DAME: Domain Adaptation for Matching Entities

Mohamed Trabelsi
mot218@lehigh.edu
Lehigh University
Bethlehem, PA, USA

Jeff Heflin
heflin@cse.lehigh.edu
Lehigh University
Bethlehem, PA, USA

Jin Cao
jin.cao@nokia-bell-labs.com
Nokia Bell Labs
Murray Hill, NJ, USA

ABSTRACT

Entity matching (EM) identifies data records that refer to the same real-world entity. Despite the effort in the past years to improve the performance in EM, the existing methods still require a huge amount of labeled data in each domain during the training phase. These methods treat each domain individually, and capture the specific signals for each dataset in EM, and this leads to overfitting on just one dataset. The knowledge that is learned from one dataset is not utilized to better understand the EM task in order to make predictions on the unseen datasets with fewer labeled samples. In this paper, we propose a new domain adaptation-based method that transfers the task knowledge from multiple source domains to a target domain. Our method presents a new setting for EM where the objective is to capture the task-specific knowledge from pretraining our model using multiple source domains, then testing our model on a target domain. We study the zero-shot learning case on the target domain, and demonstrate that our method learns the EM task and transfers knowledge to the target domain. We extensively study fine-tuning our model on the target dataset from multiple domains, and demonstrate that our model generalizes better than state-of-the-art methods in EM.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Transfer learning*.

KEYWORDS

entity matching, transfer learning, domain adaptation

ACM Reference Format:
Mohamed Trabelsi, Jeff Heflin, and Jin Cao. 2022. DAME: Domain Adaptation for Matching Entities . In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498486

1 INTRODUCTION

Entity matching (EM) identifies data records that refer to the same real-world entity. EM is an important step in data cleaning and integration [6], knowledge base enrichment [25], and entity linking [33]. Researchers have studied EM for many years in the context of data mining and integration.

In the past few years, deep learning (DL) has led to a significant improvement in multiple tasks, where DL-based methods achieved state-of-the-art (SOTA) results for text, image, and speech data. In many cases, DL models are trained end-to-end to automatically extract features and build predictive models. This significantly reduces the human effort that is needed in traditional methods for feature engineering, and gives the model the ability to capture specific features that are better than the hand-crafted ones for multiple tasks. Following the success of DL models, researchers have focused on exploring DL in data cleaning and integration. In particular, multiple DL methods have been proposed to solve the EM task [9, 11, 16, 24, 43]. Deep contextualized language models (DCLM), like BERT [8], RoBERTa [21], and DistilBERT [30] have been recently proposed to solve multiple tasks [5, 29, 35–37, 39]. Building on DCLM, Ditto [20] achieved SOTA results in EM.

Although DL methods have led to a significant improvement in the EM task, these models need a huge amount of labeled data for each domain. DL-based models are trained in a supervised setting for each dataset in EM, where a different model is obtained and is fully fine-tuned on a specific dataset. This means that existing models capture the specific signals for each dataset in EM which leads to overfitting on just one dataset. In addition, the knowledge that is learned from one dataset is not explored to better understand the EM task so that the predictions in other datasets can be made with fewer labeled samples.

In order to overcome the limitations of prior methods, we propose a new method, called ***D**omain **A**daptation for **M**atching **E**ntities* (DAME), that transfers the task knowledge from multiple source domains to a target domain. Our method presents a new setting for EM where the objective is to capture task-specific knowledge from pretraining our model using multiple source domains, then testing our model on a target domain. In our study, we are interested in two aspects of our model. First, we study the zero-shot learning (ZSL) case of DAME on the target domain. Second, we study the effect of fine-tuning our proposed model on the target domain using different percentages of training data, and we compare our fine-tuned model to SOTA methods. We formulate EM as a mixture of experts with a global shared model [13, 17, 41] where each expert is trained on an individual source domain, and the global model is trained on all domains. Then, we aggregate the features from the experts using a global model-guided attention mechanism. We train DAME with unsupervised domain adaptation (DA) loss functions [13, 41] to reduce the domain shift between the source and target domains.

In summary, we make the following contributions: (1) We propose a new DA-based method for EM. Our new formulation of EM is based on the mixture of experts where we transfer learning from multiple source domains to a target domain. (2) We study the ZSL

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498486

case on the target domain and demonstrate that our method learns the EM task and transfers the task knowledge to the target domain. (3) We extensively study fine-tuning our model on the target dataset from multiple domains, and demonstrate that our model generalizes better than SOTA methods for most of the datasets.

2 RELATED WORK

2.1 Entity matching

EM [1, 9, 16, 20, 24] is the field of research that solves the problem of finding records that refer to the same real-world entity. EM, also known as data matching, record linkage, entity resolution, etc, has been intensively studied in recent years because EM is an important step in data cleaning and integration. Given two collections of records D_1 and D_2, EM classifies a pair of entities $(e_1, e_2), \forall e_1 \in D_1, e_2 \in D_2$ into match or non-match. The records from D_1 and D_2 can have the same or different set of attributes. The value of each attribute is composed of a sequence of tokens.

Comparing all record pairs from D_1 and D_2 grows quadratically. Therefore, a set of candidate pairs $C \subset D_1 \times D_2$, where $|C| \ll |D_1 \times D_2|$ is selected in a separate step, called blocking, before running a computationally expensive algorithm for EM. After the blocking step, each record pair $(e_1, e_2) \in C$ is compared to predict a binary label indicating a match or non-match. Prior works have proposed string similarity-based methods to compare records [7, 10, 22]. Traditional supervised classifiers have been proposed to map the string similarities-based feature vector to a binary class label [4, 6]. Recently, DL-based methods have been proposed to solve EM [9, 11, 16, 20, 24, 43]. The DL methods of EM can be categorized into attribute- and record-level comparison methods. Attribute comparators predict the label of a pair of records based on the signals collected from matching values of the same attribute. DeepMatcher [24], which is the SOTA attribute-level comparator, explores multiple techniques to compute the attribute representation from word embedding, where combining both bidirectional GRU and decomposable attention [27] leads to the best results. The SOTA method in EM is a record-based comparator known as Ditto [20] which is based on DCLM. Ditto models each record by alternating between attributes and data values with two additional special tokens [COL] and [VAL]. Then, Ditto adapts the sentence pair classification setting to EM in order to compare record pairs using the special tokens [SEP] and [CLS] that are added into the input. In addition, Ditto explores domain-specific optimizations by injecting domain knowledge into the input.

2.2 Domain adaptation

DA studies the transfer of task knowledge from a single or multiple labeled source domains to an unlabeled target domain. In this paper, we are interested in the case of multiple source domains known as Multi-Source DA (MSDA). Using only unlabeled data from the target domain is known as Unsupervised DA (UDA).

