South Asian Journal of Engineering and Technology Vol.2, No.17 (2016) 280–287

ISSN No: 2454-9614

Effective Data Blending Diffusion Model for Delete Cascading Function

T. Premamala, R. MonishaJothi*, J. Amali Jasmine, M. Dhivya

Department of Information Technology,Velalar College of Engineering and Technology
Tiruchengode,Tamilnadu, India.

*Corresponding Author: T. Premamala

E-mail: tpremamala@gmail.com,

Received: 15/11/2015, Revised: 15/12/2015 and Accepted: 03/03/2016

Abstract

In a relational database, tuples are called "duplicate" if they describe the same real-world entity. If such duplicate tuples are observed, it is recommended to remove them and to replace them with one tuple that represents the joint information of the duplicate tuples to a maximal extent. This remove-and-replace operation is called a fusion operation. Within the setting of a relational database management system, the removal of the original duplicate tuples can breach referential integrity. In this study, a strategy is proposed to maintain referential integrity in a semantically correct manner, thereby optimizing the quality of relationships in the database. An algorithm is proposed that is able to propagate a fusion operation through the entire database. The algorithm is based on a framework of first and second order fusion functions on the one hand, and conflict resolution strategies on the other hand. It is shown how classical strategies for maintaining referential integrity, such as DELETE cascading, are highly specialized cases of the proposed framework. Experimental results are reported that (i) show the efficiency of the proposed algorithm and (ii) show the differences in quality between several second order fusion functions. It is shown that some strategies easily outperform DELETE cascading. **Reviewed by* **ICETSET'16** *organizing committee*

1. Introduction

In the context of relational databases, dealing with duplicates comes down to (a) identifying which tuples are duplicate and (b) replacing those tuples by a single tuple. Needless to say, in the context of a relational database, the deletion of the original-duplicate-tuples should take into account integrity constraints, in particular referential integrity, which is assumed to be satisfied in the original database. For that purpose, modern relational database systems offer the database administrator several strategies to resolve referential integrity (CASCADE, SET NULL, RESTRICT). However, in the context of data fusion, these strategies fail to propagate deletions and updates in a semantically correct manner. A correct way of propagation is to take into account that multiple tuples are fused, meaning that linked information should be fused accordingly. This way, quality of relationships is optimized. From that point of view, a framework is proposed that performs fusion of tuples and propagates this fusion operation through the entire database by making use of second order fusion functions. In this study, framework, fusion is treated modularly, meaning that it is independent from duplicate detection. As a result the framework can be used for large-scale fusion tasks, but also as a generalization of the CASCADE propagation strategy in the SQL language.

This study proposes an algorithm for maintaining referential integrity in a relational database after the fusion of duplicate tuples. The algorithm relies on a framework of fusion functions for multi-valued data (i.e.,sets), in order to cope with one-to-many relationships. A comparative study shows the difference in accuracy between several multi-valued fusers.

2. Related Work

Ivax P. Fellegi and Alan B. Sunter [1] stated that a mathematical model is developed to provide a theoretical framework for a computer-oriented solution to the problem of recognizing those records in two files which represent identical persons, objects or events (said to be matched). A comparison is to be made between the recorded characteristics and values in two records (one from each file) and a decision made as to whether or not the members of the comparison-pair represent the same person or event, or whether there is insufficient evidence to justify either of these decisions at stipulated levels of error. These three decisions are referred to as link (At), a non-link (A3), and a possible link (A2). The first two decisions are called positive dispositions.

The 2 types of error are defined as the error of the decision A3 when the members of the comparison pair are in fact unmatched, and the error of the decision A, When the members of the comparison pair are, in fact matched. The necessity for comparing the records contained in a file LA with those in a file LB in an effort to determine which pairs of records relate to the same population unit is one which arises in many contexts, most of which can be categorized as either (a) construction or maintenance of a master file for a population, or (b) merging two files in order to extend the amount of information available for population units represented in both files.

Cinzia Cappiello, Politecnico di Milano [2] describes the Guaranteeing high data quality levels is an important issue especially in information-intensive organizations. In fact, the core business of such organizations is based on the use of information for either providing personalized services or understanding and better satisfying customers' requirements.

However, poor data quality has negative impacts on almost all the enterprises. Indeed, it often implies customer dissatisfaction, increased operational cost, less effective decision-making, and a reduced ability to make and execute organizational strategies. Improving data quality often requires modifying business processes enriching them with additional activities. Such activities change on the basis of the data quality dimensions to improve. In this paper, we present a methodology to support process designers in the selection of the improvement actions to adopt in the design of business processes in order to satisfy the data quality requirements.

