Detection and Elimination of Duplicate Data using Smart Token-based Method for Airline Ticket Reservation System

Hsu Mon Mon San, Khin Lay Thwin

University of Computer studies (Hpa_an)

hsumonmonsan@gmail.com

Abstract

Data Cleaning is a process for determining whether two or more records defined differently in a database, actually represent the same real world object. During data cleaning, multiple records representing the same real life object are identified, assigned only one unique database identification, and only one copy of exact duplicate records is retained. Token formation algorithm will be efficient in handling the noisy data by expanding abbreviation, removing unimportant characters and eliminating duplicates. Attribute selection algorithm is used for the attribute selecting before the token formatting. This algorithm and token formation algorithm is used for data cleaning to reduce a complexity of data cleaning process and to clean data flexibly and effortlessly without any confusion. This paper uses smart token to increase the speed of the cleaning process and improve the quality of the data.

Keywords: Data Cleaning, Attribute selection, smart token

1. Introduction

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 [11]. Data cleaning, also called data cleansing or scrubbing. Data cleaning first detect dirty records by determing 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 [12].

Attribute selection is very important to reduce the time of the data cleaning process. An Attribute selection algorithm is effective in reducing attribute, removing irrelevant attribute, increasing speed of the data cleaning process, and improving result in clarity. An intelligent attribute selection is used as an initial step in data cleaning. There are many approaches available for selecting the attributes for the mining process to reduce dimensionality of the data warehouse. However, the recent increase of dimensionality of data poses difficulty with respect to efficiency and effectiveness in data cleaning process. The efficiency and effectiveness of attribute selection method is demonstrated through extensive comparisons with this attribute selection method using real world data of high dimensionality [13] [14].

Token-Based approach is applied in the selected attribute fields only. This attribute is selected based on the certain criteria. This attribute selection is mainly for the data cleaning process. The similarity function with long sting will take more time for the comparison process as well as it requires multi-pass approach. Token formation algorithm is used to form a token for the selected attribute fields. Token-Based approach tend to reduce the time for the comparison process and to increase the speed of the data cleaning process [6] [7].

Outline of this paper is organized as follows. Section 2 presents related work of this paper and Section 3 represents the implementation of airline ticket reservation system. Section 4 presents Attribute selection algorithm and Token formation algorithm, while section 5 presents duplicate detection and elimination with similarity function. Section 6 presents performance analysis, experimental results and section 7 presents conclusions.

2. Related Work

Bitton et al. [1] 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 and S.J Stolfo. [2][3], considered the Sorted Neighborhood Method (SNM) involves scanning the N sorted records falling within the window are compared. SNM requires w x N record comparisons. The error rate induced by SNM is critically dependent on choice of sorting keys. Multiple passes with independent sorting keys could be used to minimize the number of errors. A transitive closure over matched record pairs can be computed for combining the results of independent passes [4]. Some other methods like Priority Queue Method which is related to SNM, but sets of representative records belonging to recent clusters in the sorted record list are sorted in a priority queue. The advantage is the avoidance of the need to sort the data sources for each blocking pass, which can save significant computational time for very large data sources.

Smart Token-based data cleaning, Smart TB cleaner, which first selects efficient attributes and defines smart tokens from most important fields of records, compares and identifies duplicate records with those tokens. By using short lengthened tokens for record comparisons, a high recall/precision is achieved. After then, uses similarity match function on a data cleaning algorithm using well defined tokens from most important and useful fields of records, compares and identifies duplicate records with those tokens.

3. Implementation of Airline Ticket Reservation System

User can book air tickets from Airline ticket reservation system. Firstly, user has to choose the place and time that want to go. The system will seek the flight from airlines for presented date and time. If the system found an available flight, the customer has to fill in the passenger data. The system stored passenger' data in database. If the system found that one passenger books for two or more ticket with same time and same place, the system detect and eliminate duplicate passenger's records.