Existing approaches in UDA focus on reducing the domain shift between the source and target domains by aligning feature vectors [2, 26]. Representation learning methods have been proposed for UDA such as domain adversarial networks [32, 42]. Other representation learning methods include comparing the marginal distribution between the source and target domains in an adversarial way [13] and minimizing the covariance between the source and target representations [34]. An effective strategy in the case of MSDA is known as a mixture of experts [13, 17, 41]. Kim et al. [17] proposed to incorporate an attention mechanism to combine the predictions from multiple models trained on the source domains. Guo et al. [13] proposed a method that is based on a mixture of experts where the posteriors of the models are combined using a point-to-set Mahalanobis distance metric between an input sample and source domains. Wright and Augenstein [41] improved the performance of mixture of experts using DCLM as experts in source domains. This work follows a line of research that investigates the use of Transformers [38] in DA [14, 15, 23, 28]. Ma et al. [23] improved the performance of BERT in the target domain for natural language inference by incorporating a similarity of a given target domain to source domains with curriculum learning [3]. AdaptaBERT [15] is a BERT-based model that is proposed in the case of UDA for the sequence labeling by adding a masked language modeling in the target domain. Fine-tuning of BERT on the target domain was also shown to be effective in the sentiment analysis task [28]. Gururangan et al. [14] combines both domain and task adaptive pretraining to improve the performance of RoBERTa on NLP tasks.

3 PROBLEM STATEMENT

Our formulation of DA in EM task is based on the unsupervised multi-source DA setting which consists of K labeled source domains $\{\mathcal{S}_i\}_{i=1}^{K}$, where $\mathcal{S}_i = \left\{\left(x_j^{\mathcal{S}_i}, y_j^{\mathcal{S}_i}\right)\right\}_{j=1}^{|\mathcal{S}_i|}$ ($x_j^{\mathcal{S}_i}$ is the j-th instance of $\mathcal{S}_i$ with a label $y_j^{\mathcal{S}_i}$), and unlabeled target domain $\mathcal{T} = \left\{x_j^{\mathcal{T}}\right\}_{j=1}^{|\mathcal{T}|}$ ($x_j^{\mathcal{T}}$ is the j-th instance of $\mathcal{T}$). The objective is to learn a classifier M using labeled data from source domains and unlabeled data from the target domain so that (1) M produces accurate predictions on the target domain without fine-tuning (ZSL case), and (2) M generalizes better than SOTA methods on the target domain after partially or fully fine-tuning.

4 DOMAIN ADAPTATION FOR MATCHING ENTITIES

In this section, we introduce our proposed method DAME which is a DA-based method for matching entities. We first describe the architecture of DAME, and then present the DA-based training strategy to update the parameters of our proposed model. Finally, we present our fine-tuning strategy in the case of using labeled samples from the target domain to update DAME.

4.1 DAME architecture

There are multiple datasets that are available for the EM task. Therefore, our model is based on formulating the EM as a mixture of domain experts in the case of DA. Each expert model is trained on one source domain. We denote by f_{S_i}, the expert model that is trained on S_i. Training a mixture of experts and shared models improves the performance when multiple source domains are available as shown in prior works [13, 17, 41]. Therefore, we also add a global model g that is trained using all the source domains $\{\mathcal{S}_i\}_{i=1}^{K}$.

DCLM have been proposed in the DA setting to solve multiple tasks [14, 15, 23, 28, 41]. We propose to incorporate DCLM

in our DA-based model to solve the EM task. Each f_{S_i} and g are initialized using DistillBERT [30] which is a distilled version of BERT with fewer parameters. We choose to use DistilBERT as the main component for the expert and global models for two reasons. First, by incorporating DCLM, we compare records in their entirety which has been shown to be more effective than attribute-based comparisons. Second, DistilBERT has a reduced size and comparable performance to BERT, and our objective is to include many source domains while keeping the time and memory complexity reasonable. In general, our proposed model M has four modules:

$$M = N \circ Att \circ F \circ Rep \tag{1}$$

Rep is a representation module that produces the sequence input from a pair of records x, F is a feature extractor that produces multiple embeddings for the sequence input of the record pair x using expert models $\{f_{S_i}\}_{i=1}^{K}$ and the global model g, *Att* is an attention module that aggregates the embeddings of the expert models to produce the final multi-source embedding, and N is a classification layer that maps the final embedding to a confidence score to make a matching/non-matching decision on a record pair.

4.1.1 Representation module Rep. Each record pair $x = (e_1, e_2)$ is composed of two data entries $e_1 \in D_1$ and $e_2 \in D_2$ that correspond to candidate rows from two collections of data entries D_1 and D_2. **Both D_1 and D_2 are from the same source domain**. Each data entry $e_i = \{(\text{attr}_j, \text{val}_j)\}_{1 \le j \le C}$ is a set of attribute-value pairs denoted by $(\text{att}_j, \text{val}_j)$, where C is the number of attributes in each record. We follow the encoding of Ditto [20] for serializing data entries to produce a sequence for each record from the attribute-value pairs:

$$r_{e_i} = [\text{COL}]\text{attr}_1[\text{VAL}]\text{val}_1 \ldots [\text{COL}]\text{attr}_C[\text{VAL}]\text{val}_C \tag{2}$$

where [COL] and [VAL] are special tokens that denote the start of attributes and values, respectively. The input of EM is a pair of records $x = (e_1, e_2)$. So, *Rep* takes as input a pair of records, and produces a sequence pair of serialized entries that is given by:

$$Rep(x) = Rep((e_1, e_2)) = [\text{CLS}]r_{e_1}[\text{SEP}]r_{e_2}[\text{SEP}], \tag{3}$$

where [SEP] and [CLS] are BERT special tokens that are added into the sequence similar to the sentence pair classification setting.

4.1.2 Feature extractor F. We have $K + 1$ DistilBERT models: K expert models $\{f_{S_i}\}_{i=1}^{K}$ and a global shared model g. We use $Rep(x)$ as input to the $K + 1$ models to extract K source domain-based embeddings denoted by $f_{S_i}(Rep(x)), i = 1, \ldots, K$, and a global model-based embedding denoted by $g(Rep(x))$. The embeddings from the source domain models and the global model are extracted using the hidden state of the [CLS] token from the last Transformer block in each DistilBERT model. In conclusion, the output of F is given by:

$$F(Rep(x)) = \{f_{S_i}(Rep(x))\}_{i=1}^{K} \cup g(Rep(x)) \tag{4}$$

4.1.3 Attention module Att. When aggregating the embeddings that are extracted using F, the embeddings from the source domains and the global model should not be treated equally as there are domains that are more relevant to a given record pair x than others. We use a parameterized attention model that attends to all domains using a dot product-based attention where three parametric matrices are introduced: a query matrix $Q \in \mathbb{R}^{d \times d}$, a key matrix $K_e \in \mathbb{R}^{d \times d}$, and a value matrix $V \in \mathbb{R}^{d \times d}$, where d is the dimension of the embedding. We first concatenate all the expert embeddings from $F(Rep(x))$ to form an embedding matrix denoted by $E \in \mathbb{R}^{K \times d}$. The attention operations are defined by:

$$\begin{aligned} &\alpha = g(Rep(x))^T Q \in \mathbb{R}^{1 \times d} \\ &\mathcal{K} = EK_e \in \mathbb{R}^{K \times d} \\ &\mathcal{V} = EV \in \mathbb{R}^{K \times d} \\ &Att(Rep(x), Q, K, V) = \text{softmax}\left(\frac{\alpha \mathcal{K}^T}{\sqrt{d}}\right)\mathcal{V} \in \mathbb{R}^{1 \times d} \end{aligned} \tag{5}$$

An important design choice in our attention module *Att* is the use of the global representation $g(Rep(x))$ to map the query matrix Q to a query vector α. Given that the global model is trained on all the source domains, we expect the global model's embedding to transfer to the target domain, and by consequence we obtain more accurate attention weights in the target domain to aggregate the source domains, mainly in the zero-shot learning case. The output of the attention module is used as input to the classification layer N to predict the matching score of the input record pair x.

