DUPLICATE RECORD DETECTION FOR DATABASE CLEANSING

MARIAM REHMAN
Computer Science and Information Management Program,
Asian Institute of Technology
Pathumthani, 12120, Thailand
mariam.rehman@ait.ac.th

VATCHARAPON ESICHAIKUL
Computer Science and Information Management Program,
Asian Institute of Technology
Pathumthani, 12120, Thailand
vatchara@ait.ac.th

***Abstract*—Many organizations collect large amounts of data to support their business and decision making processes.The data collected from various sources may have data quality problems in it. These kinds of issues become prominent when various databases are integrated. The integrated databases inherit the data quality problems that were present in the source database. The data in the integrated systems need to be cleaned for proper decision making. Cleansing of data is one of the most crucial steps. In this research, focus is on one of the major issue of data cleansing i.e. "duplicate record detection" which arises when the data is collected from various sources. As a result of this research study, comparison among standard duplicate elimination algorithm (SDE), sorted neighborhood algorithm (SNA), duplicate elimination sorted neighborhood algorithm (DE-SNA), and adaptive duplicate detection algorithm (ADD) is provided. A prototype is also developed which shows that adaptive duplicate detection algorithm is the optimal solution for the problem of duplicate record detection. For approximate matching of data records, string matching algorithms (recursive algorithm with word base and recursive algorithm with character base) have been implemented and it is concluded that the results are much better with recursive algorithm with word base.**

I. INTRODUCTION

Attention towards the data cleansing is a critical issue for the organization of data in any company [13]. Data cleansing is the essential need for the organizations whose data is increasing rapidly and they want to keep pace with the growing technology to meet the emerging needs of their users. Available data needs to be clean so that good decisions can be taken on the data and customer satisfaction may also be increased.

Data cleansing is very important with respect to business perspective. If data is not accurate, complete, and consistent then decisions taken on the basis of data may be not good or can be misleading. When stand alone sources are integrated, the data quality problems are inadvertently escalated. A major issue in dirty data is the existence of duplicates [3]. The removal of duplicates is an important cleansing issue which is the focus point in this research study.

II. BACKGROUND

A. Poor Data Quality

As the data quality is getting more and more attention in organizations, companies realize that the data exist in their organization is of poor quality. However,companies are trying to get maximum benefit from their data to make better quality decisions and also showing their great concern towards the data quality issues to be resolved. Improved quality of data within the organization produces high financial return for the organizations as well. Accurate data is the most important requirement for every organization. To maintain accurate data in the organization requires more attention towards the issues of systems design, monitoring of the phases of data collection. More and more actions should be taken to correct problems from the data that have become the source of generating errors. In order to get accurate and clean data, formal data quality assurance routines and procedures are required to ensure the data quality. When heterogeneous sources are integrated then the probability of containing "dirty data" is high [3]. A major problem of dirty data is the existence of duplicates in the records. The removal of duplicates constitutes a major cleansing task

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

B. Classification of Data Quality Problems

Data quality problems are classified into single-source and multi-source problems which are further categorized into schema and instance related problems respectively.

978-0-7695-3944-7 2010
DOI 10.1109/ICMV.2009.43

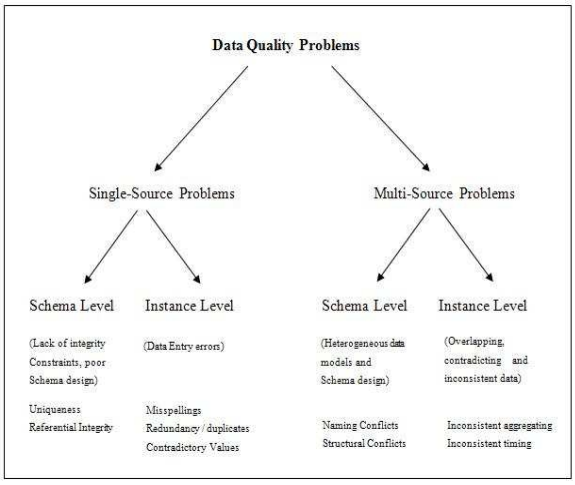

Figure 1. Classification of data quality problems in data sources

In this research, the focus is on single-source problems which are encountered at an instance level i.e. duplicate record detection as depicted in Figure 1.

