DADER: Hands-Off Entity Resolution with Domain Adaptation

Jianhong Tu
Renmin University, China
tujh@ruc.edu.cn

Xiaoyue Han
Renmin University, China
cloverhxy@mail.sdu.edu.cn

Ju Fan[*]
Renmin University, China
fanj@ruc.edu.cn

Nan Tang
QCRI, Qatar
ntang@hbku.edu.qa

Chengliang Chai
Tsinghua University, China
ccl@mail.tsinghua.edu.cn

Guoliang Li
Tsinghua University, China
liguoliang@tsinghua.edu.cn

Xiaoyong Du
Renmin University, China
duyong@ruc.edu.cn

ABSTRACT

Entity resolution (ER) is a core data integration problem that identifies pairs of data instances referring to the same real-world entities, and the state-of-the-art results of ER are achieved by deep learning (DL) based approaches. However, DL-based approaches typically require a large amount of labeled training data (*i.e.,* matching and non-matching pairs), which incurs substantial manual labeling efforts. In this paper, we introduce **DADER**, a hands-off deep ER system through domain adaptation. **DADER** utilizes multiple well-labeled source ER datasets to train a DL-based ER model for a new target ER dataset that does not have any labels or with only a few labels. To address the key challenge of domain shift, **DADER** judiciously selects labeled entity pairs from the source and then aligns distributions of the source and the target by using six popular domain adaptation strategies. **DADER** can also harness the users to gather a few labels for further improvement. We have built **DADER** as an open-sourced Python Library with intuitive APIs and demonstrated its utility on supporting hands-off ER in real-world scenarios.

PVLDB Reference Format:
Jianhong Tu, Xiaoyue Han, Ju Fan, Nan Tang, Chengliang Chai, Guoliang Li, and Xiaoyong Du. **DADER**: Hands-Off Entity Resolution with Domain Adaptation. PVLDB, 15(12): XXX-XXX, 2022.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://pypi.org/project/dader/.

1 INTRODUCTION

Entity resolution (ER) aims to determine whether two data instances refer to the same real-world entity. As a core task in data integration, ER has been extensively studied for decades [1, 3–6]. Not surprisingly, the state-of-the-art results are achieved by deep learning (DL) based approaches. A typical DL-based approach takes two data instances (or an *entity pair*) as input, and outputs a Boolean `match`/`non-match` result by formulating the task as a binary classification problem. However, DL-based ER approaches are typically *data-hungry*, *i.e.,* requiring a large amount of labeled training data. Take Ditto [4] as an example: even leveraging pre-trained language models, Ditto still needs thousands of labels to achieve good performance. Therefore, a key bottleneck for DL-based ER is the substantial labeling efforts to create enough training data.

Fortunately, the prevalence of big data provides an opportunity of *reusing* a lot of labeled ER datasets (namely *source* datasets), either from public benchmarks or within enterprises. For example, given a new *target* ER task that aims to match identical products from two e-commerce websites, say `Walmart` and `Amazon` (*i.e.,* `Walmart-Amazon`), it has the potential to largely reduce the prohibitive manual labeling cost, if we can reuse existing ER datasets in the same or relevant domains, *e.g.,* the `Abt-Buy` dataset [5].