4.2 Training strategy

In the multi-source DA setting, we have K labeled source domains $\{\mathcal{S}_i\}_{i=1}^{K}$, where $\mathcal{S}_i = \left\{\left(x_j^{\mathcal{S}_i}, y_j^{\mathcal{S}_i}\right)\right\}_{j=1}^{|\mathcal{S}_i|}$, and an unlabeled target domain $\mathcal{T} = \left\{x_j^{\mathcal{T}}\right\}_{j=1}^{|\mathcal{T}|}$. Our training phase is based on the multi-task learning setting. In each batch for the training phase, we sample B pairs of records $X_j = (x_1^{\mathcal{S}_j}, y_1^{\mathcal{S}_j}), (x_2^{\mathcal{S}_j}, y_2^{\mathcal{S}_j}), \ldots, (x_B^{\mathcal{S}_j}, y_B^{\mathcal{S}_j})$ from a given source S_j. Our loss function $\mathcal{L}$ is composed of four parts and is given by:

$$\mathcal{L}(X_j) = \lambda_1 \mathcal{L}_1(X_j) + \lambda_2 \mathcal{L}_2(X_j) + \lambda_3 \mathcal{L}_3(X_j) + \lambda_4 \mathcal{L}_4(X_j) \tag{6}$$

where λ_1, λ_2, λ_3, and λ_4 are hyperparameters that control the contribution of each loss to the final loss function $\mathcal{L}$; each of $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_4$ represents a task-specific loss.

4.2.1 Expert domain loss $\mathcal{L}_1$. f_{S_i} represents the expert model of S_i, for all $i \in 1, 2, \ldots, K$. To optimize each expert model f_{S_i}, we add a classification layer N_{S_i} that predicts the probabilities of matches and non-matches for each domain S_i. So, in total we add K classification layers. Given that X_j is sampled from the j-th domain, the domain expert loss $\mathcal{L}_1$ is given by:

$$\mathcal{L}_1(X_j) = \frac{1}{B}\sum_{l=1}^{B} CrossEnt(N_{S_j}(f_{S_j}(Rep(x_l^{\mathcal{S}_j}))), y_l^{\mathcal{S}_j}) \tag{7}$$

where *CrossEnt* denotes the cross entropy loss function.

4.2.2 Global model loss $\mathcal{L}_2$. The global model is trained on all the source domains in order to learn a universal embedding for the EM task that supports transfer to the target domain while maintaining important matching signals for each source domain. In addition, the embedding of the global model is multiplied with the query matrix Q in the attention module *Att* to compute the contribution of each source domain to the final representation. After learning how to aggregate features in the training phase on source domains, the global model guides the attention module *Att* to pick the most important source domains for the target domain during the testing phase. To optimize the global model g, we add a classification layer

Table 1: Datasets for our experiments.

Dataset	Domain	Size	% matches	nb attributes
Shoes	clothing	5,805	21.95	1
Cameras	electronics	5,255	22.03	1
Computers	electronics	8,094	22.42	1
Watches	electronics	6,413	22.85	1
DBLP-GoogleScholar	citation	28,707	18.62	4
DBLP-ACM	citation	12,363	17.95	4
Fodors-Zagats	restaurant	946	11.62	6
Beer	product	450	15.11	4
iTunes-Amazon	music	539	24.48	8
Abt-Buy	product	9,575	10.73	3
Amazon-Google	software	11,460	10.18	3
Walmart-Amazon	electronics	10,242	9.39	5

N_g that predicts the probabilities of matches and non-matches for all source domains. The global model loss $\mathcal{L}_2$ is given by:

$$\mathcal{L}_2(X_j) = \frac{1}{B}\sum_{l=1}^{B} CrossEnt(N_g(f_g(Rep(x_l^{\mathcal{S}_j}))), y_l^{\mathcal{S}_j}) \quad (8)$$

4.2.3 Meta-target loss $\mathcal{L}_3$. In DA, the objective is to incorporate multiple source domains to predict labels for samples from the target domain during the testing phase. In order to simulate the process of DA during the training phase, we use the meta-target and meta-sources similar to Guo et al. [13]. Given that X_j is sampled from the j-th domain, the meta-target is the j-th source domain and the meta-sources are $\{\mathcal{S}_i\}_{i=1,i\neq j}^{K}$. The meta-model M_{S_j} differs only on the feature extractor part F_{S_j} compared to M. M_{S_j} is given by:

$$M_{S_j} = N \circ Att \circ F_{S_j} \circ Rep \quad (9)$$

where:

$$F_{S_j}(Rep(x)) = \left\{f_{S_i}(Rep(x))\right\}_{i=1,i\neq j}^{K} \cup g(Rep(x)) \quad (10)$$

The same attention module Att is applicable to the output of the meta-feature extractor F_{S_j} where the query matrix based on the global model attends to all the expert embeddings in the key matrix regardless of the number of expert models. Finally, the meta-target loss $\mathcal{L}_3$ for the batch X_j is given by:

$$\mathcal{L}_3(X_j) = \frac{1}{B}\sum_{l=1}^{B} CrossEnt(M_{S_j}(x_l^{\mathcal{S}_j}), y_l^{\mathcal{S}_j}) \quad (11)$$

4.2.4 Adversarial loss $\mathcal{L}_4$. The global model g plays an important role in the attention module Att. Learning a domain invariant embedding from the global model makes the transfer to the target domain smoother as the attention weights should be more accurate. To obtain a domain invariant representation from g, we adapt the domain adversarial training for EM. Similar to the generative adversarial network (GAN), a min-max objective function is introduced to optimize the parameters of the generator which is the global model g and the discriminator denoted by D. The parameters of D are optimized to predict the domain of a sample x using $g(Rep(x))$, and the parameters of g are optimized to produce a confusing representation $g(Rep(x))$ for D. We alternate between updating D and g. Given that X_j is sampled from the j-th domain, in order to update D, we minimize $\mathcal{L}_D$ which is given by:

$$\mathcal{L}_D(X_j) = \frac{1}{B}\sum_{l=1}^{B} CrossEnt(D(f_g(Rep(x_l^{\mathcal{S}_j}))), j) \quad (12)$$

$\mathcal{L}_D$ is minimized with respect to only the parameters of D. Then, we set $\mathcal{L}_4(X_j) = -\mathcal{L}_D(X_j)$ to update the parameters of g when minimizing $\mathcal{L}$ (D parameters are fixed). Unlabeled samples $\mathcal{T} = \left\{x_j^{\mathcal{T}}\right\}_{j=1}^{|\mathcal{T}|}$ from the target domain can also be considered as an additional domain when updating the parameters of D and g by alternating between minimizing $\mathcal{L}_D$ and $-\mathcal{L}_D$, respectively. In this case, the total number of labels that are used in $\mathcal{L}_D$ is equal to $K+1$.

