International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-5 Issue-5, October 2015

Rule-Based Entity Resolution using Distinct Tree

Mitty Abraham, Safiya K.M

Abstract — Entity resolution identifies object referring to the same entity .Entity resolution is performed by generating rules from training set and applies them on records. Traditional ER considered each attribute value as the rule in a random fashion and performs conjunction with other rules according to length threshold .This method is very complex and tedious. Our proposed method generated rules from a distinct tree using RL method, which consider the length criteria and RNN methods which does not. Distinct tree is formed by arranging attribute and its value of records in the training set in a particular fashion .These generated rules are applied to the dataset for entity identification .Our experimental results show that the proposed method is more accurate.

Index Terms— Entity resolution, Length criteria

I. INTRODUCTION

Entity is an object which has a physical or logical existence .Relation database is a collection of these entity .Each tuple in relation represent an entity ,while each entity may contain one or more tuples.

Entity resolution identifies object referring to the same entity. It has application in government, judiciary ,public health, shopping etc., There are various task in entity resolution like deduplication ,record linkage and reference matching . Deduplication is grouping of records that related to same entity .Record linkage is the linking of records in different datasets . In reference matching incorrect records are match to correct ones in the table.

The criteria used in entity resolution is match function .Whether two records are refer to same entity is checked using match function of their one or more attribute values .If the match sore is within the threshold ,then we said these records are match .Otherwise it does not. The match functions used are exact match ,distance ,cosine, TF/IDF etc.,

To reduce number of pair wise comparison blocking methods [3] are used .The records are divided into blocks based on the blocking key. But this does not sure that all related records in the same block .The related records are more similar than others .But this break in some cases .So rule based method is used [1].In this rule generated based on attribute-value and apply these rules to record to find which entity it refer.

Organization of paper as follows: section 1 is introduction. Section 2 is related work .Section 3 is existing system. Section 4 is proposed system .Section 5 gives experimental evaluation and then to conclusion.

II. RELATED WORKS

In the work done by [8] they used smith-waterman algorithm to find relationship between dna or protein sequences. however their algorithm was domain dependent .Alvaro and Charles [6] proposed a domain independent method know as pair wise record matching.

In work [7] related bibliographic records are identified using two steps .First step is algorithm for author title clusters and second is string comparisons with n-grams .But this method leads to excessive pair wise comparison .To reduce this comparison Alvaro and Charles [6] proposed two methods .First method using union-find data structure and latter is priority queue algorithm .This method uses global distance function and the problem is even the non duplicate record shown as duplicate .

Ganesh, Jaideep and travis [5] develop a method for database integration, which includes two tasks: schema integration and entity identification. Rules for entity identification are generated manually .But this method consume more time and chance of error to occur are high.

In work [4] Active Atlas method is used for object identification .Decision tree is used to teach rules for record matching .In this we can compare only two objects at a time, which leads to large number of comparisons .To avoid such a large number of comparisons blocking methods [3] were introduced .These blocking method divide record into various blocks based on blocking key .But it is not certain whether all the related records belong to one block.

In work [2] used compact set and sparse neighbourhood methods to avoid the global distance function problem .But disadvantage is record pointed to same entity is more similar break in some case. So Lingli, Jianzhong and Hong proposed [1] a rule based method .In this rule are generated and based on this rule entity identification is done .This method produce large number of rules during its rule generation process .So we proposed a method to reduce the complexity of above one.In the proposed method rule is generate from tree by two algorithm. First take length criteria, latter one does not consider length.

Manuscript published on 30 October 2015.
***Correspondence Author(s)
Mitty Abraham, PG Scholar, Department of Computer Science and Engineering, Ilahia College of Engineering and Technology, MVPA, (Kerala). India.
Safiya K.M, Asst. Professor, Department of Computer Science and Engineering, Ilahia College of Engineering and Technology, MVPA, (Kerala). India.

III. EXISTING SYSTEM

Existing system generate rule and apply to records for entity identification. First of all the entity set is found, then we obtain a training set from entity set for rule generation. Rule sets are generated entity wise. Each attribute-value is considered as a single rule. Coverage of a rule is the objects that can be identified by satisfying the clauses of the rule. Rule is invalid if it has coverage in another entity, otherwise it is a valid rule.

The validity of above rules are check .If valid stored in R and if invalid store in L^X , L^Y. L^Y consist of all the generated invalid rules. L^X consist of invalid rule with length 1. Length is provided by the programmer. Length is used to decrease the number of attributes in the rule. After the first round of rule generation, check the length threshold. If within, conjunction of L^X & L^Y is performed and check its validity .If valid add to R, else add to L^Y. Then again check the length threshold of newly generated rules in L^Y .We continue conjunction of L^X & newly generated rules of L^Y until the length threshold is met.

