International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

Near Duplicate Detection In Relational Database

Bhagyashri. A. Kelkar [*], Prof. K. B. Manwade [**], Prof. G. A. Patil [***]

[*] *ME (CSE) Research Scholar, D. Y. Patil college of Engg. & Tech., Kolhapur, Maharashtra, India.

[**] Head of Dept, Dept. of CSE, Ashokrao Mane College of Engg, Wathar, Dist Kolhapur, Maharashtra, India.

*** Head of Dept, Dept. of CSE, D. Y. Patil College Of Engg. and Tech., Kolhapur, Maharashtra, India.

***Abstract*—Near Duplicate detection is an important precess for many database based applications. Accurately identifying duplicate entities between multiple data sources is a big challenge to organizations and researchers. To detect the approximately duplicate records that refer to the same real-world entity is important to make the database more concrete and achieve higher data quality. In this process, ideally each record must be compared with every other record in dataset for finding duplicates. It is possible to reduce search space for record comparisons by using mutual exclusion property of tuples. In this research paper we analyze two types of blocking algorithms, namely, the adaptive sorted neighborhood method (ASNM), and iterative blocking and their combination with Jaro Winkler distance for string matching. Experimental evaluation on real dataset shows that, adaptive sorted neighborhood method along with Jaro Winkler distance algorithm outperforms in terms of precision and recall and requires very less number of comparisons than iterative blocking method. The experiments also highlight that, strings matching threshold gives optimal results if value is in range of 85% to 90%.**

Keywords: Record linkage, near duplicate detection, Iterative Blocking, Adaptive sorted neighborhood method(ASNM), Jaro-Winkler Distance.

I. INTRODUCTION

In a databases based application, often there is need to consolidate data from different sources. The data integrated from the multiple sources often involves a 1% to 5% duplicate records. We call two records as nearly duplicates if they identify the same real world entity. Many industries and systems depend on the accuracy of databases for taking their competitive initiatives, strategic and operational decisions. Quality of the information stored in the databases, can have significant cost implications to a system that relies on information to function and conduct business. Low data quality results in incorrect reporting, inability to create a complete view of the customers from various segments and also results in poor customer service. Poor data quality costs billions of dollars to businesses in postage, printing, and staff overhead. Hence, data quality improvement is an essential step before establishing data warehouse.

Presence of nearly duplicate records is result of expressing an entity using different values due to inconsistent expression habit, abbreviation, type errors & different formats. Thus, data quality is often compromised by many factors, including data entry errors (e.g., Microsft instead of Microsoft), missing integrity constraints (e.g., allowing entries such as EmployeeAge = 234), and multiple conventions for recording information (i.e. lexical heterogeneity) e.g., address recorded as "44 W. 4th St." versus "44 West Fourth Street".

II. REDUCE SEARCH SPACE WITH BLOCKING

The naive method for finding nearly duplicate records is to compare all records in the database, pair wise. Obviously, this method is practically infeasible due to time complexity of $O(n^2)$. To lower the time complexity, various techniques are proposed. Blocking[2] refers to the procedure of subdividing database records into a set of mutually exclusive subsets (blocks) under the assumption that no matches occur across different blocks. In this paper, we analyze performance of iterative blocking[4] and adaptive sorted neighborhood method[3].

A. Iterative Blocking:

Most blocking techniques process blocks separately and do not exploit the results of other blocks. In iterative blocking, results of blocks are reflected to subsequently processed blocks. Blocks are now iteratively processed until no block contains any more matching records. When two records match and merge in one block, their composite may match with records in other blocks. The same pair of records may occur in multiple blocks, so once the pair is compared in one block, we can avoid comparing it in other blocks.

B. Adaptive Sorted Neighborhood Method:

In this blocking method, blocking key values adjacent to each other, but that are significantly different from each other are found using an appropriate string similarity measure. These boundary pairs of blocking keys are then used to form blocks, i.e. they mark the positions in the sorted array where one window ends and a new one starts. This approach can therefore be seen as

a combination of traditional blocking and the sorted neighborhood approach. Record pairs that were removed in the indexing step are classified as non-matches without being compared explicitly.

III. FIELD SIMILARITY COMPUTATION

Approach to reduce the computational efforts is to minimize the number of costly string comparisons that need to be made between records. Approximate string comparisons are carried out by using methods like edit distance, Jaccard similarity, Cosine similarity, matching tree etc. Jaro-Winkler algorithm[3] is found effective for fields like name and address details, while comparison functions specific for date, age, and numerical values are used for fields that contain such data. In the proposed system, performance of Jaro-Winkler metric will be evaluated on experimental dataset.

