

Data Mining in the Context of Entity Resolution

Sofus A. Macskassy
Fetch Technologies, Inc
841 Apollo Street
El Segundo, CA 90245

sofmac@fetch.com

Evan S. Gamble
Fetch Technologies, Inc
841 Apollo Street
El Segundo, CA 90245

egamble@fetch.com

ABSTRACT

We have encountered several practical issues in performing data mining on a database that has been normalized using entity resolution. We describe here four specific lessons learned in such mining and the meta-level lesson learned through dealing with these issues. The four specific lessons we describe deal with handling correlated values, getting canonical records, getting authoritative records and ensuring that relations are properly stored. The perhaps most important lesson learned is that one ought to know the kind of data mining is to be done on the data before designing the schema of the normalized database such that data specific to the mining is derivable from the database.

1. INTRODUCTION

This paper is about the practical issues we have encountered when applying data mining to a database that has been normalized using entity resolution—the process of integrating multiple pieces of data that use different formats and terminology to refer to the same entity. These kinds of databases are becoming increasingly relevant in today’s world, where there is an abundance of information about entities from multiple information sources. In addition, this information is rapidly increasing for many entities such as companies, high-profile people, etc. In order to have a coherent view of the world and in order to be able to mine this data, we need to normalize the data into some standard format as well as know when multiple records refer to the same entity. However, many details and gotchas abound when we need to do entity-centric mining of the data. We here describe our experiences and lessons learned from integrating data mining with our entity resolution technology. These issues are described in the context of a

database that has already been normalized, but the issues are equally applicable when entity resolution is performed “on the fly.”

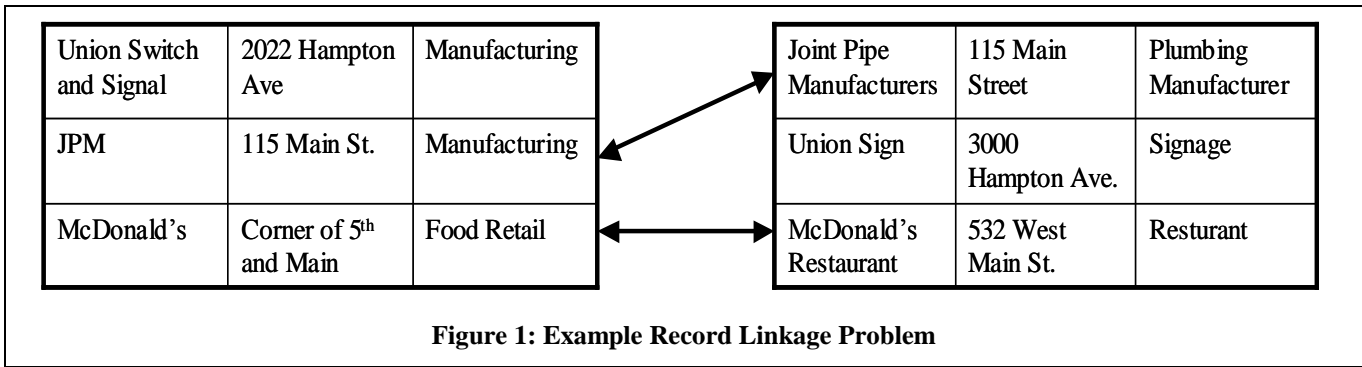
There are multiple ways to deal with record linkage and each of these have issues. In the naïve case, the goal is to merge multiple databases and end up with one consolidated record per entity. This case happens, for example, when one wants to merge multiple contact lists. A more general case, which is the focus of this paper, is when multiple records are about the same entity but cannot be consolidated into a single record. For example, when a person publishes multiple articles or when a company is mentioned many times in the news or when a football team plays multiple matches. In all of these cases, we need multiple records per entity. We need record linkage technology because the entity is likely not referred to by the exact same name across all records. At the end, we end up with a database of many records for each entity. In this paper we consider the particular case where we have a database of only one type of entity, such as a person who writes academic papers or news articles or blogs. Dealing with such a database from a data mining perspective can pose many problems, the resolutions of which are not immediately clear but are nonetheless crucial to ensure the efficacy of the data mining techniques. We consider four particular issues in this paper:

