Entity Resolution Methods–A Survey

Ammu Archa.P, Lekshmy.D.Kumar

Abstract— *In the real world, entities have two or more references in databases. Such multiple representations do not share anything in common and thus make duplicate detection a difficult task. Entity resolution or record linkage or deduplication is the process of identifying the records that refer to the same entity. Entity resolution is a challenging task particularly for entities that are highly heterogeneous and of low data quality. Due to the high importance and difficulty of the entity resolution problem, there are numerous approaches that have been proposed to solve ER problems. As there are different entity resolution approaches there is a strong need for comparative evaluations of different schemes. In this paper, different frameworks for entity resolution are studied.*

Keywords— *ER Diagram*

I. INTRODUCTION

In today's IT-based economy the use of database is very high. Matching records that relate to the same entities from several databases is recognized to be of great importance in many application domains [1]. In many applications, a real-world entity may appear in multiple data sources so that the entity may have quite different descriptions [2]. For example, in a company database an employee's details may appear in more than one table in different ways. e.g., the same employee may be represented as an employee or manager in different tables. The method of identifying same real world entity which is represented in different ways is called Entity Resolution(ER).

However there are many challenges to ER. One of the main problems is that there is no unique identifier across the databases that would connect two similar entities/data. Furthermore, the data contains errors and missing values [3]. Such problems are generally called as data heterogeneity [4]. To avoid data heterogeneity problem data cleaning [5] is performed before performing entity resolution. The main tasks of data cleaning and standardization are the conversion of the raw input data into well defined, consistent forms, and the resolution of inconsistencies in the way information is represented [6]. After Data cleaning process Entity resolution methods are applied. As there are many names for entity resolution, there are many methods for performing entity resolution. Each of the methods has advantages as well as disadvantages of its own.

II. LITERATURE SURVEY

Typing mistakes while entering data is one of the most common sources of database errors. String comparison techniques can be used to handle this type of errors. Methods like edit distance and affine gap distance [7] can be used for this task. But the records consist of multiple fields in real life situations and thus these methods cannot be efficiently used.

Revised Version Manuscript Received on June 30, 2016.

Ammu Archa.P, Student, Department of Computer Science & Engineering, SCTCE, Thiruvananthapuram, India.

Lekshmy.D.Kumar, Assistant Professor, Department of Computer Science & Engineering, SCTCE, Thiruvananthapuram, India.

In this paper, different methods that are used for matching records with multiple fields are studied. These approaches can be divided into two categories [3]:

• Methods that uses training data to "learn" how to match the records. Probabilistic approaches and supervised machine learning techniques come under this category.

• Methods that uses domain knowledge or generic distance metrics to match records. This category includes approaches that use declarative languages for matching and approaches that devise distance metrics appropriate for the duplicate detection task.

A. CART Algorithm

Breiman *et.al* [9] proposed CART algorithm for entity resolution. CART Algorithm comes under Machine learning approach. Supervised learning technique requires training data in the form of record pairs which should be prelabeled as matching or not [8]. The algorithm is based on Classification and Regression Trees. Classification and Regression Trees is a classification method which uses historical data to construct so-called decision trees. Decision trees are then used to classify new data. The tree is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample. The basic idea of tree construction is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". In this algorithm, only univariate splits are considered. At any node t, the best split is which maximizes a splitting criterion i(s,t). Splitting criteria should be in such a way that the error/impurity should be minimum. Error occurs when a non matching item is classified as a matching item and vice versa. Splitting criteria like Gini criteria [9] is used. Stopping rules control if the tree growing process should be stopped or not. Stopping rules are based on a number of parameters like user specified minimum node size, user specified maximum tree depth etc.

As the classification or regression tree is constructed, it can be used for classification of new data. The output of this stage is an assigned class or response value to each of the new observations. By set of questions in the tree, each of the new observations will get to one of the terminal nodes of the tree. A new observation is assigned with the dominating class/response value of terminal node, where this observation belongs to. Dominating class is the class that has the largest amount of observations in the current node.

B. ALIAS Algorithm

A supervised learning technique requires large number of training samples. Sarawagi and Bhamidipaty [10] developed ALIAS which is a learning-based duplicate detection system that reduces the number of training samples needed by using an additional class known as 'reject' class.

Figure 1 [10] shows the overall design of ALIAS system for deduplication. The three primary inputs to the system are Database of records (D), Initial training pairs (L) and

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Similarity functions (F).Similarity functions are set of *nf* functions each of which computes a similarity match between two records *r*1; *r*2 based on any subset of d attributes. Examples of such functions are edit-distance, soundex, abbreviation-match on text fields, and absolute difference for integer fields.