C. Duplicate Elimination Techniques

Duplicate elimination is also referred to as record linkage, data de-duplication, instance identification, object identification or record matching [17]. The goal of the duplicate elimination techniques is to reduce the clerical involvement and also to minimize *false positives* [4].

The problem of duplicate elimination is categorized into three major categories i.e. knowledge-based techniques, probabilistic techniques, and empirical techniques. In this research, focus is on empirical techniques. The idea of empirical techniques is to extract duplicates by sorting and windowing strategies, and the goal is to achieve better accuracy, good performance, and also to minimize false positives.

The techniques for detecting approximate similarities ofstring-based data are character-based similarity metrics and token-based similarity metrics.

1) Character-based similarity metrics

The problem of mismatches in databases is due to the typographical variations of entered data. The process of duplicate detection relies on approximate string matching techniques to handle such problems. Character-based similarity metrics deal with typographical errors for strings [2].

2) Token-based similarity metrics

Typographical conventions sometimes lead to rearrangement of words e.g. (“Mariam Rehman” versus “Rehman, Mariam”). To handle such cases, characterbased similarity metrics fail to get the similarity among the strings. Token-based similarity metrics were introduced to overcome the problem of character-based similarity metrics. Token-based similarity metrics focus on the string-based representation of the records. However, records consist of multiple fields [2].

The algorithms used to match records with multiple fields for duplicate record detection are the main focus of this study.

These algorithms are:

- Standard duplicate elimination algorithm
- Sorted neighborhood algorithm
- Duplicate elimination sorted neighborhood algorithm
- Adaptive duplicate detection algorithm

A comparison among these algorithms of duplicate record detection is provided in this study. After performing the comparison among the algorithms, the optimal solution is recommended for the problem of duplicate record detection.

III. METHODOLOGY

This section underlines the methodology used in this study.

A. Research Framework

In this study, it is assumed that data has already been integrated into the database which is called integrated data source. Integrated data source contains the information which is already converted into the uniform representation from various data sources.

The focus of the study is on the cleansing part. Duplicate record elimination problem is addressed by applying the techniques of duplicate record detection as shown in Figure 2. After performing cleansing on the integrated data source, cleansed data is placed into the target database which can be used for further processing of organization to avoid any false decisions.

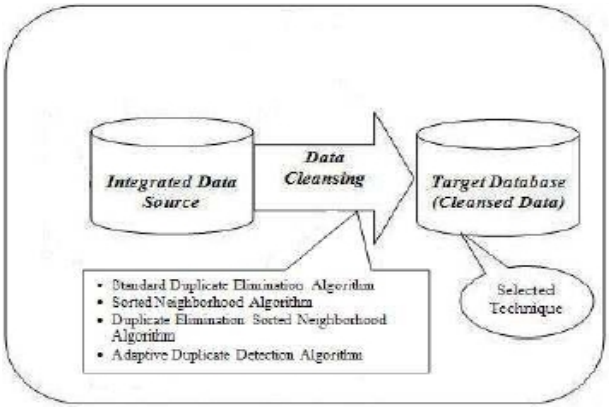

Figure 2. Figure2. Research Framework

B. Dataset

The dataset which has been used for data cleansing is collected from the Internet [11]. “Restaurant” is a data set of 867 records, containing the fields of name, address, phone, city and type. The dataset consists of 224 duplicate records in it. This dataset worked as a benchmark for evaluation of results generated by this study.

IV. RESULTS AND DISCUSSION

This section describes the evaluation methodology, marking of duplicates, evaluation, and results of algorithms.

A. Evaluation Methodology

To measure the effectiveness of algorithms, the measures to assess the accuracy of the algorithms are precision, recall and F-score. Their formulas are:

Precision = tp / tp + fp

Recall = tp / tp + fn

F-score = 2 * (Precision*Recall) / (Precision + Recall)

Where tp = true positives; fp = false positives and fn = false negatives.

B. Marking of Duplicates

For marking of duplicates, a process is written to compare the results of the algorithms with the marked data set. The accuracy of marked duplicates is evaluated and it is encountered that after marking of duplicates 222 duplicate records are identified out of 867 data records, while 224 duplicate records are available in the original source data and this is the only information available about the source data set. So, the error percentage for marking of duplicates process is: [(No. of duplicate records in the source data set – No. of duplicate records encountered after marking process /Total no. of records in the source data set) * 100] = [(224-222 / 867)*100] = 0.23% which is quite low and almost negligible.

After marking duplicates in the source data, the next step is to compare the algorithm results with this source data set and also to compute the values of precision, recall and F-score.