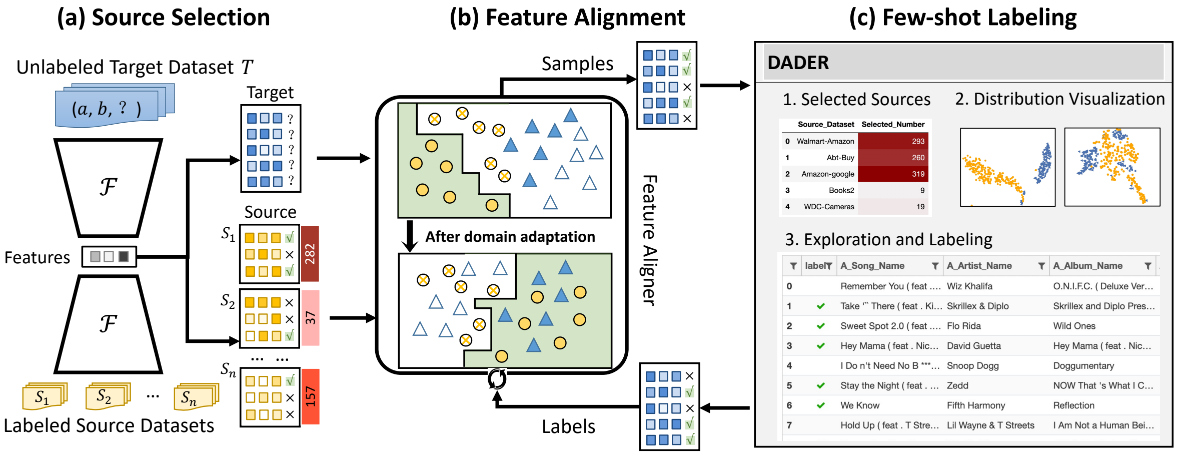
However, the straightforward strategy of *directly* reusing source ER datasets for a target ER dataset may not be effective, which may even lead to disastrous performance, due to the well-recognized *domain shift* problem [7]. An example of domain shift is illustrated in the upper rectangle of Figure 1 (b), where yellow circles and blue triangles respectively represent entity pairs from the source and the target. Take `Abt-Buy` [5] as a source and `Walmart-Amazon` [5] as a target. Since source and target datasets have different attributes, their vector-based features may not follow the same distribution, a DL-based model (the boundary line) trained on the source cannot correctly predict the target, *e.g.,* all the target pairs being predicted as `non-matches`. To address the problem, we introduce **DADER**, a hands-off deep ER system (*i.e.,* zero label for the target ER dataset) with *domain adaptation* (DA), which can align the features of source and target datasets and result in a good DL-based model for the target, as shown in the lower rectangle of Figure 1 (b). As reported in our experimental results [7], DA can improve the performance of `Walmart-Amazon` [5] by 14.2% when using `Abt-Buy` [5] as source.

An overview of DADER. Figure 1 illustrates how **DADER** achieves hands-off ER through DA that has been extensively studied in computer vision and natural language processing, yet under-explored in the ER scenarios. **DADER** consists of the following modules.

Source Selection. We do not presume that all entity pairs from the source ER datasets are equally useful to our target ER dataset. In fact, we may collect a large number of well-labeled source datasets in a variety of domains, such as product, citation and restaurant. Thus, **DADER** identifies from multiple source datasets the entity pairs that are *useful* to the target dataset. *e.g.,* the 282/37/157 entity pairs from source datasets $S_1/S_2/S_n$, as shown in Figure 1 (a). The key challenge here is how to measure the *usefulness* of each source entity pair to the target. To address this, **DADER** first uses a feature extractor $\mathcal{F}$ to map both source and target data to a high-dimensional space,

[*] Ju Fan is the corresponding author.

Proceedings of the VLDB Endowment, Vol. 15, No. 12 ISSN 2150-8097.
doi:XX.XX/XXX.XX



	Source_Dataset	Selected_Number
0	Walmart-Amazon	293
1	Abt-Buy	260
2	Amazon-google	319
3	Books2	9
4	WDC-Cameras	19

	label	A_Song_Name	A_Artist_Name	A_Album_Name
0		Remember You (feat ….	Wiz Khalifa	O.N.I.F.C. (Deluxe Ver…
1	✓	Take '` There (feat . Ki…	Skrillex & Diplo	Skrillex and Diplo Pres…
2	✓	Sweet Spot 2.0 (feat ….	Flo Rida	Wild Ones
3	✓	Hey Mama (feat . Nic…	David Guetta	Hey Mama (feat . Nic…
4		I Do n't Need No B ***…	Snoop Dogg	Doggumentary
5	✓	Stay the Night (feat . …	Zedd	NOW That 's What I C…
6	✓	We Know	Fifth Harmony	Reflection
7		Hold Up (feat . T Stre…	Lil Wayne & T Streets	I Am Not a Human Bei…

Figure 1: Overview of the DADER System. (a) *Source Selection* identifies from multiple source datasets the entity pairs relevant to the target dataset. (b) *Feature Alignment* learns from the data what is the best way of aligning distributions of source and target, such that models trained on the labeled source can be adapted to the unlabeled target. (c) *Few-shot Labeling* allows users to explore the prediction results on the target and solicits the users to provide a few labels for further improvement.

then measures similarities between source and target entity pairs, and finally selects the source entity pairs with high similarities.

