EXPLAINER: Entity Resolution Explanations

Amr Ebaid*, Saravanan Thirumuruganathan†, Ahmed Elmagarmid†, Mourad Ouzzani† and Walid G. Aref*
*Purdue University †Qatar Computing Research Institute, HBKU
{aebaid, aref}@cs.purdue.edu, sthirumuruganathan@acm.org, {aelmagarmid, mouzzani}@hbku.edu.qa

***Abstract*—Entity Resolution is a fundamental data cleaning and integration problem that has received considerable attention in the past few decades. While rule-based methods have been used in many practical scenarios and are often easy to understand, machine-learning-based methods provide the best accuracy. However, the state-of-the-art classifiers are very opaque. There has been some work towards understanding and debugging the early stages of the entity resolution pipeline, *e.g.,* blocking and generating features (similarity scores). However, there are no such efforts for explaining the model or its predictions. In this demo, we propose EXPLAINER, a tool to understand and explain entity resolution classifiers with different granularity levels of explanations. Using several benchmark datasets, we will demonstrate how EXPLAINER can handle different scenarios for a variety of classifiers.**

I. INTRODUCTION

Entity Resolution (ER, for short), *a.k.a.* Record Linkage, Entity Matching, or Duplicate Detection, identifies pairs of data instances that refer to the same real-world entity. ER has been the subject of many investigations in both industry and academia in the past few decades [1], [2]. Several recent studies [3]–[5] show that machine learning (ML)-based methods often provide state-of-the-art results for ER.

A key impediment to using these ML-based solutions in practice is that end-users are given the output (*i.e.,* the matching tuples) without sufficient explanation of why these tuples are matching. This state of affairs may hinder the use of these ML-based solutions even if they deliver the best results. With ML affecting life-altering actions nowadays – *e.g.,* loan approval, job hiring, and medical diagnosis – comes the critical need and motivation for explainability. Explanations are necessary to build trust in the decision process, and increase the adoption of (semi-)automated systems. They allow for informed human involvement and obtaining user feedback. They also help experts and developers debug errors, compare approaches, and improve functionality. Besides, this transparency is now required by new laws and regulations to justify how these decisions are made.

A recent study of explainability in data integration systems [6] concurs that there have been several approaches towards *explaining* systems (*i.e.,* explicit causal explanations) for tasks like schema matching, schema mapping, and data fusion. However, none of the current ER systems explicitly explain their results.

In this demo, we present EXPLAINER, a system that takes two input datasets to be deduplicated along with an ML model that is trained for such task, and in turn helps users understand the outcome of the ML model from various angles.

For this purpose, we adapt general-purpose explanation tools into the context of ER. While these tools provide useful instance-level or model-level explanations, those are not sufficient in the context of ER, and new techniques are needed to enable new types of ER-specific analyses. In EXPLAINER, we build upon them and extend their functionalities to provide more profound explanations and deeper analyses of their collective outcomes.

More specifically, EXPLAINER provides the following new functionalities for explaining ML-based ER:

- **Global Explanations:** A typical user could be overwhelmed by individual explanations. Hence, we post-process these explanations to help explain the overall ML model and how different features drive its predictions. Furthermore, we also derive feature importance and visualize predictions (explanations) against features values (contributions).
- **Representative Tuple Pairs.** Typically, the user is not interested in manually inspecting the explanations of all tuple pairs in order to validate the model. It is desirable to identify a small set of representatives that provide a meaningful and diverse perspective of the ML model.
- **Model Analysis:** We provide a mechanism to analyze where the model works well (true positives and true negatives), and where it does not (false positives and false negatives).
- **Differential Analysis:** One can obtain meaningful insights on how multiple ML models fare on ER task by focusing on where they disagree. This can be achieved by mining the explanations provided by each of the models for these tuple pairs.

Several challenges arise when building this framework for ER explanations. When it comes to *interpretability*, we need to provide understandable explanations to the end-user. Yet, we cannot assume any knowledge of the underlying model internals, and have to provide *model-agnostic* explanations. For *interactivity*, the framework should explain how the model would behave if certain features were different, provide flexibility to navigate through different granularity levels of explanations, and allow for comparing between different underlying models. Finally, the framework has to target different *audience types* and levels of expertise, and provide different functionalities for either regular users who want to visualize explanations of an ER model on a specific dataset, or experts who want to improve the model, engineer its features, or debug its errors.

II. System Overview

In this section, we give an overview of the typical ER pipeline, and how ExplainER weaves in to explain the model and its predictions. Notice that we are not investigating the ER pipeline itself (*e.g.,* blocking and feature selection), but are rather focusing on interpreting its predictions.