C. Evaluation of Algorithms

In this section, evaluation of algorithms is performed according to the evaluation methodology.

The recursive algorithm with word base (with some modification) and recursive algorithm with character base (with some modification) is used for approximate matching of strings along with the techniques for duplicate record detection, with the varying threshold value ranging from 10 to 100 to see the overall effect on precision, recall and F-score and also to conclude which threshold value gives the optimal results.

1) Standard duplicate elimination algorithm

After performing evaluation of standard duplicate elimination algorithm, the evaluation results of the algorithm are shown in Table 1.

TABLE I. EVALUATION OF STANDARD DUPLICATE ELIMINATION ALGORITHM

Quality Measures	Values
True Positives (TP)	52
False Positives (FP)	0
False Negatives (FN)	170
Precision	100
Recall	23.42
F-score	37.96

Table 1 shows that the value of recall is quite low and also the numbers of true positives identified in this algorithm are also very less as compared to the number of true positives (222 records) encountered after marking of duplicates process.

2) Sorted neighborhood algorithm

After performing evaluation of sorted neighborhood algorithm, the evaluation results of the algorithm are shown in Table 2.

TABLE II. EVALUATION OF SORTED NEIGHBORHOOD ALGORITHM

Quality Measures	Values
True Positives（TP）	202
Faulse Positives（FP）	0
Faulse Negatives（FN）	20
Precision	100
Recall	90.99
F-Score	95.28

Table 2 shows that the results produced by this algorithm are quite good but they are highly dependent on the generation of key.

3) Duplicate elimination sorted neighborhood algorithm

A duplicate elimination sorted neighborhood algorithm uses recursive algorithm with word base with some modification for approximate matching of strings. The varying threshold values are given as an input ranging from 10 to 100 to analyze the effect on the results and also to know at what threshold value algorithm performance is best as compared to other values. The results generated by executing this algorithm are described in Table 3. The optimal values where the algorithm performance is best are made bold in their respective Tables.

TABLE III. EVALUATION OF DUPLICATE ELIMINATION SORTED NEIGHBORHOOD ALGORITHM USING RECURSIVE ALGORITHM WITH WORD BASE

Input Value	Recursive algorithm with word base			Quality Measure Parameters		
Value	Precision	Recall	F-score	TP	FP	FN
10	57.74	99.1	72.97	220	161	2
20	57.74	99.1	72.97	220	161	2
30	57.74	99.1	72.97	220	161	2
40	95.24	99.1	97.13	220	11	2
50	95.24	99.1	97.13	220	11	2
60	95.24	99.1	97.13	220	11	2
70	100	93.24	96.5	207	0	15
80	100	93.24	96.5	207	0	15
90	100	93.24	96.5	207	0	15
100	100	93.24	96.5	207	0	15

Duplicate elimination sorted neighborhood algorithm is also executed with recursive algorithm with character base for approximate matching of strings. The results produced are described in Table 4.

TABLE IV. EVALUATION OF DUPLICATE ELIMINATION SORTED NEIGHBORHOOD ALGORITHM USING RECURSIVE ALGORITHM WITH CHARACTER BASE

Input Value	Recursive algorithm with character base			Quality Measure Parameters		
Value	Precision	Recall	F-score	TP	FP	FN
10	29.6	100	45.68	222	528	0
20	29.6	100	45.68	222	528	0
30	29.6	100	45.68	222	528	0
40	67.48	99.1	80.29	220	106	2
50	67.48	99.1	80.29	220	106	2
60	67.48	99.1	80.29	220	106	2
70	100	94.59	97.22	210	0	12
80	100	94.59	97.22	210	0	12
90	100	94.59	97.22	210	0	12
100	100	94.59	97.22	210	0	12

When duplicate elimination sorted neighborhood algorithm is executed with recursive algorithm with character base, it can be observed from the Table 3 and Table 4 that at the optimal threshold value the number of true positives have been reduced from 220 to 210 and the numbers of false negatives have been increased from 2 to 12 that's why this algorithm is not preferable as compared to the recursive algorithm with word base because the accuracy of the algorithm is reduced.

4) Adaptive duplicate detection algorithm

Adaptive duplicate detection algorithm also uses the recursive algorithm with word base with the changing value of threshold ranging from 10 to 100 for approximate matching of string contents. The results generated by the algorithm are shown in Table 5.