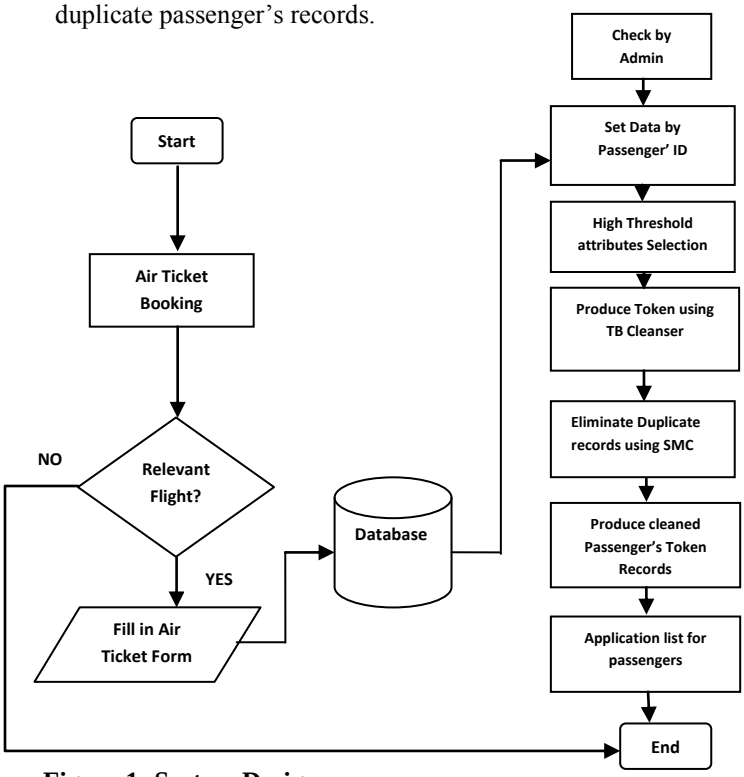

Figure 1: System Design

4. Attribute Selection Criteria

An Attribute selection is a process that chooses best attributes according to a certain criterion. This Attribute selection algorithm is used to increase the speed and improve the accuracy of the data cleaning process by removing redundant or irrelevant attribute from the database. There are three criteria are used to identify relevant attributes for the further data cleaning process:

a). Identifying key attributes

The key is an attribute or a set of attributes that uniquely identify a specific instance of the table. Every table in the data model must have a primary key whose values uniquely identify instances of the entity. The key may be primary key, candidate key, foreign key or composite key.

b). Classifying distinct and missing values

Missing character values are always same no matter whether it is expressed as one blank, or more than one blanks. Distinct is used to retrieve number of rows that have unique values for each attribute. The accuracy of the result will be poor with low distinct value and high missing value.

c). Classifying types of attributes

There are four types of attributes: nominal, ordinal, interval and ratio. The different criteria are given for each attribute types. The value of measurement types are also considered for the attribute selection. The data cleaning with numeric data will not be effective. The categorical data is efficient for the data cleaning process.

4.1. Attribute Selection Algorithm

An Attribute selection algorithm works according to the specified constraints to select the attributes for the data cleaning process. Attribute selection algorithm is presented in Figure 2.

Input : N-Attributes, no. of tuples n, relation instance r
Initialize : L - Temporary relation instance
Output : S - Selected attributes with high threshold values σ
for each attribute x_i, i<{1,2,…..N}
begin
i) Select key attribute, put into L.
ii) Select attribute collectively (Super key), then put into L.
iii) Calculate threshold σ with (σ : D /\ M /\ MT /\ S)
a. Distinct(D) value of the attribute xi if tuple $\Sigma_{i=1}^{n} t = t^i$
b. Missing(M) value of the attribute xi if tuple $\Sigma_{i=1}^{n} t^i =$ NULL
c. Measurement types(MT) of attribute (ordinal, nominal, interval, and ratio) x_i
d. Size(S) of the attribute x_i\Put into L.
iv) Select attribute with high threshold value σ_i, then put into L.
v) Ignore attribute with low threshold value
end

Figure 2: Attribute Selection Algorithm

Attribute selection algorithm first selects the relation schema R including N attributes. Then it chooses the relation instance (table) r of the relation schema R. Finally selects the attributes A_i (A_1,…..,A_N) of the relation schema R including N attributes. This Attribute selection algorithm obtains the

temporary relation schema L with attribute name, type, size, missing value, distinct value, Measurement type and Threshold value. For each attribute, read the relation tuples (records) from the selected relations instance r and find the count of missing target values of the attributes A_i and calculate the percentage value. These percentage values of missing values are stored in the temporary relation instance L. Then find the count of the distinct target values of the attributes A_i and calculate the percentage value. These percentage values of missing values are stored in the temporary relation instance L. Finally, find the measurement type of the attributes A_i and put in the temporary relation instance L for each attributes A_i. The threshold values are calculated for every target attribute A_i based on the missing values, distinct values and measurement type and put the threshold values for each attribute in the temporary relation instance L. Finally, select the attribute S from the temporary relation L based on the threshold values for the next step of the data cleaning process.