IV. RECORD SIMILARITY COMPUTATION

1. Attribute selection:

A record is usually composed of many attributes, whose similarities decide record similarity. It is vital to select the attributes of the records that participate in the record match, which represent the whole record. Attributes are selected by domain experts depending on people's comprehension of the meanings of data. If the cardinality of an attribute is approximately equal to the cardinality of dataset, then that attribute could not be a duplicate identifier. For example, title is not a duplicate identifier for name database.

2. Attribute weight allocation:

A record is usually composed of many attributes, whose contributions are different in deciding whether two pieces of records are approximate. Attribute weight is allotted according to the importance of its contribution in the process of judging approximately duplicate record. The bigger contribution the attribute makes, the bigger weight need to be allotted. In this paper, effect of attribute weights on precision and recall and F-measure is analyzed.

Field matching step results into formation of a vector that contains the numerical similarity values (F_i) calculated for each pair for selected attributes. Record similarity (RS) between two records R_1 & R_2 is

$$RS(R_1, R_2) = \frac{\sum_{i=1}^{n} F_i * W_i}{\sum_{i=1}^{n} W_i}$$

Where, W_i is weight allocated to i^{th} field and n Fields contribute to form the vector. If $RS(R_1,R_2)$ is greater than user specified threshold, record R_1 is marked as similar to R_2. This stage classifies the compared candidate record pairs into matches & non-matches. IN this paper, effect of threshold value on precision and recall, F-measure and number of record comparisons is analyzed.

V. EXPERIMENTAL RESULTS

Evaluation metrics: The effectiveness of the blocking methods and string similarity algorithms can be measured by precision, recall and F-measure metrics.

True Positive (TP): Corresponds to the number of matched detected when it is really match.

True Negative(TN): Corresponds to the number of non-matches detected when it is really non-match.

False Positive (FP): Corresponds to the number of matches detected when it is really non-match.

False Negative(FN): Corresponds to the numbers of non-matches detected when it is really match.

Precision: Precision is the fraction of true matches over the all number of candidate pairs which are classified as matches.

Recall: Recall is the fraction of matches correctly classified over the all number of matches.

F-measure: F-measure is regarded as the mean of precision and recall values.

Pairwise comparison count: This metric measures the number of pairwise comparisons performed by the algorithm. The lower the count, the more efficient the algorithm is.

Datasets –

Restaurant dataset: The experiments are done on real world Restaurant dataset. Restaurant is a standard dataset which is used in several record linkage studies. It was created by merging the information of some restaurants from two websites: Zagat (331 non-duplicate restaurants) and Fooders (533 non-duplicate restaurants). There are 864 records in this dataset and 112 of them are duplicates. Name, Address, City, Phone and Type of restaurants are attributes of this dataset. Every record has five fields of the following form

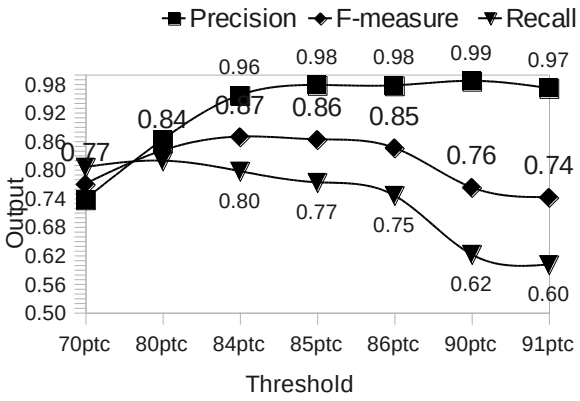
Record (RIDDLE, Restaurant data set):@data
"arnie morton's of chicago", "435 s. la cienega blv.", "los angeles", "310/246-1501", "american", '0'
"arnie morton's of chicago", "435 s. la cienega blvd.", "los angeles", "310-246-1501", "steakhouses", '0'

Implementation of iterative blocking on restaurant dataset using Jaro Winkler string similarity:

Blocks were formed on city in first pass and then on telephone number in second pass. Similarity threshold values were varied from 70% to 91%. The contributing fields i.e. name, address, city, phone and cuisine type were assigned weights as name(30%), address(30%), city(20%), cuisine_type(20%) in the calculating record similarity index . The results of Precision, Recall, F-measure & number of record comparisons are as shown in figure 1.

The results show that, at 85% of similarity threshold, F-measure, precision, recall are at optimal level. At 90% threshold, precision maximize but recall and F-measure drops as more number of records are rejected from comparisons. With increase in threshold value, number of record comparisons also increase.