In the first step, mapper moule map the initial training records in L into a pair. The mapper module takes as input a pair of records *r*1; *r*2, computes the *nf* similarity functions F and returns the result as a new record with *nf* attributes. For each duplicate pair we assign a class-label of "1" and for all the other pairs a class label of "0" is assiged. At the end of this step, a mapped training dataset *Lp* is obtained. These *Lp* instances are used to initialize the learning component of the system. The next step is to map the unlabeled record list D. The mapper is invoked on each pair of records in D*D to generate an unlabeled list of mapped records *Dp*. Next is the interactive active learning session on *Dp* with the user as the tutor. The learner chooses from the set *Dp* a subset S of n. The user is shown the set of instances S along with the current prediction of the learner. The user corrects any mistakes of the learner.

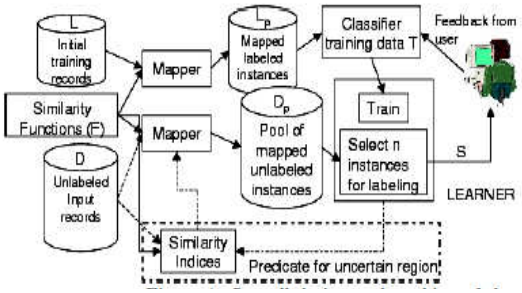

Fig. 1. Overall Architecture of ALIAS Algorithm

The newly labeled instances in S are added to the training dataset *Lp* and the active learner is retrained. The user can inspect the trained classifier and/or evaluate its Performance on a known test dataset and if it is not happy with the learner trained so far, the active learner can select another set of n instances. This process continues in a loop until the user is happy with the learnt model. In each iteration, the user aids the learner by providing new labeled data.

C. MARLIN

MARLIN (Multiply Adaptive Record Linkage with Induction) [11] employs a training-based approach for Entity Resolution. ALIAS algorithm expects the users to provide suitable training data manually, while MARLIN is based on (semi-)automatic training method. The training phase consists of two steps. First, the learnable distance metrics are trained for each data entry. The training data is obtained by taking pairs from the set of paired duplicate records. Because duplicate records may contain individual fields that are not same, training data can be noisy. However, this is not a serious problem for two reasons. First, noisy fields that are unhelpful for identifying duplicates are considered irrelevant by the classifier that combines similarities from different fields. Second, the presence of such pairs in the database shows that there is a degree of similarity between such values, and using them as training data allows the learnable metric to capture such similarities.

After distance metrics are created, they are used to compute distances for each field of duplicate and non-duplicate record pairs to obtain training data for the binary classifier in the form of vectors composed of distance features. The duplicate detection phase starts with generation of potential duplicate pairs. Since, producing all possible pairs of records and computing similarity between them in large databases is too expensive MARLIN utilizes the *canopies* clustering method [12] using Jaccard similarity, a computationally inexpensive metric based on an inverted index, to separate records into “canopies” of potential duplicates.

Learned distance metrics are then used to calculate distances for each field of each pair of potential duplicate records, thus creating distance feature vectors for the classifier. Confidence that a data belongs to the class of duplicates are estimated using the binary classifier for each candidate pair, and pairs are sorted by increasing confidence. An overall view of MARLIN is presented in Fig.2 [11].

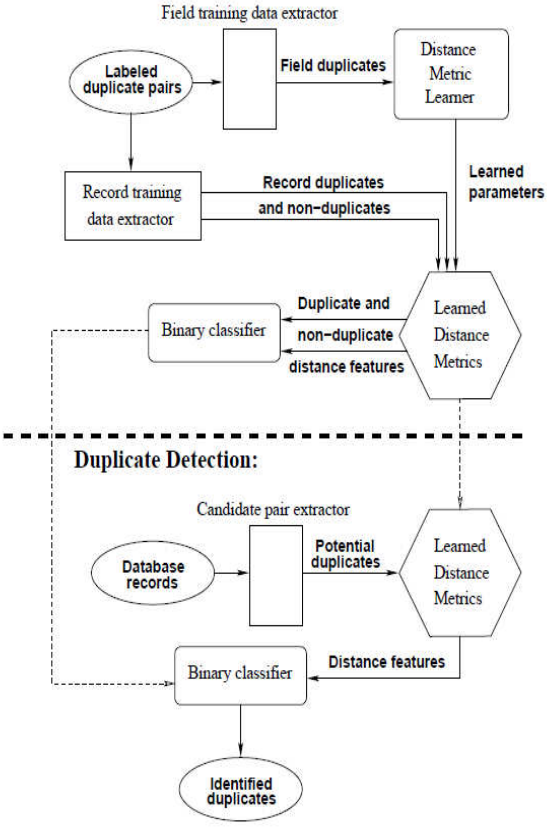

Fig. 2. Overall Design of MARLIN

D. Rule Based ER

Rule based ER is a special case of distance-based approaches in which rules are used to define whether two records matches or not. Lingli *et al* [2] proposed a rule based approach for ER problem. Each rule consists of two clauses- If clause and the then clause. ER rule set should satisfy length requirement and PR requirement. Length

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

requirement suggest that the length of ER rule created should not increase a threshold length. PR requirement suggests that the rule created should contain only positive clauses. But in some cases these requirements can be relaxed. If there are records that cannot be covered using valid positive rule, valid negative rules can be used. Fig 3 shows the overall architecture of rule based ER system.