a. **Special characters** are `, “” < > - % + _ ( ). * - $ #! [ ] ^ \ @ : ; = ? | { } ~ and etc
b. **Title or Salutation** tokens are ‘Rev ‘,’Dr’, 'Mr.’,
‘Miss’, ‘Master’, ‘Madam’, ‘Sir’, ‘Chief’, ‘Ms’, ‘Mister’,‘Shri’, ‘Drs’, ‘Dres instead of Dr.’, ‘Dr.’, ‘Mistress’, ‘Sis’,‘Sri’, ‘Dear’, ‘Judge’, ‘Justice’, ‘Sister’
c. **Ordinal forms** are ‘st’, ‘nd’, ‘rd’, and ‘th’
d. **Common abbreviations** are ‘Pvt’, ‘Ltd’, ‘Co’, ‘Rd’, ‘St’, ‘Ave’, ‘Blk’, ‘Apt’, ‘Univ’, ‘Sch’, ‘Corp’ and etc
e. **Common words** are ‘and’, ‘the’, ‘of’, ‘it’, ‘as’, ‘may’, ‘than’, ‘an’, ‘a’, ‘off’, ‘to’, ‘be’, ‘or’, ‘not’, ‘I’, ‘about’,‘are’, ‘at’, ‘by’, ‘bom’, ‘de’, ‘en’, ‘for’, ‘from’, ‘how’, ‘in’,‘is’, ‘la’, ‘on’, ‘that’, ‘this’, ‘was’, ‘what’, ‘when’, ‘where’,

Table 1: Unimportant Characters

4.2. Algorithm for Token Formation

The token is formed for each selected attribute field which has the highest rank. The following step has to be taken for the best token key before forming the token. The steps are:

a). Remove unimportant characters

The first step in the token formation is removing the unimportant character before the token formation to get smart or best token for the further data cleaning process. The unimportant tokens consist of special characters, shortcut forms or ordinal forms, common or stop words, and title or salutation tokens. The common unimportant tokens are listed in the table1.

b). Expand abbreviations using Reference Table

The use of abbreviation makes problem in the token formation. The expansion of abbreviation is important in the token formation. The some common abbreviations are listed in the Table 2. These abbreviations are stored in the log table or reference table. This table is used as reference table for the token formation and contains some common abbreviations.

c). Formation of Tokens

The different token formation rules are followed for the different kind of data. The data may be numeric, alphanumeric or alphabetic. The rules are given in the algorithm (Figure 2).

S.	No.	Shortcut Full form
1	Acc. a/c, A/C	account, account current
2	advt.	Advertisement
3	Apr.	April
4	Ave	Avenue
5	Co.	Company, country
6	Dept.	Department
7	Dep.	Departure
8	Est.	Established, estimated
9	Gov.	Government, governor
10	H.O	Head Office
11	Pvt	Private
12	Ltd	Limited
13	Rd	Road
14	Blk	Block
15	Apt	Apartment
16	St	Street

Table 2: Reference Table with sample data

1).Numeric Tokens: This numeric token formation rule is suitable for phone number, social security number, street number, apartment number, etc. First, it removes the unimportant characters and converts the character to numeric. Finally, groups the number to keep together as one token.

2).Alphabetic Tokens: This alphabetic token formation rule is well suited for the names such as contact name, customer name, produce name, book title, etc. First, it expands the abbreviations and removes the unimportant and stopping characters. Finally, takes the first character from each word, sort the selected character then groups together as one token.

```
Input: Tables with dirty data, Reference table, Selected attributes
Output: LOG table with tokens
begin
  For attribute i = 1 to last attribute, m
    For row j = 1 to last row, n
      i) remove special characters
      ii) remove shortcut forms or ordinal forms
      iii) remove common or stop words
      iv) remove title or salutation tokens
      v) remove unimportant characters
      vi) expand abbreviations using Reference table
      vii) if row(j) isnumeric then
          a. convert string into number
          b. sort or arrange the number in order
          c. form a token, then put into LOG table
      viii) if row(j) isaphanumeric then
          a. separate numeric and alphanumeric
          b. split alphanumeric into numeric and alphabetic
          c. sort numeric and alphabetic separately
          d. form a token, then put into LOG table
      ix) if row(j) isalphabetic then
          a. select the first character from each word
          b. sort these letter in a specific order
          c. stringing them together
          d. if one word is present, take first three
             character as token, then sort the characters
          e. form a token, then put into LOG table
end
```

