Learning Transferable Feature Representations Using Neural Networks

Himanshu S. Bhatt*
AI Labs, American Express
Bengaluru, India
himanshu.s.bhatt@aexp.com

Shourya Roy*
AI Labs, American Express
Bengaluru, India
shourya.roy@aexp.com

Arun Rajkumar
IIT Madras
Chennai, India
arunr@cse.iitm.ac.in

Sriranjani Ramakrishnan
Conduent Labs
Bengaluru, India
sriranjani.r@conduent.com

Abstract

Learning representations such that the source and target distributions appear as similar as possible has benefited transfer learning tasks across several applications. Generally it requires labeled data from the source and only unlabeled data from the target to learn such representations. While these representations act like a bridge to transfer knowledge learned in the source to the target; they may lead to negative transfer when the source specific characteristics detract their ability to represent the target data. We present a novel neural network architecture to simultaneously learn a *two-part representation* which is based on the principle of segregating source specific representation from the common representation. The first part captures the source specific characteristics while the second part captures the truly common representation. Our architecture optimizes an objective function which acts adversarial for the source specific part if it contributes towards the cross-domain learning. We empirically show that two parts of the representation, in different arrangements, outperforms existing learning algorithms on the source learning as well as cross-domain tasks on multiple datasets.

1 Introduction

Unsupervised domain adaptation is a sub field of machine learning where one learns from annotated data in a source domain with the aim of performing well on non-annotated data in a target domain. This attractive feature wherein the data, distributions and tasks may vary across domains has led to the widespread use of domain adaptation algorithms in several real world applications. A typical domain adaptation algorithm is provided with annotated source data and non-annotated target data

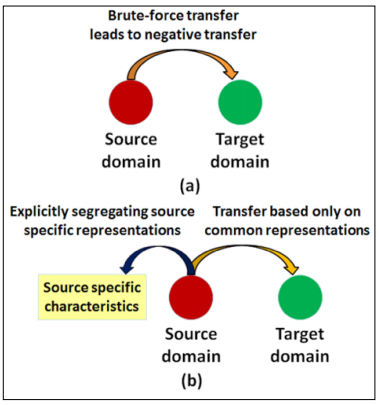

Figure 1: Illustrates the fundamental idea of learning transferable feature representations behind the proposed technique.

and it learns a 'common representation' where the source and target data distributions look similar. In this common representation, a model trained on the source data is expected to perform well on the target data as well.

While learning a common representation is useful for transferring knowledge from the source to the target domain, this may often lead to 'negative transfer' if we do not account for the fundamental question "what to transfer". It is observed that each domain has specific features that are highly discriminating only within a domain and contribute negatively if transferred across domains in a brute force manner (Pan and Yang, 2010), as shown in Figure 1. Traditional domain adaptation algorithms, being oblivious to such source specific characteristics, learn common representations which suffer from transfer loss as the source specific characteristics restrict their transferability. Moreover, it is also observed that the representation learned for domain adaptation optimizes for the performance in the target domain, often at the cost of source classification performance. While this can be justified for domain adaptation where

* Research done while working for Xerox Research Centre India.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4124–4134
Florence, Italy, July 28 - August 2, 2019.

the primary objective is maximizing the target performance, a technique that simultaneously sustains the source performance will always be preferred.

Our primary contribution is a novel neural network learning algorithm based on the principle of two-part hidden representation where individual parts can be disentangled or combined for learning tasks in different domains. We highlight some of the salient features of our algorithm:

- A novel technique for learning a two-part representation between domains. One comprising source specific and the other comprising common characteristics.
- The two-part representation behaves differently for different learning objectives:
  1. For the cross-domain task, explicitly learning the source specific representation and keeping them separate from common representation enhances the performance in the target domain.
  2. For the source learning task, the source specific and common units come together to sustain the source performance where the performance of most domain adaptation algorithms is compromised.

The proposed neural network architecture achieves this through an objective function which acts adversarial for a part of representation (source specific part) if it contributes to the cross-domain learning. Moreover, the proposed two-part representation learning approach also mitigates the possible effects of "negative transfer", as learning separate source specific and common representations evades the influence of source specific characteristics on the common representation.

2 Related Work

The problem of domain adaptation has gained a lot of attention due to its huge practical implications. Pan et al. (2010) focuses on learning a common representation minimizing the divergence between the source and target domains. Many body of work exists in literature including learning non-linear mappings (Daumé III, 2009; Pan et al., 2011; Blitzer et al., 2007; Pan et al., 2010; Barnes et al., 2018), mappings to mitigate domain divergence (Pan et al., 2010), common features (Dai et al., 2007; Dhillon et al., 2003), ensemble based approaches (Bhatt et al., 2015), subspace based methods (Gopalan et al., 2011; Gong et al., 2012; Harel and Mannor, 2010; Fernando et al., 2013) and neural networks based methods (Glorot et al., 2011; Chopra et al., 2013; Long and Wang, 2015; Tzeng et al., 2014).

A variant of unsupervised models namely marginalized stacked denoising autoencoders (mSDA) (Chen et al., 2012a) learn robust representation to input corruption noise, which is stable across changes in domains, allowing cross-domain transfer. Existing literature exploits the principle of representations generalizing across domains for classification, without labelled data from target ((Sarma et al., 2018), (Bhatt et al., 2016)) and with labelled data from target ((Zhang et al., 2018)). Our work emphasizes on domain discrimination by incorporating domain divergence and source risk minimization into the objective for learning better transferable representation without any labelled data from target domain. Another line of work aims to achieve distribution consistency between the source and target domains with linear data reconstruction such as co-regularization based augmented space (Kumar et al., 2010), coupled learning to link target-specific features to source features (Blitzer et al., 2011) and transfer of the source examples to the target and vice-versa (Zhou et al., 2016).