4.3 Fine-tuning DAME on the target domain

During fine-tuning DAME on the target domain, we only update the weights of the global model g, attention weights Att, and the classification layer N, and we keep the weights of the expert models $f_{S_1}, f_{S_2}, \ldots, f_{S_K}$ frozen. The objective of the fine-tuning step is to slightly update the parameters of DAME to incorporate dataset-specific signals related to the target domain without changing the parameters of expert models. There are multiple fine-tuning scenarios on the target domain. First, we can use all the samples from the target domain or only a limited budget of samples for fine-tuning. Second, in the case of having access to only a limited budget of samples, we can randomly choose samples, or adapt active learning (AL) selection strategies to select the most promising samples. We experiment with all the scenarios and produce AL results using methods from [12, 31, 40].

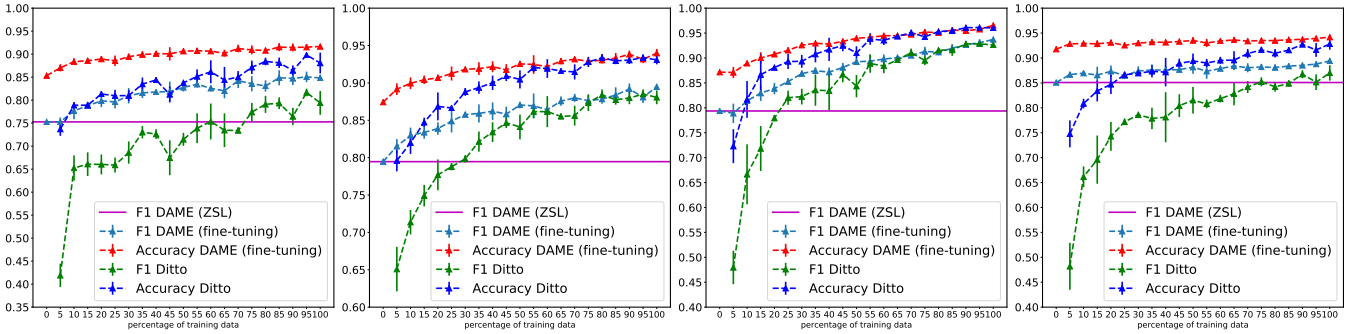
5 EVALUATION

5.1 Data collections

Table 1 represents all the 12 datasets that we use in our experiments. Datasets are collected from the entity resolution Benchmark datasets [19] and the Magellan data repository [18]. These datasets cover multiple domains including clothing, electronics, citation, restaurant, products, music, and software. Each dataset is composed of candidate pairs of records from two structured tables that have the same set of attributes. The datasets vary in the size and this simulates real-world scenarios where there are some domains that are more frequent than others. The total number of attributes in all datasets ranges from 1 to 8. The rate of matches in all datasets ranges from 9.39% to 24.48%. Clearly, there is a class imbalance in all datasets where the non-matching class is significantly larger than the matching class. Each dataset is split into training, validation, and testing, and we use the same pre-splited datasets in Ditto [20].

5.2 Baselines

We compare the performance of our proposed model against the best performing method in the category of attribute-level comparators which is DeepMatcher [24] (the previous SOTA), and the SOTA in EM which is Ditto [20]. We are interested in two aspects of our proposed model DAME. First, we evaluate the ZSL case for DAME by comparing the performance to baselines that are trained on different percentages of training data. Second, we compare the results



(a) Shoes (b) Computers (c) Watches (d) Cameras

Figure 1: Comparison of DAME results against Ditto for datasets with similar structures (Shoes, Computers, Watches, and Cameras). The plots report two evaluation metrics: F1 score and accuracy. In all figures, the light blue plot represents the F1 score of DAME, and is compared against the green plot that represents the F1 score of Ditto; the red plot represents the accuracy of DAME, and is compared against the blue plot that represents the accuracy of Ditto; the magenta color represents the F1 score of the ZSL for the target domain, which is equivalent to 0% of supervised training data from the target domain.

Method Name	Precision	Recall	F1	Accuracy
DeepMatcher [24]	**0.9489**	0.9373	0.9431	0.9789
Ditto [20]	0.9358	0.9542	0.9449	0.9793
DAME (ZSL)	0.9098	0.8579	0.8831	0.9576
DAME (full training data)	0.9354	**0.9719**	**0.9533**	**0.9850**

(a) DBLP-GoogleScholar

Method Name	Precision	Recall	F1	Accuracy
DeepMatcher [24]	0.9855	0.9869	0.9861	0.9945
Ditto [20]	**0.9865**	0.9865	0.9865	0.9951
DAME (ZSL)	0.8769	**0.9954**	0.9324	0.9741
DAME (full training data)	**0.9865**	**0.9954**	**0.9909**	**0.9971**

(b) DBLP-ACM

Table 2: DA results for EM using datasets with similar structures. (a) the target dataset is DBLP-GoogleScholar and the source dataset is DBLP-ACM; (b) the target dataset is DBLP-ACM and the source dataset is DBLP-GoogleScholar.

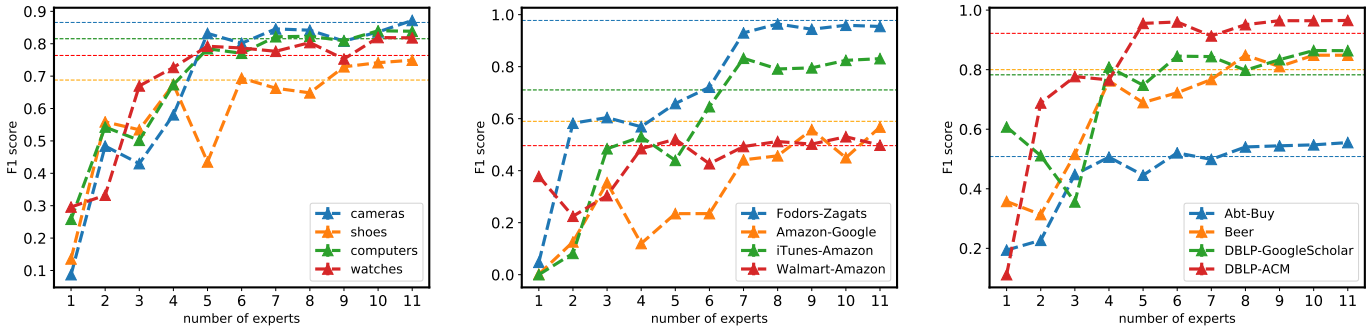
of fine-tuning DAME on the target domain against training the baselines on the target domain.