Feature Alignment. After source selection, there may be still distribution change or domain shift between the source and the target, which leads to performance degradation of the DL-based ER models. To tackle this difficulty, **DADER** learns from the data what is the best way of aligning distributions of source and target, such that models trained on the labeled source can be adapted to the unlabeled target. To this end, **DADER** adopts the most fruitful family of DA techniques, *i.e.,* learning *domain-invariant* and *discriminative* features, and allows users to systematically explore various categories of DA solutions to find the one adequate for the current target dataset. More details can be found in our research paper [7].

Moreover, **DADER** helps users understand the effect of DA via visualizations, *e.g.,* ratios of selected entity pairs in source datasets and distributions of source and target before/after feature alignment.

Few-shot Labeling. **DADER** can also solicit users to provide *a few* labels to further improve the performance. Specifically, **DADER** samples target entity pairs with predicted labels via active learning, and asks users to verify the labels.

Differences from existing solutions. Many existing works in the literature consider reducing human labeling efforts for ER, including crowdsourcing [10] and weak supervision [9]. Compared with them, **DADER** focuses on reusing existing well-labeled source ER datasets, and develops a unified framework that explores the design space of DA for ER. Our experimental results [7] show that DA is very helpful to improve ER when domain shift happens. Moreover, the performance of our approach can be maintained at a high level with a few labels, while outperforming state-of-the-art DL-based ER methods, *e.g.,* DeepMatcher [5] and Ditto [4].

Demonstration scenarios. We build **DADER** as an open-sourced Python Library in PyPI[1]. This paper demonstrates **DADER**'s utility in supporting hands-off ER by using the following scenarios. (1) We prepare over 20 well-labeled ER datasets in various domains and upload them to Hugging Face[2]. In this way, we allow different users to easily customize and access to the datasets they want to reuse. (2) For a specific target ER dataset uploaded by users, we demonstrate how the dataset benefits from **DADER** to quickly obtain match/non-match results, with zero label on the target dataset. (3) We show the intuitive APIs of **DADER** that help users easily explore the DA process, and demonstrate how **DADER** can further improve the ER performance given few-shot labeling from users. A demonstration video can be found on Youtube[3].

To summarize, we make the following contributions. (1) We develop **DADER**, a hands-off deep ER system with domain adaptation. (2) **DADER** is helpful for users to reuse well-labeled ER datasets for a new ER task, without labels or with a few labels on the target. (3) We deploy **DADER** as an open-sourced Python Library and demonstrate its utility on real ER scenarios.

2 SYSTEM OVERVIEW

Given an unlabeled target ER dataset T, **DADER** aims to determine whether each entity pair (a, b) in T refers to the same real-world entity (*i.e.,* match) or not (*i.e.,* non-match). To this end, **DADER** employs a DL-based approach that formalizes this as a binary classification problem. To reduce expensive labeling efforts for training DL models, the goal of **DADER** is to reuse a set of labeled source datasets $\{S_1, S_2, \ldots, S_n\}$ and solve the well-recognized *domain shift* problem by leveraging the *domain adaptation* (DA) techniques. Next, we will provide more details for the three modules of **DADER** as shown in Figure 1. Note that a typical ER pipeline consists of both *blocking* and *matching* phases, and this paper focuses on the matching phase.

2.1 Source Selection

The key challenge here is to determine which source entity pairs in source datasets $\{S_1, S_2, \ldots, S_n\}$ are useful to the target dataset T. To solve this, **DADER** measures *distributional similarities* between source

[1] https://pypi.org/project/dader/

[2] https://huggingface.co/datasets/RUC-DataLab/ER-dataset

[3] https://youtu.be/vnZNOzLYV3s

and target entity pairs. We introduce a feature extractor $\mathcal{F}$ to map each entity pair, either in source or target, into a high-dimensional vector (*a.k.a.* features). Following [4], we utilize pre-trained language models (*e.g.,* BERT) to implement $\mathcal{F}$. Given an entity pair (a, b), we serialize it into a sequence of tokens `[CLS]`a`[SEP]`b`[SEP]`, where `[CLS]` and `[SEP]` are special tokens in BERT. Then, we feed the token sequence to BERT and obtain a feature vector. After that, we develop two methods for selecting similar source entity pairs. (1) A classifier-based method trains a domain classifier on all the data, and selects the source entity pairs that are not easily distinguishable from the target. The classifier can be implemented by a machine learning model, such as multi-layer perceptron (MLP) or Random Forest. (2) A metric-based method uses simple distance metrics, such as cosine similarity or euclidean distance, to select source entity pairs. Furthermore, as it may be time-consuming to perform the aforementioned process for all source entity pairs, we devise an efficient TF-IDF based strategy to first filter out the source entity pairs which are not textually similar enough to the target.