Domain adversarial neural networks (DANN) (Ajakan et al., 2014; Ganin et al., 2016), closely similar in philosophy to our work, learns a single representation by using an adversarial (Liu et al., 2017) gradient reversal component for domain divergence. In DANN, the entire hidden layer contributes unanimously towards the source classification and domain divergence objective. Unlike DANN, our approach segregates the hidden layer where the two components of hidden layer are treated differently for different objectives. Both the source specific and common parts contribute positively to the source classification objective. However, for the domain divergence objective, the common part contributes positively (i.e., tries to minimize divergence by maximizing the domain regressor's loss); whereas, the source specific part contributes negatively (i.e., tries to maximize divergence by minimizing domain regressor's loss)

Generative adversarial networks (GAN) (Goodfellow et al., 2014) build generative models to synthesize samples and falls closely in the same cat-

egory due to the similar method of measuring and minimizing the discrepancy between the feature distributions. The GAN model learns the representation in generative mode while our work is based on discriminative learning.

Domain separation networks (DSN) (Bousmalis et al., 2016) inspired from shared-space component analysis, explicitly and jointly models the domain-specific (private) and shared component domain representation. DSN is based on CNN and ours is a feed-forward network based on discriminating adversarial framework. The objective function of DSN has separate losses for difference, similarity, reconstruction and task-specific, while our approach follows min-max optimization criterion minimizing domain specific component loss and maximizing shared component loss.

Jiang & Zhai (2007) also proposed a two-stage approach for domain generalization and adaptation where first stage finds the generalizable feature representation across domains and its appropriate weights. The second stage picks up features useful for the target domain using semi-supervised learning. Their approach is a semi-supervised approach which uses labelled data from source and target domains along with linear classifiers. However, our framework is unsupervised (no labeled data from target) and leverages non-linear neural network classifier.

3 Problem Formulation

Let us consider a binary classification task where $\mathcal{X} \subseteq \mathbf{R}^n$ is the input space and $\mathcal{Y} = \{0, 1\}$ is the label space. We have two different distributions over $\mathcal{X} \times \mathcal{Y}$, called the source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$. We have labeled samples from source S drawn i.i.d from $\mathcal{D}_s$ and unlabeled samples from the target T drawn i.i.d. from $\mathcal{D}_t$.

$$S = \{(x_i^s, y_i^s)\}_{i=1}^m \sim (D_s)^m;$$
$$T = \{x_i^t\}_{i=1}^{m'} \sim (D_t)^{m'}$$

where m and m' are the number of labeled source and unlabeled target samples. Let $h(\cdot)$ be the D-dimensional hidden representation of the network which is further represented as $h(\cdot) = h_{ss}(\cdot) \bigoplus h_c(\cdot)$, where $h_{ss}(\cdot)$, $h_c(\cdot)$, $\bigoplus$ represent source specific, common representations and concatenation respectively. The neural network is parametrized by $\{\mathbf{W}, \mathbf{V}, b.c\}$. Our objective is to learn two parts of the hidden layer such that the source specific characteristics $h_{ss}(\cdot)$ do not detract the ability of common representation $h_c(\cdot)$ to generalize to the target task. Let W be the weight matrix between input and hidden units. W' & W'' be the weight matrix between the input units to the common and source specific units respectively. Let $o(\cdot)$ & $o'(\cdot)$ be the domain regressor for the common and source specific representations parametrized by $\{\mathbf{u}, d\}$ & $\{\mathbf{u}', d'\}$ respectively.

4 Proposed Neural Network Architecture

The proposed neural network is a fully connected architecture, as shown in Figure 2. The emphasis of our work, in contrast to most of the previous work, is not only on modeling the similarity between the domains but also on modeling the differences i.e., the domain specific information. We propose to achieve this by learning a two part hidden layer comprising the source specific part and the common part. The network tries to optimize two objectives - a classification objective and a domain divergence objective. The classification objective tries to minimize the mis-classifications in the labeled source data while the domain divergence objective attempts to learn a representation where both the source and target domain data appears close to each other. In our network, both the source specific part and the common part contribute positively to the source classification objective (i.e., minimize the mis-classification loss). However, for the domain divergence objective, the common part contributes positively (i.e., tries to minimize divergence) whereas the source specific part contributes negatively (i.e., tries to maximize divergence). Thus, the common representation acquires domain independence and generalizable classification abilities while the source specific representation remains domain-specific and highly discriminating for the in-domain classification task.

4.1 Learning in Source Domain

A neural network architecture with one hidden layer learns the function, $h : X \rightarrow \mathbb{R}^D$, to map the input to a D-dimensional representation:

$$h(x) = sigm(\mathbf{W}x + \mathbf{b}) \in \mathbb{R}^D$$

where $h(x) = h_{ss}(x) \bigoplus h_c(x)$ and $sigm(a)\left[\frac{1}{1+\exp(-a_i)}\right]_{i=1}^{|a|}$ is parametrized by a matrix-vector pair $(\mathbf{W}, \mathbf{b}) \in \mathbb{R}^{D \times n} \times \mathbb{R}^D$.

