American-Eurasian Journal of Scientific Research 12 (5): 255-259, 2017
ISSN 1818-6785

DOI: 10.5829/idosi.aejsr.2017.255.259

Rule-Based Method for Entity Resolution Using Optimized Root Discovery (ORD)

[1]S. Liji and [2] M. Nithya

[1]ME Student, Department of Computer Science and Engineering,
VMKV Engineering College, Tamil Nadu, India
[2]Associate Professor, Department of Compute Science and Engineering,
VMKV Engineering College, Tamil Nadu, India

Abstract: Entity resolution makes out the object alluding to the same real world entity. Entity resolution is carried out by producing rules from a given input data set and applies them to records. Traditional approach randomly assumes that each attributes value as a rule and combines other rules according to the limit criteria. Traditional method is very complex and tiring. The new proposed method is experimentally more accurate and using new algorithms with the property of Optimized Root Discovery. The newly produced rules can be used for any dataset available for entity resolution or identification in an accurate way with minimum time and space complexity.

Key words: Entity resolution · Limit Criteria

INTRODUCTION

In many real-world applications, an entity may appear in multiple sources of data so that the entity may have entirely different descriptions. Entity Resolution is the problem of recognizing and linking or grouping different manifestations of the same real world entity.

Entity Resolution may also be referred to as record linkage, Duplicate detection, Reference resolution, Deduplication, Fuzzy match, Duplicate Detection, Object consolidation, Reference reconciliations, Object Identification, Identity uncertainty, Hardening Soft databases, Approximate Match, Merge/purge, Household matching, Reference matching, House holding, Entity Clustering and doubles.

Traditional ER methods get an outcome by the similarity comparison process amongst records which assumes that records pointing to the same entity are matching to each other. Anyway, such property may not hold in practice in the case of traditional ER methods. In some cases, traditional ER approaches may insufficient for this.

The match functions used in the traditional ER methods are following the match score schemes. In this method the checking of whether any two values or records point to the same entity or not takes place. If the match value is within the limit value, then the match is there. Otherwise, concludes that no match is there. Any one of the available match functions like an exact match, distance, cosine, TF/IDF can be applied.

This paper is organized as follows: Section I is an introduction. Section II is related work. Section III is the existing system. Section IV has proposed a system. Section V explains performance evaluation and finally Conclusion.

Related Work: Monge and Elkan uses an algorithm called smith-waterman domain dependent algorithm in work [1] to trace out the relation between DNA or protein sequences. Paper [2] discussed a domain independent method namely pair wise record matching.

In work [3] a solution involving two steps are proposed. One step is an algorithm for author-title clusters and other for string matching using n-grams. This method has a disadvantage that it uses a larger number of pairwise comparisons. In work done by [2], Alvaro and Charles proposed a pair of solutions. One of the solutions using union-find data structure and other one using priority queue algorithm. This is also having some cons like even the non-duplicate item found as a duplicate.

Corresponding Author: S. Liji, ME Student, Department of Computer Science and Engineering,
VMKV Engineering College, Tamil Nadu, India.

In work done by [4] describes a system of two task database integration. The integration methods include schema integration and entity identification. These methods lead to the worst complexity of time and high error rate and also it requires the manual generation of rules for entity identification.

Active atlas method is used for object identification in work [5]. A decision tree is implemented in this work but it can compare only two objects at a time and this leads to increased number of comparisons. The work [6] tries to overcome this problem using blocking methods. This method partitions the records into different blocks based on a key called blocking key. But it fails to ensure the relationship between the records and blocks.

Ganti and Motwani in [7] suggests a solution which avoids the global distance function problem but fails in some cases where record pointed to same entity breaks. Lingli, Jianzhong and Hong introduced a better method in [8] as compared to other works mentioned here, but it produces some rules in the process of rule generation. This work is the base of our work which reduces the complexity of space and time with the help of ORD.

Existing System: Existing system in [8] produces rules for the identification of a particular entity. Entity identification steps involve the identification of all the entity set and then identify the training dataset from entity set for the creation of new rules. Entity wise rule generation is done here. [8] Produces single individual rule for respective attribute-value. Another factor mentioned is the coverage of the rule. Coverage is defined as the objects that can be identified by accomplishing the clauses of the rule. There are two types of all considered valid rules and invalid rules. Valid rules are rules which have no coverage on other entity; otherwise, it is an invalid rule.

Based on the validity status obtained after checking of the generated rules, it can be stored in X, Y or Rs. All valid rules are sent to Rs and invalid rules are sent to X, Y. There is a length parameter Ln given by the developer to reduce the no of attributes in the rule. X accommodates invalid rules with Ln value 1. Other invalid rules are placed in X. After the first round of rule, creation checks the L_n threshold. If the limit satisfied conjunction of X and Y is carried out and then again examines its validity. If the status is valid, then place it in Rs otherwise place in Y. Now got new rules in Y and again check the Ln threshold of these rules in Y. Continue this conjunction process of X and new rules in Y till the Ln point is met.