Now we test whether ,it possible to identify all the objects in training set using the rules in R. If any object are left out, we then create rule for each object by conjunction of there entire attribute-value. Reduce the length of rule by eliminating each attribute-value, until it becomes invalid .There can be possibility to find one object from one or more rules .So we reduce the number of rules by greedy algorithm. In this select those rules that can be used to find more than a single object. These rule are apply to the entire data set for entity identification.

More rules are generated, so it's a complex process. Each attribute –value taken as a rule and done conjunction of rules if needed .so this lead to large number of rules. Sometime more than one rule can be used to identify a single object, So in existing system and additional step is used to avoid such rule.

IV. PROPOSED SYSTEM

Terms used in this paper.

Syntax for rule – The rule has both LHS and RHS.LHS represents the entity and RHS denotes the conjunction of clauses that can be used to identify the entity. Clause is the combination of attribute and it's value.Rule in the form shown below.

$e_1 \Rightarrow c_1 \wedge c_2 \wedge \ldots\ldots \wedge c_i$

Coverage-It used to find the validity of the rule.Coverage of a rule is the objects that can be identified by satisfying the RHS of the rule

Validity-Rule is invalid if it has coverage in another entity ,otherwise it is a valid rule.

Length requirement-It is used to decrease the number of clauses in the rule.

Distinct Tree- Tree is generated using varies attribute and its values for rule generation .It is formed by choosing the attribute, which have least number of distinct value as the parent node. This method is chosen to reduce the complexity of rule generation. Architecture of proposed system shown in figure 1.

A. *Entity Set Creation*

In this entity set is generated from raw dataset. We manually create the entity set using one or more attribute in the raw dataset.

B. *Training Set Creation*

Training set is generated by random sampling of records from the entity set according to a parameter. Parameter can be any number of records from each entity set. For example, 20% of records can be taken from each entity and the number of entity is 10.Let T be the training set and E the entity. Therefore $T_i = \max\{0.2|E_i|,1\}$.

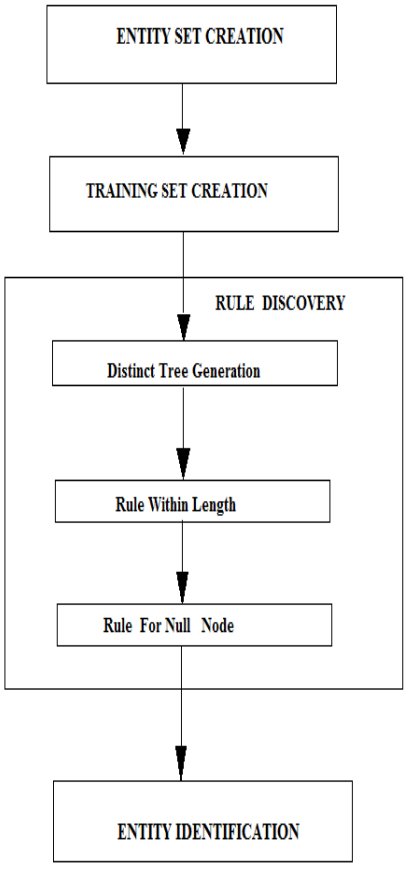

Fig 1. Architecture of proposed system

C. *Rule Discovery*

Algorithm 1 :Rule Discovery Framework

Input : length threshold l, Training set T ={T_1 ,T_2 …..T_n }

Output: Rule set R

1: T ← {T_1 ,T_2 ……..T_n}

2: T_{Tree}← DistTreeGen(T)

3: R_{Len} , N← RL (l, T_{Tree})

4: for each n_i in N do

5: R_{RNN}← RNN(n_i)

6: end for

7: R← $R_{Len} \cup R_{RNN}$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

8: Return R

Given length threshold and training set T.Tree is formed by performing the algorithm DistTreeGen using training set.Rules for entity identification using length threshold are found by RL algorithm. The rules that cannot be found by RL algorithm can be determined by RNN algorithm, which does not use the length threshold.