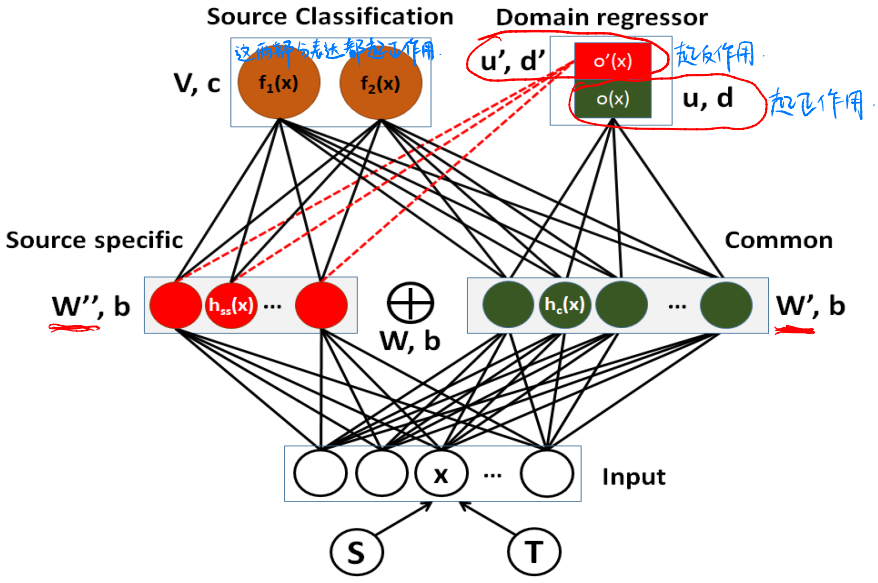

Figure 2: Illustrates the architecture to simultaneously learn the common and source specific representations.

For source classification, our network follows a standard neural network architecture where the output function $f : \mathbb{R}^D \rightarrow [0, 1]^L$ is given as:

$$f(x) = softmax(\mathbf{V}h(x) + \mathbf{c})$$

Given source examples $S = \{(x_i^s, y_i^s)\}_{i=1}^m$ and the classification loss as the negative log-probability of the correct label:

$$\ell(f(x), y) = log\frac{1}{f_y(x)}$$

Objective function for the source classification task becomes:

$$\min_{\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c}} \left[\frac{1}{m}\sum_{i=1}^{m} \ell(f(x_i^s), y_i^s)\right] \quad (1)$$

4.2 Domain Divergence

Theoretical results in transfer learning literature (Ben-David et al., 2010) show that adapting to a target domain from a source domain depends on a measure of similarity between the two. A formal measure used in this context is known as *$\mathcal{H}$-divergence*. Intuitively, it is based on the capacity of a hypothesis class $\mathcal{H}$ to distinguish between examples generated by a pair of source-target tasks.

Definition 1 *Given feature distributions of two domains, $\mathcal{D}_s$ & $\mathcal{D}_t$ and a hypothesis class $\mathcal{H}$, the $\mathcal{H}$-divergence between $\mathcal{D}_s$ and $\mathcal{D}_t$ is defined as:*

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathbf{x}^s \sim \mathcal{D}_s} [\eta(\mathbf{x}^s) = 1] - \Pr_{\mathbf{x}^t \sim \mathcal{D}_t} [\eta(\mathbf{x}^t) = 1] \right|$$

We employ a result due to Ben-David et al. (2010) where they proved that for a symmetric hypothesis class $\mathcal{H}$, one can compute an approximate empirical *$\mathcal{H}$-divergence* by running a learning algorithm on the problem of discriminating between source and target examples. For this, we construct a new dataset as:

$$\{(\mathbf{x}_i^s, 1)\}_{i=1}^m \cup \{(\mathbf{x}_j^t, 0)\}_{j=1}^{m'}.$$

where the target and source samples are labeled as 0 and 1 respectively. Then, the error (ϵ) of the classifier trained on the above dataset can be used as an approximation of *$\mathcal{H}$-divergence* termed as *$proxy - \mathcal{A}$ distance* (PAD) and is given as:

$$\hat{d}_A = 2(1 - 2\epsilon)$$

Let the common representation for the source and target samples be $h_c(S)\{h_c(x_i^s)\}_{i=1}^m$ and $h_c(T)\{h_c(x_i^t)\}_{i=1}^{m'}$ respectively. Let $\hat{d}_{\mathcal{H}}^c(h_c(S), h_c(T))$ be the empirical $\mathcal{H}$-divergence on the common representation, given as:

$$\hat{d}_{\mathcal{H}}^c(h_c(S), h_c(T)) = 2\left(1 - \min_{\eta \in \mathcal{H}} \left[\frac{1}{m}\sum_{i=1}^{m} I[\eta(h_c(x_i^s)) = 1] + \frac{1}{m'}\sum_{i=1}^{m'} I[\eta(h_c(x_i^t)) = 0]\right]\right)$$