- 1) We may need to mine characteristics of entities that are conditioned on attributes that change over time. Entity resolution preserves multiple attribute values but not pair-wise correlations.
- 2) It is not always clear whether it is possible to get a canonical record for each entity. This is needed, for example, when one wants to mine for certain demographic patterns.
- 3) We have to be very careful when dealing with multiple records that may be about the same content (such as the same academic paper or blog entry or news story) because this can have serious effects on the counting we do during basic data mining operations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DBMA '08, August 24, 2008, Las Vegas, Nevada, USA.

Copyright 2008 ACM 1-59593-439-1...\$5.00.



4) Many records contain relations between entities, such as co-authors on a paper or article, or the CEO of a company, or a journalist covering certain companies or industries. It is critical, and not always trivial, to ensure that the normalized database keeps such relational information if one would like to perform relational data mining.

The rest of this paper is outlined as follows: Section 2 describes high-level background information about our entity resolution platform, followed in Section 3 with a description of our data mining integration and a discussion of the four issues we faced. We end in Section 4 with a discussion of lessons learned and final thoughts.

2. BACKGROUND: ENTITY RESOLUTION

The process of integrating multiple pieces of data that use different formats and terminology to refer to the same entity is commonly called record linkage (or alternatively, entity resolution). This a pervasive issue when integrating multiple information sources. To illustrate the issues involved, **Figure 1** below shows two tables that list businesses, each table containing three attributes: name, address, and business type. Note that JPM and Joint Pipe Manufacturers are the same business, while Union Switch and Signal is not the same as Union Sign. We can make this judgment because JPM is clearly an acronym for Joint Pipe Manufacturers, and the two records list minor variations of the same address and business type. In contrast, Union Switch and Signal and Union Sign share only textual similarities in their name and address. Unfortunately, it is not easy to program a computer to make these types of common-sense determinations.

Fetch Technologies and the University of Southern California Information Sciences Institute have jointly developed a new record linkage approach based on statistical machine learning technology (Minton et al., 2005). To enable this technology to be used seamlessly for information aggregation purposes, we have been developing the EntityBase™ architecture (Knoblock et al., 2007). An

EntityBase is a “virtual database” which incorporates record linkage as a fundamental operation. The rest of this section provides a high-level overview of our architecture.

The work on EntityBases has focused on building the algorithms required to populate, organize, and query massive EntityBases, as described below:

- **Populating EntityBases:** Using Fetch Information Extraction technology (Fetch Agents), we aggregate data to create an EntityBase from a potentially large set of information sources, including web sites, databases, and legacy systems. Only core identifying attributes of the entities need to be collected and maintained locally. The majority of information can be maintained in the original remote sources, so that the overall system functions as a distributed, virtual database.
- **Organizing EntityBases:** The capability to consolidate, or link, multiple references to the same individual entity collected from different information sources is a central aspect of the EntityBase architecture. The architecture supports the statistical record linkage process “invisibly” as an EntityBase is populated. The system is robust to updates, enabling references to be both automatically consolidated and automatically unconsolidated as new information becomes available.
- **Querying EntityBases:** In previous research, we and others (Duschka 1997; Knoblock, Minton et al. 2001) have addressed many of the theoretical problems underlying virtual databases (i.e. mediator systems that integrate distributed, heterogeneous sources). Nevertheless, building large-scale virtual databases remains challenging in practice because it is difficult to model complex data relationships and potentially expensive to execute arbitrary queries against virtual databases. Our work addresses these specific problems. By focusing only on entities, our architecture simplifies the modeling issues and improves the tractability of query processing.