5.3 Experimental Setup

We evaluate the performance of DAME and baselines on the EM task using precision, recall, F1-score, and accuracy of predictions on the testing set. We use †, and ‡ to denote that the difference in a given evaluation metric between Ditto trained on 50% of data and DAME (ZSL) is less than 0.15, and less than 0.1, respectively. We use § to denote that either the difference between Ditto trained on 50% of data and DAME (ZSL) is less than 0.05 or DAME (ZSL) is better than Ditto trained on 50% of data. DAME is trained for 3 epochs on the source domains. We compare fine-tuning results for DAME and baselines after training for 10 epochs on the same percentage of training data from the target domain. The hyperparameters λ_1, λ_2, λ_3, and λ_4 are fine-tuned for one dataset and then kept the same for all the experiments. We distinguish 3 sets of experiments based on the structure of datasets. The first set of experiments studies DA for Shoes, Cameras, Computers, and Watches. These datasets have a unique attribute which is *title*. The second set of experiments also studies DA for datasets that have similar structures which are DBLP-GoogleScholar and DBLP-ACM. The set of attributes for these two datasets are *title*, *authors*, *venue*, and *year*. The third set of experiments is related to *DA in the wild* where we study DA using all 12 datasets regardless of the structures and domains.

5.4 Experimental results

5.4.1 DA for Shoes, Computers, Watches, Cameras. Figure 1 shows the comparison of DAME results against Ditto for Shoes, Computers, Watches, and Cameras. The caption of each subfigure represents the target domain, and the remaining 3 domains represent the source domains. Each data point represents the mean of 5 trials, and the vertical line in each data point represents the standard deviation (std). The plots report two evaluation metrics: F1 score and accuracy. In all figures, the light blue plot represents the F1 score of DAME, and is compared against the green plot that represents the F1 score of Ditto; the red plot represents the accuracy of DAME, and is compared against the blue plot that represents the accuracy of Ditto. DAME and Ditto outperform DeepMatcher for all evaluation metrics by a large margin, so that we only include DAME and Ditto results to avoid clutter in the figures. The magenta color represents the F1 score of the DAME (ZSL) for the target domain, which is equivalent to 0% of supervised training data from the target domain. We achieve high F1 scores for DAME (ZSL) for both Shoes and Cameras datasets, where the F1 score for DAME (ZSL) is equivalent to training Ditto on 72% and 85% of training data for the Shoes and Cameras, respectively. The results are lower for Computers and Watches where the F1 score of DAME (ZSL) is equivalent to Ditto trained on around 25% of training data. Figure 1 shows the results of fine-tuning DAME using different percentages of training data. Fine-tuning DAME leads to a better and more stable (smaller std in



(a) (b) (c)

Figure 2: Comparison of F1 score results with different numbers of expert domains against using global model representation during testing phase on the target domain.

Table 3: F1 results for AL after DA.

Method	Shoes	Computers	Watches	Cameras
DAME (ZSL)	0.7527	0.7946	0.7936	0.8507
DAME (full training data)	0.8483	0.8947	0.9371	0.8941
Random Sampling (5%)	0.7527	0.8181	0.8004	0.8664
Least Confidence [40] (5%)	0.7818	0.8402	0.8209	0.8745
Entropy Sampling [40] (5%)	0.7859	0.8464	0.8166	0.8748
USDE [12] (5%)	**0.7877**	0.8437	0.8151	**0.8775**
BALD [12] (5%)	0.7852	**0.8472**	**0.8313**	0.8705
K-Centers Greedy [31] (5%)	0.7674	0.8271	0.8206	0.8687
K-Means [31] (5%)	0.7527	0.8042	0.8097	0.8596
Core-Set [31] (5%)	0.7621	0.8304	0.8168	0.8734
Random Sampling (25%)	0.8120	0.8418	0.8528	0.8741
Least Confidence [40] (25%)	0.8228	0.8804	0.8677	0.8888
Entropy Sampling [40] (25%)	0.8207	0.8770	0.8740	0.8925
USDE (25%) [12]	**0.8286**	0.8741	0.8688	0.8842
BALD (25%) [12]	0.8247	**0.8835**	**0.8872**	**0.8941**
K-Centers Greedy [31] (25%)	0.8155	0.8771	0.8869	0.8780
K-Means [31] (25%)	0.8057	0.8658	0.8694	0.8737
Core-Set [31] (25%)	0.8161	0.8696	0.8812	0.8776

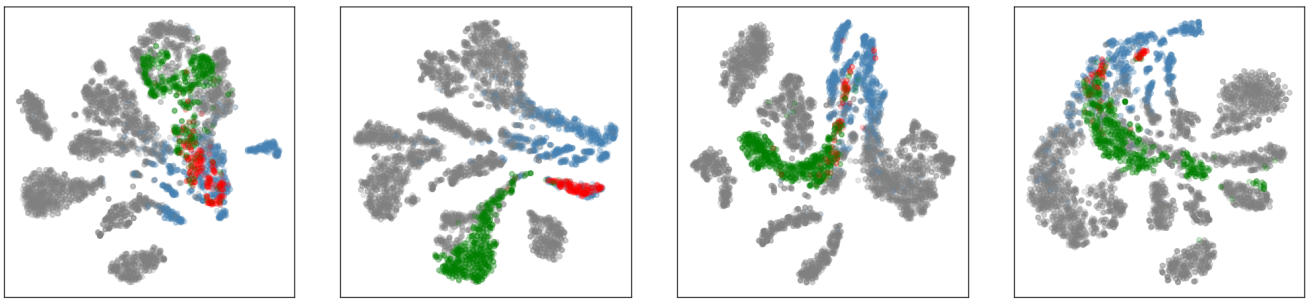
most fractions of the training data) performance than Ditto for all datasets which means that DAME generalizes better than existing methods in EM for datasets with similar structures. This can be explained by the important role of DA in learning the task so that the weights are better warmed up for EM.

5.4.2 DA for DBLP-GoogleScholar, DBLP-ACM. Table 2 summarizes the performance of different approaches on the second set of datasets with the same structure which is composed of DBLP-GoogleScholar and DBLP-ACM. In this case, we have one target dataset and one source dataset. We achieve high results for DAME (ZSL) for both datasets. In addition, fine-tuning DAME slightly increases the F1 and accuracy for both datasets. So, consistent with the first set of experiments, we conclude that DAME transfers the task knowledge from the source domains to a target domain in the case of datasets with similar structures.

5.4.3 DA in the wild. We study the case of transferring knowledge between datasets with different domains and structures. We call this setting DA in the wild which simulates real-world scenarios. Table 4 (end of the paper) shows extensive experiments on 12 datasets reporting evaluation metrics for multiple methods. DAME (ZSL) achieves a better F1 score than DeepMatcher trained with 50% of training data from the target domain for 7 out of 12 datasets. The difference between the F1 score of Ditto trained on 50% of data and DAME (ZSL) is less than 0.1, and 0.05 for 83% and 41% of datasets, respectively. By comparing the F1 score of fine-tuning all methods using 50% of training data from the target domain, we achieve SOTA results for 10 out of 12 datasets. By comparing the F1 score of fine-tuning all methods using all training data from the target domain, we achieve SOTA results for 10 out of 12 datasets. This means that DAME generalizes better than existing methods for datasets in the wild.