Now the aim is to check the possibility of the Rules in Rs that is whether it can find all the objects in the training set following the rules in Rs. If not all the objects are identified then generate a rule for the left out objects by the conjunction of their all attribute-value. One object can be resolved using more than one rule. Then the number of rules may be huge. This can be avoided using a greedy algorithm. This method supports the rules which can find more than one single object. Thus obtained rules can be applied to the entire dataset for entity resolution.

In the Existing system, the number of rules produced is high and it is observed as a complex task. Single rule is generated for each attribute-value and in some necessary situations, the conjunction is needed. This lead to the increase in some rules. More over in certain cases the same object is identified using more than one rule, so existing system need an extension for avoiding this situation.

Proposed System: Following are the terms used in this paper.

Rule Syntax: Rule consist of an RHS and LHS which represents entity and conjunction of clauses. Clause denotes the combination of attribute and its value. In this work, the attribute is also referred as a feature. Rule represented as the following form

$E_1 => C_1 ? C_2 ? C_3 ?\ldots\ldots C_I$

Scope: Represents the validity of the rule. The scope of a rule is the entities that can be resolved by the RHS.

Limit: Used to limit the number of clauses in the rule.

Optimized Tree: This is the tree created using various feature-value pairs for rule creation. It is built by selecting the feature that has a minimum number of distinct value as a parent node. Optimized tree reduces the complexity of rule generation. Figure 1 shows the Architecture of the proposed system.

Source Entity Set Creation: Source entity set is created from any raw dataset. This work follows the manual creation of the source entity set with more than one attributes in the raw dataset.

Input Data Set Creation: Input dataset is produced by random sampling method from the Source Entity set according to a particular feature.

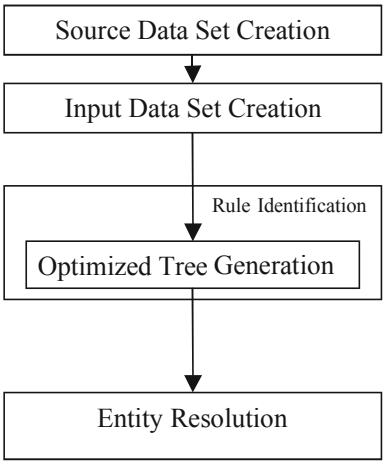

Fig. 1: Architecture of Proposed System

Rule Identification:
Algorithm 1: Rule_Identification algorithm (RI)
Input: Limit 1, T_1
Output: RuleSet RS

```
I ← {I1,I2 ……..In}
Initialize
RS = Ø
List = Ø
OpR = Op_Build_tree(I)
while List is not empty
    Node N1 = List[0]
    List.remove[0]
    R = combine N1.Parent.Value and N1.F = n.V
    if Child of R is mutually exclusive
        add R to RS
        N1.value = R
    else if R is within Limit-1
        add Children of N1 to List
    end if
end while
Create rules for NULL nodes by combining all parent attributes
Return RS
```

Rule_Identification algorithm takes the tree generated by an Op_Build_tree algorithm and limit value as the input and generates the best rule for an entity.

RI algorithm resolves rules within the given limit and also for null nodes. Null nodes represent the leaf nodes in the tree.

If we reached leaf node that indicates that the rule is not yet found, then create rules by combining all the parent node attributes.

If the parent node is assigned a rule then combining all the nodes till present node and generating the new rule R. Then checking whether the rule is satisfying the limit and whether the rule is mutually exclusive. If satisfied adding rule to the node otherwise null value is assigned.

Optimized Tree Generation:
Algorithm 2: Op_Build_tree
Input: Input Data Set I = $\{I_1, I_2 \ldots. I_n\}$
Output: T_1

```
Initialize
List = Ø
Tree Node OpR = new Tree Node (I)
Add OpR to List
While (List is not Empty)
    N= List [0]
    Remove N from List
    F = FindNextAttribute(member,FList)
    if F is not null
        Add F to FList
        VL = Distinct_Values (F, N)
        for each value V in VL
            Create a node N1 = Node (F, V, N)
                add N1 as Child of N
                add N1 to List
        end for
    end if
end while
Return T1
```

Procedure FindNextAttribute(Member, Flist)

```
Sel = null
for each Feature F in Member's Feature not in FList
    if(Sel==null| Distinct_Count(F)<Distinct_Count(Sel))
        Sel = F
    end if
end for
Return Sel
end Procedure
```

Op_Build_tree algorithm is formed to produce an optimized root tree for every item set along with their feature values or attribute values. Procedure FindNextAttribute plays an important role in this algorithm. It selects the best feature F with a minimum number of distinct values. Distinct_Count is used to find the count of distinct values. There is presently three

values assigned for each node created. They are a feature or attribute F, value V and parent node N. RI algorithm add a rule to each node. VL or Value List consists of best Feature's values. Updating of FList is done by removing N from List.

Entity Resolution: Rules are generated from the input dataset. These generated rules applied to the entire dataset we have and identify the desired entity. All rules are assigned with an individual weight and here it is assumed as 1. In certain cases, an object can be identified by the rules of other entity. This case is solved with the selection of entity with maximum weight. The weight of an entity is the sum of the weight of rules that are fulfilled by the entity.