Algorithm 1 Learning Two-Part Hidden Representation for Neural Network

Input: Samples $S = \{(x_i^s, y_i^s)\}_{i=1}^m$ and $T = \{x_i^t\}_{i=1}^{m'}$, hidden layer size l with n_s source specific and n_c common nodes, adaptation parameter λ, learning rate α.
Output: neural network {**W,V,b,c**}.
Initialization: W, V $\leftarrow$ $random\_init(l)$
$\boldsymbol{b, c, u, d, u', d'} \leftarrow \mathbf{0}$
while stopping criteria is not met **do**
  for i from **1** to m **do**
    #Forward Propagation
    $\boldsymbol{h(x_i^s)} \leftarrow \boldsymbol{\sigma(b + W x_i^s)}$
    $\boldsymbol{f(x_i^s)} \leftarrow \boldsymbol{softmax(c + V h(x_i^s))}$
    # where $\boldsymbol{h(x_i^s) = h_c(x_i^s) + h_{ss}(x_i^s)}$
    #Backpropagation
    $\boldsymbol{\triangle_c} \leftarrow \boldsymbol{-(e(y_i^s) - f(x_i^s))}$
    $\boldsymbol{\triangle_V} \leftarrow \boldsymbol{\triangle_c h(x_i^s)^\top}$
    $\boldsymbol{\triangle_b} \leftarrow \boldsymbol{(V^\top \triangle_c) \odot h(x_i^s) \odot (1 - h(x_i^s))}$
    $\boldsymbol{\triangle_w} \leftarrow \boldsymbol{\triangle_b \cdot (x_i^s)^\top}$
    # where $\boldsymbol{h(x_i^s) = h_c(x_i^s) + h_{ss}(x_i^s)}$
    #Domain adaptation regularizer...
    #...from current domain - common representation
    $\boldsymbol{o(x_i^s)} \leftarrow \boldsymbol{\sigma(d + u^\top h_c(x_i^s))}$
    $\boldsymbol{\triangle_d} \leftarrow \boldsymbol{\lambda(1 - o(x_i^s))}$; $\boldsymbol{\triangle_u} \leftarrow \boldsymbol{\lambda(1 - o(x_i^s))h_c(x_i^s)}$
    $\boldsymbol{tmp} \leftarrow \boldsymbol{\lambda(1 - o(x_i^s))u \odot h_c(x_i^s) \odot (1 - h_c(x_i^s))}$
    $\boldsymbol{\triangle_b} \leftarrow \boldsymbol{\triangle_b + tmp}$; $\boldsymbol{\triangle_{w'}} \leftarrow \boldsymbol{\triangle_{w'} + tmp \cdot (x_i^s)^\top}$
    #...from current domain - source specific representation
    $\boldsymbol{o'(x_i^s)} \leftarrow \boldsymbol{\sigma(d' + u'^\top h_{ss}(x_i^s))}$
    $\boldsymbol{\triangle_{d'}} \leftarrow \boldsymbol{\lambda(1 - o'(x_i^s))}$
    $\boldsymbol{\triangle_{u'}} \leftarrow \boldsymbol{\lambda(1 - o'(x_i^s))h_{ss}(x_i^s)}$
    $\boldsymbol{tmp} \leftarrow \boldsymbol{\lambda(1 - o'(x_i^s))u \odot h_{ss}(x_i^s) \odot (1 - h_{ss}(x_i^s))}$
    $\boldsymbol{\triangle_b} \leftarrow \boldsymbol{\triangle_b + tmp}$; $\boldsymbol{\triangle_{w'}} \leftarrow \boldsymbol{\triangle_{w'} + tmp \cdot (x_i^s)^\top}$
  end for
  for i from **1** to m **do**
    #...from other domain - common representation
    $\boldsymbol{j} \Leftarrow \boldsymbol{uniform\_integer(1, ..., m')}$
    $\boldsymbol{h_c(x_j^t)} \Leftarrow \boldsymbol{\sigma(b + W x_j^t)}$
    $\boldsymbol{o(x_j^t)} \leftarrow \boldsymbol{\sigma(d + u^\top h_c(x_j^t))}$
    $\boldsymbol{\triangle_d} \leftarrow \boldsymbol{\triangle_d - \lambda o(x_j^t)}$;$\boldsymbol{\triangle_u} \leftarrow \boldsymbol{\triangle_u - \lambda o(x_j^t) h_c(x_j^t)}$
    $\boldsymbol{tmp} \leftarrow \boldsymbol{-\lambda o(x_j^t) u \odot h_c(x_j^t) \odot (1 - h_c(x_j^t))}$
    $\boldsymbol{\triangle_b} \leftarrow \boldsymbol{\triangle_b + tmp}$; $\boldsymbol{\triangle_{w'}} \leftarrow \boldsymbol{\triangle_{w'} + tmp \cdot (x_j^t)^\top}$
    #...from other domain - source specific representation
    $\boldsymbol{j} \Leftarrow \boldsymbol{uniform\_integer(1, ..., m')}$
    $\boldsymbol{h_{ss}(x_j^t)} \Leftarrow \boldsymbol{\sigma(b + W x_j^t)}$
    $\boldsymbol{o'(x_j^t)} \leftarrow \boldsymbol{\sigma(d' + u'^\top h_{ss}(x_j^t))}$
    $\boldsymbol{\triangle_{d'}} \leftarrow \boldsymbol{\triangle_{d'} + \lambda o'(x_j^t)}$
    $\boldsymbol{\triangle_{u'}} \leftarrow \boldsymbol{\triangle_{u'} + \lambda o'(x_j^t) h_{ss}(x_j^t)}$
    $\boldsymbol{tmp} \leftarrow \boldsymbol{\lambda o'(x_j^t) u' \odot h_{ss}(x_j^t) \odot (1 - h_{ss}(x_j^t))}$
    $\boldsymbol{\triangle_b} \leftarrow \boldsymbol{\triangle_b + tmp}$; $\boldsymbol{\triangle_{w''}} \leftarrow \boldsymbol{\triangle_{w''} + tmp \cdot (x_j^t)^\top}$
    #Update neural network parameters
    $\boldsymbol{W} \leftarrow \boldsymbol{W - \alpha \triangle_w}$; $\boldsymbol{V} \leftarrow \boldsymbol{V - \alpha \triangle_v}$
    $\boldsymbol{W'} \leftarrow \boldsymbol{W' - \alpha \triangle_{w'}}$; $\boldsymbol{W''} \leftarrow \boldsymbol{W'' - \alpha \triangle_{w''}}$
    $\boldsymbol{b} \Leftarrow \boldsymbol{b - \alpha \triangle_b}$; $\boldsymbol{c} \Leftarrow \boldsymbol{c - \alpha \triangle_c}$
    #Update domain classifier parameters
    $\boldsymbol{u} \leftarrow \boldsymbol{u + \alpha \triangle_u}$; $\boldsymbol{d} \leftarrow \boldsymbol{d + \alpha \triangle_d}$
    $\boldsymbol{u'} \leftarrow \boldsymbol{u' + \alpha \triangle_{u'}}$; $\boldsymbol{d'} \leftarrow \boldsymbol{d' + \alpha \triangle_{d'}}$
  end for
end while