5.4.4 Expert models vs Global model. Figure 2 shows the comparison of F1 score results with different numbers of expert domains against using the global model representation during the testing phase on the target domain in the case of ZSL. The x-axis represents the number of experts that we use for predictions. For example, if the number of experts is equal to 6, it means that we randomly choose 6 experts and we drop the remaining 5 experts. Each data point in Figure 2 represents an average of 5 trials. The dashed line represents the F1 score for the global model. For 10 out of 12 datasets, combining multiple experts using the attention network *Att* leads to better results than the global model. Figure 2 shows that the fewest number of experts needed to outperform the global model was 5 (DBLP-ACM); the most required was 11 (Cameras). Overall, we obtain better F1 scores for the mixture of experts when we increase the number of experts. This means that the experts help to better understand the EM task, and therefore transfer the learned task knowledge to the unseen target domain.

5.4.5 DAME with Active learning. So far, we have discussed the performance of fine-tuning DAME using randomly selected samples from the target domain. To improve the results of fine-tuning our model, we investigate multiple AL selection techniques given a limited budget of labeled instances. Table 3 shows the results of multiple AL selection methods applied to the DAME (ZSL) model. The starting point is our DA-based model which is not fine-tuned on the target domain, and the best performance corresponds to DAME fine-tuned on all training data from the target domain. We



(a) Computers (b) DBLP-ACM (c) Amazon-Google (d) Walmart-Amazon

Figure 3: The t-SNE visualization of the final embeddings for the target and source domains after DA in the wild (ZSL case). The gray and blue colors represent randomly selected data points from the source domains with label 0 and label 1, respectively; The green and red colors represent randomly selected data points from the testing set of the target domain with label 0 and label 1, respectively. 12 domains are used in each experiment, where the caption of each subfigure represents the target domain, and the 11 remaining datasets represent the source domains.

report results using two budget levels: 5% and 25% of the training data from the target domain. The simplest baseline is Random Sampling. The remaining baselines can be categorized into two groups: the confidence-based baselines which are: Least Confidence [40], Entropy Sampling [40], Uncertainty Sampling with Dropout Estimation (USDE) [12], and Bayesian Active Learning Disagreement (BALD) [12]; and the embedding-based baselines which are K-Centers Greedy [31], K-Means [31], and Core-Set [31]. The selection of samples in the first group is based on the confidence scores of the training data from the target domain that are computed using the DAME (ZSL) model. For example, for a budget of b samples, Least Confidence corresponds to the top b samples with the lowest confidence level. Multiple predictions for a given sample are needed for USDE and BALD to compute the uncertainty functions, and we obtain these different predictions by activating the dropout layers during the inference phase on the target domain. The second group is based on the embeddings of samples from the target domain that are obtained using the DAME (ZSL) model. Clustering of the input space is then applied to determine centers of clusters or core sets. Table 3 shows that the confidence-based methods lead to better results than the embedding-based methods. In particular, when we select 25% of samples using the BALD method for the Cameras dataset, we achieve the same F1 score of a fully fine-tuned DAME model using all training data from the target domain. This indicates that the predictions from the classification layer N of our model M accurately reflect the data points where DA was unsuccessful. Therefore, by fine-tuning on these samples from the training data, our model generalizes better on the testing set of the target domain.

5.4.6 Visualization. We show the embedding of DAME in the case of ZSL. Figure 3 shows the t-SNE visualization of the final embeddings for the target and source domains after DA in the wild (ZSL case). We only show the embeddings of four domains due to the space limitation in the paper, but we notice similar patterns for all the datasets. The gray and blue colors represent randomly selected data points from the source domains with a label 0 and label 1, respectively; the green and red colors represent randomly selected data points from the testing set of the target domain with a label 0 and label 1, respectively. 12 domains are used in each experiment, where the caption of each subfigure in Figure 3 represents the target domain, and the 11 remaining datasets represent the source domains. The best case is to have a mixture of blue and red dots which represent the matching class for the source and target domains, respectively, and a mixture of gray and green dots which represent the non-matching class for the source and target domains, respectively. This means that we transfer the task knowledge from sources to the target domain for both labels. For example, for Computers and DBLP-ACM, we obtain embeddings that respect the matching and non-matching classes as shown in Figure 3 (a) and (b), respectively. On the other hand, for Amazon-Google and Walmart-Amazon, there are green dots that are closer to the blue dots than the gray dots as shown in Figure 3 (c) and (d), respectively, and this leads to incorrect predictions for DAME (ZSL).

6 CONCLUSIONS

We have shown that our proposed model transfers learning from multiple source domains to an unseen target domain in the EM task. We formulate the EM task as a mixture of experts that capture task-specific knowledge from pretraining on multiple source domains and testing on a target domain. We evaluate DAME in two aspects. First, we study the ZSL case on the target domain and demonstrate that DAME learns the EM task and transfers knowledge to the target domain. Second, we study fine-tuning DAME on the target domain and demonstrate that DAME generalizes better than SOTA methods for most of the datasets. We showed that our results hold in two scenarios which are EM for datasets with similar structures and EM in the wild. Our experimental section contains extensive experiments over 12 datasets with different domains, sizes and structures. In addition, we showed the importance of selecting a specific set of samples in the fine-tuning of the target domain by studying AL methods with limited budget. Future work includes extending our model to pairs of records with different sets of attributes, and enriching our DA-based model with external knowledge, such as knowledge graphs, to better understand the EM task and therefore transfer more knowledge to the target domain.

Table 4: DA results for EM in the wild.

