Technical Perspective: Toward Building Entity Matching Management Systems

Wang-Chiew Tan
Recruit Institute of Technology
wangchiew@recruit.ai

Entity matching, also known as *entity resolution* or *reference reconciliation*, is to identify when two (different) representations refer to the same real-world entity. Overcoming the entity matching problem is often a key step in today's data preparation and integration pipeline before useful data can be produced for analysis. For example, to understand how many potential new customers there may be, a company may wish to integrate an internal repository of customer profiles to an externally sourced dataset that contains profiles of users (e.g., Twitter data). A successful entity matching process would need to discern when two heterogeneous customer profiles may actually refer to the same customer and also for the opposite, when two seemingly identical customer profiles may actually not be the same customer. For example, it is not obvious whether or not the these two records:

(`D. Smith`, `IBM Yorktown`, ...)
(`S., David`, `International Business Machine`, ...)

refer to the same person and one may need to understand the remaining values in the customer profiles before a final decision can be made. An entity matching outcome is largely dependent on the features that are selected by the user or learned (if training data is available) for determining whether a match is successful. Different features and measures may lead to different outcomes for the two records above. Similarly, a different training data used for training the entity matching model may lead to a different outcome. On top of this, for an entity matching workflow to be useful, more often than not, it has to scale to large datasets.

Magellan is a fairly recent entity matching system developed at the University of Wisconsin that overcomes several limitations of existing solutions to entity matching. There are two important attributes of Magellan that make it particularly useful and "easy" for end users to develop entity matching solutions and this paper describes how Magellan has been successfully used by several such end-user groups.

First, Magellan has a rich set of libraries for users to carry out the entire entity matching pipeline which may involve several substeps such as data cleaning, visualization, in addition to blocking, and matching. For example, Magellan provides libraries for different types of string matching functions (a basic building block in entity matching). And because Magellan is developed in Python, it can also leverage publicly available Python libaries (more than 130,000 libraries are available for data science, see pypi.python.org) for other tasks such as exploration, visualization, and cleaning.

Second, Magellan provides how-to guides that describe how to approach the development of entity matching workflows. The how-to guides are useful because they describe step-by-step instructions with examples that illustrate the example methodology and functionalities available in Magellan. In addition, they illustrate critical substeps that are sometimes overlooked by users in designing an entity matching workflow. For example, during the design of an entity matching workflow for large datasets, one often *downsamples* the dataset so that the resulting dataset is smaller and allows for faster testing. However, care has to be taken to ensure that the downsampled dataset is representative of the original dataset and Magellan provides supporting tools to help with the downsampling process.

The user works in the *development stage* with the downsampled dataset. When ready, the user moves the final entity matching workflow to the *production stage* where it will be executed on the original dataset with supporting software libraries for scaling the operation such as running MapReduce/Spark jobs in a parallel and distributed setting.

The paper largely describes the development stage by delineating a few main steps in the development of entity matching workflows that uses supervised learning. Magellan provides a tool for downsampling which samples data intelligently to ensure a reasonable number of matches exists in the downsampled dataset. After this, the downsampled data is *blocked* to remove tuples that are highly unlikely to match. Blocking helps further reduce the number of candidate matches to consider and can considerably speed up the overall entity matching process. Magellan provides a debugger tool for users to examine the tuples that are eliminated by the blocker. A general rule of thumb is that if there are only a few matches among the eliminated tuples, then the blocker has achieved a sufficiently high reall. From the remaining tuples, the next step samples a set of candidate matching tuples and the user labels the candidates as match/no-match. Magellan provides tools to ensure that there are sufficiently many true matches in the sampled data. After this, Magellan automatically generates a set of features from the labeled data and converts each candidate pair of tuples into a feature vector. By training and cross validation, it selects a matcher with the the highest estimated accuracy from among those supplied by Magellan. A debugging step then follows to examine the mistakes of the matcher and improve upon it as needed and the process can be repeated.