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis et al. [3] stated in the real world, entities have two or more representations in databases. Duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task. Errors are introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors. In this paper, we present a thorough analysis of the literature on duplicate record detection. We cover similarity metrics that are commonly used to detect similar field entries and present an extensive set of duplicate detection algorithms that can detect approximately duplicate records in a database. And also cover multiple techniques for improving the efficiency and scalability of approximate duplicate detection algorithms. We conclude with coverage of existing tools and with a brief discussion of the big open problems in the area.

E. F. CODD [4] stated that Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information. Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data.

Felix Naumann, Alexander Bilke et al [5] described heterogeneous and dirty data is abundant. It is stored under different, often opaque schemata, it represents identical real-world objects multiple times, causing duplicates, and it has missing values and conflicting values. Without suitable techniques for integrating and fusing such data, the data quality of an integrated system remains low. We present a suite of methods, combined in a single tool, that allows ad-hoc, declarative fusion of such data by employing schema matching, duplicate detection and data fusion.

Guided by a SQL-like query against one or more tables, we proceed in three fully automated steps: First, instance-based schema matching bridges schematic heterogeneity of the tables by aligning corresponding attributes. Next, duplicate detection techniques find multiple representations of identical real-world objects. Finally, data fusion and conflict resolution merges each duplicate into a single, consistent, and clean representation.

3. Existing System

The existing system proposes an algorithm for maintaining referential integrity in a relational database after the fusion of duplicate tuples. The algorithm relies on a framework of fusion functions for multi-valued data (i.e., sets), in order to cope with one-to-many relationships. A comparative study shows the difference in accuracy between several multi-valued fusers. Experimental results show the complexity in terms of the number of clusters, the number of linked tuples and the recursive depth. Fusion propagation and Recursive propagation algorithms are carried out for Propagation of Data fusion. The existing system also investigates how DELETE operations that are the consequence of a data fusion operation can be propagated through a relational database with semantically correctness.

In the existing stud only similar word contents are checked for duplicate records. And the semantic similarity between the words is not taken for duplication record identification. It only two records are fused during duplication removal. Multiple records are not fused as single record.

4. Proposed System

Like existing system, the propagation of data fusion is carried out here also. In addition, different words with same meaning are also fed into database and taken during duplicate record elimination. So if the same concept is present in two records with different words or sentences, then they are treated as single record. Before finding the duplicate, all the records are checked for such similarity and replaced with one meaning. Then fusion process is continued for duplicate record elimination.

- Different word contents are checked for duplicate records.
- Semantic similarity between the words is taken for duplication record identification.
- Multiple records are fused as single record.

4. 1. Proposed Methodology

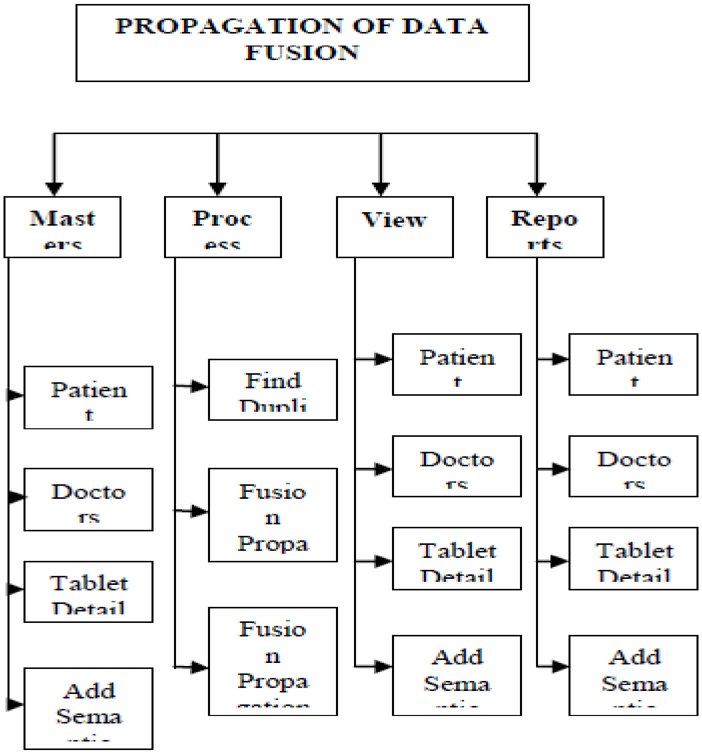
Fusion Propagation Algorithm:

Require: R^*, R with FK^* the foreign key of R^*

Require:

Require: F and F^*

```
For all $t$ € □do [ All the duplicate records are taken and iterated]
    $St = \{t^* \mid t^*€ R^*\}$ [Record from new relation is assigned to St]
    $St^* = St$ [St is assigned to t*]
    For all $t^*€ St^*$ [All the records are taken for St* and iterated]
        $T^*[FK^*] = F(\Delta)\ [K]$ [New Foreign key is assigned based on old available foreign keys. Here the removed records primary key will be omitted]
    End For
End For
$T\Delta = \{St^*\ [K^*] \mid t€\ \Delta$ [Now the duplicate records those are fused containing the correct keys are taken as TΔ]
$= F^*(T\Delta]$ [Fusion function for new records]
```



10) = £ ΦNon-key columns are fused]

11) For all t € Δ do [All the duplicate records are taken]

12) R = (R* \ St) [Incorrect records are eliminated]*

13) End For

14) R = (R* UΦ) [This new Schema R* added with Non-key columns fused values]*

15) The algorithm eliminates the old foreign key values after the fusion is carried out.

The algorithm performs semantic similarity between the words is taken for duplication record identification. So the two records with different sentences but with same meaning are considered as duplicate records and they will be fused as one record now.

5. Results and Discussions

S.NO	Number of Data set (n)	SR-Data Fusion (Count)	MR-SE-Data Fusion (Count)

1	100	22	30
2	200	72	83
3	300	175	202
4	400	205	278
5	500	313	367
6	600	376	452
7	700	415	505
8	800	489	543
9	900	532	605
10	1000	601	698

Table 1.1 Performances Analysis- SR and MR-SE Data Fusion Algorithm

The table 1.1 shows the performance analysis of Single Record and Multi Record-Semantic data fusion algorithm. The table contains the values of Number of dataset used, SR(Single Record) Data Fusion Count and Multi Record-Semantic data fusion count.

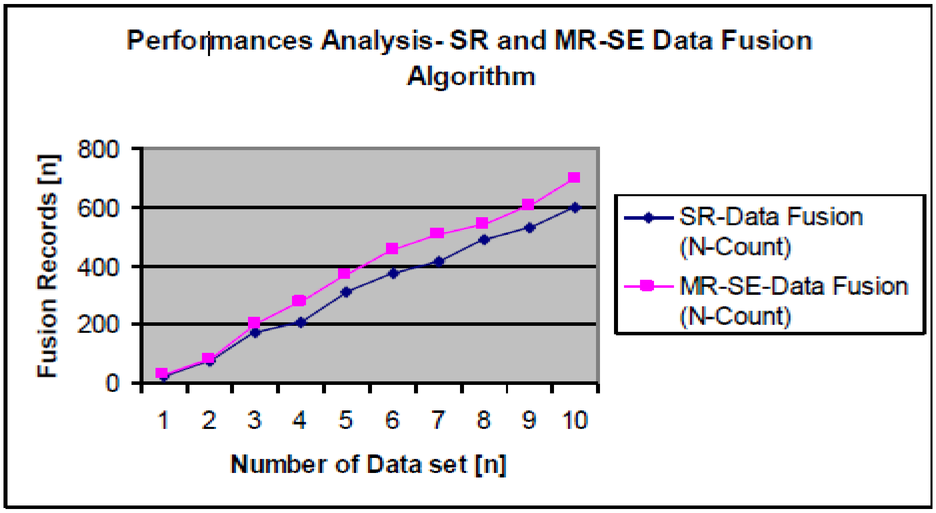


Fig 1.1 Performances Analysis- SR and MR-SE Data Fusion Algorithm

The figure 1.1 shows the performance analysis of Single Record and Multi Record-Semantic data fusion algorithm. The figure shows the values of Number of dataset used, SR(Single Record) Data Fusion Count and Multi Record-Semantic data fusion count.

S.NO	Number of Data set (n)	SR-Data Fusion (%)	MR-SE-Data Fusion (%)
1	100	45.45	33.33
2	200	27.77	24.09
3	300	17.14	14.85
4	400	19.51	14.38
5	500	15.97	13.62
6	600	15.95	13.27
7	700	16.86	13.86
8	800	16.35	14.73
9	900	16.91	14.87
10	1000	16.63	14.32

Table 1.2 Error Rate Analysis- SR and MR-SE Data Fusion Algorithm

Table 1.2 contains the values of number of dataset (n), percentage of single record data fusion and percentage of multiple record data fusion with semantic words. It describes the error rate analysis of the existing and proposed methodologies.