where $I[\cdot]$ is the indicator function and $\eta(\cdot)$ is a hypothesis function from $\mathcal{H}$. To estimate the "min" part of the above equation, we use a logistic regression model that predicts the probability that a given input (using the common representation) is from the source domain D_S^x (denoted by $z = 1$) or the target domain D_T^x (denoted by $z = 0$):

$$p(z = 1|\phi) = o(\phi) sigm(d + u^T \phi)$$

where ϕ is either $h_c(x^s)$ or $h_c(x^t)$ and $o(\cdot)$ is the domain (logistic) regressor on the common representation with loss function $\ell^d(\cdot, \cdot)$ defined as:

$$\ell^d(o(\cdot), z) = -z \log(o(\cdot)) - (1 - z) \log(1 - o(\cdot)) \quad (2)$$

Similarly, the divergence on the source specific representation $\hat{d}_{\mathcal{H}}^{ss}(h_{ss}(S), h_{ss}(T))$ is given as:

$$\hat{d}_{\mathcal{H}}^{ss}(h_{ss}(S), h_{ss}(T)) = 2\left(1 - \min_{\eta \in \mathcal{H}} \left[\frac{1}{m}\sum_{i=1}^{m} I[\eta(h_{ss}(x_i^s)) = 1] + \frac{1}{m'}\sum_{i=1}^{m'} I[\eta(h_{ss}(x_i^t)) = 0]\right]\right)$$

The "min" part of above equation is estimated using the domain regressor for the source specific representation, $o'(\phi')sigm(d' + u'^T\phi')$, where ϕ' is either $h_{ss}(x^s)$ or $h_{ss}(x^t)$ and $\ell^{d'}(\cdot, \cdot)$ is its loss, defined similar to Eq. 2.

4.3 The Learning Algorithm

Adding domain regressor terms to the objective of Eq. 1, we get the final objective function as:

$$\min_{W,V,b,c}\Big[\frac{1}{m}\sum_{i=1}^{m}\ell(f(x_i^s),y_i^s)+$$

$$\lambda\max_{W',u,b,d}\Big(+\frac{1}{m}\sum_{i=1}^{m}\ell^d(o(x_i^s),1)+\frac{1}{m'}\sum_{i=1}^{m'}\ell^d(o(x_i^t),0)\Big)+$$

$$\lambda\min_{W'',u',b,d'}\Big(+\frac{1}{m}\sum_{i=1}^{m}\ell^{d'}(o'(x_i^s),1)+\frac{1}{m'}\sum_{i=1}^{m'}\ell^{d'}(o'(x_i^t),0)\Big)\Big]$$

where the hyper-parameter $\lambda > 0$ is the domain adaptation regularization term that controls the trade-off between the source risk and the domain divergence terms. In other words, it controls how much weight mass is put on generalizable common representation v/s the source specific representation.

The optimization problem involves minimization with respect to some parameters and maximization with respect to the others. We use a stochastic gradient descent (SGD) approach which samples a pair of source and target example x_i^s , x_i^t and updates all the parameters of the neural network. The first term in the objective represents the source classification loss and updates for its associated parameters, i.e. $\{\mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}\}$, follow the negative of the gradient to minimize this loss. The second term in the objective represents the loss of $o(\cdot)$ which is the domain regressor on the common representation. This term is maximized so as to diminish the ability of domain regressor to detect whether the sample belongs to the source or the target domain using the common representation. This makes both the domains look similar by minimizing the divergence. Note, minimizing the domain divergence is equivalent to maximizing the loss of the domain regressor. Therefore, the associated parameters, $\{\mathbf{W}', \mathbf{u}, \mathbf{d}\}$, are updated in the direction of the gradient (since we maximize with respect to them, instead of minimizing). The last term in the objective represents the loss of $o'(\cdot)$ which is the domain regressor on the source specific representation. We want the source specific representation to make the two domains appear largely distinct and hence, minimize the loss of its domain regressor. The updates for its associated parameters, $\{\mathbf{W}'', \mathbf{u}', \mathbf{d}'\}$, follows the negative of the gradient. The algorithm is detailed in Algorithm 1 where $e(y)$ represents a one-hot vector, consisting of all 0s except for a 1 at position y and $\odot$ represents the element-wise product.

Table 1: Collections from the OSM dataset.

Target	Description	# Unlabelled	# Labelled
Col1	Mobile support	22645	5650
Col2	Obama Healthcare	36902	11050
Col3	Microsoft kinnect	20907	3258
Col4	X-box	36000	4580

5 Experimental Evaluation

The effectiveness of the proposed technique which learns source specific and common shared representations between domains is evaluated for a cross-domain sentiment classification task.

5.1 Datasets

The first dataset used in this research is the Amazon review dataset (Blitzer et al., 2007) which has four domains each comprising user reviews about Books (**B**), DVDs (**D**), Kitchen appliances (**K**) and Electronics (**E**) respectively. Each domain has 2000 reviews in-total with equal number of positive and negative reviews. Each review is encoded in 5000 dimensional feature vectors of unigrams/bigrams pre-processed to tf-idf vectors. The performance is compared on 12 different cross-domain classification tasks on the Amazon review dataset and is reported as the classification accuracy for binary classification. For each task, 1400 labeled reviews from one domain constitute the source and 1400 unlabeled reviews from a different domain constitute the target. Unseen non-overlapping 200 and 400 reviews from the target domain are used as the validation and test set.