1. *Distinct Tree Generation*

Algorithm 2 : DistTreeGen

Input: Training set T ={T_1 ,T_2 ……..T_n }

Output : T_{Tree}
1: Initialize
2: Sel_attr=Ø
3: for each r_i in T_i
4: records ← r_i
5: end for
6: T_{Treei} ←Perform Group (records, Sel_attr)
7: Return T_{Tree} ={T_{Tree1} ∪ T_{Tree2}∪ … .∪T_{Treen} }

8:procedure Perform Group(records, Sel_attr)
9: if (Sel_attr < no: of attributes) then
10: Best_attr, Values←Get Best Attr (records)
11: for each value in Values do
12: n_i ← NodeGen(Best_attr,value)
13: T_{Treei}← n_i
14: PerformGroup(records,Sel_attr)
15: end for
16: end if
17: return T_{Treei}

18:end procedure

19: Procedure NodeGen(Best_attr,value)
20: Sel_attr = Best_attr + Sel_attr
21: n_i ← (Best_attr,value)
22: for each r_i in records
23: if(Best_attr value of(r_i) ==value) then
24: records ← r_i
25: end if
26: end for
27: return n_i
28:end procedure

Dist Tree Gen algorithm is used to generate tree for each entity set using their attribute and value. In Perform Group procedure Get Best Attr is used to find the attribute with least number of distinct values and update attribute list by removing that attribute. The result of Get Best Attr are attribute and it's values which are stored into Best_attr and Values respectively .Node is generated using the Node Gen procedure. Perform Group procedure recursively calls until all the attributes are considered. Node Gen procedure generate node using Best_attr and value .It update the records based on node.

2.Rule Within Length

Algorithm 3 :RL
Input : l, T_{Tree}
Output : N, R_{Len}
1: Initialize
2: for each n_i in T_{Treei}
3: re = GetParent(n_i)
4: if (re ! = Ø) then
5: n_i . rule = re
6: else
7: re = GetRule(n_i)
8: if (|re| < = l & re is valid) then
9: n_i . rule = re
10: else
11: n_i . rule = Ø
12: N_i← n_i
13: end if

14: end if
15: end for
16: Return N = {N_1∪ N_2∪ ……∪ N_n}
17:Return R_{Len}={T_{Tree1}.rule ∪ T_{Tree2}.rule ∪…∪ T_{Treen}.rule}

18:Procedure GetParent(n_i)
19: while (n_i != Ø)
20: if(n_i.rule!= Ø) then
21: return n_i.rules
22: n_i = parent of n_i
23: end if
24: end while
25: return null
26 end procedure
27: Procedure Get Rule(n_i)
28: re = Ø
29: while (parent of n_i != Ø)
30: r = n_i
31: if(re == Ø) then
32: re = r
33: else
34: re = re ∧ r
35: end if
36: n_i = parent of n_i
37: end while
38: return re
39: end procedure

RL algorithm used to generate rules for entity identification by length threshold. We use Get Parent procedure for node to find whether rule for parent is set .If the Get Parent procedure returns null then perform Get Rule procedure .Otherwise we assign the rule of parent to the current node .In Get Rule procedure , rule generated by conjunction of nodes till the current node .After that we check whether the length of rule lies within the threshold and rule is valid. If true assign the rule to the node. Otherwise null value is assigned to the node.

2. *Rule For Null Node*

For this method we find rules for the nodes which has null set as their rule. In this procedure to find the rules are as same as RL algorithm except for two things. One is that we do not use the length threshold criteria. Second is that if we could not found the rule even after reaching the leaf node ,we then perform conjunction of the entire nodes in the branch.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

D. Entity Identification

In this we apply the rules generated from training set to the entire dataset for entity identification. Also we assign 1 as the weight of each rule. Sometimes there can be a situation where an object can be identified by the rules of different entity .In such a case the entity with maximum weight is chosen .The weight of entity is the summation of weight of rules that are satisfied by the object.

V. EXPERIMENTAL EVALUATION

We perform experiment to determine the accuracy, false negative and time required to complete the program of our proposed algorithm. Dataset contain medical details of different persons .Training set is derived from the dataset according to the parameter given by the user. Our algorithms are implemented using java programming .The experiments are performed on a corei3PC ,running in windows 7.

Proposed method is compared against R-ER [1].we considered parameters like rule creation time, false negative and accuracy. Compare to R-ER our proposed method generate rules faster. Taking false negative criteria our method is better than R-ER.F-measure is used for accuracy. It is the harmonic mean of recall and precision. Precision representing number of tuples correctly identified to the number of tuples return by method. Recall representing number of tuples correctly identified to the number of relevant tuples. Experimental result shows that our method is more accurate. Figures 2, 3 &4 shows the time ,false negative and accuracy respectively against training percentage.