Figure 2: Token Formation Algorithm

3).Alphanumeric Tokens: This alphanumeric token rule is suited for address, product code etc. First, it splits alphanumeric into numeric and alphabetic, sorts the divided token and then groups numeric and alphabetic separately. Finally, the tokens in the field are grouped together to get token as one field.

Table3 produces token key for the address field. The alphanumeric token rule is used in this table. First, it splits the alphanumeric into numeric and alphabetic and then it uses alphabetic rule. Finally, it combines together to get token key.

Customer ID	Address	Token Key
P00001	No.402, Rno.202, Kannar Road, Botataung Tsp, Yangon	402202KBY
P00002	No.20,B.E.H.S(1) Road, Hpa_an	201BEHSHA
P00003	No.6, Pyay Road, Sat Thwar Taw Village, Hmawbi Tsp, Yangon	6PSTTHY
P00004	No.6,Pyay Road, Sat Thwar Taw Village, Hmawbi Tsp, Yangon	6PSTTHY
P00005	No.90, Pyi Thar Yar Quarter,No.2 street, Meikhtila	902PTYM

Table 3: Formation of Token for the address field

4.3. Maintaining LOG Table

Token formation algorithm is used to form a token for the selected attributes. The formed tokens are stored in the LOG table. This LOG table is a temporary table to store tokens of the selected attribute field values. The comparison of records will be take place in the LOG Table to find duplicates. The sample LOG table with smart token is described in Table4.

Customer ID	**Name Key**	**Address Key**	**Mail Key**
P00001	HMMS	402202KBY	88hsumon
P00002	KKLA	201BEHSH	kyawkyawlinaung
P00003	SWM	6PSTTHY	90susu
P00004	AWP	11151YBTY	90aung
P00005	CLY	902PTYM	msshonlaiye

Table 4: LOG Table with Smart Tokens

5. Duplicate Detection and Elimination with Similarity Function

The main cleaning tasks are accomplished in this step. For this step, need to define the Similarity Match Count (SMC). Given two records R_1 and R_2 having m pairs of token fields, R_1t_1, R_1t_2,……., R_1t_m ; R_2t_1, R_2t_2, ….., R_2t_m. The SMC is the number, n of corresponding token fields that match divided by total number, m of token fields and ranges from 0.00 to 1.00. The value of SMC of a match is used to determine whether R1 and R2 is (i) Perfect match (if SMC is 1.0). (ii) Near perfect match (if SMC is between 0.66 and 0.99), (iii) May be match (if SMC is between 0.33 and 0.67) and (iv) no match at all (if SMC is less than 0.33). When the SMC results in a "may be match", a function further computes the "Similarity Match Ratio", SMR of each of the pairs of tokens that did not match exactly. SMR is a character level comparison that is used to determine whether the token pair matches or not. Given two tokens t_1 and t_2 with m and n characters respectively also, given that the number of characters common in t_1 and t_2 is c. SMR is defined as 2C/n+m .Token t1 and t2 are considered a match if and only if SMR >= 0.67.Once the SMR of tokens are used to determine the number of tokens that match, the SMC of the records is now computed in order to declare the records a match or not. Finally the duplicate results identified by each of the token tables and integrated to obtain the list of record duplicates. After then, need to eliminate the duplication. The final result is duplicate-free and cleaned table.

6. Performance Analysis

The performance of each algorithm was measured against four parameters, namely, (i) recall (RC), (ii) false-positive error (FPE), (iii) reverse false-positive error (RFP) and (iv) 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 **(x/y)*100**. False positive error is a ratio of wrongly identified duplicates. Formally, False-positive errors, **FPE = (number of wrongly identified duplicates/total number of identified duplicates)*100**. R**everse false-positive error, (RFP)** the number of duplicates that a given algorithm could not identify. Formally**, RFP = (number of duplicates that escaped identification/total number of duplicates)*100.** A good data cleaning algorithm should: (i) have a high recall, (ii) have a very low (better if zero) FPE, hence high precision, (iii) a very low (better if zero) RFP.

