EMSS: Entity Matching in a Semi-Supervised way

Shaokai Wang
University of Waterloo

Jia R. Wu
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, Data Tamer, JedAI to name a few. [7, 13, 11].

However, traditional supervised learning methods in entity matching have a problem. They generally lack labels for training. Usually there are two ways to solve the problem: training on other labeled datasets, or labeling the training dataset manually. The first approach may fail to train a representative generalized model if the two training datasets are not similar. The second one requires a non-trivial amount 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 [12].

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 selects the most informative sample to be labeled by the human and training is conducted with these samples. We found a method called Dedupe.io which has applied active learning in entity matching, but we found that they have low recall.

In our paper, we first provide a comparison of Entity Matching across popular packages such as Magellan and Dedupe.io [2, 7]. Comparisons are conducted with two datasets ABT_BUY and RESTAURANTS. Additionally, we provide a fully reproducible and containerized environment for execution requiring no external dependencies other than Docker [9]. Finally, we propose EMSS(Entity Matching in a Semi-Supervised way) by extending 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 structured csv tables, our aim is to combine them into a single table. During the combination, our method 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 [8, 6, 5]. Optimizations to reduce the amount of user effort in labeling with respect to a time cleaning budget have been explored [1]. Other works have attempted to distribute and parallelize the blocking step [3].

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 between 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 the training set. J is the 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 of dedupe in this paper. The user workflow of Dedupe.io is very similar to Magellan: attributes are defined for blocking purposes and a learnable matching function is trained. However, Dedupe.io utilizes an active learning system to obtain the label: it first uses a distance matrix to find most probable same entity pairs to be labeled and then finds the most informative pairs to be labeled. Based on these labeled data, a resultant clustering component will produce a dataset with the most probable duplicates clustered together.

According to our survey, Dedupe.io is the only EM method that applies active learning. We found that the precision of Dedupe.io is usually very high but the recall is very low when there is not sufficiently labeled data. To our understanding, the reason is that Dedupe.io uses clustering to find the same entity rather than using a classifier to judge whether it is the same entity. This will make Dedupe.io consider less pairs.

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 few 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

Active learning is a semi-supervised machine learning framework that can interactively obtain labels for training dataset. The reason we need active learning is that there are always available datasets but the labels for them are not always available. In entity matching problems, usually it is very hard to have large labeled datasets to train a machine learning model. With the same limited labeled data, if one data contains more information, it is more likely to train a stronger model. Hence, we propose Entity Matching in a Semi-Supervised way, or EMSS for brevity. We will apply active learning to improve Magellan.

4.3 EMSS

EMSS divides the labeling process into several rounds. For each round, it asks a human a batch of pairs to be the same or not. At the first round, it randomly selects pairs from the whole dataset. For the next several rounds, EMSS will first use the trained model to test the whole dataset. Based on the result, there are three ways of sampling: random sampling(RS), less confident(LC) and breaking ties(BT). Random sampling is like what we do at first round: randomly select a batch of pairs. Less confident is the selection of samples which are less confident, that is, the probability value of any class is very low. Breaking ties is the boundary region selection between two classes, that is, the probability value of most probable two classes are very close.

5 Results

5.1 Datasets

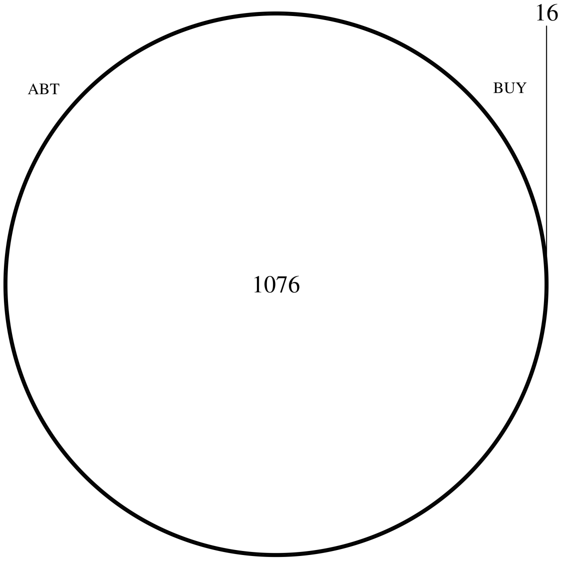

Figure 1: ABT_BUY Entity Overlap

The **ABT_BUY** dataset represents two records kept of electronics. There are 4 attributes in this dataset, [**id, name, description, price**]. This is a dataset referring to electronics and their prices. Attributes name and description are highly variable and price may be absent in tuples between the two tables. As referenced by Figure 1, ABT_BUY primarily consists of overlapping entities. There are a total of 1076 overlapping entities between ABT and BUY with 16 entities in BUY that are not present in ABT.

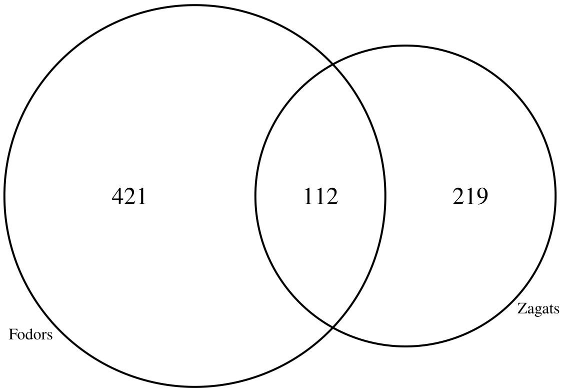

Figure 2: Restaurants Entity Overlap

The **RESTAURANTS** dataset represents restaurants listed by two different review platforms. There are 6 attributes in this dataset, [**id, name, addr, city, phone, type**]. All attributes in both CSVs are present and differences are not too variable between the two. As highlighted by Figure 4, there are only 112 common entities between the two, with 421 unique entities in Fodors and 219 unique entities in Zagats.