Figure 1. PRECISION, F-MEASURE,RECALL OBTAINED WITH ITERATIVE BLOCKING, JARO-WINKLER SIMILARITY AND UNEQUAL WEIGHT COMBINATION ON RESTAURANT DATASET



Implementation of adaptive sorted neighborhood with Jaro-Winkler string similarity:

In the first pass, sort key was created by concatenating first two characters of each of following fields : name, address, city, phone and cuisine type in order. In the second pass sort key was created by concatenating first two characters of fields city, phone, cuisine type, name and address in sequence. The data was then sorted on sort key 1 in first pass and on sort key 2 in second pass.

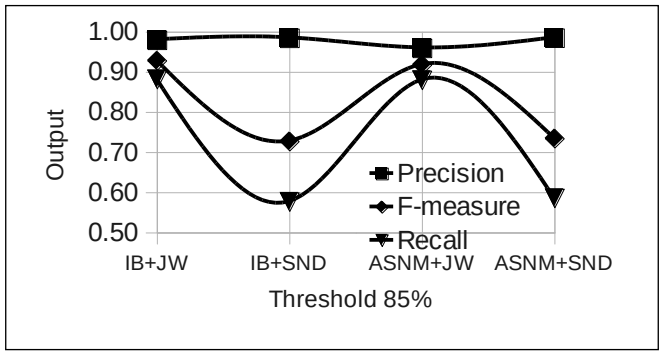
A fixed window size of 10 records is defined initially and changed adaptively when a match is found. Comparisons of records take place within the window only. First record in result set is selected as base record and compared with remaining records one after the other within the window. Jaro-Winkler string similarity is used to find mathcing index of the records. If a matching record is found, i.e. match index > threshold value, remaining records in the window were bypassed and next record of current base record is marked as new base record. If the new record is already matched with some other record, it need not be processed further and so bypassed and next base record is selected for processing. This process is continued till all records exhaust. The results of Precision, Recall, F-measure and number of comparisons required are as shown in following table.

TABLE I. RESULTS OF ADAPTIVE SORTED NEIGHBORHOOD METHOD WITH JARO-WINKLER SIMILARITY AND EQUAL FIELD WEIGHTS

Threshold %	Precision	Recall	F-measure	#Comparisons
70	0.79	0.87	0.83	4258
80	0.85	0.88	0.86	4307
84	0.93	0.88	0.90	4382
85	0.96	0.88	0.92	4406
86	0.97	0.86	0.91	4430
90	0.97	0.76	0.85	4520
91	0.98	0.74	0.84	4544

Above results show that, at 85% of similarity threshold, F-measure, precision, recall reach optimum values. At 91% threshold, precision maximizes but recall and F-measure drop as more number of records are rejected from comparisons. Adaptive approach results in tremendous drop in number of comparisons.

Figure 2. COMPARATIVE CHART OF ITERATIVE BLOCKING(IB) AND ADAPTIVE SORTED NEIGHBORHOOD(ASNM) IN COMBINATION WITH JARO-WINKLER(JW) AND SOUNDEX(SND) SIMILARITY



Results obtained by combination of iterative blocking and adaptive sorted neighborhood method with Jaro Winkler and Soundex at threshold value of 0.85 & equal weight assigned to contributing fields are shown in Fig. 1. It shows that Jaro-Winkler string similarity works well than Soundex for maximizing precision, recall and F-measure.

Figure 3. RECORD COMPARISONS REQUIRED FOR ITERATIVE BLOCKING(IB) AND ADAPTIVE SORTED NEIGHBORHOOD(ASNM) IN COMBINATION WITH JARO-WINKLER(JW) AND SOUNDEX(SND) SIMILARITY

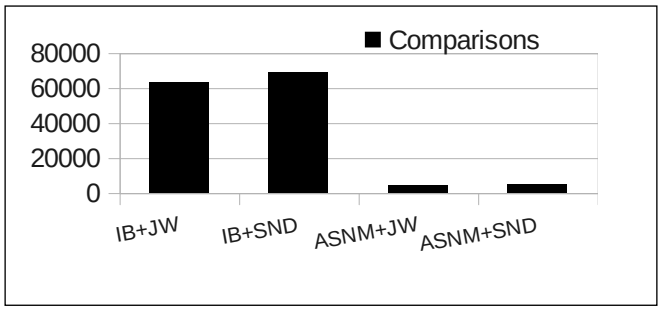


Fig. 3 shows that, adaptive sorted neighborhood method reduces number of record comparisons by factor of 93%.