2.2 Feature Alignment

The key challenge is how to align distributions of the source and target datasets, such that models trained on the source can predict well for the target. **DADER** adopts the most popular and fruitful family of DA techniques on learning *domain-invariant* and *discriminative* features for both source and target datasets. Along this line, **DADER** develops six popular DA strategies falling into three categories, which will be briefly described below. Please find more details about these strategies in our research paper [7].

Discrepancy-based approaches use statistical metrics to minimize the domain distribution discrepancy between source and target datasets. **DADER** provides two metrics: Maximum Mean Discrepancy (MMD) and K-orders, which have been shown very effective. Specifically, during training, these methods compute statistical distribution discrepancy on top of the features of source and target datasets, and reduce the discrepancy via back propagation, so as to make distributions of source and target as close as possible.

Adversarial-based approaches use adversarial training to adjust feature extractor $\mathcal{F}$ to map source and target data into an appropriate feature space. To this end, the approaches implement the feature aligner as a domain classifier, which is trained to distinguish features from source or target, while the feature extractor $\mathcal{F}$ tries to generate the features that can confuse this domain classifier. In this way, we can align distributions between source and target to reduce domain shift. **DADER** realizes three representative adversarial-based methods, namely gradient reversal layer (GRL), inverted labels GAN (InvGAN) and InvGAN + Knowledge distillation (KD).

Reconstruction-based approaches introduce an auxiliary *unsupervised reconstruction task.* Specifically, the feature aligner is used as the decoder to reconstruct the input of feature extractor $\mathcal{F}$, to ensure that features contain useful and shared information across the source and target datasets. **DADER** provides the Encoder-Decoder (ED) networks to support this strategy on ER.

2.3 Few-shot Labeling

The previous Source Selection and Feature Alignment are totally *hands-off, i.e.,* the system can automatically complete the entity matching task without human intervention. **DADER** can adopt few-shot learning to further improve the performance of ER results on the target with human interaction. To this end, **DADER** uses *active learning* strategies to sample a small subset of target entity pairs, and asks the users to verify their labels. After gathering users' feedbacks, **DADER** can utilize the limited number of target labels to refine our models, which would result in further improvement on the ER results. This few-shot labeling process can repeat multiple rounds until the user is satisfied with the results.

3 DEMONSTRATION SCENARIOS

In this section, we will walk through a concrete example to demonstrate how to use **DADER** for hands-off ER.

Unlabeled target ER dataset. We will use `iTunes-Amazon`, a dataset about music from Magellan [2]. It contains two tables from websites *iTunes* and *Amazon*, respectively. Each entity in the tables consists of ten attributes (*e.g.,* Song-Name, Artist-Name, Price). Note that we presume that this ER dataset does not have *any* label.

