Chapter 6
Entity Resolution on Multiple Relations

ABSTRACT

Entity resolution is a central issue in data quality management. It has been proven extremely useful in data fusion, inconsistency and inaccuracy detection, knowledge extraction, and data repairing. Nevertheless, in the real world, entities often have two or more representations in databases. The complex structures of data introduce new challenges and make entity resolution much harder than record matching on relational data. Entity resolution on complex data is to find data objects that refer to the same real-world entity and to cluster such objects together. This chapter introduces an overview of recent advances in the study of entity resolution on complex data. At first, the authors make a definition of this problem. Then the authors discuss similarity measures and reasonable algorithm to solve the problem. Finally, the authors validate the resolution by the various experiments on different kind of datasets.

INTRODUCTION

Databases play an important role in today’s IT-based economy. As studied in (Elmagarmid, Ipeirotis & Verykios, 2007), many industries and systems depend on the accuracy of database to carry out operations in. However, it’s impossible to have absolute accurate data in the real world. The information stored in databases contains a varied of different kinds of errors such as incorrectness, inconsistency, duplication, incompleteness or data being out of date. Data often lack a unique identifier which consists of linking – in relational terms, joining – two or more tables on their key fields. Thus, data quality is often compromised by many factors, including data entity errors (e.g. “Wei Wan” instead of “Wei Wang”), missing integrity constraints (e.g. “age = 234”), and multiple conventions for recording information (e.g. “W. Wei” equals to “Wei Wang”). Therefore, the quality of information stored in the database can have significant cost implications to a system that relies on information to function and conduct business. Entity resolution is a central issue in data quality management, which is one of the most popular research directions.
The data in applications is probably that with complex structure, such as XML information in enterprises, in websites and social networks, chemical and biological databases and graph data. This was further explored in (Wang & Fan, 2011). Often, there are a variety of ways of referring to the same underlying entity. To use complex data effectively in practice, necessary techniques must be in place to improve the quality of the data. Although, traditional approaches to entity resolution and deduplication use a variety of attribute similarity measures, which often based on approximate string-matching criteria. In fact, determining that two records refer to the same individual may in turn allow us to make additional inferences. The problem involves resolving multiple types of entities at the same time using the different relations that are observed among them. In this chapter, we will define the problem and propose similarity measures and some reasonable algorithms on this problem.

**PROBLEM DEFINITION**

In the relational entity resolution problem, we have some collection of references to entities and from this set of references we would like to identify the unique collection of individuals or entities to which they should be mapped. The entity resolution on multiple relations has varied classification methods. According to the result of resolution, entity identification on complex data can classify into *pairwise entity resolution* and *group-wise entity resolution*. While according to the target of the resolution, entity identification on multiple relations includes resolution on XML, resolution on graph data and resolution on complex networks. Entity resolution on multiple relations can be applied on information integration (Chapter 14) and healthcare information management (Chapter 17).

**Pairwise Entity Resolution**

- Pairwise entity identification is to recognize whether two data objects and refers to the same real-world entity. Before going through this problem, we list the meanings of some symbols we will use frequently here.
- Lower case characters e, t and r denote entities, types and references.
- Upper case character labels like e.A and r.E to denote variables, while e.a and r.e denote values of them (shot-hand for e.A = a).
- Character $\mathcal{R} = \{r_i\}$ denotes the set of references, where each reference $r$ has its attributes $r.A$.
- Character $\mathcal{T} = \{t_i\}$ denotes the set of the references’ possible types.
- Character $\mathcal{E} = \{e_i\}$ denotes the set of entities the references correspond to, so that each reference $r$ has its entity $r.E$.
- Character $\mathcal{H} = \{h_i\}$ denotes the set of hyper-edges to describe co-occurrence of references, which was further explored in (Bhattacharya & Getoor, 2005; Bhattacharya & Getoor, 2007).

We should note that each $r$ in $\mathcal{R}$ has an observable type and $r.e.T = r.T$, which means that a reference $r$ and its corresponding entity $r.E$ are necessarily of the same type. And it is true that the entities and their attributes are often not observable. What we observe are the attributes $r.A$ of individual references, which can be imagined being generated by some distortion process from the attributes of the corresponding entity $r.E$. Our target is to recover the set of underlying entities $\mathcal{E}$ and figure out the unobservable attributes and types by the observed attributes and types from given references.