Real customer dataset: Real dataset having 10644 customer records was loaded into mysql dataset. The data contains following fields :

Title, full name, address line 1, address line 2, address line 3, City, state code, pin code, date of birth, sex code, interest category, mother tongue

Dataset have 2167 duplicates blocks and total of 7056 duplicate records.

Implementation of iterative blocking using Jaro-Winkler string similarity:

The data was preprocessed for removing known wrong values in city and title fields using update queries. No specific field can be selected to form blocks. So blocking key was formed by concatenating first four characters of full name and address line 1. First record in the block was marked as base record of the block. All other records were compared one by one with base record to compute record similarity.Similarity threshold values were varied from 70% to 99%. All the contributing fields were assigned equal weights while calculating record similarity index. The results of Precision, Recall, F-measure & number of comparisons required are as shown following tables.

TABLE II: RESULTS OF ITERATIVE BLOCKING WITH JARO-WINKLER SIMILARITY AND EQUAL FIELD WEIGHTS

Threshold %	Precision	Recall	F-measure	#Comparisons
70	0.7	0.47	0.56	3021245
80	0.5	0.5	0.5	3588053
84	0.78	0.54	0.64	4155076
85	0.80	0.55	0.65	4437091
86	0.81	0.56	0.66	4723404
90	0.87	0.64	0.74	6050503
91	0.89	0.65	0.75	6332849
93	0.9	0.68	0.78	6832347
95	0.91	0.69	0.79	7262777
97	0.92	0.69	0.79	7700536
99	0.93	0.67	0.78	8337269

TABLE III:RESULTS OF ITERATIVE BLOCKING USING JARO-WINKLER SIMILARITY AND UNEQUAL FIELD WEIGHTS

Threshold %	Precision	Recall	F-measure	#Comparisons
70	0.73	0.51	0.60	2381521
80	0.79	0.57	0.66	3418051
84	0.83	0.62	0.71	4235334
85	0.85	0.64	0.73	4713467
86	0.87	0.66	0.75	5280880
90	0.96	0.79	0.86	7218565
91	0.97	0.80	0.88	7599389
93	0.99	0.82	0.90	8149949
95	1.0	0.82	0.90	8546981
97	1.0	0.80	0.89	8944531
99	1.0	0.77	0.87	9349063

Comparison of table II with table III indicate that, assigning unequal weights for forming similarity index results in improved precision, recall & F-measure and increase in number of record comparisons.

Implementation of adaptive sorted neighborhood with Jaro-Winkler string similarity:

Blocking key was formed by concatenating first four characters of full name and first two characters of address line 1. The data was then sorted based on sort key 1.

Results show that, at 93% of similarity threshold, F-measure, precision, recall reach optimum values. At 95% threshold, precision maximizes but recall and F-measure drop as more number of records are rejected from comparisons. Adaptive approach results in tremendous drop in number of comparisons. Adaptive sorted neighborhood method reduces number of record comparisons by factor of 82%.

TABLE IV. RESULTS OF ADAPTIVE SORTED NEIGHBORHOOD METHOD WITH JARO-WINKLER SIMILARITY AND EQUAL FIELD WEIGHTS

Threshold %	Precision	Recall	F-measure	#Comparisons
70	0.81	0.57	0.67	53678
80	0.82	0.58	0.68	54999
84	0.85	0.62	0.71	57488
85	0.87	0.63	0.73	58650
86	0.88	0.64	0.74	59655
90	0.93	0.70	0.80	63683
91	0.93	0.71	0.81	64396
93	0.94	0.73	0.82	65653
95	0.95	0.72	0.82	66719
97	0.95	0.71	0.81	67856
99	0.95	0.69	0.80	69308

If highest weight-age is given to name field, following results are obtained.

TABLE V. RESULTS OF ADAPTIVE SORTED NEIGHBORHOOD METHOD USING JARO-WINKLER SIMILARITY AND UNEQUAL FIELD WEIGHTS

Threshold %	Precision	Recall	F-measure	#Comparisons
70	0.78	0.58	0.66	47351
80	0.80	0.60	0.69	46000
84	0.86	0.66	0.75	53992
85	0.88	0.68	0.77	55757
86	0.91	0.72	0.80	57717
90	0.97	0.81	0.88	64288
91	0.98	0.82	0.89	65485
93	0.99	0.83	0.90	67220
95	1.0	0.81	0.90	68503
97	1.0	0.80	0.89	69637
99	1.0	0.77	0.87	70838

Dbgen Dataset:

Synthetic dataset generated from data generator software - dbgen having 10252 records and 501 duplicates blocks. The data contains following fields :

social security number, first name, middle name, last name, street number, address, apartment number, city, state, zip code

First blocking key selected was state field and second blocking key was social security number field. Results show that, on dbgen dataset, giving highest weightage to name field produces optimal results with slight increase in number of comparisons. Iterative blocking and Jaro Winkler similarity achieves highest precision of 1.00, highest F-measure of 0.94, highest recall of 0.89 and requires only 0.13% comparisons against one to one comparisons of 10.5 core comparisons when highest weight is given to name and address fields. At 62% similarity threshold recall , F-measure, precision are maximum.