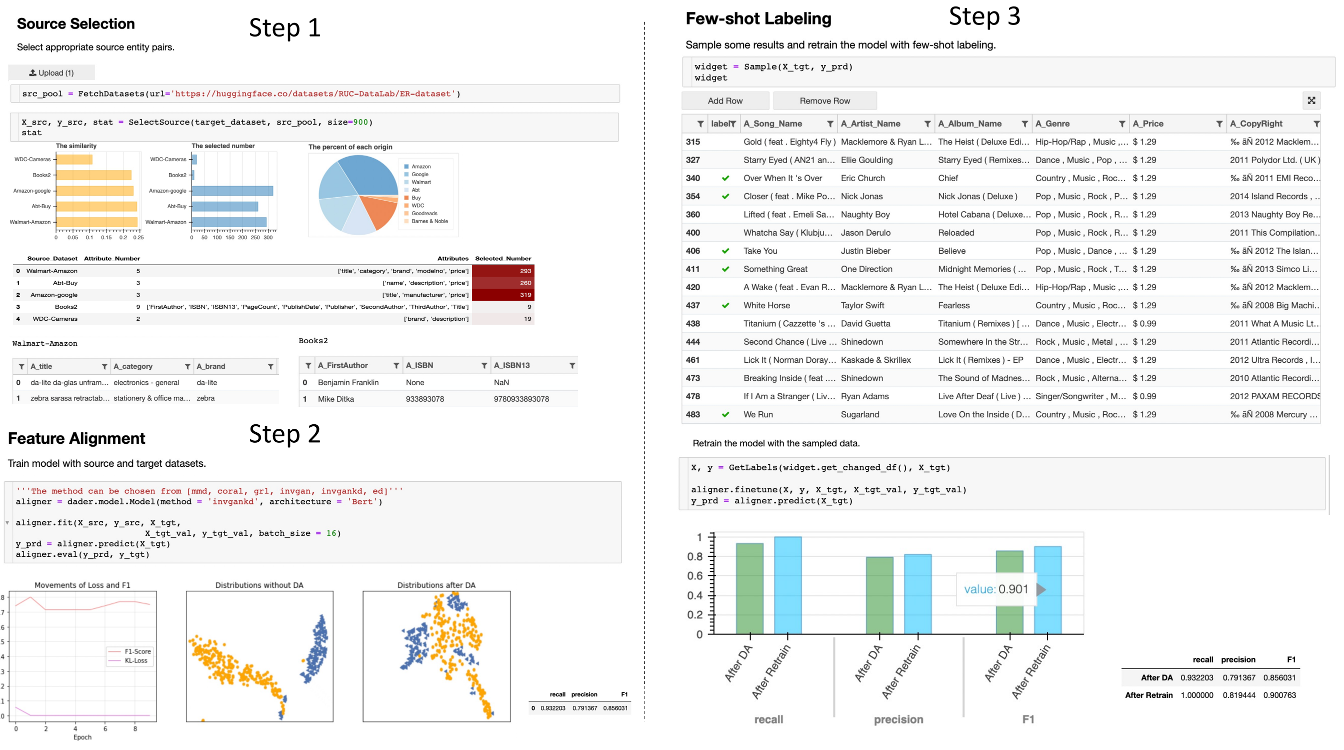
Labeled source ER datasets. We prepare over 20 well-labeled ER datasets in various domains and upload them in Hugging Face, as described in Section 1. These datasets cover ten domains, including product, movie, book, citation, restaurant, and so on.

Goals. We have three demonstration goals. **(G1) DADER** can automatically select labeled source entity pairs from multiple datasets that are possibly from different domains, *a.k.a. Source Selection.* **(G2) DADER** can align the features of source and target entity pairs, *a.k.a. Feature Alignment.* **DADER** will then train a binary classifier using selected labeled source entity pairs, which will be directly applied to unlabeled target entity pairs, *a.k.a. Hands-off ER.* **(G3) DADER** also offers intuitive interfaces that allow the audience to give a small number of labels for the target dataset to further improve the model performance, *a.k.a. Few-shot labeling.*

Next we describe the following concrete steps corresponding to the above three goals, as shown in Figure 2.

Step 1: Source Selection. We first use the "Upload" button to upload our target dataset `iTunes-Amazon`. Then, we fetch our well-labeled source ER datasets from the aforementioned repository in Hugging Face. Next, we call the "`SelectSource`" function with parameters, such as the maximum number of selected source tuple pairs. This function will automatically select source entity pairs that are useful to the target. In order to help users understand how source tuples are selected, we use histograms to visualize the numbers of selected data pairs from different source datasets. As illustrated in Figure 2, we can see that **DADER** selects a large number of data pairs from `Amazon-Google`, `Walmart-Amazon` and `Abt-Buy`, which are highly similar with our target dataset `iTunes-Amazon`. Interestingly, the pie chart in the figure shows that most of the selected data instances are from e-commerce website *Amazon.* The main reason is that our target dataset `iTunes-Amazon` also contains many music records from `Amazon`, and the selected data pairs are similar to the target, *e.g.,* having similar vocabularies. Furthermore, if users are not satisfied with the result, they can customize their own source selectors via our intuitive APIs to achieve desired results.

