EMSS: Entity Matching in a Semi-Supervised way

Jia R. Wu
University of Waterloo

Shaokai Wang
University of Waterloo

Abstract

1 Introduction

Entity matching (EM), also known as Entity Resolution (ER), in the world of data management refers to resolving duplicate entities to a single entity. Matching may be done in a probabilistic or a deterministic (rule-based) manner. According to our survey, many end-to-end machine learning based methods have been applied to this problem such as Magellan and Dedupe.io.

However, traditional supervised learning method in entity matching has a problem: lacking of labels on training dataset. Usually there are two ways to solve the problem: training on other labeled dataset, or labeling the training dataset manually. The first one may not able to train a good model if the two training dataset is not similar. The second one needs a lot of human labor. To find a good solution to this problem, we propose to use active learning, a semi-supervised learning method, to do the labeling.

Active learning in the context of machine learning describes the process of a user actively providing labels for model training. This paradigm of active human labeling is referred to as human-in-the-loop machine learning, which is, the model select the most informative sample to be labeled by human and train on these sample, repeatedly. We found a method called Dedupe.io has applied active learning in entity matching, but they are also facing a problem of low recall.

In this paper, we first provide a comparison of Entity Matching across popular packages such as Magellan and Dedupe.io [2, 3]. Comparisons are conducted with two datasets BUY and RESTAURANT (add names for these later). Additionally, we provide a fully reproducible and containerized environment for execution requiring no external dependencies other than Docker [5]. Finally, we extend Magellan with semi-supervised learning and demonstrate that comparable accuracy for EM can be achieved with less samples.

We make the following contributions:

- An evaluation of EM across common systems
- A framework for reproducible execution
- An active-learning extension for EM (semi-supervised)

2 Problem definition

In this work, we focus on structured entity matching when resolving two csv tables. Given two csv tables with same structure, our aim is to combine them into a single table. During the combination, our method can can detect the same entity that appears in both two csv tables and delete one of them, which is also called deduplication. After the combination, it will return a table that has information of two tables and has no duplicate entities.

3 Related Work

There has been much work done throughout the years to address entity matching. Efforts ranging from traditional machine learning to simple rule based entity matching have been investigated [4]. Optimizations to reduce the amount of user effort with respect to a time cleaning budget have been explored [1].

3.1 Magellan

Magellan is described as an end-to-end EM framework as it provides the user with a toolset to conduct EM from the beginning of the workflow through the end. These tools include user guides, strategies for blocking, sampling and matching. The following is a description of the Magellan workflow.

Initially, Magellan computes the cartesian product of both tables. From this resultant set of ordered pairs, an overlap blocker is applied for a resulting set C. The blocking technique used within Magellan is an overlap blocker. Attributes are tokenized into q-grams and blocked together if they share 1 minimum overlap bewteen q-grams. Subsequently, a sample denoted by S of 450 tuples is taken from this ordered set C. Users are expected to add labels to these, denoting if they are identical entities or not.

I and J are computed from this labeled set S.

I is training set. J is testing set.

Feature vectors are then extracted from tables I and J. An example of a feature in this instance would be a function that maps a tuple pair into a value. If the attributes are textual, a feature could be a 3-gram jaccard score. Sample set S is thus split into training and testings sets, and six various machine learning strategies are applied to match values. The six included are Decision Tree, Support Vector Machine, Random Forest, Logistic Regression, Linear Regression and Naive Bayes. Each matcher is evaluated on the test set J and the test set I for precision and recall.

3.2 Dedupe.io

Dedupe.io is available as both an open source and as a paid service for EM. We will focus on the open source implementation implementation of dedupe in this paper. The user workflow of Dedupe.io is very similar to Magellan. Attributes are defined for blocking purposes, an active labeling system is conducted, a learnable matching function is trained, and a resultant clustering component will produce a dataset with the most probable duplicates clustered together.

4 Methods

4.1 Docker

Docker is a virtual-machine like system where each container (vm) offers a consistent and reproducible execution environment. Docker-compose is a standard for defining different Docker containers. By offering our code in a Git repository paired with a Docker configuration for each tool, each project can be pulled into a host machine and run + evaluated with less than 2 lines of code. Additionally, the dependencies for each tool are different, with Magellan and Dedupe.io requiring differing numpy versions in addition to other differences. Thus, with our docker-compose setup we are able to evaluate all our tools simultaneously on a single host machine.

4.2 Active Learning

Our method applied active learning on Magellan: In Magellan, the user

5 Results

Have a section on why the DBLP-ACM benchmark was used.

Describe the performance of using ActiveLearning in Magellan.

Add Key Magellan Result/Conclusion Here

6 Challenges

Describe the ease of use of each system here. Describe the potential performance of each system here.

Both Magellan and Dedupe.io had documentation however Dedupe.io was documented towards developers and Magellan was documented for researchers. Thus it was not clear what blocking function was used for the RecordLink class. From manual code inspection it turns out that the base function is similar

7 Threats To Validity

8 Future Work

9 Acknowledgments

10 Availability

All relevant scripts and data can be retireved from the following repository:

`https://github.com/JRWu/cs848w20`

Notes

[1]The predefined settings can be located here: https://github.com/filebench/filebench/wiki/Predefined-personalities

References

[1] AO, J., AND CHIRKOVA, R. Effective and efficient data cleaning for entity matching. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (New York, NY, USA, 2019), HILDA19, Association for Computing Machinery.

[2] BILENKO, M. Learnable similarity functions and their applications to clustering and record linkage.

[3] KONDA, P., DAS, S., SUGANTHAN GC, P., DOAN, A., ARDALAN, A., BALLARD, J. R., LI, H., PANAHI, F., ZHANG, H., NAUGHTON, J., ET AL. Magellan: Toward building entity matching management systems. *Proceedings of the VLDB Endowment 9*, 12 (2016), 1197–1208.

[4] KÖPCKE, H., AND RAHM, E. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering 69*, 2 (2010), 197–210.

[5] MERKEL, D. Docker: Lightweight linux containers for consistent development and deployment. *Linux J. 2014*, 239 (Mar. 2014).