A. Entity Resolution

Let R and R' be two relations with aligned schema $\{A_1, A_2, \ldots, A_m\}$. Furthermore, let $t[A_j]$ be the value of Attribute A_j on Tuple t. Given all distinct tuple pairs $(t, t') \in R \times R'$, ER aims to identify the pairs of tuples that refer to the same real-world entities.

Blocking. Typically, ER solutions run *blocking* methods first, which generate a candidate set $C \subseteq R \times R'$ that includes tuple pairs that are likely to match.

Training/Testing Data. Most, if not all, ER solutions need *training/testing data*, which can be formalized as follows: A labeled dataset is a set of triplets $L \subseteq R \times R' \times \{0, 1\}$, where Triplet $(t, t', 1)$ (resp. $(t, t', 0)$) denotes that Tuples t and t' are (resp. are not) duplicates.

B. Explanations for Entity Resolution

While machine learning provides amazing results in many applications, a common reservation against its use is the lack of transparency and understanding of why a decision is made by a given ML algorithm. Thus, explaining ML algorithms has been the subject of intense research activity.

Some models (*e.g.,* Decision Trees and Linear Models) are interpretable, but many other models are harder to understand. To explain a black-box model, model-agnostic tools either learn an interpretable model on the predictions of the underlying model, or alter the model inputs to monitor its changes.

Some tools focus on how different features contribute to every single instance prediction (*local explanations*), while other tools compute the combined feature importance, or summarize the model as a whole (*global explanations*).

In this demo, we leverage various general-purpose explanation tools for explaining ML-based ER. LIME [7] explains the predictions of any classifier by approximating the classifier locally with an interpretable model for perturbed inputs. It also presents a set of representative instances, selected via submodular optimization, as an explanation of the whole model. Anchors [8] is another tool that targets local explanations by providing explanations based on *if-then* rules (anchors) that sufficiently anchors the prediction locally, such that changes to the rest of the feature values of the instance do not change its predicted class. BRL [9] aims to output global explanations that consist of a series of *if-then-else* statements. These rules discretize the feature space into a series of simple interpretable decision statements. Finally, Skater [10], by datascience.com, uses a combination of algorithms to clarify the relationships between the data a model receives and the output it produces.

While these tools do provide local and global explanations, we build upon their collective multiple-granularity explanations to support further use-cases in the ER context.

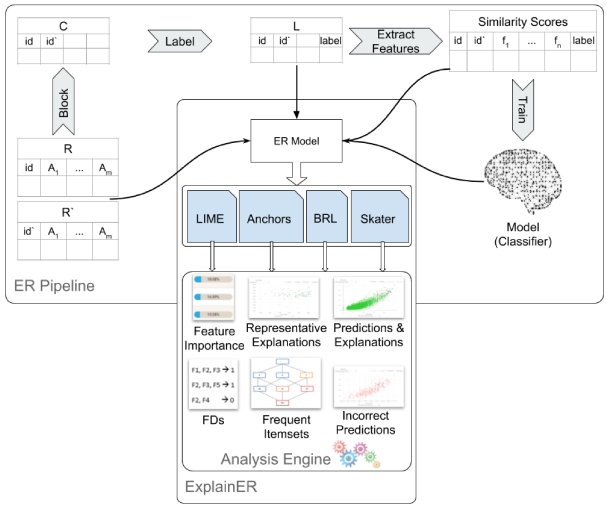


Fig. 1. ER Pipeline and ExplainER Architecture

ExplainER takes in the tuple pairs, labeled data, features and trained model, and processes the explanations from the underlying tools to output more profound analyses. Several examples will be highlighted in the demo. An overview of the ER pipeline and ExplainER architecture is shown in Fig. 1.

III. Demonstration Overview

We propose to showcase ExplainER[1] in action. The audience will choose among various datasets and classifiers, for which ExplainER will provide explanations at multiple granularities. Different scenarios will help users understand the dataset and the classifier, figure out the important features and fine-tune them, and inspect when and why the model performs badly and address such shortcomings.

Due to space limitations, we only show examples on the DBLP-ACM dataset. However, in the actual demo, we will showcase several multi-domain benchmark datasets [11], [12] that have been frequently used in ER research, along with different classifiers from Magellan [3]. Since ExplainER deals with classifiers as black-boxes and does not assume any knowledge of the underlying models, it can be easily extended to work with others without loss of generality.