Figure 2 shows the primary components of our EntityBase architecture. An information gathering layer consists of software agents (built and executed using the Fetch Agent

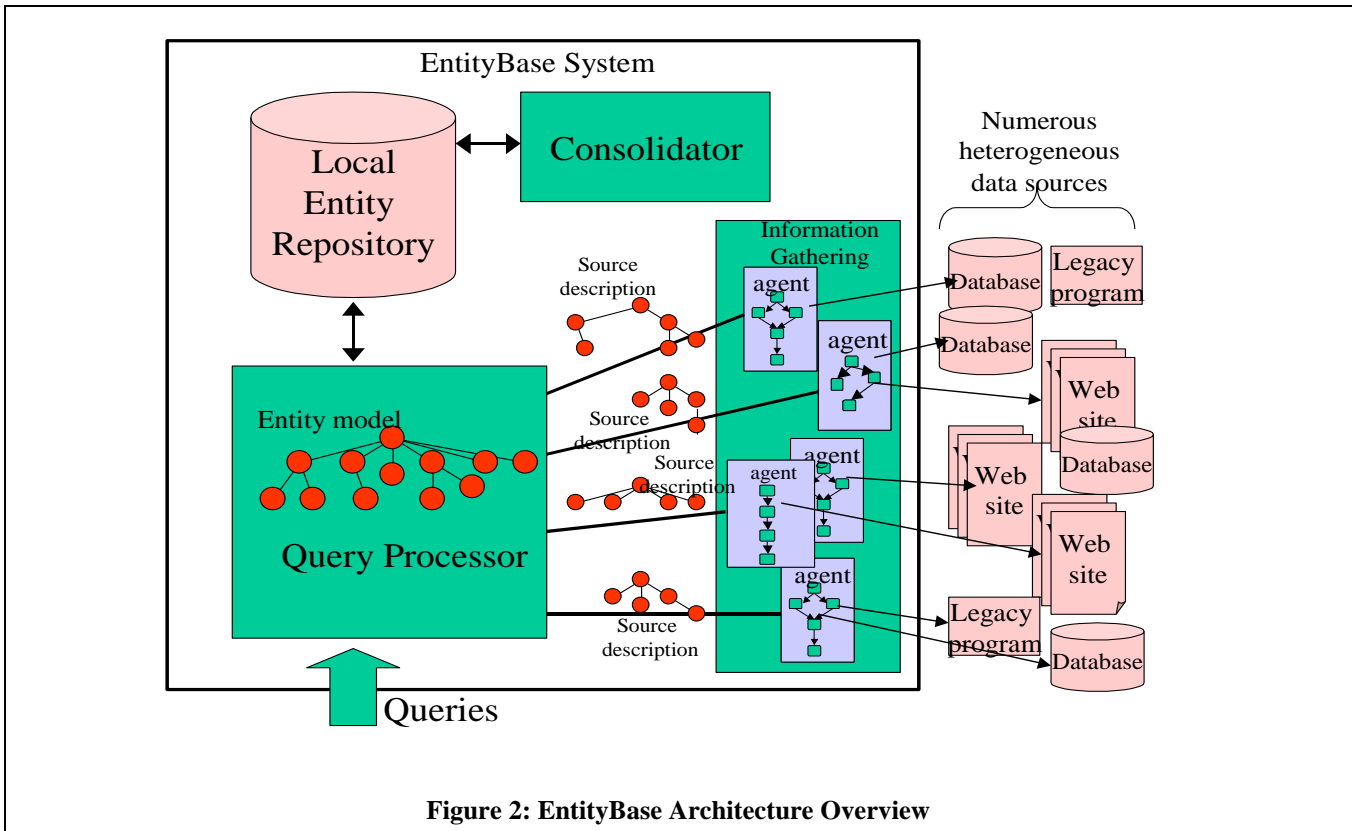


Figure 2: EntityBase Architecture Overview

Platform, our commercial product) that can collect data from heterogeneous sources, including document streams, remote databases, and web sites. A query processor accepts information requests from users and other applications. The query processor in turn can issue standing queries to the agents so that they harvest data on a scheduled basis. It can also task agents to dynamically collect data on demand.