TABLE VI. RESULTS OF ITERATIVE BLOCKING WITH JARO-WINKLER SIMILARITY AND UNEQUAL FIELD WEIGHTS

Threshold %	Precision	Recall	F-measure	#Comparisons
70	1.00	0.77	0.77	248720
80	1.00	0.77	0.69	46000
84	1.00	0.77	0.75	53992
85	1.00	0.77	0.77	55757
86	1.00	0.77	0.80	57717
90	1.00	0.76	0.88	64288
91	1.00	0.82	0.89	65485
93	1.00	0.83	0.90	67220
95	1.00	0.81	0.90	68503
97	1.00	0.80	0.89	69637
99	1.00	0.77	0.87	70838

VI. CONCLUSION

Following points highlight the conclusion drawn out of the work:

Whole record need not be compared to identify duplicates. Fields which represent the whole record must be used for the similarity vector computation than comparing all the fields. Those fields can be identified as the fields having higher cardinality. Weights assigned to contributing fields play important role in efficiency of the process. Higher weights must be assigned to the most discriminating fields in the record. Highest weight must be assigned to the field having lowest cardinality in the whole dataset. This weight combination results into higher precision, recall and F-measure than that of equal weights. Similarity threshold used to compute similarity vector also affects the precision, recall, F measure values. On real dataset like 'Bank' similarity threshold of 95% results in optimum results, whereas on restaurant dataset, similarity threshold of 85% results into optimum results. i.e. similarity threshold value yielding optimal results is dependent on database characteristics. On average similarity threshold of 90% produces optimal results across datasets. Iterative and adaptive sorted neighborhood blocking methods perform nearly equal in terms of F-measure, precision and recall. Adaptive sorted neighborhood method requires very less number of record comparisons than that of iterative blocking. Adaptive sorted neighborhood method is useful for large datasets. The overhead is to find proper sort key. Jaro-Winkler similarity function always outperforms than Soundex similarity and Jaro-Winkler similarity is best suited for name and address data.

References

[1] Xiaochun Yang Bin Wang Guoren Wang Ge Yu Key "RSEARCH: Enhancing Keyword Search in Relational Databases Using Nearly Duplicate Records" (2010 IEEE. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering)

[2] Ahmed K. Elmagarmid, Senior Member, IEEE, Panagiotis G. Ipeirotis, Member, IEEE Computer Society, and Vassilios S. Verykios, Member, IEEE Computer Society Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S.Verykios. "Duplicate Record Detection: A Survey"(IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 19, NO. 1, JANUARY 2007)

[3] Su Yan, Dongwon Lee, Min-Yen Kan, and Lee C. Giles. "Adaptive sorted neighborhood methods for efficient record linkage." In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, 2007.

[4] Steven Euijong Whang, David Menestrina, Georgia Koutrika, Martin Theobald, and Hector Garcia-Molina. "Entity resolution with iterative blocking". In Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2009.

[5] A. E. Monge and C. Elkan, "An efficient domain-independent algorithm for detecting approximately S.E. Whang, D. Menestrina, G. Koutrika, M. Theobald, H. Garcia-Molina, Entity resolution with iterative blocking, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09), 2009, pp. 219–232

[6] Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg. "Adaptive name matching in information integration" IEEE Intelligent Systems, 18(5):16-23, Sep/Oct 2003.

AUTHORS

Bhagyashri A. Kelkar received the B.E. in Computer Sci. & Engg. from Walchand College of Engg., Sangli in 1995. She is currently pursuing M.E. degree in Comp. Science & Engg.

Prof. K. B. Manwade received M.Tech. degree in computer science from Shivaji University, Kolhapur. He is working as an Head of Computer sceience Department in Ashokrao Mane Group of Institutes, Wathar.

Prof. G. A. Patil received ME degree in computer sci. & engg. From Walchand College Sangli. He is working as head of department in D. Y. Patil College of engg. & tech., Kolhapur

.