.27
0.88	Basic Algo	87.56	11	4
0.88	Lee's Algo	83.23	13	10

7. Conclusion

It is important for the University Administration that they do not allow one student at more than one program. Detecting duplicate record is very important for the application like the system. Token-Based Data Cleaning Technique is used to solve the problem. According to the experimental results, Token-based algorithm outperforms over other algorithms.

8. References

[1] A.E Monge and C.P Elkan. The Field Matching Problems: Algorithms and Applications. Proceedings of the 2nd Int'l Conference on Knowledge and Data Mining pp 267 - 270, 1996.

[2] D. Bitton and D.J. Dewitt. Duplicate Record Elimination in Large Data Files. ACM Transactions on Database Systems, Vol. 8, No. 2, PP 255 - 265, June 1983.

[3] H. Galharda and D. Florescu and D. Shasha and E. Simon and C. Saita. Declarative Data Cleaning: Language, Model and Algorithms. Proceedings of the 27th VLDB conference, Roma, Italy, 2001.

[4] M.A Hernandez and S.J Stolfo. The Merge/Purge Problem for Large Databases. In Proceedings of the ACM SIGMOD Int'l Conference on Management of Data, pp 127 - 138, May 1995.

[5] M.A Hernandez and S.J Stolfo. Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge Discovery, 2, 9 - 37 1998.

[6] M.L. Lee and L. Hongjun and W.L Tok and T.K Yee. Cleansing Data for Mining and Warehousing. In Proceedings of the 10th Int'l Conference on Database and Expert Systems Applications (DEXA 99), Florence, Italy, August 1999.

[7] V. Raman and J.M. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation. Proceedings of the 27th VLDB conference, Roma, Italy, 2001.