Information about the entities that populate the EntityBase is stored in a local entity repository. A key component is the consolidator, which manages the organization of the local repository, integrating data retrieved from multiple databases into distinct entities. The repository contains only critical data that needs to be stored locally for efficiency, security and maintenance purposes. This includes key attributes required to identify and link the individual entities in the EntityBase. Auxiliary information about the entities is retrieved via the agents as needed, reducing local storage requirements and ensuring that the data retrieved is up-to-date. The consolidator uses statistical inference techniques to determine when to merge multiple entities (e.g., when it is determined that “Steven Minton, CTO of Fetch Technologies”, is the same person as “Professor Steven N. Minton, at USC/ISI”). The consolidator thus automatically manages the organization of the EntityBase.

Queries to the system are described in terms of an entity model – a “schema” describing the classes of data that

compose the EntityBase. For each agent there is a source description specifying how the data retrieved by the agent maps into the entity model. The query processor relies on these source descriptions to determine which agent(s) can retrieve the data necessary to answer the query

3. DATA MINING WITH ENTITYBASES

We have been creating EntityBases in various domains, including academics and journalists, and are currently working on creating an EntityBase in the aerospace domain. In the academic domain, we have a large amount of online data from sources such as DBLP¹, CiteSeer² and many online journal web-sites. We acquired historical data from DBLP and CiteSeer and have been crawling many journal web-sites to create a large database of academic publications. In the journalist domain, we have created a large EntityBase of journalists for a commercial customer, where we have information on a per-article basis, where each article record contains information such as what industries or companies are mentioned in the article. In the aerospace domain, we acquired data about companies in the

¹ <http://www.informatik.uni-trier.de/~ley/db/>

² <http://citeseer.ist.psu.edu/>

aerospace domain, the planes they own and the flights these planes have taken since late 2007. We have recently begun data mining these EntityBases, and have in the process hit various barriers. In particular, we will here discuss four such barriers, each time outlining the issues with the EntityBase representation and, if applicable, how we addressed the issue.

3.1 Correlated Attributes

When mining across entities, we may need to find patterns in entity characteristics that are conditioned on various attributes of the entities. For example, in the aerospace domain, we need to mine for characteristics of planes and their flight behavior (such as how fast they were generally flying as well as the variance of these observed speeds).³ In particular, we need to condition the observables on attributes of the planes themselves. However, it turns out that some attributes of a plane change over time, such as the kind of accessories it carries on a particular flight. For example, sometimes some of the planes would have accessories for RVSM (Reduced Vertical Separation Minimum) certification, and sometimes they would not.⁴ It may be that this accessory has some bearing on the speed and we would like to take that into account. A changing attribute will have multiple values in the consolidated entity, but EntityBases do not maintain pair-wise correlations between the multiple values of different attributes.

We have considered three ways in which this can be dealt with: (1) ignore the changing attribute, (2) define a heuristic to choose a 'default' value or (3) consider the entities to be different for different values of the attribute. In the aerospace domain we decided to ignore the attribute as this is the simplest and because our data mining in this domain is still early stages. However, as this data matures and as our data mining capabilities matures, this is an issue that is very likely to appear again. The problems with the second and third approaches are that domain experts are needed to give insight into the correct approach to take for each possible attribute and the data mining techniques that we would like to apply to it. In other words, expensive resources need to be brought in to better understand how different choices might affect the results. Often, the experts themselves may not know the correct answer, which makes this particular issue all the more difficult.

³ This issue is still theoretical as we have not yet tested our data mining capabilities with the aerospace EntityBase, as the schema of this EntityBase is still being designed. However, the issue discussed here is still very real and needs to be addressed in this schema.

⁴ This certification needs two independent altimetry systems, an autopilot capable of holding altitude within 65 feet, an altitude alerting system and a mode C transponder.

We raise the issue here in the hopes that this will open a discussion on how to deal with correlated attributes in entities.