Step 2: Feature Alignment. We directly call the encapsulated package to train the models with four lines of code. Here, as an example, we use the default BERT model and InvGAN+KD strategy (see



Source Selection

Step 1

Select appropriate source entity pairs.

```
src_pool = FetchDatasets(url='https://huggingface.co/datasets/RUC-DataLab/ER-dataset')
```

```
X_src, y_src, stat = SelectSource(target_dataset, src_pool, size=900)
stat
```

	Source_Dataset	Attribute_Number	Attributes	Selected_Number
0	Walmart-Amazon	5	['title', 'category', 'brand', 'modelno', 'price']	293
1	Abt-Buy	3	['name', 'description', 'price']	260
2	Amazon-google	3	['title', 'manufacturer', 'price']	319
3	Books2	9	['FirstAuthor', 'ISBN', 'ISBN13', 'PageCount', 'PublishDate', 'Publisher', 'SecondAuthor', 'ThirdAuthor', 'Title']	9
4	WDC-Cameras	2	['brand', 'description']	19

Walmart-Amazon

	A_title	A_category	A_brand
0	da-lite da-glas unfram…	electronics - general	da-lite
1	zebra sarasa retractab…	stationery & office ma…	zebra

Books2

	A_FirstAuthor	A_ISBN	A_ISBN13
0	Benjamin Franklin	None	NaN
1	Mike Ditka	933893078	9780933893078

Feature Alignment

Step 2

Train model with source and target datasets.

```
'''The method can be chosen from [mmd, coral, grl, invgan, invgankd, ed]'''
aligner = dader.model.Model(method = 'invgankd', architecture = 'Bert')

aligner.fit(X_src, y_src, X_tgt,
                        X_tgt_val, y_tgt_val, batch_size = 16)
y_prd = aligner.predict(X_tgt)
aligner.eval(y_prd, y_tgt)
```

	recall	precision	F1
0	0.932203	0.791367	0.856031

Few-shot Labeling

Step 3

Sample some results and retrain the model with few-shot labeling.

```
widget = Sample(X_tgt, y_prd)
widget
```

	label	A_Song_Name	A_Artist_Name	A_Album_Name	A_Genre	A_Price	A_CopyRight
315		Gold (feat . Eighty4 Fly)	Macklemore & Ryan L…	The Heist (Deluxe Edi…	Hip-Hop/Rap , Music ,…	$ 1.29	‰ ãÑ 2012 Macklem…
327		Starry Eyed (AN21 an…	Ellie Goulding	Starry Eyed (Remixes…	Dance , Music , Pop , …	$ 1.29	2011 Polydor Ltd. (UK)
340	✓	Over When It 's Over	Eric Church	Chief	Country , Music , Roc…	$ 1.29	‰ ãÑ 2011 EMI Reco…
354	✓	Closer (feat . Mike Po…	Nick Jonas	Nick Jonas (Deluxe)	Pop , Music , Rock , P…	$ 1.29	2014 Island Records , ..
360		Lifted (feat . Emeli Sa…	Naughty Boy	Hotel Cabana (Deluxe…	Pop , Music , Rock , R…	$ 1.29	2013 Naughty Boy Re…
400		Whatcha Say (Klubju…	Jason Derulo	Reloaded	Pop , Music , Rock , R…	$ 1.29	2011 This Compilation..
406	✓	Take You	Justin Bieber	Believe	Pop , Music , Dance , …	$ 1.29	‰ ãÑ 2012 The Islan…
411	✓	Something Great	One Direction	Midnight Memories (…	Pop , Music , Rock , T…	$ 1.29	‰ ãÑ 2013 Simco Li…
420		A Wake (feat . Evan R…	Macklemore & Ryan L…	The Heist (Deluxe Edi…	Hip-Hop/Rap , Music ,…	$ 1.29	‰ ãÑ 2012 Macklem…
437	✓	White Horse	Taylor Swift	Fearless	Country , Music , Roc…	$ 1.29	‰ ãÑ 2008 Big Machi..
438		Titanium (Cazzette 's …	David Guetta	Titanium (Remixes) […	Dance , Music , Electr…	$ 0.99	2011 What A Music Lt..
444		Second Chance (Live …	Shinedown	Somewhere In the Str…	Rock , Music , Metal , …	$ 1.29	2011 Atlantic Recordi…
461		Lick It (Norman Doray…	Kaskade & Skrillex	Lick It (Remixes) - EP	Dance , Music , Electr…	$ 1.29	2012 Ultra Records , I…
473		Breaking Inside (feat ….	Shinedown	The Sound of Madnes…	Rock , Music , Alterna…	$ 1.29	2010 Atlantic Recordi…
478		If I Am a Stranger (Liv…	Ryan Adams	Live After Deaf (Live) …	Singer/Songwriter , M…	$ 0.99	2012 PAXAM RECORDS
483	✓	We Run	Sugarland	Love On the Inside (D…	Country , Music , Roc…	$ 1.29	‰ ãÑ 2008 Mercury …