TABLE V. EXPERIMENTAL EVALUATION FOR ADAPTIVE DUPLICATE DETECTION ALGORITHM USING RECURSIVE ALGORITHM WITH WORD BASE

Input Value	Recursive algorithm with word base			Quality Measure Parameters		
Value	Precision	Recall	F-score	TP	FP	FN
10	28.17	98.2	43.78	218	556	4
20	28.17	98.2	43.78	218	556	4
30	53.75	100	69.92	222	191	0
40	53.75	100	69.92	222	191	0
50	85.06	100	91.93	222	39	0
60	85.06	100	91.93	222	39	0
70	96.7	92.34	94.47	205	7	17
80	96.7	92.34	94.47	205	7	17
90	100	54.95	70.93	122	0	100
100	100	54.95	70.93	122	0	100

TABLE VI. EXPERIMENTAL EVALUATION FOR ADAPTIVE DUPLICATE DETECTION ALGORITHM USING RECURSIVE ALGORITHM WITH CHARACTER BASE

Input Value	Recursive algorithm with character base			Quality Measure Parameters		
Value	Precision	Recall	F-score	TP	FP	FN
10	25.78	100	41	222	639	0
20	25.78	100	41	222	639	0
30	34.86	99.55	51.64	221	413	1
40	34.86	99.55	51.64	221	413	1
50	61.9	99.55	76.34	221	136	1
60	61.9	99.55	76.34	221	136	1
70	94.98	93.69	94.33	208	11	14
80	94.98	93.69	94.33	208	11	14
90	100	62.16	76.67	138	0	84
100	100	62.16	76.67	138	0	84

In the case of adaptive duplicate detection algorithm, when recursive algorithm with word base is used for approximate matching of strings, the results are more accurate as it can be observed from the Table 5 that precision at the optimal threshold values is 96.7% and recall is 92.34%. When the algorithm is executed with recursive algorithm with character base the results are less accurate because the false positives have been increased from 7 to 11 as shown in Table 6 and precision achieved is 94.98% and recall is 93.69%.

D. Performance Evaluation

The algorithms performance is tested on the restaurant data set. The objective of this experiment is to consider the time complexity element of algorithms at the optimal threshold value which have been observed from the results.

TABLE VII. PERFORMANCE EVALUATION

Algorithm	Standard duplicate elimination algorithm	Sorted neigh-borhood algorithm	Duplicate elimination sorted neigh-borhood algorithm	Adaptive duplicate detection algorithm
Recursive algorithm with word base	2.06	0.28	0.38	0.42
Recursive algorithm with character base	2.06	0.28	2.02	1.56

It can be inferred from the Table 7 that the performance of the algorithms is quite good which is measured in seconds as they take very less time for execution.

V. Conclusion

The biggest challenge for this study was to improve theresults in terms of performance and accuracy of the algorithms to achieve better results as compared to the previous work done in the domain of duplicate record detection. It is concluded from the results that the error percentage of adaptive duplicate detection algorithm is very low (1.96%) so these results lead to the accuracy of the algorithm as the number of false positives have also been reduced drastically from 39 to 7; it is one of the main objective for this study to investigate at the optimal threshold value of 70 and 80 which has been obtained from the detailed results in case of adaptive duplicate detection algorithm.

Recursive algorithm with word base (with some modification) has been used for approximate matching. This algorithm has its shortcoming that the algorithm is not symmetrical. In this study, the specified shortcoming of the algorithm has been overcome by introducing the concept of swapping among the strings. If the symmetrical approach is implemented then it can increase the time complexity of the algorithm which is of course not a good solution in terms of performance of the algorithm. The focus of the study is to improve the accuracy and efficiency of the algorithm that is a reason why symmetrical functionality has not been implemented. By modifying this algorithm for approximate match, the results are more accurate; however, very small percentage of error exists in the resultant dataset and overall precision, and recall is also relatively good, as the results produced in this research are also compared with the other related work in this domain.

For approximate string matching, it has been concluded that the results produced by the recursive algorithm with word base are better than the results produced by recursive algorithm with character base. As for the recursive algorithm with character base, the numbers of false positives have been increased and also percentage of precision and recall has been reduced in general. This algorithm has also increased the execution time of the algorithm as well.