Target dataset	Method	Precision	Recall	F1	Accuracy
Fodors-Zagats	DAME (ZSL)	0.9565§	1.0000§	0.9777§	0.9947§
	DeepMatcher[24] (50% training data)	0.9360±0.0559	0.8333±0.0428	0.8801±0.0334	0.9735±0.0074
	Ditto [20] (50% training data)	**1.0000**	0.9545	0.9767	**0.9947**
	DAME (50% training data)	0.9565	**1.0000**	**0.9777**	**0.9947**
	DeepMatcher[24] (full training data)	0.9092±0.0756	0.9848±0.0214	0.9437±0.0423	0.9858±0.0108
	Ditto [20] (full training data)	**1.0000**	0.9545	0.9767	0.9947
	DAME (full training data)	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Beer	DAME (ZSL)	0.7368§	1.000§	0.8484§	0.9450§
	DeepMatcher[24] (50% training data)	**0.8095±0.0673**	0.4047±0.0336	0.5396±0.0448	0.8937±0.0103
	Ditto [20] (50% training data)	0.7211±0.0288	0.6428	0.6794±0.0128	0.9065±0.0054
	DAME (50% training data)	0.7801 ±0.0433	**1.000**	**0.8758±0.0273**	**0.9560±0.0109**
	DeepMatcher[24] (full training data)	0.8183±0.0581	0.7142±0.0583	0.7588±0.0193	0.9304±0.0051
	Ditto [20] (full training data)	**0.8174±0.0396**	0.9285±0.0714	0.8660±0.0089	**0.9560**
	DAME (full training data)	0.7801	**1.000**	**0.8758**	**0.9560**
iTunes-Amazon	DAME (ZSL)	0.6750	1.000§	0.8059‡	0.8807‡
	DeepMatcher[24] (50% training data)	0.9005±0.0226	0.7901±0.0698	0.8406±0.0464	0.9266±0.0198
	Ditto [20] (50% training data)	0.8685±0.0114	0.8518±0.0370	0.8594±0.0132	0.9311±0.0045
	DAME (50% training data)	**0.9333±0.0666**	**0.9629**	**0.9467±0.0344**	**0.9724±0.0183**
	DeepMatcher[24] (full training data)	0.9139±0.0149	0.9135±0.0174	0.9135±0.0088	0.9571±0.0043
	Ditto [20] (full training data)	0.9282±0.0317	0.9259±0.0370	0.9258±0.0027	0.9633
	DAME (full training data)	**0.9807±0.0192**	**0.9259**	**0.9524±0.0090**	**0.9770±0.0045**
Abt-Buy	DAME (ZSL)	0.4545	0.6796†	0.5447	0.8778‡
	DeepMatcher[24] (50% training data)	0.6978±0.0416	0.5355±0.0397	0.6033±0.0112	0.9244±0.0021
	Ditto [20] (50% training data)	0.7916±0.0312	0.7839±0.0169	0.7870±0.0069	0.9543±0.0028
	DAME (50% training data)	**0.7960±0.0078**	**0.7864±0.0097**	**0.7911±0.0088**	**0.9553±0.0018**
	DeepMatcher[24] (full training data)	0.7382±0.0214	0.6181±0.0127	0.6725±0.0082	0.9352±0.0022
	Ditto [20] (full training data)	**0.9206±0.0095**	0.7864±0.0097	**0.8481±0.0015**	**0.9697**
	DAME (full training data)	0.8243±0.0252	**0.8592±0.0097**	0.8410±0.0084	0.9650±0.0026
Amazon-Google	DAME (ZSL)	0.5431	0.6453§	0.5898‡	0.9084§
	DeepMatcher[24] (50% training data)	0.5623±0.0395	0.5327±0.0708	0.5416±0.0161	0.9085±0.0063
	Ditto [20] (50% training data)	**0.7055±0.0037**	0.6709±0.017	**0.6877±0.0107**	**0.9378±0.0015**
	DAME (50% training data)	0.6339±0.0327	**0.7435±0.0683**	0.6809±0.0099	0.9291±0.0032
	DeepMatcher[24] (full training data)	0.7002±0.0281	0.6011±0.0344	0.6454±0.0082	0.9326±0.0017
	Ditto [20] (full training data)	0.6709±0.0077	**0.8098±0.0064**	0.7338±0.0020	0.9400±0.0010
	DAME (full training data)	**0.7046±0.0038**	0.7692±0.0213	**0.7353±0.0076**	**0.9435±0.0006**
Shoes	DAME (ZSL)	0.6798§	0.8135§	0.7407§	0.8450§
	DeepMatcher[24] (50% training data)	0.6346±0.0250	0.7163±0.0303	0.6719±0.0041	0.8096±0.0082
	Ditto [20] (50% training data)	0.7137±0.0240	0.7559±0.0881	0.7301±0.0290	0.8496±0.0046
	DAME (50% training data)	**0.8234±0.0189**	**0.8423±0.0084**	**0.8325±0.0055**	**0.9077±0.0046**
	DeepMatcher[24] (full training data)	0.6908±0.0366	0.7988±0.0162	0.7400±0.0179	0.8468±0.0158
	Ditto [20] (full training data)	0.7569±0.0377	0.8389±0.0118	0.7950±0.0155	0.8819±0.0129
	DAME (full training data)	**0.8421±0.0222**	**0.8796±0.0152**	**0.8600±0.0043**	**0.9220±0.0041**
Computers	DAME (ZSL)	0.7957§	0.8729§	0.8325§	0.9043§
	DeepMatcher[24] (50% training data)	0.5762±0.0239	0.7547±0.0536	0.6529±0.0315	0.7820±0.0169
	Ditto [20] (50% training data)	0.8020±0.0085	**0.9080±0.0083**	0.8517±0.0085	0.9139±0.0050
	DAME (50% training data)	**0.8303±0.0268**	0.9063±0.0234	**0.8659±0.0039**	**0.9234±0.0045**
	DeepMatcher[24] (full training data)	0.7002±0.0258	0.8350±0.0356	0.7614±0.0270	0.8576±0.0157
	Ditto [20] (full training data)	**0.8682**	0.9147±0.0117	0.8908±0.0055	0.9389±0.0027
	DAME (full training data)	0.8630±0.0076	**0.9264±0.0033**	**0.8935±0.0025**	**0.9398±0.0018**
Watches	DAME (ZSL)	0.7267†	0.9124§	0.8090‡	0.8834‡
	DeepMatcher[24] (50% training data)	0.6997±0.0260	0.7274±0.0478	0.7126±0.0314	0.8415±0.0151
	Ditto [20] (50% training data)	0.8664±0.0037	0.8996±0.0054	0.8827±0.0045	0.9352±0.0024
	DAME (50% training data)	**0.8691±0.0196**	**0.9160±0.0109**	**0.8917±0.0051**	**0.9397±0.0039**
	DeepMatcher[24] (full training data)	0.7771±0.0093	0.8309±0.0169	0.8030±0.0087	0.8896±0.0044
	Ditto [20] (full training data)	**0.9145±0.0030**	0.9178±0.0164	0.9161±0.0097	0.9545±0.0049
	DAME (full training data)	0.9010±0.0038	**0.9470±0.0091**	**0.9234±0.0023**	**0.9575±0.0009**
Cameras	DAME (ZSL)	0.8376§	0.8958§	0.8657§	0.9243§
	DeepMatcher[24] (50% training data)	0.5896±0.0063	0.6863±0.0630	0.6328±0.0275	0.7842±0.0058
	Ditto [20] (50% training data)	0.7585± 0.0694	0.8628±0.0607	0.8020±0.0127	0.8831±0.0175
	DAME (50% training data)	**0.8801 ±0.0312**	**0.8871±0.0295**	**0.8825±0.0011**	**0.9356±0.0028**
	DeepMatcher[24] (full training data)	0.6986±0.0280	0.7847±0.0075	0.7388±0.0159	0.8486±0.0124
	Ditto [20] (full training data)	0.8573±0.0075	0.9062±0.0173	0.8809±0.0042	0.9333±0.0014
	DAME (full training data)	**0.8917±0.0013**	**0.9070±0.0017**	**0.8963±0.0001**	**0.9432**
Walmart-Amazon	DAME (ZSL)	0.3558	0.9015§	0.5102	0.8369†
	DeepMatcher[24] (50% training data)	0.6938±0.0171	0.5474±0.0217	0.6118±0.0167	0.9346±0.0023
	Ditto [20] (50% training data)	**0.8501±0.0206**	0.7098±0.0466	0.7721±0.0191	0.9607±0.0017
	DAME (50% training data)	0.8082±0.0019	**0.8083±0.0103**	**0.8082±0.0061**	**0.9638±0.0009**
	DeepMatcher[24] (full training data)	0.6971±0.0183	0.6010±0.0223	0.6448±0.0067	0.9376±0.0011
	Ditto [20] (full training data)	**0.8883±0.0459**	0.7694±0.0336	**0.8227±0.0004**	**0.9687±0.0014**
	DAME (full training data)	0.8615±0.0090	**0.7875±0.0207**	0.8226±0.0071	0.9680±0.0007
DBLP-GoogleScholar	DAME (ZSL)	0.9077§	0.8490‡	0.8737‡	0.9499§
	DeepMatcher[24] (50% training data)	0.9347±0.0034	0.9439±0.0074	0.9385±0.0019	0.9770±0.0006
	Ditto [20] (50% training data)	0.9356±0.0030	**0.9448±0.0065**	0.9385±0.0016	0.9771±0.0005
	DAME (50% training data)	**0.9367±0.0019**	0.9411±0.0037	**0.9389±0.0028**	**0.9771±0.0010**
	DeepMatcher[24] (full training data)	**0.9489±0.0014**	0.9373±0.0018	0.9431±0.0016	0.9789±0.0006
	Ditto [20] (full training data)	0.9358±0.0025	**0.9542±0.0009**	0.9449±0.0008	0.9793±0.0003
	DAME (full training data)	0.9392±0.0023	0.9537±0.0032	**0.9464±0.0003**	**0.9798**
DBLP-ACM	DAME (ZSL)	0.8661†	0.9854§	0.9219‡	0.9651§
	DeepMatcher[24] (50% training data)	0.9787±0.0098	0.9763±0.0056	0.9774±0.0020	0.9919±0.0008
	Ditto [20] (50% training data)	**0.9865±0.0066**	**0.9865±0.0011**	**0.9865±0.0027**	**0.9951±0.0010**
	DAME (50% training data)	0.9787±0.0055	0.9831±0.0011	0.9809±0.0033	0.9931±0.0012
	DeepMatcher[24] (full training data)	0.9855±0.0056	0.9869±0.0022	0.9861±0.0039	0.9945±0.0014
	Ditto [20] (full training data)	**0.9865±0.0011**	0.9865±0.0022	0.9865±0.0016	0.9951±0.0006
	DAME (full training data)	0.9865±0.0032	**0.9868±0.0033**	**0.9866**	**0.9951**