6.1 Experimental Result

The results of four case studies are given in this section and used a small sized input data to enable us evaluate the output of the experiments.

Experiment 1: 20 rows of records, 4 pairs of duplicates, trivial data dirt - the results as CASE 1 of Table 5.

Experiment 2: 40 rows of records, 7 pairs of duplicates, slightly less trivial data dirt – Table 5, CASE 2.

Experiment 3: 80 rows of records, 10 pairs of duplicates, advance data dirt - results are in Table 5, CASE 3.
Experiment 4: 120 rows of records, 14 pairs of duplicates, advance data dirt - results in Table 5, CASE 4.

	(RC)	(FPE)	(RFP)
Case1			
Smart TB Cleanser	**100**	**0**	**0**
Simple TB Cleanser	**75**	**0**	**25**
Case2			
Smart TB Cleanser	**100**	**0**	**0**
Simple TB Cleanser	**75**	**25**	**0**
Case3			
Smart TB Cleanser	**100**	**0**	**0**
Simple TB Cleanser	**50**	**25**	**25**
Case 4			
Smart TB Cleanser	**75**	**0**	**25**
Simple TB Cleanser	**25**	**0**	**75**

Table 5: Performance Analysis

7. Conclusion

This paper used similarity function to detect and eliminate duplication by using well-defined tokens for detecting and removing duplicate records. Customers can buy air tickets easily by using this web-based system. This system support users in airline ticket reservation by using token-based approach. User can reply air ticket demands correctly and efficiently. Future work should consider applying this token-based cleaning technique on unstructured, and semi-structured data.

REFERENCES

[1] 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.

[2] Galhardas, H.; Florescu, D.; Shasha, D.; Simon, E.: Declaratively cleaning your data using AJAX. In Journee Bases de Donnees, Oct. 2000. http://caravel.inria.fr/~galharda/BDA.ps.

[3] Galhardas, H.; Florescu, D.; Shasha, D.; Simon, E.: AJAX: An Extensible Data Cleaning Tool. Proc. ACM SIGMOD Conf., p. 590, 2000.

[4] M.A. Hernandez and S.J. Stolfo. Real-world data is dirty: data cleansing and the merge/purge problem. Journal of Data Mining and Knowledge Discovery, 1(2), 1998.

[5] Hernandez, M.A. & Stolfo, S.J. The merge/purge problem for large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 127-138), 1995.

[6] Ohanekwu, T.E. & Ezeife, C.I. (2003, January). A token based data cleaning technique for data warehouse systems. *Proceedings of the International Workshop on Data Quality in Cooperative Information Systems* (pp. 21-26), held in conjunction with the *9th International Conference on Database Theory* (ICDT 2003), Siena, Italy.

[7] C.I. Ezeife and Timothy E. Ohanekwu, "Use of Smart Tokens in Cleaning Integrated Warehouse Data", *the International Journal of Data Warehousing and Mining (IJDWM),* Vol. 1, No. 2, pp. 1-22, Ideas Group Publishers, April-June 2005.

[8] Lee, M.L., Hongjun, L., Tok, W.L., & Yee, T.K (1999). Cleansing data for mining and warehousing. *Proceedings of the 10th International Conference on Database and Expert Systems Applications* (DEXA 99), Florence, Italy.

[9] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of VLDB,* Hong Kong, 2002.

[10] J.Jebamalar and V.Saravanan : Handling Noisy Data using Attribute Selection and Smart Tokens(2009)

[11] W.W. Cohen and J.Richman. Learning to Match and Cluster Large High-Dimensional Data Integration. In SIGKDD'02, 2002.

[12] Rawshan Basha: Using well defined tokens in similarity function for record matching in data cleaning techniques (2005)

[13] G. H. John, R. Kohavi, and K. Pfleger, Irrelevant Features and the Subset Selection Problem, Proc. Of the 11th Int'l Conf. on Machine Learning, pages 121-129, Morgan Kaufmann, 1944.

[14]. H.Liu and H. Motoda, Feature Selection for Knowledge Discovery and Data Mining Kluwer, Boston, 1998.