The second dataset is from Twitter.com comprising tweets about the products and services in different domains and is referred to as online social media (OSM) dataset. Table 1 lists different collections where the tweets are collected based on user-defined keywords captured in a listening engine which then crawls the social media and fetches comments matching the keywords. This dataset being noisy and comprising short-text is more challenging than the other dataset. We use labelled comments from the source and unlabelled comments from the target for learning. While reporting the performance on the target, we used the

Table 2: Comparing the cross-domain performance of different approaches on the Amazon Review dataset. D→B represents the performance of an algorithm on unlabeled target domain B with D as labeled source domain.

Method	D→ B	E→ B	K→ B	B→ D	E→ D	K→ D	D→ E	B→ E	K→ E	D→ K	B→ K	E→ K
SS	43.2	47.3	47.0	43.5	47.4	46.7	48.6	47.6	48.4	51.2	51.0	49.6
NN	76.8	71.4	74.2	71.4	72.0	67.3	71.6	72.4	77.6	76.5	77.8	79.6
SVM	77.9	73.2	75.4	73.2	73.5	70.4	73.6	71.5	78.2	77.7	78.8	82.3
SCL	78.7	75.3	76.8	78.2	75.0	73.1	75.3	75.8	84.0	77.1	79.3	85.4
SFA	80.5	75.9	76.6	77.6	75.3	74.2	75.4	77.0	84.2	78.1	80.3	85.8
PJNMF	81.8	77.2	78.8	79.4	76.3	75.8	76.4	77.8	84.4	79.0	81.6	86.4
SDA	81.1	76.6	76.8	78.2	75.4	75.4	75.8	77.4	83.8	78.4	80.8	87.2
mSDA	81.3	77.6	78.5	79.5	76.5	76.4	75.4	77.2	83.6	78.5	81.2	88.2
TLDA	81.5	78.0	80.6	79.8	76.6	76.4	76.2	78.0	84.2	79.4	81.8	87.6
BTDNNs	81.9	78.6	81.2	80.0	77.9	76.2	76.8	78.6	85.2	80.5	82.7	88.3
SS+Common	78.8	76.7	77.3	74.4	77.8	73.6	74.4	76.8	80.4	78.6	80.2	83.5
DANN	79.5	77.4	78.2	76.3	78.4	76.3	75.2	77.2	81.4	78.9	80.6	85.8
DSN	81.5	78.9	79.0	78.3	79.5	77.4	76.0	78.3	83.4	79.5	81.4	87.7
Proposed	**83.2**	**81.8**	**83.8**	**81.3**	**81.8**	**82.2**	**82.4**	**83.2**	**86.0**	**86.2**	**88.4**	**89.9**
Gold-standard	*84.6*	*84.6*	*84.6*	*83.4*	*83.4*	*83.4*	*86.7*	*86.7*	*86.7*	*90.2*	*90.2*	*90.2*

Table 3: Comparing the cross-domain performance of different approaches on the OSM dataset.

Method	Col2→1	Col3→1	Col4→1	Col1→2	Col3→2	Col4→2	Col1→3	Col2→3	Col4→3	Col1→4	Col2→4	Col3→4
SS	35.0	39.4	35.6	32.8	40.2	38.6	40.7	41.9	42.5	45.0	44.9	42.4
NN	66.4	65.2	68.3	65.8	66.8	63.8	65.2	67.2	68.2	67.3	67.2	68.1
SVM	67.1	63.2	64.3	62.6	64.3	60.4	62.8	63.2	65.8	68.2	69.3	72.4
SCL	68.2	67.5	67.2	67.1	67.3	64.1	64.5	65.3	72.1	68.8	70.1	73.6
SFA	71.3	67.6	67.8	69.1	70.2	67.8	68.2	68.4	74.2	69.5	72.3	76.3
PJNMF	72.0	67.2	68.3	70.4	70.5	68.4	69.3	69.1	74.8	70.0	72.5	74.8
SDA	71.5	66.3	67.6	68.2	69.3	70.2	67.6	68.3	68.7	72.4	69.3	72.6
mSDA	72.1	67.5	68.2	69.0	70.4	70.8	68.3	69.1	69.2	73.0	70.2	73.1
TLDA	72.4	67.8	68.6	69.7	71.1	71.5	68.8	69.8	70.0	73.8	70.7	73.8
BTDNNs	73.1	68.3	69.0	70.2	71.6	72.1	69.4	70.2	70.6	74.2	71.3	74.2
SS+Common	68.7	67.9	67.7	67.5	67.8	64.9	65.0	65.7	72.6	69.4	70.7	74.2
DANN	69.6	69.5	69.8	70.0	68.7	66.2	66.3	66.6	73.4	70.6	71.4	75.7
DSN	72.9	68.6	69.4	70.5	72.0	72.2	69.5	70.3	70.8	74.3	71.5	74.6
Proposed	**77.6**	**74.5**	**75.5**	**76.2**	**77.8**	**78.2**	**75.2**	**75.7**	**76.1**	**80.1**	**77.9**	**80.9**
Gold-standard	*78.2*	*78.2*	*78.2*	*79.1*	*79.1*	*79.1*	*81.0*	*81.0*	*81.0*	*81.4*	*81.4*	*81.4*

comments for which the actual labels are available; however, label information is used only as ground truth to report the performance. The comments were pre-processed by converting it to lowercase followed by stemming. Further, feature selection was based on document frequency (DF = 5) which reduces the number of features as well as speed up the learning task.

5.2 Experimental Protocol

Performance of proposed architecture is compared with standard neural network architecture with one hidden layer ("NN") (as described in Eq. 1) and a support vector machine ("SVM") (Chih-Wei Hsu and Lin, 2003) with linear kernel where the training is performed on labelled source domain and performance is reported on the target domain. "Gold-standard" refers to target domain supervised performance of the SVM. The performance is further compared with popular shared representation learning approaches for domain adaptation including Structural Correspondance Learning ("SCL") (Blitzer et al., 2006),(Blitzer et al., 2007), Spectral Feature Alignment ("SFA") (Pan et al., 2010) and "PJNMF" (Zhou et al., 2015).