Performance Evaluation: Performance is an important factor in any case where accuracy is concerned. We performed an experiment to determine the advantages of our proposed algorithm. We used a dataset where medical diagnosis details of various patients are available. Input data set is derived from the dataset according to the particular feature given by the user. The proposed algorithms are implemented using the Java programming on a corei3PC with Windows 7 OS.

Our method is discussed as the extension of the R-ER in [1]. So the comparison is done with the new method and R-ER. Time, false negative and accuracy are the chosen parameters. Performance evaluation shows that our one is better. Figure 2, 3 and 4 represents the Rule Generation Time (RGT), False Negative (FN) and accuracy measure(A-Measure) plotted against the input percentage. A-Measure is used for accuracy

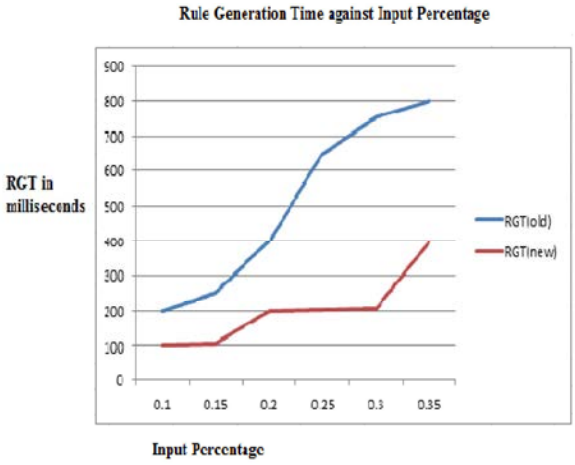

Fig. 2: RGT Plotted Against Input Percentage

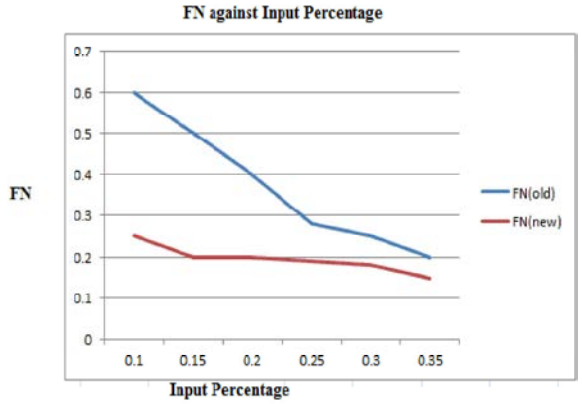

Fig. 3: FN Plotted Against Input Percentage

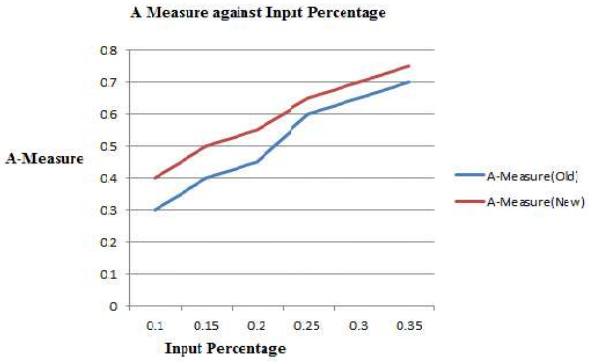

Fig. 4: A-Measure Plotted Against Input Percentage

CONCLUSION

In our work an entity is resolved using rules which satisfy the limit and mutual exclusive property. Optimized tree is generated using the Op_Build_tree algorithm. Rules for leaf nodes are also considered in RI algorithm. Our result evaluation under performance evaluation points that our scheme is more acceptable. This work can be used for effective Prediction or identification of real world entities with the use of generated rules.

REFERENCES

1. Alvaro, E. Monge and Charles P. Elkan, 1996. The field matching problem: Algorithms and applications, KDD-96 Proceedings.
2. Monge and C. Elkan., 1997. An Efficient Domain Independent Algorithm For Detecting Approximately Duplicate Database Records. In Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery, Arizona,

3. Jeremy A. Hylton, 1996. Identifying and Merging Related Bibliographic Records, M.I.T. Laboratory for Computer Science Technical,
4. Ganesh, M., Jaideep Srivastava and Travis Richardson, 1996. Mining Entity-Identification Rules For Database Integration, KDD-96 Proceedings
5. Sheila Tejada, Craig A. Knoblock and Steven Minton, 2001. Learning Object Identification Rules For Information Integration, Information Systems, 26(8): 607-633.
6. Baxter, R., P. Christen and T. Churches, 2003. A comparison of fast blocking methods for record linkage. In Proceedings of the ACM SIGKDD workshop on data cleaning, record linkage and object identification.
7. Chaudhuri, S., V. Ganti and R. Motwani, 2005. Robust identification of fuzzy duplicates, in Proc. 21st Int. Conf. Data Eng., pp: 865-876.
8. LingliLi, JianzhongLi and Hong Gao, 2015. Rule Based Method For Entity Resolution, IEEE Trans.Knowl. Data.Eng. 27(1): 250-263.