3.2 Canonical Records

The process of entity resolution links several records to a single entity, but does not create a canonical record with a single canonical value for each attribute. When mining for patterns in an EntityBase, we often need a single representative value for an attribute, but it is not clear how to choose a representative among the values, or whether to combine the values in a normalized form. For example, a name attribute may have three values from different sources: "S. Minton," "Minton, S.," and "Steven Menton." The canonical value might be "S. Minton" or "Steven Minton."

One way to choose among the values is to use the same scoring function used for record linkage to see which attribute value or which source record has the highest scores in relation to the others. If there is some temporal aspect to the data sources the most recent records may be chosen as canonical, at least for some attributes. Alternatively, a normalized form may be more representative of all the values.

3.3 Authoritative Records

In basic data mining operations on an EntityBase, we often need to count records associated with a given entity. Because the EntityBase may have been constructed from source databases that have been revised or augmented over time, there may be multiple records that refer to the same real-world object, only one of which is considered authoritative. For example, in the journalist domain, we need to count distinct articles associated with a given journalist, such as how many of the journalist's articles mention a given industry or company. Multiple records may refer to the same article, because revisions to the article may have been added as new records rather than alterations of the existing record.

We have learned that to the extent it is possible to identify multiple records with the same real-world object, this identification should be preserved during the EntityBase creation. This sort of auxiliary information is not critical to the primary purpose of an EntityBase, which is very fast consolidation of massive amounts of data, and hence may be a burden to the highly optimized Local Entity Repository, but it is critical to basic data mining on the entities, and therefore should be made available in some way, perhaps via agents to the original source databases.

3.4 Entity Relations

An EntityBase does not directly represent relations between entities, such as co-authors on a paper or article, or the CEO of a company, or a journalist covering certain companies or industries. Relations between entities may be indirectly represented as attributes of entities that refer to some identifying characteristic of other entities, but the record linkage algorithms that perform entity resolution do not yet recognize such attributes as inter-entity relations.

Given that current EntityBases do not directly represent relations, if one would like to perform relational data mining on an EntityBase, for example discovering patterns in a co-author graph, relations between entities must be derivable in some way from entity attributes. This is only trivial in the case where an attribute's values uniquely identify records that have been consolidated into an existing entity. Often, though, the attribute values only weakly identify an entity, for example by a name in various formats, in which case an additional record linkage process must be applied to the already-consolidated database to derive the inter-entity relations.

4. CONCLUSIONS

We have learned several lessons from mining EntityBases, i.e. databases that have been normalized by entity resolution. Primarily we learned that great care must be taken in the EntityBase schema design, so that information necessary for the needed mining operations is either preserved in the EntityBase or derivable as auxiliary information. This implies that to a large extent the mining operations must be known in advance of construction of the EntityBase.

5. ACKNOWLEDGMENTS

This work was sponsored in part by the Office of Naval Research under award number N00014-07-C-0923. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of Naval Research or the U.S. Government.

6. REFERENCES

- [1] Duschka, O. M. and M. R. Genesereth (1997). Answering Recursive Queries Using Views. Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Tucson, Arizona.
- [2] Knoblock, C. A., J. L. Ambite, K. Ganesan, S. Minton, G. Barish, E. Gamble, C. Nanjo, K. See, C. Shahabi and C.-C. Chen (2007). EntityBases: Compiling, Organizing and Querying Massive Entity Repositories. International Conference on Artificial Intelligence, Las Vegas, NV.
- [3] Knoblock, C. A., S. Minton, J. L. Ambite, N. Ashish, I. Muslea, A. G. Philpot and S. Tejada (2001). "The Ariadne approach to web-based information integration." International Journal of Cooperative Information Systems (IJCIS) Special Issue on Intelligent Information Agents: Theory and Applications 10(1-2): 145-169.
- [4] Minton S., Nanjo C., Knoblock C., Michalowski M., and Michelson M. (2005) "A Heterogeneous Field Matching Method for Record Linkage" The Fifth IEEE International Conference on Data Mining.