We also compared the performance with "DANN" (Ajakan et al., 2014), stacked Denoising Auto-encoders ("SDA") (Glorot et al., 2011), and marginalized SDA ("mSDA") (Chen et al., 2012b) and transfer learning with deep auto-encoders ("TLDA") (Pan et al., 2008) , "BTDNNs" (Zhou et al., 2016) and "DSN" (Bousmalis et al., 2016) which are some of the popular approaches in cross-domain sentiment analysis. The performance is also compared with different components of the learned representations i.e. source specific ("SS"), common ("Proposed"), and "SS+common" representations. For SDA, mSDA, TLDA, BTDNNs, SS, SS+common and the proposed, a standard SVM is trained on the learned representation and is applied to predict the sentiment labels for target data.

Training is done using stochastic gradient descent (SGD) with minibatch size of 50. The initial learning rate was fixed at 0.01 and then empirically varied to find optimal value as 0.0001. Epochs were fixed at 25, above which gradients were found to saturate. The hyperparameter λ was varied in the range [0,1].

5.3 Results and Analysis

Results in Table 2 show the efficacy of the proposed neural network architecture for learning common shared representation while limiting the source specific representation from negatively effecting their generalizable capabilities in the target domain. Results suggest that the learned common representation, referred to as "Proposed", consistently outperforms other existing algorithms for all cross-domain sentiment analysis task on the Amazon review dataset (Blitzer et al., 2007). The source specific (SS) representation performs consistently poor at the all target tasks as they are trained to emphasize only on the source task. Results also suggest that combining source specific representation with the common representation, referred to as "SS+Common" leads to a lower performance than the common representation alone. This validates our assertion that combining source specific characteristics with common representation negatively effects the generalization capabilities of the common representation in the target domain. The proposed method also surpasses BTDNNs (state-of-the-art) which focuses on the feasibility of transfer between domains with a linear data reconstruction for distribution consistency. Contrary to the proposed two-part representations, it suggests that enforcing distribution consistency across all hidden units suppresses the discriminating information which results in lower classification performance for BTDNNs. The proposed approach even outperforms deep learning based methods (SDA, mSDA and TLDA) as these approaches learn the unified domain-invariable feature representation by combining the source domain and target domain data which may not separate out the domain-specific features from the commonality of domains. On the contrary, the objective used in the paper is based on the min-max optimization criterion that minimizes the domain specific component loss as well as maximizes the shared component loss. In other words, the proposed approach not only models the similarity between domains but also models and mitigates the source domain specific information, thus leading to better cross-domain performance.

Results in Table 3 compare the performance of all algorithms on the OSM dataset. We observe that the overall performance of all the algorithms is lower on the OSM dataset, as compared to the first dataset, as it is more challenging due to short and noisy text. Both Tables 2 & 3 demonstrate that the domain adaptation methods perform better than the baselines and "SS" representation which suggests that transferring knowledge across domains benefits the cross-domain sentiment classification task. The improvements achieved by the proposed technique, which reaches closest to the target domain supervised performance "Gold-standard", is consistently better than the existing algorithms as it explicitly keeps away any source specific components from the learned common representation so as to yield the best generalization on the target domain.

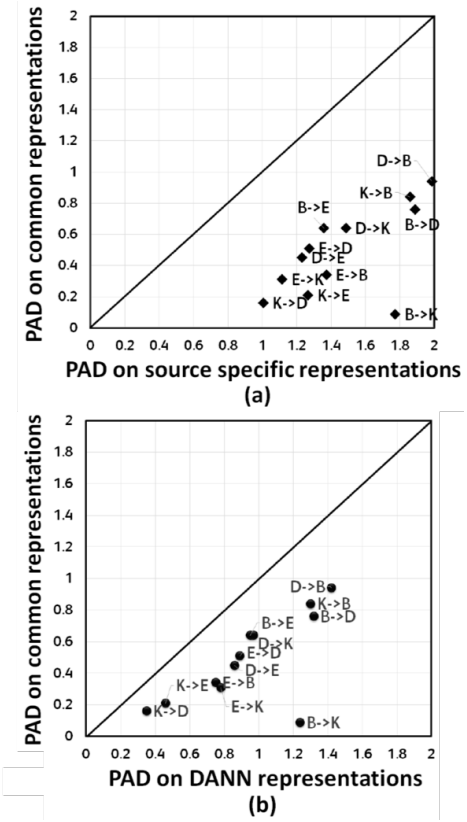

Figure 3: Compares the $proxy - \mathcal{A}$ distances (PAD) computed on the common representations learned using the proposed technique v/s the (a) source specific and (b) representations learned using DANN.

5.3.1 The Common Representation:

The primary objective of the common representation is to make the source and target distributions appear similar. In other words, these representation should be such that it becomes arduous to distinguish between the source and target examples for a model trained on this representation. We compute $proxy - \mathcal{A}$ distance (PAD) between two domains, as explained in Eq. 2. Figure 3 il-

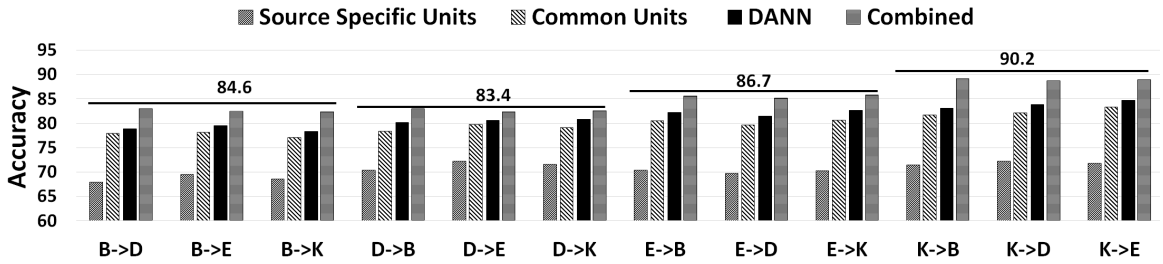

Figure 4: Compares the performance on the source classification task. For example, B→D here represent the performance of an algorithm on the source domain B when the representations are learned with B as labeled source and D as unlabeled target domain.

lustrates that the learned common representation leads to a lower PAD when compared with either the source specific representation or the representation learned using DANN. A low PAD between domains for a given representation signifies that the divergence between the domains is reduced.