The different scenarios of the demo are as follows: 1) *Global Explanations* to present a global interpretation of an ER model via post-processing the instance-level explanations. 2) *Model Analysis* to provide a deeper understanding and highlight when the model works well and when it does not. 3) *Differential Analysis* to compare two different classifiers with a focus on where they differ.

A. Global Explanations

While several frameworks provide local explanations for predictions on instance-level, our goal in this scenario is to provide a global understanding of the ER model in whole. To

[1] https://dcx.cs.purdue.edu

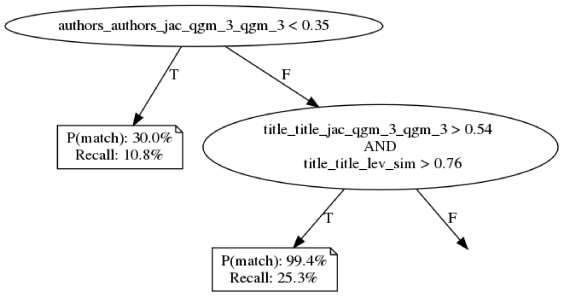
that end, EXPLAINER uses different channels to communicate various aspects and properties of the model to the end user (examples are shown in Fig. 2):

- *Feature Importance* provided by Skater [10] or derived from feature contributions over all local explanations (Fig. 2a). This helps understand how significant each feature is, and how much it affects the model decisions.
- Approximating the model as a BRL [9] (Fig. 2b). This helps compare features importance and visualize how each guides the model through the feature space.
- Plotting all predictions against feature values to visualize different clusters and inspect predictions in each cluster.
- Plotting all local explanations against feature weights to visualize how effectively different features can distinguish matches from non-matches.
- Choosing a set of *representative explanations* as a summary of all explanations (Fig. 2c), another shot at global understanding of the classification model. While SP-LIME [7] calculates top-K diverse explanations, it assumes equal contributions for features, and does not distinguish between important and unimportant ones. We use the K-Medoids algorithm [13] in order to take feature weights into consideration as well. The K-Medoids algorithm chooses representatives among explanations themselves, attempting to minimize the distance (in feature weights) between those in the same cluster.

All of these can help the non-expert user have an overall perspective of the ER model and identify important features and their corresponding contributions to the classifier. They also allow experts to carry out feature engineering and proceed further with improving and fine-tuning the model.

Name	Left Attribute	Right Attribute	Similarity Function	Importance
year_year_exm	year	year	Exact Match	18.08%
title_title_mel	title	title	Monge-Elkan Algorithm	16.89%
year_year_lev_dist	year	year	Levenshtein Distance	15.08%

(a) Feature Importance



(b) Bayesian Rule List (BRL)

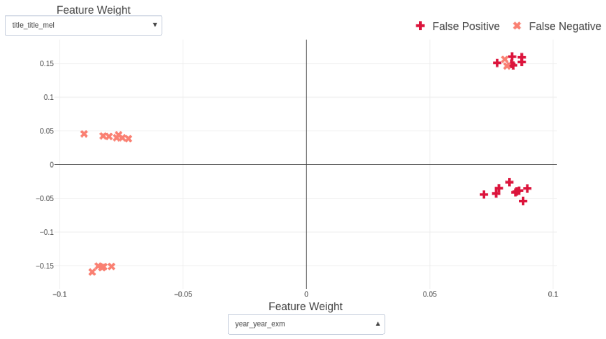
(c) Feature weights for representative explanations

Fig. 2. Global Explanations for Random Forest classifier on DBLP-ACM

B. Model Analysis

In this scenario, the goal is to have a deeper understanding of the model and features via analysis on the explanations. Using individual basic explanations, we construct more advanced informed interpretations (examples are shown in Fig. 3):

- Inspecting every individual explanation by LIME [7] and Anchors [8]. This can help answer why a specific instance is a match (true positive) or a non-match (true negative), and more interestingly explain erroneous predictions (false positives and false negatives).
- Visualizing representative explanations of incorrect predictions, *i.e.,* false positives and false negatives, to highlight where the model fails (Fig. 3a).
- To highlight possible correlations between features and explain which sets of features contributed together towards a prediction, we mine *frequent itemsets* and *association rules* [14] from explanations formatted as vectors of feature weights (Fig. 3b).