Retrain the model with the sampled data.

```
X, y = GetLabels(widget.get_changed_df(), X_tgt)

aligner.finetune(X, y, X_tgt, X_tgt_val, y_tgt_val)
y_prd = aligner.predict(X_tgt)
```

	recall	precision	F1
After DA	0.932203	0.791367	0.856031
After Retrain	1.000000	0.819444	0.900763

Figure 2: Demonstration scenarios of DADER on dataset `iTunes-Amazon` in the Music domain.

Section 2). Users can adjust parameters as required. To show the changes of feature distributions before and after feature alignment, we use t-SNE [8] to map the features of source and target datasets into a two-dimensional space and plot all entity pairs in the space. From the two plots, we can clearly see that source and target entity pairs are obviously more mixed when we apply DA, which indicates the domain shift reduction. To show the performance of the models, we draw the F1-score curve of target validation set during training. Based on this, the user can know the training process of the model. The training process takes around three minutes to converge. Finally, The F1-score achieved by **DADER** on the `iTunes-Amazon` dataset is 0.86, with *zero* label from the target dataset.

Step 3: Few-shot Labeling. We sample some predicted results of target to users via active learning, as described in Section 2.3. Each entity pair with a green tick indicates that the corresponding predicted label is matching, while the rest ones are non-matching. Users can view the data and update labels by clicking the corresponding rows. Then these sampled data with predicted labels or manually annotated true labels will be used to retrain the models. This step can repeat multiple rounds until the user is satisfied with the results. Here for our `iTunes-Amazon` example, we sample a set of 50 predicted results in one round and update labels of **only three** entity pairs in the sample, which then improves the F1-score of the predicted ER results from 0.86 to 0.90.

ACKNOWLEDGMENTS

This work was partly supported by NSF of China (62072458, 62122090, 62072461, U1911203), Medical AI research and development project of PLAGH (2019MBD-046), and the Special Funds for Central Universities to Build World-class Universities (Disciplines) and Guide the Development of Distinctive Features - Basic Research Funds for Central Universities" of Renmin University of China, 2022.

REFERENCES

[1] Mikhail Bilenko and Raymond J Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *SIGKDD*. 39–48.

[2] AnHai Doan, Pradap Konda, Paul Suganthan GC, Yash Govind, Derek Paulsen, Kaushik Chandrasekhar, Philip Martinkus, and Matthew Christie. 2020. Magellan: toward building ecosystems of entity matching solutions. *Commun. ACM* 63, 8 (2020), 83–91.

[3] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *Proc. VLDB Endow.* 11, 11 (2018), 1454–1467.

[4] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584* (2020).

[5] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *SIGMOD*. 19–34.

[6] Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Synthesizing entity matching rules by examples. *Proc. VLDB Endow.* 11, 2 (2017), 189–202.

[7] Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Chengliang Chai, Guoliang Li, Ruixue Fan, and Xiaoyong Du. 2022. Domain Adaptation for Deep Entity Resolution. In *SIGMOD*. 443–457.

[8] Laurens Van Der Maaten. 2013. Barnes-hut-sne. *arXiv preprint arXiv:1301.3342* (2013).

[9] Renzhi Wu, Prem Sakala, Peng Li, Xu Chu, and Yeye He. 2021. Demonstration of Panda: A Weakly Supervised Entity Matching System. *Proc. VLDB Endow.* 14, 12 (2021), 2735–2738.

[10] Jingru Yang, Ju Fan, Zhewei Wei, Guoliang Li, Tongyu Liu, and Xiaoyong Du. 2018. Cost-Effective Data Annotation using Game-Based Crowdsourcing. *Proc. VLDB Endow.* 12, 1 (2018), 57–70.