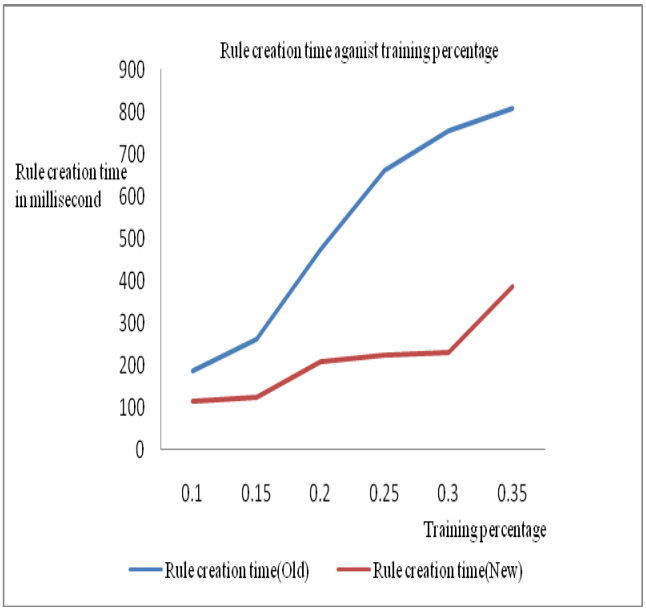

Fig 2.Rule creation time plotted against training percentage

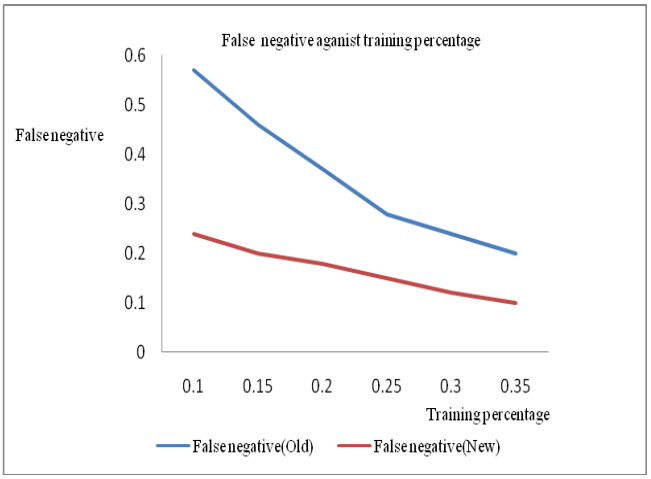

Fig 3.False negative plotted against training percentage

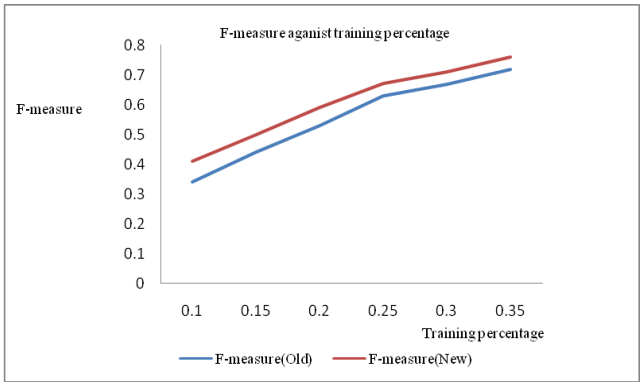

Fig 4.F-measure plotted against training percentage

VI. CONCLUSION

In our paper rules are used for entity resolution. Rules are generated from distinct tree structure using RL and RNN methods. Former one use length criteria and latter does not. Distinct tree is generated from attributes and it's values of records. Our experimental results shows that proposed method is more accurate. Future Work is to include human perspectives for discovery quality rules.

ACKNOWLEDGMENT

The authors wish thanks to the Management and Principal and Head of the Department (CSE) of Ilahia College of Engineering and Technology for their support and help in completing this work.

REFERENCES

1. Lingli Li, Jianzhong Li, and Hong Gao, "Rule-Based Method for Entity Resolution," IEEE Transactions On Knowledge And Data Engineering, vol. 27, no. 1, january 2015.
2. S. Chaudhuri, V. Ganti, and R. Motwani, "Robust identification of fuzzy duplicates," in Proc. 21st Int. Conf. Data Eng., 2005, pp. 865–876.
3. R. Baxter, P. Christen, and T. Churches," A comparison of fast blocking methods for record linkage". In Proceedings of the ACM SIGKDD workshop on data cleaning, record linkage, and object identification, August 2003.
4. Sheila Tejada, Craig A. Knoblock, and Steven Minton,"Learning Object Identification Rules For Information Integration" Information Systems vol. 26, no. 8, pp. 607-633, 2001.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

5. M. Ganesh,Jaideep Srivastava and Travis Richardson"Mining Entity-Identification Rules For Database Integration", KDD-96 Proceedings,1996.
6. Monge and C. Elkan.," An Efficient Domain Independent Algorithm For Detecting Approximately Duplicate Database Records". In Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery, Arizona, May 1997.
7. Jeremy A. Hylton," Identifying and Merging Related Bibliographic Records" M.I.T. Laboratory for Computer Science Technical , June 1996
8. Alvaro E. Monge and Charles P. Elkan ,"The field matching problem: Algorithms and applications" KDD-96 Proceedings,1996

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)