Adaptive duplicate detection is the best choice for the specified problem of duplicate record detection. This algorithm does not require domain knowledge. The algorithm is only dependent on the queue size to define how many records to be compared which is the requirement for the execution of the algorithm. For this algorithm, high importance is given to those fields of the database that have major impact to identify records as duplicates. As in the case of adaptive duplicate detection, fields that are used for comparison are name, address, phone, city and type. So, these fields have major impact on the results of the algorithm. These important fields are identified after having the domain knowledge about the dataset.

Adaptive duplicate detection algorithm outperforms the other algorithms in terms of accuracy but not in efficiency as this algorithm takes more time for execution because of both activities (exact and approximate match). This algorithm is domain independent and the variation proposed for approximate matching improves the accuracy of the results.

Acknowledgment

All thanks to Allah the most Gracious, most Compassionate, by His Blessings who make me able to complete this research work.

References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [1]. Where appropriate, include the name(s) of editors of referenced books. The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first . . .”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] A. D. Chapman, Principals and methods of data cleaning: *Primary Species and Species-Occurrence Data,* version 1.0. Report for the Global Biodiversity Information Facility, Copenhagen: Australia (2005).

[2] A. K. Elmagarmi, P. G. Ipeirotis and V. S.Verykios, Duplicate record detection: a survey. *In proceedings of IEEE transactions on knowledge and data engineering*, *19* (1), 1 – 16. New York: IEEE Educational activities department (2007).

[3] A. O. Udechukwu, Domain independent deduplication in data warehouse cleaning. (Master Thesis, University of Windsor, 2002). Canada:University of Windsor (2002).

[4] C. Batini and M. Scannapieco, Data Quality: Concepts, methodologies and techniques, *International journal on Information Quality, 1*(4), 444 – 450 (2006).

[5] D. G. Brizan and A. U. Tansel, A survey of entity resolution and record linkage methodologies, *Communications of the IIMA, 6* (3), 46 – 56 (2006).

[6] E. Monge, Matching Algorithms within a Duplicate Detection System. *In proceedings of IEEE Data Engineering Bulletin, 23* (4), 14 – 20. Chiba, Japan: IEEE (2000).

[7] E. Rahm and H. I. Do, Data Cleaning: Problems and current approaches. *In proceedings of IEEE Bulletin of the Technical Committee on Data Engineering, 23* (4), 3 – 13. Germany: IEEE(2000).

[8] H. Muller and J. C. Freytag, Problems, Methods and Challenges in comprehensive Data Cleansing. Technical Report, Humboldt University Berlin:Germany (2003).

[9] J. E. Olson, *Data Quality: The Accuracy Dimension*. *(5th ed.)*. Morgan Kaufmann (2003).

[10] J. I. Maletic and A. Marcus, Data Cleansing: Beyond integrity analysis. *In proceedings of the Conference on Information Quality (IQ2000),* 200– 209. Toronto: Boston Press (2000).

[11] M. Bilenko, RIDDLE, Repository of information on duplicate detection, record linkage, and identity uncertainty(2002).http://www.cs.utexas.edu/users/ml/riddle/index.html

[12] P. Christen and K. Goiser, Quality and Complexity Measures for Data Linkage and De-duplication. *In Guillet, F. & Hamilton, H. J. ed. Quality Measures in Data Mining, Springer Studies in Computational Intelligence*, *43*, 127 - 151. Heidelberg: Springer Link (2007).

[13] P. Neely, Data Quality tools for Data Warehousing: a small sample survey. *In proceedings of MIT conference on Information Quality,* Center for technology in government. University at Albany / SUNY: Germany (1998).

[14] P. Ravikumar and W. W. Cohen, A hierarchical graphical model for record linkage. In *20th Conf. Uncertainty in Artificial Intelligence (UAI '04),* 454 – 461. Banff, Canada: AUAI Press (2004).

[15] S. O. I. Pei, A comparative study of Record Matching Algorithms. (Master Thesis, RWTH Aachen, Germany and University of Edinburgh, 2008). Scotland: Germany and University of Edinburgh (2008).

[16] S. Tejada, A. Knoblock and S. Minton, Learning object identification rules for information integration. *Journal of Information Systems, 26* (8), 607 – 633 (2001).

[17] W. E. Winkler, Matching and Record Linkage. In B. G. Cox et al. (ed.), *Business Survey Methods*. 355 – 384. New York: John Wiley (1995).

[18] W. W. Cohen and J. Richman, Learning to match and cluster large high-dimensional data sets for data integration. In *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '02),* 475 – 480. Alberta, Canada: ACM (2002).