REFERENCES

[1] Nils Barlaug and Jon Atle Gulla. 2021. Neural Networks for Entity Matching: A Survey. *ACM Trans. Knowl. Discov. Data* 15, 3 (2021), 52:1–52:37.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of Representations for Domain Adaptation. In *Advances in Neural Information Processing Systems 19*, Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (Eds.). MIT Press, 137–144.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML*, Vol. 382. ACM, 41–48.

[4] Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 39–48.

[5] Zhiyu Chen, Mohamed Trabelsi, Jeff Heflin, Yinan Xu, and Brian D. Davison. 2020. Table Search Using a Deep Contextualized Language Model. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 589–598.

[6] Peter Christen. 2008. Febrl -: an open source data cleaning, deduplication and record linkage system with a graphical user interface. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1065–1068.

[7] Peter Christen. 2012. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[9] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. *Proc. VLDB Endow.* 11, 11 (2018), 1454–1467.

[10] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Trans. Knowl. Data Eng.* 19, 1 (2007), 1–16.

[11] Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. End-to-End Multi-Perspective Matching for Entity Resolution. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. ijcai.org, 4961–4967.

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1183–1192.

[13] Jiang Guo, Darsh J. Shah, and Regina Barzilay. 2018. Multi-Source Domain Adaptation with Mixture of Experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 4694–4703.

[14] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. Association for Computational Linguistics, 8342–8360.

[15] Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 4237–4247.

[16] Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource Deep Entity Resolution with Transfer and Active Learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 5851–5861.

[17] Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain Attention with an Ensemble of Experts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*. 643–653.

[18] Pradap Konda, Sanjib Das, Paul Suganthan G. C., AnHai Doan, Adel Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeffrey F. Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. 2016. Magellan: Toward Building Entity Matching Management Systems. *Proc. VLDB Endow.* 9, 12 (2016), 1197–1208.

[19] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proc. VLDB Endow.* 3, 1 (2010), 484–493.

[20] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endow.* 14, 1 (2020), 50–60. https://doi.org/10.14778/3421424.3421431

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019).

[22] Jiaheng Lu, Chunbin Lin, Jin Wang, and Chen Li. 2019. Synergy of Database Techniques and Machine Learning Models for String Similarity Search and Join. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*. ACM, 2975–2976.

[23] Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain Adaptation with BERT-based Domain Classification and Data Selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP*. Association for Computational Linguistics, 76–83.

[24] 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 *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. ACM, 19–34.

[25] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33.

[26] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*. ACM, 751–760.

[27] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, 1-4, 2016*. The Association for Computational Linguistics, 2249–2255.

[28] Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020*. European Language Resources Association, 4933–4941.

[29] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. FAQ Retrieval Using Query-Question Similarity and BERT-Based Query-Answer Relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1113–1116.

[30] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108 (2019).

[31] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[32] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. [n.d.]. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18)*. 4058–4065.

[33] Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Trans. Knowl. Data Eng.* 27, 2 (2015), 443–460.

[34] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of Frustratingly Easy Domain Adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016*. AAAI Press, 2058–2065.

[35] Mohamed Trabelsi, Jin Cao, and Jeff Heflin. 2020. Semantic Labeling Using a Deep Contextualized Language Model. *CoRR* abs/2010.16037 (2020).

[36] Mohamed Trabelsi, Jin Cao, and Jeff Heflin. 2021. SeLaB: Semantic Labeling with BERT. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*. IEEE, 1–8.

[37] Mohamed Trabelsi, Zhiyu Chen, Brian D. Davison, and Jeff Heflin. 2021. Neural ranking models for document retrieval. *Inf. Retr. J.* 24, 6 (2021), 400–444.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*. 5998–6008.

[39] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 353–355.

[40] Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*. IEEE, 112–119.

[41] Dustin Wright and Isabelle Augenstein. 2020. Transformer Based Multi-Source Domain Adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Association for Computational Linguistics, 7963–7974.

[42] Yuan Zhang, Regina Barzilay, and Tommi S. Jaakkola. 2017. Aspect-augmented Adversarial Networks for Domain Adaptation. *Trans. Assoc. Comput. Linguistics* 5 (2017), 515–528.

[43] Chen Zhao and Yeye He. 2019. Auto-EM: End-to-end Fuzzy Entity-Matching using Pre-trained Deep Models and Transfer Learning. In *The World Wide Web Conference, WWW 2019*. ACM, 2413–2424.