Rule Based Method for Entity Resolution
Hemant Halwai\textsuperscript{1} Ajay Mahajan\textsuperscript{2} Nilesh Pawar\textsuperscript{3}
\textsuperscript{1,2,3}Department of Computer Engineering
\textsuperscript{1,2,3}AISSMS IOIT

Abstract— Entity Resolution is to distinguish the representations referring to the same real world entity in one or more databases. This project aims at developing a different approach or method for Entity Resolution than the traditional ER(Entity Resolution) methods as it is based on pairwise similarity comparison assuming that the records belonging to the same entity are much more similar to each other than the others. However this assumption does not always hold in practice resulting into incorrect ER result. We present a new class of rules which when applied to every record, we identify which entity the record refers to.

Key words: S Entity resolution, Data cleaning, Rule learning

I. INTRODUCTION
Entity resolution is one of many importation operations for data quality management, information retrieval, and data management. Entity resolution is to distinguish the representations referring to the same real-world entity in one or more databases and recognize all different real-world entities in the databases. Our project aims at developing a rule based method for ER which can be considered as complementary to the traditional ER methods based on pairwise similarity comparison thereby increasing the accuracy and overcoming the limitations of traditional ER methods. Basically by applying rules to each record, we identify which entity the record refers to.

We propose an effective and efficient rule discovery algorithm which will generate a new class of rules which could describe the complex matching conditions between records and entities.

II. PROBLEM STATEMENT

A. Problem Definition:
In many applications, a real-world entity may appear in multiple data sources so that the entity may have quite different descriptions. For example, there are several ways to represent a person’s name or a mailing address. Thus, it is necessary to identify the records referring to the same real-world entity, which is called Entity Resolution (ER). Traditional ER approaches obtain a result based on similarity comparison among records, assuming that records referring to the same entity are more similar to each other (compact set property [6]).

\[
\text{Similarity}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}
\]

However, such property may not hold so traditional ER approaches cannot identify records correctly in some cases. Hence we aim at developing a rule based method for ER which will increase the accuracy of Entity Resolution task by working as complimentary to the similarity based methods.

B. Goals and Objective:
The work of entity resolution can be broadly divided into three categories. Pairwise ER. Most works on ER focus on record matching, which involves comparing record pairs and identifying whether they match. A major part of work on record matching focuses on similarity functions. Non pairwise ER. The research on non-pairwise ER includes clustering strategies and classifiers. Most strategies solve ER based on the relationship graph among records. This paper aims at the aforementioned problems, and contributions like the syntax and semantics of the rules for ER are designed, and the independence, consistency, completeness and validity of the rules are defined and analyzed. Similarly a rule maintaining method is proposed when entity information is changed.

C. Software Context:
Entity resolution has been studied in a variety of contexts, using several approaches. We introduce three dimensions for the solution space of entity resolution, namely type of method input, objective of method and type of method. Concerning the type of input data, we consider the cases of tabular, tree and graph data. We study entity resolution approaches with respect to their particular objectives, i.e. effectiveness, efficiency and scalability, and their methods, by distinguishing them between blocking, iterative and learning ones.

Type of method input: Different types of input data impose different solutions for the problem of entity resolution. In particular, due to the high structured property of tabular data, for computing similarities between entity descriptions of this type, it suffices to compare the values of their common attributes. For the case of tree data, e.g. XML, where a hierarchical structure exists, i.e. ancestor and descendant relations, the similarity of values, used to compare two entity descriptions, is affected by the similarity of their ancestors and descendants.

Objective of method: We discern entity resolution methods with respect to their main objectives between those that aim at effectiveness, efficiency, and scalability. Algorithms focusing on effectiveness, try to find as many true matches and as few false matches as possible, while algorithms focusing on efficiency aim at resolving the given entity descriptions as fast as possible, usually by reducing the huge number of redundant comparisons. Scalable entity resolution methods are those that can cope with Big Data, typically, by parallelizing the task of entity resolution and distributing it to multiple computational resources.

Type of method: Methods that perform entity resolution are categorized as blocking, iterative, or learning. Blocking-based approaches group together, in the same block, entity descriptions that are close to each other, denoting possible matches. Iterative algorithms are based on the idea that identifying some matching entity descriptions can lead to new matches. Learning-based approaches typically use an initial set of training data, annotated as matches and non-matches, and then classify the entity descriptions in these two categories, using statistical inference.
D. **Major Constraint:**
1) Proper implementation of database.
2) Availability: The data should be available to multiple users in a distributed system.
3) Memory must be sufficient to support large datasets.

**III. ARCHITECTURE**

A. **Basic Flow of Events:**
1) Initially the admin will maintain the dataset.
2) Our algorithm will create a class of rules.
3) Input will be taken from user.
4) The given query is compared with the generated rules and suitable output is generated.

**IV. CONCLUSION**

In most ER scenarios, the logic for resolving records evolves over time, as the application itself evolves and as the expertise for comparing records improves. In this paper we have explored a fundamental question: when and how can we base a resolution on a previous result as opposed to starting from scratch? We have answered this question in two commonly-used contexts, record comparisons based on Boolean predicates and record comparisons based on distance (or similarity) functions. We identified two properties of ER algorithms, rule monotonic and context free (in addition to order independence and general incremental), that can significantly reduce runtime at evolution time. We also categorized several popular ER algorithms according to the four properties. In some cases, computing an ER result with a new rule can be much faster if certain partial results are materialized when the original ER result (with the old rule) is computed. We studied how to take advantage of such materializations, and how they could be computed efficiently by piggybacking the work on the original ER computation.

For future work, we would like to extend our techniques to more general cases. For instance, how to discover ER-rules when the operator for each attribute is not given? We would also like to consider how to incorporate human resources, such as Crowd, into our rule-discovery framework to improve the quality of rules.

**REFERENCES**