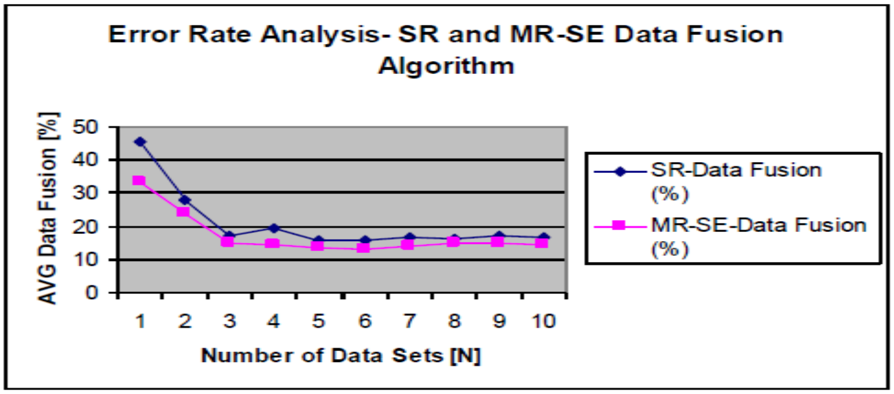


Figure 1.2 Error Rate Analysis- SR and MR-SE Data Fusion Algorithm

Figure 1.2 shows the error rate analysis of the existing and proposed methodologies. It shows the values of number of dataset (n), percentage of single record data fusion and percentage of multiple record data fusion with semantic words.

6. Conclusion

In this paper it has been analysis the DELETE operations that are consequence of a data fusion operation, can be propagated through a relational database with semantically correctness. A proposed framework of first and second order fusion functions has been developed and a distinction was made between backward and forward propagation. While backward propagation can be avoided with reasonable assumptions by intelligent design of the first order fusion function, forward propagation was shown to be more easily. A propagation algorithm was proposed that relies on second order fusion functions to recursively fuse sets of linked tuples. Conflict resolution strategies are used to resolve conflicts in relationship attributes. It is shown the DELETE cascading strategy is a highly specialized case of the propagation algorithm. Analysis of complexity shows an acceptable computational overhead to guarantee semantically correctness with data fusion.

7. Scope for Future Enhancement

In feature a database-centric platform for building information fusion applications offers many advantages. These include tight integration of individual components, security, scalability, and high availability. Current trends in RDBMSs are moving towards providing all key components for delivering comprehensive state-of-the-art information fusion applications. As illustrated above for a hyper-spectral satellite image information fusion problem, these features provide great flexibility and analytic power. By leveraging an existing RDBMS-based technology stack, a full-fledged information fusion application can be developed in a reasonably short time and at low development cost.

References

[1] Bronselaer, D. V. Britsom, and G. D. Tr e, “A framework for multiset merging,” Fuzzy Sets Syst., vol. 191, pp. 1–20, 2012.

[2] Bronselaer and G. D. Tr e, “Weak preservation of multi-valued fusion,” in Proc. 8th Eur. Soc. Fuzzy Logic Technol., 2013, pp. 514–520.

[3] Bronselaer and G. D. Tr e, “Aspects of object merging,” in Proc. Annu. Meeting North Amer. Fuzzy Inform. Process., Toronto, ON, Canada, Jul. 2010, pp. 27–32.

[4] J. Bleiholder and F. Naumann, “Data fusion,” ACM Comput. Surv., vol. 41, no. 1, pp. 1–41, 2008.

[5] A. Bronselaer, G. De Tr e, and D. V. Britsom, “Robustness of multiset merge functions,” in Proc. IPMU Conf., 2012, pp. 481–490.

[6] Elmagarmid, P. Ipeirotis, and V. Verykios, “Duplicate record detection: A survey,” IEEE Trans. Knowl. Data Eng., vol. 19, no. 1, pp. 1–16, Jan. 2007

[7] Bhattacharya and L. Getoor, “Collective entity resolution in relational data,” ACM Trans. Knowl. Discovery Data, vol. 1, p. 2007, 2006.

[8] Arasu, C. R e, and D. Suciu, “Large-scale deduplication with constraints using dedupalog,” in Proc.

[9] IEEE Int. Conf. Data Eng., 2009, pp. 952–963.

[10] L. Getoor. (2012) [Online]. Available: http://www.umiacs.umd.edu/getoor/tutorials/er vldb2012.pdf (2011) [Online]. Available: ISO/IEC 9075-1:2011: Information technology database languages SQL part1: Framework (SQL/framework).