5.3.2 The Source Specific Representation:

We evaluate the performance of different parts of the learned representation on the source domain. Optimizing the target domain performance is indeed the primary objective for domain adaptation; however, existing domain adaptation algorithms generally exhibit a lower performance on the source. We empirically demonstrate that the proposed method for learning source specific and common parts of the hidden layer sustains a higher level of performance in the source as well.

Results in Figure 4 compares the performance of the different representations on the source domain. We compare the performance of the source specific representation and the common representation learned using the proposed approach with the representation learned using DANN (Ajakan et al., 2014) and the skyline source domain performance. Results suggest that while the two individual parts of the learned representation yield lower source domain performance, the combined source specific and common representation ("combined") outperforms the source domain performance of the representation learned using DANN. This signifies that the two parts of the learned representation learn complementary characteristics i.e. source specific and general.

5.3.3 Source Specific & Common Units:

While learning the two part representation with our neural network architecture, the number of source specific and common units in the hidden layer is an important factor to influence the cross-domain performance. In our experiments, we observed that when the source and target domains were similar (as measured by the PAD), hidden layer with a higher portion of common v/s source specific units resulted in better cross-domain performance as compared to when the source and target domains were dissimilar. This intuitively suggests that for similar domains there are more commonalities than domain specific characteristics and hence, a higher number of common units is required to capture this commonality. Similarly, we observed that for not so similar domains, the source specific units dominate the number of common units.

6 Conclusion & Future Work

The paper proposed a novel neural network learning algorithm based on the principle of learning a two-part representation where each part optimizes for different objective. One part captures the source specific characteristics that are discriminating for learning in the source domain. The other part captures the common representation between the source and target domain pair which contributes to both source domain learning as well as generalizes to the unlabelled target domain task. The major contribution of this work is to learn the common shared representation between domains by explicitly disentangling the source specific characteristics so as not to detract the capabilities of common representation for the cross-domain task. In the cross-domain task, the common part of the representation performs best when it is isolated from the source specific part. On the contrary, both the source specific and common parts of the representation come along for efficient performance in the source domain task. Finally, we demonstrated the efficacy of the proposed approach for cross-domain classification on different datasets.

References

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. 2014. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2018. Projecting embeddings for domain adaption: Joint modeling of sentiment analysis in diverse domains. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 818–830. Association for Computational Linguistics.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

Himanshu Sharad Bhatt, Arun Rajkumar, and Shourya Roy. 2016. Multi-source iterative adaptation for cross-domain classification. In *In Proceedings of International Joint Conference on Artificial Intelligence*.

Himanshu Sharad Bhatt, Deepali Semwal, and Shourya Roy. 2015. An iterative similarity based adaptation technique for cross-domain text classification. In *Proceedings of Conference on Natural Language Learning*, pages 52–61.

J. Blitzer, R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 120–128.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of Association for Computational Linguistics*, pages 440–447.

John Blitzer, Sham Kakade, and Dean Foster. 2011. Domain adaptation with coupled subspaces. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 173–181, Fort Lauderdale, FL, USA. PMLR.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *arXiv preprint arXiv:1608.06019*.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012a. Marginalized denoising autoencoders for domain adaptation.

Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. 2012b. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1627–1634, USA. Omnipress.

Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.

S Chopra, S Balakrishnan, and R Gopalan. 2013. Dlid: Deep learning for domain adaptation by interpolating between domains. *ICML Workshop on Challenges in Representation Learning*.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Co-clustering based classification for out-of-domain documents. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 210–219.

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

I. S. Dhillon, S. Mallela, and D. S Modha. 2003. Information-theoretic co-clustering. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 89–98.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2013. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2960–2967.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *In Proceedings of the International Conference on Machine Learning*.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2011. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE.

Maayan Harel and Shie Mannor. 2010. Learning from multiple outlooks. *arXiv preprint arXiv:1005.0027*.

Jing Jiang and ChengXiang Zhai. 2007. A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 401–410. ACM.

Abhishek Kumar, Avishek Saha, and Hal Daume. 2010. Co-regularization based semi-supervised domain adaptation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 478–486. Curran Associates, Inc.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1–10.

Mingsheng Long and Jianmin Wang. 2015. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Sinno Jialin Pan, , Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Sinno Jialin Pan, James T Kwok, and Qiang Yang. 2008. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682.

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 International Conference on World Wide Web*, pages 751–760.

Prathusha K. Sarma, Yingyu Liang, and William A. Sethares. 2018. Domain adapted word embeddings for improved sentiment classification. *CoRR*, abs/1805.04576.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Qi Zhang, Xuanjing Huang, Minlong Peng, and Yu-Gang Jiang. 2018. Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2505–2513.

Guangyou Zhou, Tingting He, Wensheng Wu, and Xiaohua Tony Hu. 2015. Linking heterogeneous input features with pivots for domain adaptation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1419–1425.

Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. 2016. Bi-transferring deep neural networks for domain adaptation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 322–332. Association for Computational Linguistics.