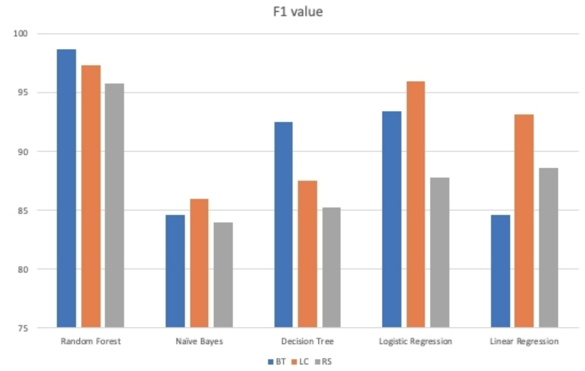

Figure 3: F1 Scores of EMSS on RESTAURANTS dataset

This figure is a description of the F1 scores on the RESTAURANTS dataset. The batch size was 4 and the number of active learning rounds was 5.

EMSS boosts the F1 score on all classifiers except for Linear Regression.

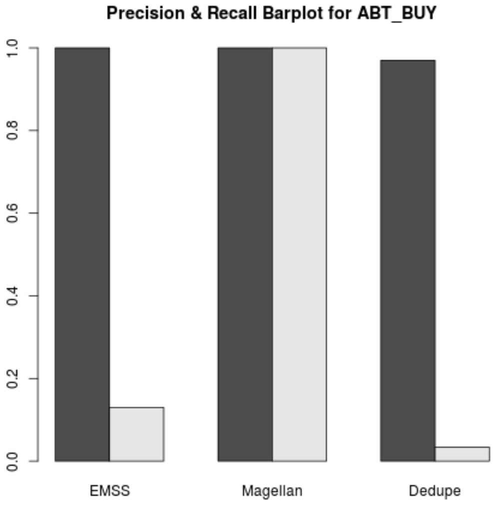

Figure 4: Precision vs Recall Barplot. Precision is on Left, Recall is on Right.

Figure 4 depicts the Precision vs Recall graph for EMSS, Magellan and Dedupe.io on the ABT_BUY dataset. Base Magellan has some strange performance where it lists both Precision and Recall as 1.

EMSS outperforms Dedupe.io on the ABT_BUY dataset when a low number of training samples are used. (n=20)

6 Challenges

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. Despite having claimed that their documentation is very good, we had to consult the libraries they used such as scikit-learn and pandas to understand what functions in Magellan did.

7 Threats To Validity

As we only used two different datasets, we are unable to say that active learning works well across multiple different domains. Additionally, the entities within each dataset are known to be mostly duplicates, thus we cannot say with confidence that our method can resolve datasets where there are few duplicates better than standard Magellan or Dedupe.io. Future work would be to vary the proportion of duplicates across pairs of datasets that need to be resolved.

We did not experiment with aggregation of multiple different blocking functions. A future direction of work would be to conduct an empirical evaluation of permutations of different blocking functions and investigate whether or not our observations change.

The entity matcher in Magellan is the best performing matcher from one of the following: Decision Tree, Support Vector Machine, Random Forest, Logistic Regression, Linear Regression and Naive Bayes. Despite how optimized the training data selection becomes for each of these methods, the downstream matching of entities will be limited by the same drawbacks of each of the aforementioned machine learning methods. In structured data, the assumptions of these methods generally hold, however there are other methods that work better on unstructured data such as DeepER and others [4, 10].

8 Availability

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

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

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] Chu, X., Ilyas, I. F., and Koutris, P. Distributed data deduplication. *Proceedings of the VLDB Endowment 9*, 11 (2016), 864–875.

[4] Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., and Tang, N. Deeper–deep entity resolution. *arXiv preprint arXiv:1710.00597* (2017).

[5] Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering 19*, 1 (2007), 1–16.

[6] Fan, W., Ma, S., Tang, N., and Yu, W. Interaction between record matching and data repairing. *J. Data and Information Quality 4*, 4 (May 2014).

[7] 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.

[8] Köpcke, H., and Rahm, E. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering 69*, 2 (2010), 197–210.

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

[10] Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., and Raghavendra, V. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data* (New York, NY, USA, 2018), SIGMOD 18, Association for Computing Machinery, p. 1934.

[11] Papadakis, G., Tsekouras, L., Thanos, E., Giannakopoulos, G., Palpanas, T., and Koubarakis, M. The return of jedai: End-to-end entity resolution for structured and semi-structured data. *Proc. VLDB Endow. 11*, 12 (Aug. 2018), 19501953.

[12] Settles, B. Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[13] Stonebraker, M., Bruckner, D., Ilyas, I. F., Beskales, G., Cherniack, M., Zdonik, S. B., Pagan, A., and Xu, S. Data curation at scale: the data tamer system.