(a) Representative explanations of incorrect predictions

Feature Itemset	Support
year_year_anm + authors_authors_mel + title_title_cos_dlm_dc0_dlm_dc0	92.34%
authors_authors_cos_dlm_dc0_dlm_dc0 + year_year_exm + authors_authors_mel	86.23%
authors_authors_jac_qgm_3_qgm_3 + year_year_exm	79.15%

(b) Features Frequent Itemsets

Fig. 3. Model Analysis for Random Forest classifier on DBLP-ACM

TABLE I
WRONG LABELS IN DBLP-ACM DATASET

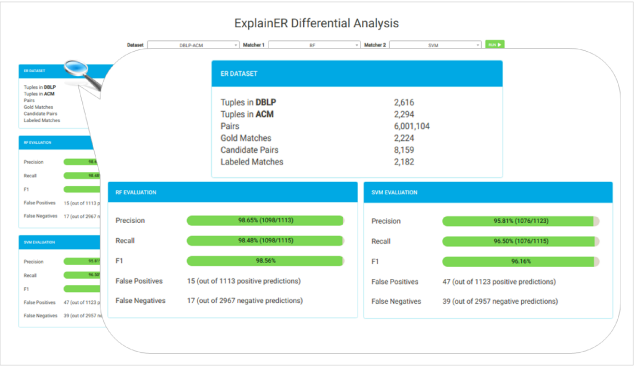
title	authors	venue	year
Reminiscences on Influential Papers	Richard T. Snodgrass	SIGMOD Record	1998
Reminiscences on influential papers	Richard Snodgrass	ACM SIGMOD Record	1998
XPath processing in a nutshell	Reinhard Pichler *et al*	SIGMOD Record	2003
XPath processing in a nutshell	Georg Gottlob *et al*	ACM SIGMOD Record	2003

- Mining *FD (CFD) rules* [15] from explanations' feature contributions formatted as binary feature vectors, to compare against rule-based global explanations and feature importance results.

The above approaches towards batch explanations allow to zoom in beyond local explanations and contrast against global explanations. Additionally, they can help spot and tackle issues in the dataset itself, for example, dealing with heterogeneous values in the same attribute, *e.g.,* the currency of price attributes in the Amazon-Google dataset, or detecting the more serious issue of wrong labels. In Table I, we show a couple of examples for pairs correctly classified as matches, although wrongly labeled as non-matches in the dataset, discovered while inspecting false positives explanations.

C. Differential Analysis

In this scenario, we have two different models $M1$ and $M2$, and the goal is to understand how they compare against each other. We investigate both models predictions, generate explanations for each pair of tuples for both models, and highlight where they disagree. The user specifies the input dataset and two classifiers, and is then provided with different comparisons of their predictions and explanations (Fig. 4a). Like other scenarios, we can still explain each model from a global perspective (*e.g.,* feature importance, BRL and representative explanations), or look closely at the explanations and the model analysis (*e.g.,* local explanations, incorrect predictions and features frequent itemsets), but more importantly we can easily inspect those predictions where the two models agree and where they do not (Fig. 4b).



(a) One dataset and two ER models

(b) Predictions where the two models disagree

Fig. 4. Differential Analysis for RF and SVM classifiers on DBLP-ACM

ACKNOWLEDGMENT

Walid G. Aref's research is partly supported by the National Science Foundation under Grant Number III-1815796.

REFERENCES

[1] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.

[2] P. Christen, *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media, 2012.

[3] P. Konda, S. Das, P. Suganthan GC, A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. Naughton *et al.*, "Magellan: Toward building entity matching management systems," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1197–1208, 2016.

[4] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra, "Deep learning for entity matching: A design space exploration," in *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 19–34.

[5] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, "Distributed representations of tuples for entity resolution," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1454–1467, 2018.

[6] X. Wang, L. Haas, and A. Meliou, "Explaining data integration," *Data Engineering*, p. 47, 2018.

[7] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[8] ——, "Anchors: High-precision model-agnostic explanations," in *AAAI Conference on Artificial Intelligence*, 2018.

[9] B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[10] P. Choudhary, A. Kramer, and c. datascience.com team, "Skater: Model interpretation library," Mar. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1198885

[11] H. Köpcke, A. Thor, and E. Rahm, "Benchmark datasets for entity resolution," https://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution.

[12] S. Das, A. Doan, P. S. G. C., C. Gokhale, and P. Konda, "The magellan data repository," https://sites.google.com/site/anhaidgroup/useful-stuff/data.

[13] L. Kaufman and P. Rousseeuw, *Clustering by means of medoids*. North-Holland, 1987.

[14] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th international conference on Very Large Data Bases, VLDB*, vol. 1215, 1994, pp. 487–499.

[15] W. Fan, F. Geerts, J. Li, and M. Xiong, "Discovering conditional functional dependencies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 5, pp. 683–698, 2011.