Rules are generated from the database which is given as input. Each attribute-value pair is considered as a rule and the coverage is checked. Coverage is the number of records in the database that can be satisfied using a particular rule. If the rule does not cover all the records in the database then the record that is not covered is considered and that attribute-value pair is considered as a rule. This process is continued until all the records in the database are covered. After generating the rules minimum rule is generated ie, in some rules some clauses may be irrelevant. Even after removing those particular clauses, the coverage of the rule will remain unchanged. So, such clauses are removed from the rule set. If no valid PR can be used to cover all the records in the database, negative clauses are also considered. After generating all valid minimum rule set, Entity Resolution can be performed using these generated rules.

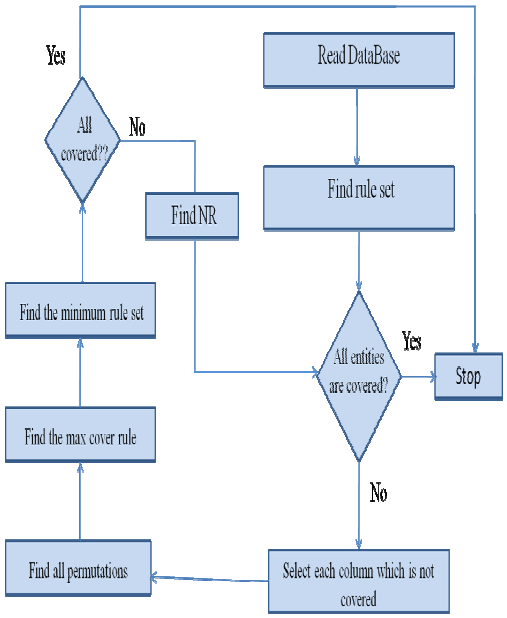

Fig. 3. Architecture of Rule Based ER

III. CONCLUSION

Entity resolution (ER) is the problem of matching records that represent the same real-world entity and then merging the matching records. Correctly merging these records and the information they represent is an essential step in producing data of sufficient quality for mining ER is a well known problem that arises in many applications. The high importance and difficulty of the entity resolution problem has triggered a huge amount of research on different variations of the problem and numerous approaches have been proposed especially for structured data. Due to the high number and diversity of different entity resolution approaches there is a strong need for comparative evaluations of different schemes. Comparing different approaches one can conclude that ER based on rules are more efficient. In rule based ER different rules are generated from the given table is used to refer the entities. This class of ER-rules is capable to describe the complex matching conditions between records and entities

REFERENCES

[1] Peter Christen, *"A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication"*, ieee transactions on knowledge and data engineering, vol. 24, no. 9, september 2012 1537

[2] Lingli Li, Jianzhong Li, and Hong Gao, *"Rule-Based Method for Entity Resolution"*, IEEE trans on knowledge and data engineering, vol. 27, no. 1, January 2015.

[3] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios, *"Duplicate Record Detection"*, IEEE January 2007.

[4] A. Chatterjee and A. Segev, *"Data Manipulation in Heterogeneous Databases"*, ACM SIGMOD Record, vol. 20, no. 4, pp. 64-68, Dec. 1991.

[5] IEEE Data Eng. Bull., S. Sarawagi, ed., *"special issue ondata cleaning"*, vol. 23, no. 4, Dec. 2000.

[6] T. Churches, P. Christen, K. Lim, and J. X. Zhu, *"Preparation of name and address data for record linkage using hidden Markov models"*, Biomed Central Medical Informatics and Decision Making, 2(9), 2002.

[7] L. Breiman, J.H. Friedman, R.A. Olshen, , and C.J Stone. *"Classification and Regression Trees"*. Wadsworth, Belmont", Ca, 1983.

[8] H.B. Newcombe, J.M. Kennedy, S. Axford, and A. James, *"Automatic Linkage of Vital Records"*, vol 130, Science, no. 3381, pp. 954-959, Oct. 1959.

[9] Breiman, L., Friedman, J.H., Olshen, R., and Stone, C.J., 1984,*"Classification and Regression Tree Wadsworth & Brooks/Cole Advanced Books & Software"*, Pacific California.

[10] S. Sarawagi and A. Bhamidipaty, *"Interactive Deduplication Using Active Learning,"* Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '02), pp. 269-278, 2002.

[11] Mikhail Bilenko and Raymond J. Mooney, *"Adaptive Duplicate Detection Using Learnable String Similarity Measures"*, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003), Washington DC, pp.39-48, August, 2003

[12] A. K. McCallum, K. Nigam, and L. Ungar, *"Efficient clustering of high-dimensional data sets with application to reference matching"*., Boston, MA, Aug. 2000.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.